ESPOL /DISK

```
%#########################################################################
%
%                    B-5700 ESPOL COMPILER
%                         MARK XVI.0.00
%                         OCT  1, 1974
%
%#########################################################################
%
  COMMENT: * TITLE: B5500/B5700 MARK XVI SYSTEM RELEASE                 *
           * FILE ID: SYMBOL/ESPOL TAPE ID: SYMBOL1/FILE000             *
           * THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION      *
           * AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED            *
           * EXCEPT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON          *
           * WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF            *
           * BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232             *
           *                                                            *
           * COPYRIGHT (C)  1971, 1972, 1974                            *
           *       BURROUGHS CORPORATION                                *
           *  AA320206   AA393180   AA332366                           *;
COMMENT####################################################################
                    ERROR MESSAGES
#########################################################################
%
ERROR NUMBER   ROUTINE:ERROR MESSAGE
 000           BLOCK:  DECLARATION NOT FOLLOWED BY SEMICOLON.
 001           BLOCK:  IDENTIFIER DECLARED TWICE IN SAME BLOCK.
 002           PROCEDUREDEC: SPECIFICATION PART CONTAINS
                             IDENTIFIER NOT APPEARING IN
                             FORMAL PARAMETER PART.
 003           BLOCK:  NON-IDENTIFIER APPEARS IN IDENTIFIER
                       LIST OF DECLARATION.
 004           PROCEDUREDEC: STREAM PROCEDURE DECLARATION
                             PRECEDED BY ILLEGAL DECLARATOR.
 005           PROCEDUREDEC: PROCEDURE DECLARATION PRECEDED
                             BY ILLEGAL DECLARATOR.
 006           PROCEDUREDEC: PROCEDURE IDENTIFIER USED BEFORE
                             IN SAME BLOCK(NOT FORWARD).
 007           PROCEDUREDEC: PROCEDURE IDENTIFIER NOT FOLLOWED
                             BY ( OR SEMICOLON IN PROCEDURE
                             DECLARATION.
 008           PROCEDUREDEC: FORMAL PARAMETER LIST NOT FOLLOWED
                             BY ).
 009           PROCEDUREDEC: FORMAL PARAMETER PART NOT FOLLOWED
                             BY SEMICOLON.
 010           PROCEDUREDEC: VALUE PART CONTAINS IDENTIFIER
                             WHICH DID NOT APPEAR IN FORMAL
                             PARAPART.
```

```
00001000 T  0000
00001010 T  0000
00001020 T  0000
00001030 T  0000
00001040 T  0000
00001050 T  0000
00001060 T  0000
00001070 T  0000
00001072 T  0000
00001073 T  0000
00001074 T  0000
00001075 T  0000
00001076 T  0000
00001077 T  0000
00001078 T  0000
00001079 T  0000
00001080 T  0000
00001081 T  0000
00001082 T  0000
00001110 T  0000
00001120 T  0000
00001130 T  0000
00001140 T  0000
00002000 T  0000
00003000 T  0000
00004000 T  0000
00005000 T  0000
00006000 T  0000
00007000 T  0000
00008000 T  0000
00009000 T  0000
00010000 T  0000
00011000 T  0000
00012000 T  0000
00013000 T  0000
00014000 T  0000
00015000 T  0000
00016000 T  0000
00017000 T  0000
00018000 T  0000
00019000 T  0000
00020000 T  0000
00021000 T  0000
00022000 T  0000
00023000 T  0000
00024000 T  0000
00025000 T  0000
```

```
011          PROCEDUREDEC: VALUE PART NOT ENDED BY SEMICOLON.                    00026000 T  0000
012          PROCEDUREDEC: MISSING OR ILLEGAL SPECIFICATION                      00027000 T  0000
                           PART.                                                 00028000 T  0000
013          PROCEDUREDEC: OWN USED IN ARRAY SPECIFICATION.                      00029000 T  0000
014          PROCEDUREDEC: SAVE USED IN ARRAY SPECIFICATION.                     00030000 T  0000
015          BLOCK:  DECLARATION PRECEDED BY ILLEGAL DECLARATOR.                 00031000 T  0000
016          ARRAYDEC:     ARRAY ID IN DECLARATION NOT FOLLOWED                  00032000 T  0000
                           BY ( .                                                00033000 T  0000
017          ARRAYDEC:     LOWER BOUND IN ARRAY DEC NOT                          00034000 T  0000
                           FOLLOWED BY : .                                       00035000 T  0000
018          ARRAYDEC:     BOUND PAIR LIST NOT FOLLOWED BY ].                    00036000 T  0000
019          ARRAYSPEC:    ILLEGAL LOWER BOUND DESIGNATOR IN                     00037000 T  0000
                           ARRAY SPECIFICATION.                                  00038000 T  0000
020          BLOCK:        OWN APPEARS IMMEDIATELY BEFORE                        00039000 T  0000
                           IDENTIFIER(NO TYPE).                                  00040000 T  0000
021          BLOCK:        SAVE APPEARS IMMEDIATELY BEFORE                       00041000 T  0000
                           IDENTIFIER(NO TYPE).                                  00042000 T  0000
022          BLOCK:        STREAM APPEARS IMMEDIATELY BEFORE                     00043000 T  0000
                           IDENTIFIER(THE WORD PROCEDURE LEFT                    00044000 T  0000
                           OUT).                                                 00045000 T  0000
023          BLOCK:        DECLARATOR PRECEDED ILLEGALLY BY                      00046000 T  0000
                           ANOTHER DECLARATOR.                                   00047000 T  0000
024          PROCEDUREDEC: LABEL CANNOT BE PASSED TO FUNCTION.                   00048000 T  0000
025          BLOCK:        DECLARATOR OR SPECIFIER ILLEGALLY                     00049000 T  0000
                           PRECEDED BY OWN OR SAVE OR SOME                       00050000 T  0000
                           OTHER DECLARATOR.                                     00051000 T  0000
026          FILEDEC:      MISSING ( IN FILE DEC.                                00052000 T  0000
027          FILEDEC:      NO. OF BUFFERS IN FILE DEC MUST BE                    00053000 T  0000
                           AN UNSIGNED INTEGER.                                  00054000 T  0000
028          FILEDEC:      ILLEGAL BUFFER PART OR SAVE FACTOR                    00055000 T  0000
                           IN FILE DEC.                                          00056000 T  0000
029          FILEDEC:      MISSING ) IN FILE DEC.                                00057000 T  0000
030          PROCEDUREDEC: PROCEDURE TYPE AT ACTUAL DECLARATION                  00058000 T  0000
                           TIME DIFFERENT THAN AT FORWARD DEC.                   00059000 T  0000
031          LISTDEC:      MISSING ( IN LISTDEC.                                 00060000 T  0000
032          FORMATDEC:    MISSING ( IN FORMAT DEC.                              00061000 T  0000
033          SWITCHDEC:    SWITCH DEC DOES NOT HAVE ← OR                         00062000 T  0000
                           FORWARD AFTER IDENTIFIER.                             00063000 T  0000
034          SWITCHFILEDEC:MISSING ← AFTER FILED.                                00064000 T  0000
035          SWITCHFILEDEC:NON FILE ID APPEARING IN DECLARATION                  00065000 T  0000
                           OF SWITCHFILE.                                        00066000 T  0000
036          SUPERFORMATDEC:FORMAT ID NOT FOLLOWED BY ←   .                      00067000 T  0000
037          SUPERFORMATDEC:MISSING ( AT START OF FORMATPHRASE .                 00068000 T  0000
038          SUPERFORMATDEC:FORMAT SEGMENT >1022 WORDS.                          00069000 T  0000
040          SEGMENT:      SAVE CODE EXCEEDS 4080 WHICH KERNAL CAN H/L           00069100 T  0000
050     ANYWHERE:   OUT OF RANGE OF C RELATIVE ADDRESSING FOR CONSTANT           00069500 T  0000
051     BLOCK    :  ILLEGAL F RELATIVE ADDRESS EXP IN DECLARATION                00069510 T  0000
052  BLOCK:    PROCEDURE WHOSE BODY NOT A BLOCK                                  00069520 T  0000
053       ARRAYDEC:  CANT FIND RIGHT BRACKET IN SAVE ARRAY DEC                   00069530 T  0000
054       ARRAYDEC: FILL PART OF SAVE ARRAY DEC LONGER THAN SIZE                 00069540 T  0000
056       ARRAYDEC: ILLEGAL DIMENSION INDICATOR IN ARRAY DEC                     00069560 T  0000
057       SEGMENTSTART:SAVE STORAGE NOT ALLOWED WITH INTRINSIC OPTION            00069570 T  0000
098            IOSTMT:  ILLEGAL SPECIFIER IN SCOPE STMT: MUST BE ≥15.           00069980 T  0000
099            INLINE:  EXTRA : IN STREAM HEAD.                                 00069990 T  0000
100            ANYWHERE:  UNDECLARED IDENTIFIER.                                00070000 T  0000
101            CHECKER:  AN ATTEMPT HAS BEEN MADE TO ADDRESS AN                 00071000 T  0000
                     IDENTIFIER WHICH IS LOCAL TO ONE PROCEDURE AND GLOBAL      00072000 T  0000
```

```
                    TO ANOTHER.  IF THE QUANTITY IS A PROCEDURE NAME OR          00073000 T  0000
                    AN OWN VARIABLE THIS RESTRICTION IS RELAXED.                 00074000 T  0000
102                 AEXP:  CONDITIONAL EXPRESSION IS NOT OF ARITHMETIC TYPEH     00075000 T  0000
103                 PRIMARY:  PRIMARY MAY NOT BEGIN WITH A QUANTITY OF THIS      00076000 T  0000
                    TYPE.                                                        00077000 T  0000
104                 ANYWHERE:  MISSING RIGHT PARENTHESIS.                        00078000 T  0000
105                 ANYWHERE:  MISSING LEFT PARENTHESIS.                         00079000 T  0000
106                 PRIMARY:  PRIMARY MAY NOT START WITH DECLARATOR.             00080000 T  0000
107                 BEXP:  THE EXPRESSION IS NOT OF BOOLEAN TYPE.                00081000 T  0000
108                 EXPRSS:  A RELATION MAY NOT HAVE CONDITIONAL EXPRESSIONS     00082000 T  0000
                    AS THE ARITHMETIC EXPRESSIONS.                               00083000 T  0000
109                 BOOSEC,SIMPBOO, AND BOOCOMP:  THE PRIMARY IS NOT BOOLEAN.    00084000 T  0000
110                 BOOCOMP: A NON-BOOLEAN OPERATOR OCCURS IN A BOOLEAN          00085000 T  0000
                    EXPRESSION.                                                  00086000 T  0000
111                 BOOPRIM:  NO EXPRESSION (ARITHMETIC, BOOLEAN, OR DESIGNA-    00087000 T  0000
                    TIONAL) MAY BEGIN WITH A QUANTITY OF THIS TYPE.              00088000 T  0000
112                 BOOPRIM:  NO EXPRESSION (ARITHMETIC, BOOLEAN, OR DESIGNA-    00089000 T  0000
                    TIONAL) MAY BEGIN WITH A DECLARATOR.                         00090000 T  0000
113                 PARSE:  EITHER THE SYTAX OR THE RANGE OF THE LITERALS FOR    00091000 T  0000
                    A CONCATENATE OPERATOR IS INCORRECT.                         00092000 T  0000
114                 DOTSYNTAX:  EITHER THE SYNTAX OR THE RANGE OF THE LITERALS   00093000 T  0000
                    FOR A PARTIAL WORD DESIGNATOR IS INCORRECT.                  00094000 T  0000
115                 DEXP:  THE EXPRESSION IS NOT OF DESIGNATIONAL TYPE.          00095000 T  0000
116                 IFCLAUSE:  MISSING THEN.                                     00096000 T  0000
117                 BANA:  MISSING LEFT BRAKET.                                  00097000 T  0000
118                 BANA:  MISSING RIGHT BRAKET.                                 00098000 T  0000
119                 COMPOUNDTAIL:  MISSING SEMICOLON OR END.                     00099000 T  0000
120                 COMPOUNDTAIL:  MISSING END.                                  00100000 T  0000
121                 ACTUALPARAPART:  AN INDEXED FILE MAY BE PASSED BY NAME       00101000 T  0000
                    ONLY AND ONLY TO A STREAM PROCEDURE - THE STREAM             00102000 T  0000
                    PROCEDURE MAY NOT DO A RELEASE ON THIS TYPE PARA-            00103000 T  0000
                    METER.                                                       00104000 T  0000
122                 ACTUALPARAPART:  STREAM PROCEDURE MAY NOT HAVE AN            00105000 T  0000
                    EXPRESSION PASSED TO IT BY NAME.                             00106000 T  0000
123                 ACTUALPARAPART:  THE ACTUAL AND FORMAL PARAMETERS DO NOT     00107000 T  0000
                    AGREE AS TO TYPE.                                            00108000 T  0000
124                 ACTUALPARAPART:  ACTUAL AND FORMAL ARRAYS DO NOT HAVE SAME   00109000 T  0000
                    NUMBER OF DIMENSIONS.                                        00110000 T  0000
125                 ACTUALPARAPART:  STREAM PROCEDURES MAY NOT BE PASSED AS A    00111000 T  0000
                    PARAMETER TO A PROCEDURE.                                    00112000 T  0000
126                 ACTUALPARAPART:  NO ACTUAL PARAMETER MAY BEGIN WITH A        00113000 T  0000
                    QUANTITY OF THIS TYPE.                                       00114000 T  0000
127                 ACTUALPARAPART:  THIS TYPE QUANTITY MAY NOT BE PASSED TO A   00115000 T  0000
                    STREAM PROCEDURE.                                            00116000 T  0000
128                 ACTUALPARAPART:  EITHER ACTUAL AND FORMAL PARAMETERS DO      00117000 T  0000
                    NOT AGREE AS TO NUMBER, OR EXTRA RIGHT PARENTHESIS.          00118000 T  0000
129                 ACTUALPARAPART:  ILLEGAL PARAMETER DELIMITER.                00119000 T  0000
130                 RELSESTMT:  NO FILE NAME.                                    00120000 T  0000
131                 DOSTMT:  MISSING UNTIL.                                      00121000 T  0000
132                 WHILESTMT:  MISSING DO.                                      00122000 T  0000
133                 LABELR:  MISSING C OLON.                                     00123000 T  0000
134                 LABELR:  THE LABEL WAS NOT DECLARED IN THIS BLOCK.           00124000 T  0000
135                 LABELR:  THE LABEL HAS ALREADY OCCURED.                      00125000 T  0000
136                 FORMATPHRASE: IMPROPER FORMAT EDITING PHRASE.                00126000 T  0000
137                 FORMATPHRASE: A FORMAT EDITING PHRASE DOES NOT HAVE AN       00127000 T  0000
                    INTEGER WHERE AN INTEGER IS REQUIRED.                        00128000 T  0000
138                 FORMATPHRASE:  THE WIDTH IS TOO SMALL IN E OR F EDITING      00129000 T  0000
```

```
                PHRASE.                                                  00130000 T  0000
139             TABLE:  DEFINE IS NESTED MORE THAN EIGHT DEEP.           00131000 T  0000
140             NEXTENT:  AN INTEGER IN A FORMAT IS GREATER THAN 1023.   00132000 T  0000
141             SCANNER:  INTEGER OR IDENTIFIER HAS MORE THAN 63         00133000 T  0000
                CHARACTORS.                                              00134000 T  0000
142             DEFINEGEN:  A DEFINE CONTAINS MORE THAN 2047 CHARACTORS  00135000 T  0000
                (BLANK SUPPRESSED).                                      00136000 T  0000
143             COMPOUNDTAIL:  EXTRA END.                                00137000 T  0000
144             STMT:  NO STATEMENT MAY START WITH THIS TYPE IDENTIFIER. 00138000 T  0000
145             STMT:  NO STATEMENT MAY START WITH THIS TYPE QUANTITY.   00139000 T  0000
146             STMT:  NO STATEMENT MAY START WITH A DECLARATOR - MAY BE 00140000 T  0000
                A MISSING END OF A PROCEDURE OR A MISPLACED              00141000 T  0000
                DECLARATION.                                             00142000 T  0000
147             SWITCHGEN:  MORE THAN 256 EXPRESSIONS IN A SWITCH        00143000 T  0000
                DECLARATION.                                             00144000 T  0000
148             GETSPACE:  MORE THAN 1023 PROGRAM REFERENCE TABLE CELLS  00145000 T  0000
                ARE REQUIRED FOR THIS PROGRAM.                           00146000 T  0000
149             GETSPACE:  MORE THAN 255 STACK CELLS ARE REQUIRED FOR THIS 00147000 T  0000
                PROCEDURE.                                               00148000 T  0000
150             ACTUALPARAPART:  CONSTANTS MAY NOT BE PASSED BY NAME TO  00149000 T  0000
                STREAM PROCEDURES.                                       00150000 T  0000
151             FORSTMT:  IMPROPER FOR INDEX VARIABLE.                   00151000 T  0000
152             FORSTMT:  MISSING LEFT ARROW FOLLOWING INDEX VARIABLE.   00152000 T  0000
153             FORSTMT:  MISSING UNTIL OR WHILE IN STEP ELEMENT.        00153000 T  0000
154             FORSTMT:  MISSING DO IN FOR CLAUSE.                      00154000 T  0000
155             IFEXP:  MISSING ELSE                                     00155000 T  0000
156             LISTELEMENT:  A DESIGNATIONAL EXPRESSION MAY NOT BE A LIST 00156000 T  0000
                ELEMENT.                                                 00157000 T  0000
157             LISTELEMENT:  A ROW DESIGNATOR MAY NOT BE A LISTELEMENT  00158000 T  0000
158             LISTELEMENT:  MISSING RIGHT BRAKET IN GROUP OF ELEMENTS  00159000 T  0000
159             PROCSTMT:  ILLEGAL USE OF PROCEDURE OR FUNCTION IDENTIFIER 00160000 T  0000
160             PURGE:  DECLARED LABEL DOES NOT OCCUR.                   00161000 T  0000
161             PURGE:  DECLARED FORWARD PROCEDURE DOES NOT OCCUR.       00162000 T  0000
163             ZIPSTMT:  MISSING COMMA IN ZIP STATEMENT                 00163000 T  0000
163             FORMATPHRASE:  THE WIDTH OF A FIELD IS MORE THAN 63.     00164000 T  0000
200             EMIT:   SEGMENT TOO LARGE ( > 4093SYLLABLES).            00165000 T  0000
201             SIMPLE VARIABLE:  PARTIAL WORD DESIGNATOR NOT  LEFT-MOST 00166000 T  0000
                IN A LEFT PART LIST.                                     00167000 T  0000
202             SIMPLE VARIABLE:  MISSING . OR ← .                       00168000 T  0000
203             SUBSCRIPTED VARIABLE:  WRONG NUMBER OF SUBSCRIPTS IN A ROW 00169000 T  0000
                DESIGNATOR.                                              00170000 T  0000
204             SUBSCRIPTED VARIABLE:  MISSING ] IN A ROW DESIGNATOR.    00171000 T  0000
205             SUBSCRIPTED VARIABLE:  A ROW DESIGNATOR APPEARS OUTSIDE OF 00172000 T  0000
                AN ACTUAL PARAMETER LIST OR FILL STATEMENT.              00173000 T  0000
206             SUBSCRIPTED VARIABLE:  MISSING ].                        00174000 T  0000
207             SUBSCRIPTED VARIABLE:  MISSING [.                        00175000 T  0000
208             SUBSCRIPTED VARIABLE:  WRONG NUMBER OF SUBSCRIPTS.       00176000 T  0000
209             SUBSCRIPTED VARIABLE:  PARTIAL WORD DESIGNATOR NOT LEFT- 00177000 T  0000
                MOST IN A LEFT PART LIST.                                00178000 T  0000
210             SUBSCRIPTED VARIABLE:  MISSING , OR ← .                  00179000 T  0000
211             VARIABLE:  PROCEDURE ID USED OUTSIDE OF SCOPE IN LEFT PART. 00180000 T  0000
250             STREAM STMT:ILLEGAL STREAM STATEMENT.                    00181000 T  0000
251             ANY STREAM STMT PROCEDURE:  MISSING ←.                   00182000 T  0000
252             INDEX:  MISSING + OR - .                                 00183000 T  0000
253             INDEX:  MISSING NUMBER OR STREAM VARIABLE.               00184000 T  0000
254             EMITC:  NUMBER>63 OR NUMBER OF LABELS+LOCALS+FORMALS>63.  00185000 T  0000
255             DSS:  MISSING STRING IN DS← LIT STATEMENT.               00186000 T  0000
```

```
256         RELEASES: MISSING PARENTHESIS OR FILE IDENTIFIER IS NOT                  00187000 T  0000
                A FORMAL PARAMETER.                                                  00188000 T  0000
257         GOTOS,LABELS,OR JUMPS: LABEL SPECIFIED IS NOT ON THE SAME                00189000 T  0000
                NEST LEVEL AS A PRECEDING APPEARANCE OF THE                          00190000 T  0000
                LABEL.                                                               00191000 T  0000
258         LABELS: MISSING ;.                                                       00192000 T  0000
259         LABELS: LABEL APPEARS MORE THAN ONCE.                                    00193000 T  0000
260         GOTOS: MISSING LABEL IN A GO TO OR JUMP OUT TO STATEMENT.                00194000 T  0000
261         JUMPS: MISSING OUT IN JUMP OUT STATEMENT.                                00195000 T  0000
262         NESTS: MISSING PARENTHESIS.                                              00196000 T  0000
263         IFS:MISSING SC IN IF STATEMENT.                                          00197000 T  0000
264         IFS: MISSING RELATIONAL IN IF STATEMENT.                                 00198000 T  0000
265         IFS: MISSING ALPHA,DC OR STRING IN IF STATEMENT.                         00199000 T  0000
266         IFS: MISSING THEN INIF STATEMENT.                                        00200000 T  0000
267         FREDFIX: THERE ARE GO TO STATEMENTS IN WHICH THE LABEL IS                00201000 T  0000
                UNDEFINED.                                                           00202000 T  0000
268         EMITC:  A REPEAT INDEX ≥ 64 WAS SPECIFIED OR TOO MANY                    00203000 T  0000
                FORMAL PARAMETERS,LOCALS AND LABELS.                                 00204000 T  0000
269         TABLE: A CONSTANT IS SPECIFIED WHICH IS TOO LARGE                        00205000 T  0000
                OR TOO SMALL.                                                        00206000 T  0000
281         DBLSTMT:  MISSING (.                                                     00207000 T  0000
282         DBLSTMT:  TOO MANY OPERATORS.                                            00208000 T  0000
283         DBLSTMT:  TOO MANY OPERANDS.                                             00209000 T  0000
284         DBLSTMT:  MISSING , .                                                    00210000 T  0000
285         DBLSTMT:  MISSING ) .                                                    00211000 T  0000
300         FILLSTMT:  THE IDENTIFIER FOLLOWING "FILL" IS NOT                        00212000 T  0000
                AN ARRAY IDENTIFIER.                                                 00213000 T  0000
301         FILLSTMT:  MISSING "WITH" IN FILL STATEMENT.                             00214000 T  0000
302         FILLSTMT:  IMPROPER FILL ELEMENT.                                        00215000 T  0000
303         FILLSTMT:  NON-OCTAL CHARACTER IN OCTAL FILL.                            00216000 T  0000
304         FILLSTMT:  IMPROPER ARRAY ROW DESIGNATOR IN FILL.                        00217000 T  0000
305         FILLSTMT:  DATA IN FILL EXCEEDS 1023 WORDS.                              00218000 T  0000
306         FILLSTMT:  ODD NUMBER OF PARENTHESES IN FILL.                            00218110 T  0000
400         MERRIMAC:MISSING FILE ID IN MONITOR DEC.                                 00219000 T  0000
401         MERRIMAC:MISSING LEFT PARENTHESIS IN MONITOR DEC.                        00220000 T  0000
402         MERRIMAC:IMPROPER SUBSCRIPT FOR MONITOR LIST ELEMENT.                    00221000 T  0000
403         MERRIMAC:IMPROPER SUBSCRIPT EXPRESSION DELIMITER IN                      00222000 T  0000
                MONITOR LIST ELEMENT.                                                00223000 T  0000
404         MERRIMAC:IMPROPER NUMBER OF SUBSCRIPTS IN MONITOR LIST                   00224000 T  0000
                ELEMENT.                                                             00225000 T  0000
405         MERRIMAC:LABEL OR SWITCH MONITORED AT IMPROPER LAVEL.                    00226000 T  0000
406         MERRIMAC:IMPROPER MONITOR LIST ELEMENT.                                  00227000 T  0000
407         MERRIMAC:MISSING RIGHT PARENTHESIS IN MONITOR DECLARATION.               00228000 T  0000
408         MERRIMAC:IMPROPER MONITOR DECLARATION DELIMITER.                         00229000 T  0000
409         DMUP:MISSING FILE IDENTIFIER IN DUMP DECLARATION.                        00230000 T  0000
410         DMUP:MISSING LEFT PARENTHESIS IN DUMP DECLARATION.                       00231000 T  0000
411         DMUP:SUBSCRIPTED VARIABLE IN DUMP LIST HAS WRONG NUMBER OF               00232000 T  0000
                SUBSCRIPTS.                                                          00233000 T  0000
412         DMUP:SUBSCRIPTED VARIABLE IN DUMP LIST HAS WRONG NUMBER OF               00234000 T  0000
                SUBSCRIPTS.                                                          00235000 T  0000
413         DMUP:IMPROPER ARRAY DUMP LIST ELEMENT.                                   00236000 T  0000
414         DMUP:ILLEGAL DUMP LIST ELEMENT.                                          00237000 T  0000
415         DMUP:MORE THAN 100 LABELS APPEAR AS DUMP LIST ELEMENTS                   00238000 T  0000
                IN ONE DUMP DECLARATION.                                             00239000 T  0000
416         DMUP:ILLEGAL DUMP LIST ELEMENT DELIMITER.                                00240000 T  0000
417         DMUP:MISSING DUMP LABEL IN DUMP DECLARATION.                             00241000 T  0000
418         DMUP:MISSING COLON IN DUMP DECLARATION.                                  00242000 T  0000
```

```
419         DMUP:IMPROPER DUMP DECLARATION DELIMITER.                                  00243000 T  0000
420         READSTMT:MISSING LEFT PARENTHESIS IN READ STATEMENT.                       00244000 T  0000
421         READSTMT:MISSING LEFT PARENTHESIS IN READ REVERSE                          00245000 T  0000
                 STATEMENT.                                                            00246000 T  0000
422         READSTMT:MISSING FILE IN READ STATEMENT.                                   00247000 T  0000
423         READSTMT:IMPROPER RELEASE INDICATOR.                                       00248000 T  0000
424         READSTMT:IMPROPER FILE DELIMITER IN READ STATEMENT                         00249000 T  0000
425         READSTMT:IMPROPER FORMAT DELIMITER IN READ STATEMENT.                      00250000 T  0000
426         READSTMT:IMPROPER DELIMITER FOR SECOND PARAMETER IN READ                   00251000 T  0000
                 STATEMENT.                                                            00252000 T  0000
427         READSTMT:IMPROPER ROW DESIGNATOR IN READ STATEMENT.                        00253000 T  0000
428         READSTMT:IMPROPER ROW DESIGNATOR DELIMITER IN READ                         00254000 T  0000
                 STATEMENT.                                                            00255000 T  0000
429         READSTMT:MISSING ROW DESIGNATOR IN READ STATEMENT.                         00256000 T  0000
430         READSTMT:IMPROPER DELIMITER PRECEEDING THE LIST IN A READ                  00257000 T  0000
                 STATEMENT.                                                            00258000 T  0000
431         HANDLETHETAILENDOFAREADORSPACESTATEMENT:IMPROPER END OF                     00259000 T  0000
                 FILE LABEL IN READ OR SPACE STATEMENT.                                00260000 T  0000
432         HANDLETHETAILENDOFAREADORSPACESTATEMENT:IMPROPER PARITY                     00261000 T  0000
                 LABEL IN READ OR SPACE STATEMENT.                                     00262000 T  0000
433         HANDLETHETAILENDOFAREADORSPACESTATEMENT:MISSING RIGHT                       00263000 T  0000
                 BRACKET IN READ OR SPACE STATEMENT.                                   00264000 T  0000
434         SPACESTMT:MISSING LEFT PARENTHESIS IN SPACE STATEMENT.                     00265000 T  0000
435         SPACESTMT:IMPROPER FILE IDENTIFIER IN SPACE STATEMENT.                     00266000 T  0000
436         SPACESTMT:MISSING COMMA IN SPACE STATEMENT.                                00267000 T  0000
437         SPACESTMT:MISSING RIGHT PARENTHESIS IN SPACE STATEMENT.                    00268000 T  0000
438         WRITESTMT:MISSING LEFT PARENTHESIS IN A WRITE STATEMENT.                   00269000 T  0000
439         WRITESTMT:IMPROPER FILE IDENTIFIER IN A WRITE STATEMENT.                   00270000 T  0000
440         WRITESTMT:IMPROPER DELIMITER FOR FIRST PARAMETER IN A                      00271000 T  0000
                 WRITE STATEMENT.                                                      00272000 T  0000
441         WRITESTMT:MISSING RIGHT BRACKET IN CARRIAGE CONTROL PART                   00273000 T  0000
                 OF A WRITE STATEMENT.                                                 00274000 T  0000
442         WRITESTMT:ILLEGAL CARRIAGE CONTROL DELIMITER IN A WRITE                    00275000 T  0000
                 STATEMENT.                                                            00276000 T  0000
443         WRITESTMT:IMPROPER SECOND PARAMETER DELIMITER IN WRITE                     00277000 T  0000
                 STATEMENT.                                                            00278000 T  0000
444         WRITESTMT:IMPROPER ROW DESIGNATOR IN A WRITE STATEMENT.                    00279000 T  0000
445         WRITESTMT:MISSING RIGHT PARENTHESIS AFTER A ROW DESIGNATOR                 00280000 T  0000
                 IN A WRITE STATEMENT.                                                 00281000 T  0000
446         WRITESTMT:MISSING ROW DESIGNATOR IN A WRITE STATEMENT.                     00282000 T  0000
447         WRITESTMT:IMPROPER DELIMITER PRECEEDING A LIST IN A WRITE                  00283000 T  0000
                 STATEMENT.                                                            00284000 T  0000
448         WRITESTMT:IMPROPER LIST DELIMITER IN A WRITE STATEMENT.                    00285000 T  0000
449         READSTMT:IMPROPER LIST DELIMITER IN A READ STATEMENT.                      00286000 T  0000
450         LOCKSTMT:MISSING LEFT PARENTHESIS IN A LOCK STATEMENT.                     00287000 T  0000
451         LOCKSTMT:IMPROPER FILE PART IN A LOCK STATEMENT.                           00288000 T  0000
452         LOCKSTMT:MISSING COMMA IN A LOCK STATEMENT.                                00289000 T  0000
453         LOCKSTMT:IMPROPER UNIT DISPOSITION PART IN A LOCK                          00290000 T  0000
                 STATEMENT.                                                            00291000 T  0000
454         LOCKSTMT:MISSING RIGHT PARENTHESIS IN A LOCK STATEMENT.                    00292000 T  0000
455         CLOSESTMT:MISSING LEFT PARENTHESIS IN A CLOSE STATEMENT.                   00293000 T  0000
456         CLOSESTMT:IMPROPER FILE PART IN A CLOSE STATEMENT.                         00294000 T  0000
457         CLOSESTMT:MISSING COMMA IN A CLOSE STATEMENT.                              00295000 T  0000
458         CLOSESTMT:IMPROPER UNIT DISPOSITION PART IN A CLOSE                        00296000 T  0000
                 STATEMENT.                                                            00297000 T  0000
459         CLOSESTMT:MISSING RIGHT PARENTHESIS IN A CLOSE STATEMENT.                  00298000 T  0000
460         RWNDSTMT:MISSING LEFT PARENTHESIS IN A REWIND STATEMENT.                   00299000 T  0000
```

```
                461          RWNDSTMT:IMPROPER FILE PART IN A REWIND STATEMENT.                    00300000 T  0000
                462          RWNDSTMT:MISSING RIGHT PARENTHESIS IN A REWIND STATEMENT.            00301000 T  0000
                463          BLOCK:A MONITOR DECLARATION APPEARS IN THE SPECIFICATION             00302000 T  0000
                                  PART OF A PROCEDURE.                                            00303000 T  0000
                464          BLOCK:A DUMP DECLARATION APPEARS IN THE SPECIFICATION PART           00304000 T  0000
                                  OF A PROCEDURE.                                                 00305000 T  0000
                465          INLINE: MISSING PARAMETER IDENTIFIER IN INLINE STREAM                00305001 T  0000
                                  STATEMENT PARAMETER LIST.                                       00305002 T  0000
               500           .ID:  NEEDS DOUBLE PERIOD FOR PRTE IF PAST 512                       00305100 T  0000
                520      TABLE: STRING LONGER THAN ONE WORD (48 BITS).                            00306200 T  0000
                521      TABLE: STRING CONTAINS A NON-PERMISSIBLE CHARACTER.                      00306300 T  0000
                600      DOLLARCARD: NUMBER EXPECTED.                                             00400000 T  0000
                601      DOLLARCARD: OPTION IDENTIFIER EXPECTED.                                  00401000 T  0000
                602      DOLLARCARD: TOO MANY USER-DEFINED OPTIONS.                               00403000 T  0000
                603      DOLLARCARD: UNRECOGNIZED WORD OR CHARACTER.                              00404000 T  0000
                604      DOLLARCARD: MISMATCHED PARENTHESES.                                      00405000 T  0000
                605      DOLLARCARD:  $ IN CARD COLUMN 1 FOR OMIT CARD                            00406000 T  0000
                610      READACARD: SEQUENCE ERROR.                                               00410000 T  0000
                611      READACARD: ERROR LIMIT HAS BEEN EXCEEDED.                                00411000 T  0000
                ;                                                                                 00490000 T  0000
               BEGIN COMMENT OUTERMOST BLOCK;                                                     00500000 T  0000
PRT(22) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
PRT(23) = *OUTER BLOCK DESCRIPTOR*
PRT(24) = *SEGMENT DESCRIPTOR*
                                                                  START OF SEGMENT **********     2
          INTEGER ERRORCOUNT; COMMENT NUMBER OF ERROR MSGS, MCP WILL TYPE                         00501000 T  0000
PRT(25) = ERRORCOUNT
                        SYNTX ERR AT EOJ IF THIS IS NON-ZERO, MUST BE @R+25;                      00502000 T  0000
          INTEGER SAVETIME; COMMENT SAVE-FACTOR FOR CODE FILE, GIVEN BY MCP.                      00503000 T  0000
PRT(26) = SAVETIME
                              IF COMPILE & GO =0, FOR SYNTAX, =-1. MUST BE AT R+26;               00504000 T  0000
          INTEGER CARDNUMBER; % SEQ # OF CARD BEING PROCESSED.                                    00504100 T  0000
PRT(27) = CARDNUMBER
          INTEGER CARDCOUNT;    % NUMBER OF CARDS PROCESSED.                                     00504150 T  0000
PRT(30) = CARDCOUNT
           BOOLEAN BUILDLINE;                                                                     00504700 T  0000
PRT(31) = BUILDLINE
              COMMENT RR1-RR11 ARE USED IN SOME PROCEDURES IN                                     00505000 T  0000
              PLACE OF LOCALS TO SAVE STACK SPACE;                                                00506000 T  0000
          REAL RR1,RR2,RR3,RR4,RR5,RR6,RR7,RR8,RR9,RR10,RR11;                                     00507000 T  0000
PRT(32) = RR1
PRT(33) = RR2
PRT(34) = RR3
PRT(35) = RR4
PRT(36) = RR5
PRT(37) = RR6
PRT(40) = RR7
PRT(41) = RR8
PRT(42) = RR9
PRT(43) = RR10
PRT(44) = RR11
              COMMENT SOME OF THE RRI ARE USED TO PASS FILE INFORMATION                           00508000 T  0000
                   TO THE MAIN BLOCK;                                                             00509000 T  0000
              COMMENT EXAMIN RETURNS THE CHARACTER AT ABSOLUTE ADDRESS NCR;                       00510000 T  0000
          REAL STREAM PROCEDURE EXAMIN(NCR); VALUE NCR;                                           00511000 T  0000
PRT(45) = EXAMIN
                    BEGIN SI=NCR;DI=LOC EXAMIN;DI=DI+7; DS=CHR END;                               00512000 T  0000
```

```
                INTEGER STREAM PROCEDURE GETF(Q);VALUE Q;                                   00523000 T  0002
PRT(46) = GETF
                    BEGIN SI←LOC GETF; SI←SI-7;DI←LOC Q;DI←DI+5;                             00524000 T  0002
                        SKIP 3 DB; 9(IF SB THEN DS←SET ELSE DS←RESET; SKIP SB);             00525000 T  0003
                        DI←LOC Q;SI←Q;DS←WDS;SI←Q;GETF←SI                                   00526000 T  0005
                    END GETF;                                                               00527000 T  0006



                COMMENT START SETTING UP FILE PARAMETERS;                                   00528000 T  0007
                        IF EXAMIN(RR11←GETF(3)←"Y08") ≠12 THEN RR1←5 ELSE                   00529000 T  0007
                        BEGIN RR1←2;RR2←150 END;                                            00530000 T  0013
                        IF EXAMIN(RR11←5) ≠12 THEN RR3←4 ELSE                               00531000 T  0016
                        BEGIN RR3←2; RR4←150 END;                                           00532000 T  0020
                        IF EXAMIN(RR11←10)=12 THEN                                          00533000 T  0022
                            BEGIN RR5←2;RR6←10;RR7←150 END ELSE                             00534000 T  0024
                        BEGIN RR5←1;RR6←56;RR7←10 END;                                      00535000 T  0027
                        IF EXAMIN(RR11←15)=12 THEN                                          00536000 T  0029
                            BEGIN RR8←10;RR9←150 END ELSE                                   00537000 T  0032
                            BEGIN RR8←56;RR9←10 END;                                        00538000 T  0034
                BEGIN COMMENT MAIN BLOCK;                                                   01000000 T  0036
                INTEGER OPINX; % USED FOR INDEXING INTO OPTIONS ARRAY.                      01000800 T  0036
PRT(47) = *SEGMENT DESCRIPTOR*
                                                                START OF SEGMENT **********     3
PRT(50) = OPINX
                BOOLEAN SETTING; % USED BY DOLLARCARD FOR AN OPTION"S SETTING,               01000802 T  0000
PRT(51) = SETTING
                 INTEGER NEWINX, ADDVALUE, BASENUM, TOTALNO;                                01000860 T  0000
PRT(52) = NEWINX
PRT(53) = ADDVALUE
PRT(54) = BASENUM
PRT(55) = TOTALNO
                DEFINE OPARSIZE = 200 #;                                                    01000902 T  0000
                ARRAY OPTIONS[0:OPARSIZE];                                                  01000904 T  0000
PRT(56) = OPTIONS
                BOOLEAN OPTIONWORD;                                                         01000910 T  0002
PRT(57) = OPTIONWORD
                DEFINE    CHECKBIT        =  1#,                                            01000920 T  0002
                          DEBUGBIT        =  2#,                                            01000930 T  0002
                          DECKBIT         =  3#,                                            01000940 T  0002
                          FORMATBIT       =  4#,                                            01000950 T  0002
                          INTBIT          =  5#,                                            01000960 T  0002
                          LISTABIT        =  6#,                                            01000970 T  0002
                          LISTBIT         =  7#,                                            01000980 T  0002
                          LISTPBIT        =  8#,                                            01000990 T  0002
                          MCPBIT          =  9#,                                            01001000 T  0002
                          MERGEBIT        = 10#,                                            01001010 T  0002
                          NESTBIT         = 11#,                                            01001020 T  0002
                          NEWBIT          = 12#,                                            01001030 T  0002
                          NEWINCLBIT      = 13#,                                            01001040 T  0002
                          OMITBIT         = 14#,                                            01001050 T  0002
                          PRINTDOLLARBIT  = 15#,                                            01001060 T  0002
```

```
                PRTBIT          = 16#,                                          01001070 T  0002
                PUNCHBIT        = 17#,                                          01001080 T  0002
                PURGEBIT        = 18#,                                          01001090 T  0002
                SEGSBIT         = 19#,                                          01001100 T  0002
                SEQBIT          = 20#,                                          01001110 T  0002
                SEQERRBIT       = 21#,                                          01001120 T  0002
                SINGLBIT        = 22#,                                          01001130 T  0002
                STUFFBIT        = 23#,                                          01001140 T  0002
                VOIDBIT         = 24#,                                          01001150 T  0002
                VOIDTBIT        = 25#,                                          01001160 T  0002
                USEROPINX       = 26#;                                          01001170 T  0002
COMMENT         IF A NEW COMPILER-DEFINED OPTION IS ADDED, CHANGE USEROPINX     01001180 T  0002
                AND ADD OPTION IN DEFINES BELOW, IN DOLLARCARD, AND IN          01001190 T  0002
                FILL STATEMENT IN INITIALIZATION OF COMPILER;                   01001200 T  0002
DEFINE          CHECKTOG        = OPTIONWORD.[CHECKBIT:1] #,                    01001210 T  0002
                DEBUGTOG        = OPTIONWORD.[DEBUGBIT:1] #,                    01001220 T  0002
                DECKTOG         = OPTIONWORD.[DECKBIT:1] #,                     01001230 T  0002
                FORMATOG        = OPTIONWORD.[FORMATBIT:1] #,                   01001240 T  0002
                INTOG           = OPTIONWORD.[INTBIT:1] #,                      01001250 T  0002
                LISTATOG        = OPTIONWORD.[LISTABIT:1] #,                    01001260 T  0002
                LISTOG          = OPTIONWORD.[LISTBIT:1] #,                     01001270 T  0002
                LISTPTOG        = OPTIONWORD.[LISTPBIT:1] #,                    01001280 T  0002
                MCPTOG          = OPTIONWORD.[MCPBIT:1] #,                      01001290 T  0002
                MERGETOG        = OPTIONWORD.[MERGEBIT:1] #,                    01001300 T  0002
                NESTOG          = OPTIONWORD.[NESTBIT:1] #,                     01001310 T  0002
                NEWTOG          = OPTIONWORD.[NEWBIT:1] #,                      01001320 T  0002
                NEWINCL         = OPTIONWORD.[NEWINCLBIT:1] #,                  01001330 T  0002
                OMITTING        = OPTIONWORD.[OMITBIT:1] #,                     01001340 T  0002
                PRINTDOLLARTOG  = OPTIONWORD.[PRINTDOLLARBIT:1] #,              01001350 T  0002
                PRTOG           = OPTIONWORD.[PRTBIT:1] #,                      01001360 T  0002
                PUNCHTOG        = OPTIONWORD.[PUNCHBIT:1] #,                    01001370 T  0002
                PURGETOG        = OPTIONWORD.[PURGEBIT:1] #,                    01001380 T  0002
                SEGSTOG         = OPTIONWORD.[SEGSBIT:1] #,                     01001390 T  0002
                SEQTOG          = OPTIONWORD.[SEQBIT:1] #,                      01001400 T  0002
        COMMENT SEQTOG INDICATES RESEQUENCING IS TO BE DONE;                    01001410 T  0002
                SEQERRTOG       = OPTIONWORD.[SEQERRBIT:1] #,                   01001420 T  0002
                SINGLTOG        = OPTIONWORD.[SINGLBIT:1] #,                    01001430 T  0002
                STUFFTOG        = OPTIONWORD.[STUFFBIT:1] #,                    01001440 T  0002
                VOIDING         = OPTIONWORD.[VOIDBIT:1] #,                     01001450 T  0002
                VOIDTAPE        = OPTIONWORD.[VOIDTBIT:1] #,                    01001460 T  0002
                DUMMY           = #;                                            01001470 T  0002
        BOOLEAN NOHEADING;      % TRUE IF DATIME HAS NOT BEEN CALLED.           01001480 T  0002
PRT(60) = NOHEADING
        BOOLEAN NEWBASE;        % NEW BASENUM FOUND ON A NEW $-CARD.            01001490 T  0002
PRT(61) = NEWBASE
        BOOLEAN LASTCRDPATCH;   % NORMALLY FALSE, SET TO TRUE WHEN THE          01001500 T  0002
PRT(62) = LASTCRDPATCH
                                % LAST CARD FROM SYMBOLIC LIBRARY READ          01001510 T  0002
                                % IS PATCHED FROM THE CARD READER.             01001520 T  0002
        INTEGER XMODE;          % TELLS DOLLARCARD HOW TO SET OPTIONS.          01001530 T  0002
PRT(63) = XMODE
        BOOLEAN DOLLARTOG;      % TRUE IF SCANNING A DOLLAR CARD.               01001540 T  0002
PRT(64) = DOLLARTOG
        INTEGER ERRMAX;         % COMPILATION STOPS IF EXCEEDED.                01001550 T  0002
PRT(65) = ERRMAX
        BOOLEAN SEQXEQTOG;      % GIVE SEQ. NO. WHEN DS-ING OBJ.                01001560 T  0002
PRT(66) = SEQXEQTOG
```

```
                BOOLEAN LISTER;         % LISTOG OR LISTATOG OR DEBUGTOG.
PRT(67) = LISTER
                ALPHA MEDIUM;           % INPUT IS: T,C,P,CA,CB,CC.
PRT(70) = MEDIUM
                INTEGER MYCLASS;        % USED IN DOLLARCARD EVALUATION.
PRT(71) = MYCLASS
                REAL BATMAN;            % USED IN DOLLARCARD EVALUATION.
PRT(72) = BATMAN
                 ARRAY SPECIAL[0:31];
PRT(73) = SPECIAL
                    COMMENT THIS ARRAY HOLDS THE INTERNAL CODE FOR THE SPECIAL
                            CHARACTORS: IT IS FILLED DURING INITIALIZATION;

                ARRAY INFO [0:127,0:255];
PRT(74) = INFO
                    COMMENT INFO CONTAINS ALL THE INFORMATION ABOUT A GIVEN IDENTIFIER
                            OR RESERVED WORD.  THE FIRST WORD OF A GIVEN ENTRY IS
                            THE INTERNAL CODE ( OR ELBAT WORD AS IT IS USUALLY
                            CALLED).  THE SECOND WORD CONTAINS THE FORWARD BIT (IN
                            [1:1]) FOR PROCEDURES, THE LINK TO PREVIOUS ENTRY (IN
                            [4:8]), THE NUMBER OF CHARACTORS IN THE ALPHA REPRESENTA-
                            TION (IN [12:6]), AND THE FIRST 5 CHARACTORS OF ALPHA.
                            SUCCEDING WORDS CONTAIN THE REMAINING CHARACTORS OF ALPHA,
                            FOLLOWED BY ANY ADDITIONAL INFORMATION. THE ELBAT WORD
                            AND THE ALPHA FOR ANY QUANTITY ARE NOT SPLIT ACROSS A ROW
                            OF INFO.  FOR PURPOSES OF FINDING AN IDENTIFIER OR
                            RESERVED WORD THE QUANTITIES ARE SCATTERED INTO 125
                            DIFERENT LISTS OR STACKES, WHICH STACK CONTAINS A QUANTITY
                            IS GIVEN BY TAKING NAAAAA MOD 125 WHERE N IS THE NUMBER
                            OF CHARACTORS AND AAAAA IS THE FIRST 5 CHARACTORS OF
                            ALPHA, FILLED IN WITH ZEROS FROM THE RIGHT IF NEEDED.
                            THIS NUMBER IS CALLED THE SCRAMBLE NUMBER OR INDEX.
                            THE FIRST ROW OF INFO IS USED FOR OTHER PURPOSES.  THE
                            RESERVED WORDS OCCUPY THE SECOND ROW. IT IS FILLED DURING
                            INITIALIZATION;
COMMENT                         INFO FORMAT
  FOLLOWING IS A DESCRIPTION OF THE FORMAT OF ALL TYPES OF ENTRIES
  ENTERED IN INFO:
     THE FIRST WORD OF ALL ENTRIES IS THE ELBAT WORD.
     THE INCR FIELD ([27:8]) CONTAINS AN INCREMENT WHICH WHEN
     ADDED TO THE CURRENT INDEX INTO INFO YELDSAN INDEX TO ANY
     ADDITIONAL INFO (IF ANY) FOR THIS ENTRY.
      E.G. IF THE INDEX IS  IX  THEN  INFO[(IX+INCR),LINKR,(IX+INCR),
          LINKC] WILL CONTAIN THE FIRST WORD OF ADDITIONAL INFO.
     THE LINK FIELD OF THE ELBAT WORD IN INFO IS DIFFERENT FROM
     THAT OF THE ENTRY IN ELBAT PUT IN BY TABLE,THE ENTRY IN ELBAT
     POINTS TO ITS OWN LOCATION (RELATIVE) IN INFO.
     THE LINK IN INFO POINTS TO THE PREVIOUS ENTRY E.G.,THE
     LINK FROM STACKHEAD WHICH THE CURRENT ENTRY REPLACED.
     FOR SIMPLICITY,I WILL CONSIDER INFO TO BE A ONE DIMENSIONAL
     ARRAY,SO THAT THE BREAKING UP OF THE LINKS INTO ROW AND COLUMN
     WILL NOT DETRACT FROM THE DISCUSSION.
        ASSUME THAT THREE IDENTIFIERS A,B,AND C "SCRAMBLE" INTO
     THE SAME STACKHEAD LOCATION IN THE ORDER OF APPEARANCE.
       FURTHER ASSUME THERE ARE NO OTHER ENTRIES CONNECTED TO
     THIS STACKHEAD INDEX. LET THIS STACKHEAD LOCATION BE
                          S[L]
```

```
01001570 T  0002
01001580 T  0002
01001590 T  0002
01001600 T  0002
01003000 T  0002
01004000 T  0005
01005000 T  0005
01006000 T  0005
01007000 T  0005
01008000 T  0007
01009000 T  0007
01010000 T  0007
01011000 T  0007
01012000 T  0007
01013000 T  0007
01014000 T  0007
01015000 T  0007
01016000 T  0007
01017000 T  0007
01018000 T  0007
01019000 T  0007
01020000 T  0007
01021000 T  0007
01022000 T  0007
01023000 T  0007
01024000 T  0007
01025000 T  0007
01026000 T  0007
01027000 T  0007
01028000 T  0007
01029000 T  0007
01030000 T  0007
01031000 T  0007
01032000 T  0007
01033000 T  0007
01034000 T  0007
01035000 T  0007
01036000 T  0007
01037000 T  0007
01038000 T  0007
01039000 T  0007
01040000 T  0007
01041000 T  0007
01042000 T  0007
01043000 T  0007
01044000 T  0007
01045000 T  0007
01046000 T  0007
01047000 T  0007
01048000 T  0007
01049000 T  0007
```

```
NOW THE DECLARATION
     BEGIN REAL A,B,C  IS ENCOUNTERED
    IF THE NEXT AVAILABLE INFO SPACE IS CALLED NEXTINFO
 THEN  A IS ENTERED AS FOLLOWS:(ASSUME AN ELBAT WORD T HAS BEEN
                                CONSTRUCTED FOR A)
                       T.LINK← S[L], (WHICH IS ZERO AT FIRST).
INFO[NEXTINFO]←T.      S[L]←NEXTINFO.
                       NEXTINFO←NEXTINFO+NUMBER OF WORDS IN THIS
                                ENTRY.
   NOW  S[L] POINTS TO THE ENTRY FOR A IN INFO AND THE ENTRY
            ITSELF CONTAINS THE STOP FLAG ZERO.
  B IS ENTERED SIMILARLY TO A.
   NOW  S[L] POINTS TO THE ENTRY FOR B AND IT POINTS TO THE
                       ENTRY FOR A.
  SIMILARLY,AFTER C IS ENTERED
            S[L] POINTS TO C,WHOSE ENTRY POINTS TO B WHOSE ENTRY
              POINTS TO A.
THE SECONB WORD OF EACH ENTRY IN INFO IS MADE UP AS FOLLOWS:
   FWDPT =[1:1],THIS TELLS WHETHER A PROCEDURE WAS DECLARED
               FORWARD,IT IS RESET AT THE TIME OF ITS ACTUAL
               FULL DECLARATION.
   PURPT =[4:8] THIS GIVES A DECREMENT WHICH GIVES THE RELATIVE
               INDEX TO THE PREVIOUS INFO ENTRY WHEN SUBTRACTED
               FROM THE CURRENT ENTRY INDEX.
  [12:6] TELLS THE NUMBER OF CHARACTERS IN THE ENTRY.(<64)
  [18:30] CONTAINS THE FIRST FIVE ALPHA CHARACTERS OF THE ENTRY
         AND SUCCEEDING WORDS CONTAIN ALL OVERFLOW IF NEEDED.
         THESE WORDS CONTAIN 8 CHARACTERS EACH,LEFT JUSTIFIED.
        THUS,AN ENTRY FOR  SYMBOL FOLLOWED BY AN ENTRY
                       FOR  X      WOULD APPEAR AS FOLLOWS:
    INFO[I] =      ELBATWRD      (MADE FOR SYMBOL)
        I+1 =      OP6SYMBO     (P DEPENDS ON PREVIOUS ENTRY)
        I+2 =      L
        I+3 =      ELBATWRD      (MADE FOR X)
        I+4 =      031X
      THIS SHOWS THAT INFO[I-P] WOULD POINT TO THE BEGINNING OF
      THE ENTRY BEFORE SYMBOL, AND
      INFO[I+3-3] POINTS TO THE ENTRY FOR SYMBOL.
ALL ENTRIES OF IDNETIFIERS HAVE THE INFORMATION DESCRIBED ABOVE
THAT IS,THE ELBAT WORD FOLLOWED BY THE WORD CONTAING THE FIRST
FIVE CHARACTERS OF ALPHA,AND ANY ADDITIONAL WORDS OF ALPHA IF
 NECESSARY.
THIS IS SUFFICIENT FOR ENTRIES OF THE FOLLOWING TYPES,
       REAL
       BOOLEAN
       INTEGER
       ALPHA
       FILE
       FORMAT
       LIST
   OTHER ENTRIES REQUIRE ADDITIONAL INFORMATION.
ARRAYS:
    THE FIRST WORD OF ADDITIONAL INFO CONTAINS THE NUMBER OF
    DIMENSIONS(IN THE LOW ORDER PART).[40:8]
   EACH SUCCEEDING WORD CONTAINS INFORMATION ABOUT EACH LOWER
   BOUND IN ORDER OF APPEARANCE,ONE WORD FOR EACH LOWER BOUND.
   THESE WORDS ARE MADE UP AS FOLLOWS:
```

```
01050000 T  0007
01051000 T  0007
01052000 T  0007
01053000 T  0007
01054000 T  0007
01055000 T  0007
01056000 T  0007
01057000 T  0007
01058000 T  0007
01059000 T  0007
01060000 T  0007
01061000 T  0007
01062000 T  0007
01063000 T  0007
01064000 T  0007
01065000 T  0007
01066000 T  0007
01067000 T  0007
01068000 T  0007
01069000 T  0007
01070000 T  0007
01071000 T  0007
01072000 T  0007
01073000 T  0007
01074000 T  0007
01075000 T  0007
01076000 T  0007
01077000 T  0007
01078000 T  0007
01079000 T  0007
01080000 T  0007
01081000 T  0007
01082000 T  0007
01083000 T  0007
01084000 T  0007
01085000 T  0007
01086000 T  0007
01087000 T  0007
01088000 T  0007
01089000 T  0007
01090000 T  0007
01091000 T  0007
01092000 T  0007
01093000 T  0007
01094000 T  0007
01095000 T  0007
01096000 T  0007
01097000 T  0007
01098000 T  0007
01099000 T  0007
01100000 T  0007
01101000 T  0007
01102000 T  0007
01103000 T  0007
01104000 T  0007
01105000 T  0007
01106000 T  0007
```

```
          [23:12] =ADD OPERATOR SYLLABLE (0101) OR                                  01107000 T  0007
                   SUB OPERATOR SYLLABLE (0301) CORRESPONDING                       01108000 T  0007
                   RESPECTIVELY TO WHETHER THE LOWER BOUND IS                       01109000 T  0007
                   TO BE ADDED TO THE SUBSCRIPT IN INDEXING OR                      01110000 T  0007
                   SUBTRACTED.                                                      01111000 T  0007
          [35:11] =11 BIT ADDRESS OF LOWER BOUND,IF THE LOWER BOUND                 01112000 T  0007
                   REQUIRES A PRT OR STACK CELL,OTHERWISE THE BIT                   01113000 T  0007
                   35 IS IGNORED AND THE NEXT TEN BITS([36:10])                     01114000 T  0007
                   REPRESENT THE ACTUAL VALUE OF THE LOWER BOUND                    01115000 T  0007
          [46:2] =00 OR 10  DEPENDING ON WHETHER THE [35:11] VALUE                  01116000 T  0007
                   IS A LITERAL OR OPERAND,RESPECTIVELY.                            01117000 T  0007
PROCEDURES:                                                                         01118000 T  0007
     THE FIRST WORD OF ADDITIONAL INFO CONTAINS THE NUMBER OF                       01119000 T  0007
     PARAMETERS [40:8]                                                              01120000 T  0007
     IF A STREAM PROCEDURE THEN THIS WORD CONTAINS ALSO IN                          01121000 T  0007
     [13:11] ENDING  PRT ADDRESS FOR LABELS,                                        01122000 T  0007
     [ 7:6]  NO OF LABELS REQUIRING PRT ADDRESSES, AND [1:6] NUMBER                 01123000 T  0007
     OF LOCALS.                                                                     01124000 T  0007
     SUCCEEDING WORDS (ONE FOR EACH FORMAL PARAMETER,IN ORDER                       01125000 T  0007
                       OF APPEARANCE IN FORMAL PARAPART) ARE                        01126000 T  0007
     ELBAT WORDS SPECIFYING TYPE OF EACH PARAMETER AND WHETHER                      01127000 T  0007
     VALUE OR NOT([10:1]).                                                          01128000 T  0007
     THE ADDRESS([16:11]) IS THE F= ADDRESS FOR EACH.                               01129000 T  0007
     IF THE PARAMETER IS AN ARRAY THEN THE INCR FIELD([27:8])                       01130000 T  0007
     CONTAINS THE NUMBER OF DIMENSIONS,OTHERWISE INCR IS MEANINGLESS.               01131000 T  0007
     LINK([35:13]) IS MEANINGLESS.                                                  01132000 T  0007
     IF A STREAM PROCEDURE THEN THE CLASS OF EACH PARAMATER IS                      01133000 T  0007
     THAT OF LOCAL ID OR FILE ID, DEPENDING ON WHETHER OR NOT A RELEASE             01134000 T  0007
     IS DONE IN THE STREAM PROCEDURE.                                               01135000 T  0007
LABELS:                                                                             01136000 T  0007
     AT DECLARATION TIME THE ADDITIONAL INFO CONTAINS 0.  THE SIGN                  01137000 T  0007
     BIT TELLS WHETHER OR NOT THE DEFINITION POINT HAS BEEN REACHED.                01138000 T  0007
     IF SIGN = 0, THEN [36:12] CONTAINS AN ADDRESS IN CODEARRAY OF A                01139000 T  0007
     LIST OF FORWARD REFERENCES TO THIS LABEL.  THE END OF LIST FLAG IS             01140000 T  0007
     0.  IF SIGN =0, THEN [36:12] CONTAINS L FOR THIS LABEL.                        01141000 T  0007
SWITCHES:                                                                           01142000 T  0007
     THE FIELD [36:12] CONTAINS L FOR THE BEGINNING OF SWITCH DECLAR-               01143000 T  0007
     ATION.  [24:12] CONTAINS L FOR FIRST SIMPLE REFERENCE TO SWITCH.               01144000 T  0007
     IF SWITCH IS NOT SIMPLE, IT IS MARKED FORMAL.  HERE SIMPLE MEANS               01145000 T  0007
     NO POSSIBILITY OF JUMPING OUT OF A BLOCK.                                  ;   01146000 T  0007
          DEFINE MON    =[ 1: 1]#,                                                01147000 T  0007
                 CLASS  =[ 2: 7]#,                                                01148000 T  0007
                 FORMAL=[ 9: 1]#,                                                 01149000 T  0007
                 VO     =[10: 1]#,                                                01150000 T  0007
                 LVL    =[11: 5]#,                                                01151000 T  0007
                ADDRESS=[16:11]#,                                                 01152000 T  0007
                 INCR   =[27: 8]#,                                                01153000 T  0007
                 LINK   =[35:13]#,                                                01154000 T  0007
                 LINKR  =[35: 5]#,                                                01155000 T  0007
                 LINKC  =[40: 8]#;                                                01156000 T  0007
COMMENT THESE DEFINES ARE USED TO PICK APART THE ELBAT WORD.                        01157000 T  0007
          MON IS THE BIT WHICH IS ON IF THE QUANTITY IS MONITORED.                  01158000 T  0007
          CLASS IS THE PRINCIPAL IDENTIFICATION OF A GIVEN                          01159000 T  0007
              QUANTITY.                                                             01160000 T  0007
          FORMAL IS THE BIT WHICH IS ON IF THE QUANTITY IS A FORMAL                 01161000 T  0007
              PARAMETER.                                                            01162000 T  0007
          VO IS THE VALUE-OWN BIT. IF FORMAL = 1 THEN THE BIT                       01163000 T  0007
```

```
                DISTINGUISHES VALUE PARAMETERS FROM OTHERS. IF                  01164000 T  0007
                FORMAL = 0 THEN THE BIT DISTINGUISHES OWN VARIABLES             01165000 T  0007
                FROM OTHERS.                                                    01166000 T  0007
            LVL GIVES THE LEVEL AT WHICH A QUANTITY WAS DECLARED.               01167000 T  0007
            ADDRESS GIVES THE STACK OR PRT ADDRESS.                             01168000 T  0007
            INCR GIVES A RELATIVE LINK TO ANY ADDITIONAL INFORMATION            01169000 T  0007
                NEEDED, RELATIVE TO THE LOCATION IN INFO.                       01170000 T  0007
            LINK CONTAINS A LINK TO THE LOCATION IN INFO IF THE                 01171000 T  0007
                QUANTITY LIES IN ELBAT, OTHERWISE IT LINKS TO THE               01172000 T  0007
                NEXT ITEM IN THE STACK. ZERO IS AN END FLAG.                    01173000 T  0007
            LINKR AND LINKC ARE SUBDIVISIONS OF LINK.;                          01174000 T  0007
COMMENT  CLASSES FOR ALL QUANTITIES  -  OCTAL CLASS IS IN COMMENT;              01175000 T  0007
   COMMENT   CLASSES FOR IDENTIFIERS;                                           01176000 T  0007
DEFINE  UNKNOWNID       =00#,    COMMENT 000;                                   01177000 T  0007
        STLABID         =01#,    COMMENT 001;                                   01178000 T  0007
        LOCLID          =02#,    COMMENT 002;                                   01179000 T  0007
        DEFINEDID       =03#,    COMMENT 003;                                   01180000 T  0007
        LISTID          =04#,    COMMENT 004;                                   01181000 T  0007
        FRMTID          =05#,    COMMENT 005;                                   01182000 T  0007
        SUPERFRMTID     =06#,    COMMENT 006;                                   01183000 T  0007
        REALSUBID       =07#,    COMMENT 007;                                   01184000 T  0007
        SUBID           =08#,    COMMENT 010;                                   01185000 T  0007
        SWITCHID        =09#,    COMMENT 011;                                   01186000 T  0007
        PROCID          =10#,    COMMENT 012;                                   01187000 T  0007
        INTRNSICPROCID  =11#,    COMMENT 013;                                   01188000 T  0007
        STRPROCID       =12#,    COMMENT 014;                                   01189000 T  0007
        BOOSTRPROCID    =13#,    COMMENT 015;                                   01190000 T  0007
        REALSTRPROCID   =14#,    COMMENT 016;                                   01191000 T  0007
        ALFASTRPROCID   =15#,    COMMENT 017;                                   01192000 T  0007
        INTSTRPROCID    =15#,    COMMENT 017;                                   01193000 T  0007
        BOOPROCID       =17#,    COMMENT 021;                                   01194000 T  0007
        REALPROCID      =18#,    COMMENT 022;                                   01195000 T  0007
        ALFAPROCID      =19#,    COMMENT 023;                                   01196000 T  0007
        INTPROCID       =19#,    COMMENT 023;                                   01197000 T  0007
        BOOID           =21#,    COMMENT 025;                                   01198000 T  0007
        REALID          =22#,    COMMENT 026;                                   01199000 T  0007
        ALFAID          =23#,    COMMENT 027;                                   01200000 T  0007
        INTID           =23#,    COMMENT 027;                                   01201000 T  0007
        BOOARRAYID      =25#,    COMMENT 031;                                   01202000 T  0007
        REALARRAYID     =26#,    COMMENT 032;                                   01203000 T  0007
        ALFAARRAYID     =27#,    COMMENT 033;                                   01204000 T  0007
        INTARRAYID      =27#,    COMMENT 033;                                   01205000 T  0007
        NAMEID          =30#,    COMMENT 036;                                   01205200 T  0007
        INTNAMEID       =31#,    COMMENT 037;                                   01205400 T  0007
        LABELID         =32#,    COMMENT 040;                                   01206000 T  0007
COMMENT   CLASSES FOR PRIMARY BEGINNERS;                                        01207000 T  0007
        TRUTHV          =33#,    COMMENT 041;                                   01208000 T  0007
        NONLITNO        =34#,    COMMENT 042;                                   01209000 T  0007
        LITNO           =35#,    COMMENT 043;                                   01210000 T  0007
        STRNGCON        =36#,    COMMENT 044;                                   01211000 T  0007
        LEFTPAREN       =37#,    COMMENT 045;                                   01212000 T  0007
        POLISHV         =38#,    COMMENT 046;                                   01212100 T  0007
        ASTRISK         =39#,    COMMENT 047;                                   01212200 T  0007
COMMENT   CLASS FOR ALL DECLARATORS;                                            01213000 T  0007
        DECLARATORS     =40#,    COMMENT 050;                                   01214000 T  0007
COMMENT   CLASSES FOR STATEMENT BEGINNERS                                       01215000 T  0007
        DOUBLEV         =42#,    COMMENT 052;                                   01222000 T  0007
```

```
      FORV              =43#,     COMMENT 053;                                01223000 T  0007
      WHILEV            =44#,     COMMENT 054;                                01224000 T  0007
      DOV               =45#,     COMMENT 055;                                01225000 T  0007
      UNTILV            =46#,     COMMENT 056;                                01226000 T  0007
      ELSEV             =47#,     COMMENT 057;                                01227000 T  0007
      ENDV              =48#,     COMMENT 060;                                01228000 T  0007
      SEMICOLON         =50#,     COMMENT 062;                                01230000 T  0007
      IFV               =51#,     COMMENT 063;                                01231000 T  0007
      GOV               =52#,     COMMENT 064;                                01232000 T  0007
      IOCLASS           =53#,     COMMENT 065;                                01233000 T  0007
      BEGINV            =54#,     COMMENT 066;                                01234000 T  0007
COMMENT   CLASSES FOR STREAM RESERVED WORDS;                                    01235000 T  0007
      SIV               =55#,     COMMENT 067;                                01236000 T  0007
      DIQ               =56#,     COMMENT 070;                                01237000 T  0007
      CIV               =57#,     COMMENT 071;                                01238000 T  0007
      TALLYV            =58#,     COMMENT 072;                                01239000 T  0007
      DSV               =59#,     COMMENT 073;                                01240000 T  0007
      SKIPV             =60#,     COMMENT 074;                                01241000 T  0007
      JUMPV             =61#,     COMMENT 075;                                01242000 T  0007
      DBV               =62#,     COMMENT 076;                                01243000 T  0007
      SBV               =63#,     COMMENT 077;                                01244000 T  0007
      TOGGLEV           =64#,     COMMENT 100;                                01245000 T  0007
      SCV               =65#,     COMMENT 101;                                01246000 T  0007
      LOCV              =66#,     COMMENT 102;                                01247000 T  0007
      DCV               =67#,     COMMENT 103;                                01248000 T  0007
      LOCALV            =68#,     COMMENT 104;                                01249000 T  0007
      LITV              =69#,     COMMENT 105;                                01250000 T  0007
      TRNSFER           =70#,     COMMENT 106;                                01251000 T  0007
COMMENT   CLASSES FOR VARIOUS MISCELLANEOUS QUANTITIES;                         01252000 T  0007
      COMMENTV          =71#,     COMMENT 107;                                01253000 T  0007
      FORWARDV          =72#,     COMMENT 110;                                01254000 T  0007
      STEPV             =73#,     COMMENT 111;                                01255000 T  0007
      THENV             =74#,     COMMENT 112;                                01256000 T  0007
      TOV               =75#,     COMMENT 113;                                01257000 T  0007
      VALUEV            =76#,     COMMENT 114;                                01258000 T  0007
      WITHV             =77#,     COMMENT 115;                                01259000 T  0007
      COLON             =78#,     COMMENT 116;                                01260000 T  0007
      COMMA             =79#,     COMMENT 117;                                01261000 T  0007
      CROSSHATCH        =80#,     COMMENT 120;                                01262000 T  0007
      LFTBRKET          =81#,     COMMENT 121;                                01263000 T  0007
      PERIOD            =82#,     COMMENT 122;                                01264000 T  0007
      RTBRKET           =83#,     COMMENT 123;                                01265000 T  0007
      RTPAREN           =84#,     COMMENT 124;                                01266000 T  0007
      AMPERSAND         =85#,     COMMENT 125;                                01266500 T  0007
COMMENT   CLASSES FOR OPERATORS;                                                01267000 T  0007
      HEXOP             =86#,     COMMENT 126;                                01268000 T  0007
      BITOP             =87#,     COMMENT 127;                                01269000 T  0007
      ISOLATE           =88#,     COMMENT 130;                                01270000 T  0007
      OPERATOR          =89#,     COMMENT 131;                                01271000 T  0007
      NOTOP             =90#,     COMMENT 132;                                01272000 T  0007
      ASSIGNOP          =91#,     COMMENT 133;                                01273000 T  0007
      EQVOP             =92#,     COMMENT 134;                                01274000 T  0007
      OROP              =93#,     COMMENT 135;                                01275000 T  0007
      ANDOP             =94#,     COMMENT 136;                                01276000 T  0007
      RELOP             =95#,     COMMENT 137;                                01277000 T  0007
      ADOP              =96#,     COMMENT 140;                                01278000 T  0007
      MULOP             =97#,     COMMENT 141;                                01278500 T  0007
```

```
%         STRING          =99#,     COMMENT 143;                                  01278600 T  0007
     COMMENT   SUBCLASSES FOR DECLARATORS (KEPT IN ADDRESS);                       01279000 T  0007
          OWNV             =01#,     COMMENT  01;                                 01280000 T  0007
          SAVEV            =02#,     COMMENT  02;                                 01281000 T  0007
          BOOV             =03#,     COMMENT  03;                                 01282000 T  0007
          REALV            =04#,     COMMENT  04;                                 01283000 T  0007
          ALFAV            =05#,     COMMENT  05;                                 01284000 T  0007
          INTV             =05#,     COMMENT  05;                                 01285000 T  0007
          LABELV           =07#,     COMMENT  07;                                 01286000 T  0007
          DUMPV            =08#,     COMMENT  10;                                 01287000 T  0007
          SUBV             =09#,     COMMENT  11;                                 01288000 T  0007
          OUTV             =10#,     COMMENT  12;                                 01289000 T  0007
          INV              =11#,     COMMENT  13;                                 01290000 T  0007
          MONITORV         =12#,     COMMENT  14;                                 01291000 T  0007
          SWITCHV          =13#,     COMMENT  15;                                 01292000 T  0007
          PROCV            =14#,     COMMENT  16;                                 01293000 T  0007
          ARRAYV           =15#,     COMMENT  17;                                 01294000 T  0007
          NAMEV            =16#,     COMMENT  20;                                 01295000 T  0007
          FILEV            =17#,     COMMENT  21;                                 01296000 T  0007
          STREAMV          =18#,     COMMENT  22;                                 01297000 T  0007
          DEFINEV          =19#;     COMMENT  23;                                 01298000 T  0007
DEFINE DDES          =  8#,                                                        01299000 T  0007
       ADES          = 28#,                                                        01299010 T  0007
       PDES          = 29#,                                                        01299020 T  0007
       LDES          = 30#,                                                        01299030 T  0007
       CHAR          = 31#,                                                        01299040 T  0007
       FACTOP        = ASTRISK#,                                                   01299100 T  0007
       OPERATORS     = HEXOP#,                                                     01299200 T  0007
       FILEID        = 0#,                                                         01299300 T  0007
       MAXINTRINSIC  = 150#, % USED IN BUILDING INTABLE @ 09414120                 01299400 T  0007
       INTRINSICADR  = (MAXINTRINSIC DIV 30)#; % RESERVES SEG FOR INTABLE          01299500 T  0007
                                                                                   01300000 T  0007
 REAL TIME1;
PRT(75) = TIME1
                                                                                   01300100 T  0007
 BOOLEAN ASTOG;
PRT(76) = ASTOG
                                                                                   01300200 T  0007
 BOOLEAN SAF;
PRT(77) = SAF
                                                                                   01301000 T  0007
 INTEGER SCRAM;
PRT(100) = SCRAM
    COMMENT SCRAM CONTAINS THE SCRAMBLE INDEX FOR THE LAST IDENTIFIER              01302000 T  0007
            OR RESERVED WORD SCANNED;                                              01303000 T  0007
 ALPHA ARRAY ACCUM[0:10];                                                          01304000 T  0007
PRT(101) = ACCUM
    COMMENT ACCUM HOLDS THE ALPHA AND CHARACTER COUNT OF THE LAST                  01305000 T  0010
            SCANNED ITEM IN A FORM COMPATIBLE WITH ITS APPEARANCE                  01306000 T  0010
            IN INFO, THAT IS ACCUM[1] = 00NAAAAA, ACCUM[I] , I> 1,                 01307000 T  0010
            HAS ANY ADDITIONAL CHARACTERS, ACCUM[0] IS USED FOR                    01308000 T  0010
            THE ELBAT WORD BY THE ENTER ROUTINES;                                  01309000 T  0010
 ARRAY STACKHEAD[0:125];                                                           01310000 T  0010
PRT(102) = STACKHEAD
    COMMENT STACKHEAD[N] CONTAINS AN INDEX INTO INFO GIVING THE TOP                01311000 T  0012
            ITEM IN THE N-TH STACK;                                                01312000 T  0012
 INTEGER COUNT;                                                                    01313000 T  0012
PRT(103) = COUNT
    COMMENT COUNT CONTAINS THE NUMBER OF CHARACTORS OF THE LAST ITEM               01314000 T  0012
            SCANNED;                                                               01315000 T  0012
 ALPHA Q;                                                                          01316000 T  0012
```

```
PRT(104) = Q
                    COMMENT Q CONTAINS ACCUM[1] FOR THE LAST IDENTIFIER OR RESERVED          01317000 T  0012
                            WORD SCANNED;                                                   01318000 T  0012
                  ARRAY ELBAT[0:75]; INTEGER I, NXTELBT;                                    01319000 T  0012
PRT(105) = ELBAT
PRT(106) = I
PRT(107) = NXTELBT
                    COMMENT ELBAT IS AN ARRAY HOLDING ELBAT WORDS FOR RECENTLY SCANNED      01320000 T  0015
                            QUANTITIES.  THE TABLE ROUTINE MAINTAINS THIS ARRAY,            01321000 T  0015
                            (ELBAT IS TABLE SPELLED BACKWARDS.)  THE TABLE ROUTINE          01322000 T  0015
                            GUARANTIES THAT ELBAT ALWAYS CONTAINS THE ELBAT WORDS           01323000 T  0015
                            FOR THE LAST 10 QUANTITIES SCANNED.  NXTELBT IS AN INDEX        01324000 T  0015
                            POINTING TO THE NEXT AVAILABLE WORD IN ELBAT.  I IS AN          01325000 T  0015
                            INDEX USED BY THE REST OF THE COMPILER TO FETCH THINGS          01326000 T  0015
                            FROM ELBAT.  I IS ALSO MAINTAINED BY THE TABLE ROUTINE;         01327000 T  0015
                  INTEGER ELCLASS;                                                          01328000 T  0015
PRT(110) = ELCLASS
                    COMMENT ELCLASS USUALLY CONTAINS ELBAT[I].CLASS;                        01329000 T  0015
                  INTEGER FCR, NCR, LCR,TLCR,CLCR;                                          01330000 T  0015
PRT(111) = FCR
PRT(112) = NCR
PRT(113) = LCR
PRT(114) = TLCR
PRT(115) = CLCR
                  INTEGER MAXTLCR;                                                          01331000 T  0015
PRT(116) = MAXTLCR
                    COMMENT FCR CONTAINS ABSOLUTE ADDRESS OF THE FIRST CHARACTOR OF         01332000 T  0015
                            THE CARD IMAGE CURRENTLY BEING SCANNED, NCR THE ADDRESS         01333000 T  0015
                            OF THE NEXT CHARACTOR TO BE SCANNED, AND LCR THE LAST           01334000 T  0015
                            CHARACTOR (COLUMN 73), TLCR AND CLCR CONTAIN ADDRESS OF         01335000 T  0015
                            THE LAST CHARACTER IN THE TAPE AND CARD BUFFERS.  MAXTLCR       01336000 T  0015
                            IS THE MAXIMUM OF TLCR WHEN THE INPUT IS BLOCKED;               01337000 T  0015
                   ARRAY TEN[-46:69];                                                       01340000 T  0015
PRT(117) = TEN

                                                                                            01341000 T  0017
                   DEFINE PRTBASE=129#,PRTOP=896#;  COMMENT PASE AND TOP OF PRT;            01342000 T  0017
                  ARRAY PRT[PRTBASE:PRTOP];                                                 01343000 T  0017
PRT(120) = PRT
                  INTEGER DISKADR,CORADR; COMMENT GLOBALS FOR PROGDESCBLDR;                 01344000 T  0020
PRT(121) = DISKADR
PRT(122) = CORADR
                  INTEGER SGAVL;COMMENT NEXT AVAILABLE SEGMENT NUMBER;                      01369000 T  0020
PRT(123) = SGAVL
                  INTEGER SGNO;COMMENT THIS IS THE CURRENT SEGMENT NUMBER;                  01370000 T  0020
PRT(124) = SGNO
                       ARRAY  COP,WOP[0:127];                                               01371000 T  0020
PRT(125) = COP
PRT(126) = WOP
                    COMMENT  THE EMIT ROUTINES PLACE EACH SYLLABLE INTO THE EDOC ARRAY      01372000 T  0023
                            AS SPECIFIED BY "L".                                            01373000 T  0023
                            IF THE DEBUGTOG IS TRUE COP AND WOP ARE FILLED WITH             01374000 T  0023
                            THE BCD FOR THE OPERATORS,OTHERWISE THEY ARE NOT USED;          01375000 T  0023
                   REAL  LASTENTRY ;                                                        01376000 T  0023
PRT(127) = LASTENTRY
                    COMMENT LASTENTRY IS USED BY EMITNUM AND CONSTANTCLEAN, IT POINTS       01377000 T  0023
                            INTO INFO[0,*] AT THE NEXT AVAILABLE CELL FOR CONSTANTS;        01378000 T  0023
                   BOOLEAN MRCLEAN ;                                                        01379000 T  0023
```

```
PRT(130) = MRCLEAN
                    COMMENT  NO  CONSTANTCLEAN ACTION TAKES PLACE WHILE MRCLEAN IS
                             FALSE. THIS FEATURE IS USED BY BLOCK BECAUSE OF THE
                             POSSIBILITY THAT CONSTANTCLEAN WILL USE INFO[NEXTINFO]
                             DURING AN ARRAY DECLARATION ;
                  REAL GT1,GT2,GT3,GT4,GT5;
PRT(131) = GT1
PRT(132) = GT2
PRT(133) = GT3
PRT(134) = GT4
PRT(135) = GT5
                  INTEGER GTI1;
PRT(136) = GTI1
                    COMMENT THESE VARIABLES ARE USED FOR TEMPORARY STORAGE;
                  INTEGER RESULT;
PRT(137) = RESULT
                    COMMENT THIS VARIABLE IS USED FOR A DUAL PURPOSE BY THE TABLE
                             ROUTINE AND THE SCANNER.  THE TABLE ROUTINE USES THIS
                             VARIABLE TO SPECIFY SCANNER OPERATIONS AND THE SCANNER
                             USES IT TO INFORM THE TABLE ROUTINE OF THE ACTION TAKEN;
                  INTEGER LASTUSED;
PRT(140) = LASTUSED
                    COMMENT LASTUSED IS A VARIABLE THAT CONTROLS THE ACTION OF
                             READACARD, THE ROUTINE WHICH READS CARDS AND INITIALIZES
                             OR PREPARES THE CARD FOR THE SCANNER.
                          LASTUSED  LAST CARD READ FROM
                          --------  ------------------
                              1     CARD READER ONLY, NO TAPE.
                              2     CARD READER, TAPE AND CARD MERGE.
                              3     TAPE, TAPE AND CARD MERGE.
                              4     INITIALIZATION ONLY, CARD ONLY.
                  ;
                  BOOLEAN LINKTOG;
PRT(141) = LINKTOG
                      COMMENT LINKTOG IS FALSE IF THE LAST THING EMITTED IS A LINK,
                              OTHERWISE IT IS TRUE;
                  INTEGER LEVEL,FRSTLEVEL,SUBLEVEL,MODE;
PRT(142) = LEVEL
PRT(143) = FRSTLEVEL
PRT(144) = SUBLEVEL
PRT(145) = MODE
                    COMMENT THESE VARIABLES ARE MAINTAINED BY THE BLOCK ROUTINE TO KEEP
                             TRACK OF LEVELS OF DEFINITION.  LEVEL GIVES THE DEPTH OF
                             NESTING IN DEFINITION, WHERE EACH BLOCK AND EACH PROCEDURE
                             GIVES RISE TO A NEW LEVEL.  SUBLEVEL GIVES THE LEVEL OF
                             THE PARAMETERS OF THE PROCEDURE CURRENTLY BEING COMPILED.
                             FRSTLEVEL IS THE LEVEL OF THE PARAMETERS OF THE MOST
                             GLOBAL OF THE PROCEDURES CURRENTLY BEING COMPILED.  MODE
                             IS THE CURRENT DEPTH OF THE PROCEDURE IN WHICH WE ARE
                             NESTED (AT COMPILE TIME);
                  BOOLEAN ERRORTOG;
PRT(146) = ERRORTOG
                    COMMENT ERRORTOG IS TRUE IF MESSAGES ARE CURRENTLY ACCEPTABLE TO THE
                             ERROR ROUTINES,  ERRORCOUNT IS THE COUNT OF ERROR MSSGS;
                  BOOLEAN ENDTOG;     COMMENT ENDTOG TELLS THE TABLE TO ALLOW
PRT(147) = ENDTOG
                                      COMMENT TO BE PASSED BACK TO COMPOUNDTAIL;
```

```
01380000 T  0023
01381000 T  0023
01382000 T  0023
01383000 T  0023
01384000 T  0023
01384500 T  0023
01385000 T  0023
01386000 T  0023
01387000 T  0023
01388000 T  0023
01389000 T  0023
01390000 T  0023
01391000 T  0023
01392000 T  0023
01393000 T  0023
01394000 T  0023
01394500 T  0023
01394600 T  0023
01395000 T  0023
01396000 T  0023
01397000 T  0023
01398000 T  0023
01398300 T  0023
01399000 T  0023
01400000 T  0023
01401000 T  0023
01402000 T  0023
01403000 T  0023
01404000 T  0023
01405000 T  0023
01406000 T  0023
01407000 T  0023
01408000 T  0023
01409000 T  0023
01410000 T  0023
01411000 T  0023
01412000 T  0023
01413000 T  0023
01414000 T  0023
01415000 T  0023
01416000 T  0023
```

```
                BOOLEAN STREAMTOG;                                                                  01417000 T  0023
PRT(150) = STREAMTOG
                   COMMENT STREAMTUG IS TRUE IF WE ARE COMPILING STREAM STATEMENT.  IT              01418000 T  0023
                           IS USED TO CONTROL COUMPOUNDTAIL;                                       01419000 T  0023
                DEFINE FS = 1#, FP = 2#, FL = 3#, FR =4#;                                          01420000 T  0023
                   COMMENT THESE DEFINES ARE USED WHEN CALLING THE VARIABLE ROUTINE.               01421000 T  0023
                           THEIR PURPOSES IS TO TELL VARIABLE WHO IS CALLING.                      01422000 T  0023
                           THEIR MEANING IS:                                                       01423000 T  0023
                           FS MEANS FROM STATEMENT,                                                01424000 T  0023
                           FP MEANS FROM PRIMARY,                                                  01425000 T  0023
                           FL MEANS FROM LIST,                                                     01426000 T  0023
                           FR MEANS FROM FOR;                                                      01427000 T  0023
                INTEGER L;                                                                         01428000 T  0023
PRT(151) = L
                   COMMENT   L IS THE LOCATION OF THE NEXT SYLLABLE TO BE EMITTED;                 01429000 T  0023
                DEFINE BLOCKCTR = 16#, JUNK = 17 #, XITR = 18 #, LSTRTN = 19#;                     01430000 T  0023
                DEFINE ATYPE =3#, BTYPE=ATYPE#,DTYPE=ATYPE#;                                       01452000 T  0023
                BOOLEAN TB1;                                                                       01457000 T  0023
PRT(152) = TB1
                   COMMENT TB1 IS A TEMPORARY BOOLEAN VARIABLE;                                    01458000 T  0023
                INTEGER JUMPCTR;                                                                   01459000 T  0023
PRT(153) = JUMPCTR
                   COMMENT   JUMPCTR IS A VARIABLE USED FOR COMMUNICATION BETWEEN BLOCK            01460000 T  0023
                             AND GENGO.  IT GIVES HIGHEST LEVEL TO WHICH A JUMP HAS                01461000 T  0023
                             BEEN MADE FROM WITHIN A THE PRESENTLY BEING COMPILED                  01462000 T  0023
                             SEGMENT.  THE BLOCK COMPILES CODE TO INCREMENT AND DECRE-             01463000 T  0023
                             MENT THE BLOCKCTR ON THE BASIS OF JUMPCTR AT COMPLETION               01464000 T  0023
                             OF COMPILATION OF A SEGMENT - I.E. THE BLOCKCTR IS TALLIED            01465000 T  0023
                             IF LEVEL = JUMPCTR;                                                   01466000 T  0023
                                                                                                   01467000 T  0023
                                                                                                   01468000 T  0023
                                                                                                   01469000 T  0023
                                                                                                   01470000 T  0023
                                                                                                   01471000 T  0023
                REAL STLB;
PRT(154) = STLB
                   COMMENT STLB IS USED BY VARIABLE AND ACTUALPARAPART TO COMMUNICATE              01472000 T  0023
                           THE LOWER BOUND INFORMATION FOR THE LAST DIMENSION OF THE               01473000 T  0023
                           ARRAY INVOLVED IN A ROW DESIGNATOR.  THE FORMAT OF THE                  01474000 T  0023
                           INFORMATION IS THAT OF INFO.  STLB IS ALSO SOMETIMES USED               01475000 T  0023
                           FOR TEMPORARY STORAGE;                                                  01476000 T  0023
                DEFINE BUMPL = L+L+2#;                                                             01477000 T  0023
                   COMMENT BUMPL IS USED MOSTLY TO PREPARE A FORWARD JUMP;                         01478000 T  0023
                DEFINE IDMAX =   LABELID#;                                                         01479000 T  0023
                   COMMENT IDMAX IS THE MAXIMUM CLASS NUMBER FOR IDENTIFIERS;                      01480000 T  0023
                INTEGER DEFINECTR,DEFINEINDEX;                                                     01481000 T  0023
PRT(155) = DEFINECTR
PRT(156) = DEFINEINDEX
                REAL  JOINFO,        COMMENT POINTS TO PSEUDO LABEL FOR JUMP OUTS;                 01482000 T  0023
PRT(157) = JOINFO
                   LPRT, COMMENT SHOWS LOCATION OF THE LAST LABEL IN THE PRT ;                     01483000 T  0023
PRT(160) = LPRT
                  NESTLEVEL,         COMMENT  COUNTS NESTING FOR GU TO AND JUMP OUTS;              01484000 T  0023
PRT(161) = NESTLEVEL
                   JUMPLEVEL;        COMMENT NUMBER OF LEVELS TO BE JUMPED OUT;                    01485000 T  0023
PRT(162) = JUMPLEVEL
                COMMENT THE REALS ABOVE ARE FOR STREAM STATEMENT;                                  01486000 T  0023
                ARRAY  MACRO[0:35];                                                                01487000 T  0023
```

```
PRT(163) = MACRO
                COMMENT MACRO IS FILLED WITH SYLLABLES FOR STREAM STATEMENT;                01488000 T  0026
            REAL  P,     COMMENT CONTAINS NUMBER OF FORMALS FOR STREAM PROCS;               01489000 T  0026
PRT(164) = P
                  Z;     COMMENT CONTAINS 1ST WORD OF INFO FOR STREAM FUNCTIONS;            01490000 T  0026
PRT(165) = Z
                ARRAY NEWTAPBUF[0:9];                                                      01490510 T  0026
PRT(166) = NEWTAPBUF
            SAVE ARRAY DEFINEARRAY[0:23];                                                  01491000 T  0028
PRT(167) = DEFINEARRAY
                COMMENT   THESE VARIABLES ARE USED TO CONTROL ACTION OF THE DEFINE.        01492000 T  0031
                          DEFINECTR COUNTS DEPTH OF NESTING OF DEFINE-# PAIRS.             01493000 T  0031
                          THE CROSSHATCH PART OF THE TABLE ROUTINE USES DEFINECTR           01494000 T  0031
                          TO DETERMINE THE MEANING OF A CROSSHATCH.  DEFINEINDEX IS        01495000 T  0031
                          THE NEXT AVAILABLE CELL IN THE DEFINEARRAY.  THE DEFINE-         01496000 T  0031
                          ARRAY HOLDS THE ALPHA OF THE DEFINE BEING RECREATED AND          01497000 T  0031
                          THE PREVIOUS VALUES OF LASTUSED, LCR, AND NCR;                   01498000 T  0031
             INTEGER BEGINCTR;                                                             01499000 T  0031
PRT(170) = BEGINCTR
                COMMENT   BEGINCTR GIVES THE NUMBER OF UNMATCHED BEGINS.  IT IS USED       01500000 T  0031
                          FOR ERROR CONTROL ONLY;                                          01501000 T  0031
            INTEGER DIALA,DIALB;                                                           01502000 T  0031
PRT(171) = DIALA
PRT(172) = DIALB
                COMMENT THESE VARIABLES GIVE THE LAST VALUE TO WHICH A AND B WERE        01503000 T  0031
                          DIALED.  THIS GIVES SOME LOCAL OPTIMIZATION.  EMITD            01504000 T  0031
                          WORRIES ABOUT THIS.  OTHER ROUTINES CAUSE A LOSS OF MEMORY     01505000 T  0031
                          BY SETTING DIALA AND DIALB TO ZERO;                            01506000 T  0031
           BOOLEAN RRB1;  COMMENT RRB1---RRBN ARE BOOLEAN VARIABLES THAT SERVE THE       01522000 T  0031
PRT(173) = RRB1
                     SAME FUNCTION AS RR1---RRN FOR REAL VARIABLES.  SEE                 01523000 T  0031
                     COMMENT AT RR1;                                                     01524000 T  0031
                BOOLEAN RRB2;  COMMENT SEE COMMENT AT RRB1 DECLARATION;                  01525000 T  0031
PRT(174) = RRB2
          DEFINE ARRAYMONFILE = [27:11]#;  COMMENT ARRAYMONFILE IS THE DEFINE FOR        01526000 T  0031
                                           THE ADDRESS OF THE FILE DESCRIPTOR IN         01527000 T  0031
                                           THE FIRST WORD OF ADDITIONAL INFO;            01528000 T  0031
          DEFINE SVARMONFILE = [37:11]#;  COMMENT MONITORFILE IS THE DEFINE FOR          01529000 T  0031
                                          THE ADDRESS OF THE FILE DESCRIPTOR IN          01530000 T  0031
                                          INFO FOR MONITORED SIMPLE VARIABLES;           01531000 T  0031
          DEFINE NODIMPART = [40:8]#;  COMMENT THE FIRST ADDITIONAL WORD OF INFO         01532000 T  0031
                                       FOR ARRAYS CONTAINS THE NUMBER OF DIMENSIONS      01533000 T  0031
                                       IN NODIMPART;                                     01534000 T  0031
          DEFINE LABLMONFILE = [13:11]#;  COMMENT LABLMONFILE DESIGNATES THE BIT         01535000 T  0031
                                          POSITION IN THE FIRST WORD OF ADDITIONAL       01536000 T  0031
                                          INFO THAT CONTAINS THE MONITOR FILE            01537000 T  0031
                                          ADDRESS FOR LABELS;                            01538000 T  0031
          DEFINE SWITMONFILE = [13:11]#;  COMMENT SWITMONFILE DESIGNATES THE BIT         01539000 T  0031
                                          POSITION IN THE FIRST WORD OF ADDITIONAL       01540000 T  0031
                                          INFO THAT CONTAINS THE MONITOR FILE            01541000 T  0031
                                          ADDRESS FOR LABELS;                            01542000 T  0031
          DEFINE FUNCMONFILE = [27:11]#;  COMMENT FUNCMONFILE DESIGNATES THE BIT         01543000 T  0031
                                          POSITION IN THE FIRST WORD OF ADDITIONAL       01544000 T  0031
                                          INFO THAT CONTAINS THE MONITOR FILE            01545000 T  0031
                                          ADDRESS FOR LABELS;                            01546000 T  0031
          DEFINE DUMPEE = [2:11]#;  COMMENT THE DUMPEE FIELD IN THE FIRST                01547000 T  0031
                                    ADDITIONAL WORD OF INFO FOR LABELS CONTAINS          01548000 T  0031
```

```
                                    THE ADDRESS OF THE COUNTER THAT IS INCREMENTED          01549000 T  0031
                                    EACH TIME THE LABEL IS PASSED IF THAT LABEL             01550000 T  0031
                                    APPEARS IN A DUMP DECLARATION;                         01551000 T  0031
                DEFINE DUMPOR = [24:11]#;  COMMENT THE DUMPOR FIELD IN THE FIRST           01552000 T  0031
                                    ADDITIONAL WORD OF INFO FOR LABELS CONTAINS            01553000 T  0031
                                    THE ADDRESS OF THE ROUTINE THAT IS GENERATED           01554000 T  0031
                                    FROM THE DUMP DECLARATION THAT IN TURN CALLS           01555000 T  0031
                                    THE PRINTI ROUTINE;                                    01556000 T  0031
                DEFINE SUBOP=48#;                                                          01556500 T  0031
                 FILE OUT CODE DISK SERIAL[1:1](1,1023);                                   01556900 T  0031
PRT(175) = CODE
                FILE IN CARD(RR1,10,RR2);                                                  01557000 T  0035
PRT(176) = CARD
                FILE OUT LINE DISK SERIAL[20:2400](RR3,15,RR4,SAVE 10);                    01558000 T  0039
PRT(177) = LINE
PRT(200) =
PRT(201) = FILE ATTRBUTS
                 ARRAY LIN[0:20]; COMMENT PRINT OUTPUT BUILT IN LIN;                       01559010 T  0046
PRT(202) = LIN
                INTEGER DA;                                                                01559020 T  0049
PRT(203) = DA
                SAVE FILE OUT NEWTAPE DISK SERIAL[20:2400](RR5,RR6,RR7,SAVE 1);            01560000 T  0049
PRT(204) = NEWTAPE
                FILE IN TAPE "OCRDIMG"(2,RR8,RR9);                                         01561000 T  0056
PRT(205) = TAPE
                SAVE ARRAY CBUFF,TBUFF[0:9]; % INPUT BUFFERS.                              01561056 T  0060
PRT(206) = CBUFF
PRT(207) = TBUFF
                FILE OUT CODISK DISK SERIAL [20:600] (2,30,300);                           01561300 T  0064
PRT(210) = CODISK
                FILE OUT DISK DISK [1:2100] "MCP""DISK"(3,30,300,SAVE 99);                 01561400 T  0068
PRT(211) = DISK
                DEFINE    MCPTYPE               = 63#,                                   01561410 T  0075
                          DCINTYPE              = 62#,                                   01561420 T  0075
                          TSSINTYPE             = 61#;                                   01561430 T  0075
                COMMENT  E$POL CODE FILES ARE UNIQUELY TYPED IN THEIR FILE               01561440 T  0075
                         HEADERS.   HEADER[4],[36:6] IS THE FIELD USED TO CONTAIN        01561450 T  0075
                         THE TYPE;                                                       01561460 T  0075
                FILE OUT DECK 0 (2,10);                                                  01561500 T  0075
PRT(212) = DECK
                FILE STUFF DISK SERIAL[20:150](2,10,30,SAVE 15);                         01561600 T  0079
PRT(213) = STUFF
                ARRAY TWXA[0:16];                                                        01561700 T  0086
PRT(214) = TWXA
                 REAL C;                                                                 01562000 T  0089
PRT(215) = C
                    COMMENT  C CONTAINS ACTUAL VALUE OF LAST CONSTANT SCANNED;           01563000 T  0089
                 REAL T;                                                                 01564000 T  0089
PRT(216) = T
                    COMMENT  T IS A TEMPORARY CELL;                                      01565000 T  0089
                 INTEGER TCOUNT;                                                         01566000 T  0089
PRT(217) = TCOUNT
                    REAL STACKCT;                                                        01566010 T  0089
PRT(220) = STACKCT
                    COMMENT  TCOUNT IS A VARIABLE WHICH HOLDS A PREVIOUS VALUE OF COUNT  01567000 T  0089
                              FOR THE USE OF CONVERT;                                    01568000 T  0089
                 DEFINE LASTSEQUENCE = 145#,                                             01569000 T  0089
```

```
                    LASTSEQROW = 2#;                                                    01570000 T  0089
                                                                                        01571000 T  0089
                                                                                        01572000 T  0089
                                                                                        01573000 T  0089
                                                                                        01574000 T  0089
                                                                                        01575000 T  0089
                                                                                        01576000 T  0089
                                                                                        01577000 T  0089
                                                                                        01578000 T  0089
                                                                                        01579000 T  0089
                                                                                        01580000 T  0089
                                                                                        01581000 T  0089
                                                                                        01582000 T  0089
                                                                                        01583000 T  0089
                REAL FOULED;                                                            01583100 T  0089
PRT(221) = FOULED
                                                                                        01584000 T  0089
                                                                                        01585000 T  0089
                BOOLEAN
                      FUNCTOG,    COMMENT TELLS WHETHER PROCEDURE BEING DECLARED IS A    01586000 T  0089
PRT(222) = FUNCTOG
                                          FUNCTION;                                     01587000 T  0089
                      P2,         COMMENT GENERALY TELLS WHETHER OWN WAS SEEN;          01588000 T  0089
PRT(223) = P2
                      P3,         COMMENT TELLS WHETHER SAVE WAS SEEN;                  01589000 T  0089
PRT(224) = P3
                      VONF,       COMMENT VALUE OR OWN FIELD OF ELBAT WORD;             01590000 T  0089
PRT(225) = VONF
                      FORMALF,    COMMENT FORMAL FIELD OF ELBAT WORD;                   01591000 T  0089
PRT(226) = FORMALF
                          PTOG,   COMMENT TELLS THAT FORMAL PARAPART IS BEING PROCESSD; 01592000 T  0089
PRT(227) = PTOG
                SPECTOG,                                                                01593000 T  0089
PRT(230) = SPECTOG
                      STOPENTRY,  COMMENT THIS MAKES THE ENTRY PROCEDURE ENTER ONLY     01594000 T  0089
PRT(231) = STOPENTRY
                                          ONE ID AND THEN EIXT;                         01595000 T  0089
                      AJUMP;      COMMENT TELLS WHETHER A JUMP IS HANGING;              01596000 T  0089
PRT(232) = AJUMP
                BOOLEAN STOPDEFINE;                                                     01597000 T  0089
PRT(233) = STOPDEFINE
                INTEGER MAXSAVE;                                                        01598000 T  0089
PRT(234) = MAXSAVE
                    COMMENT THIS CONTAINS THE SIZE OF THE MAXIMUM SAVE ARRAY            01599000 T  0089
                    DECLARED. IT IS USED TO HELP DETERMINE STORAGE REQUIREMENTS         01600000 T  0089
                    FOR THE PROGRAM PARAMETER BLOCK FOR THE OBJECT PROGRAM;             01601000 T  0089
                                                                                        01602000 T  0089
                 REAL
                      KLASSF,     COMMENT CLASS IN LOW ORDER 7 BITS;                    01603000 T  0089
PRT(235) = KLASSF
                      ADDRSF,     COMMENT ADDRESS IN LOW ORDER 11 BITS;                 01604000 T  0089
PRT(236) = ADDRSF
                      LEVELF,     COMMENT LVL IN LOW ORDER 5 BITS;                      01605000 T  0089
PRT(237) = LEVELF
                      LINKF,      COMMENT LINK IN LOW ORDER 13 BITS;                    01606000 T  0089
PRT(240) = LINKF
                      INCRF,      COMMENT INCR CN LOW ORDER 8 BITS;                     01607000 T  0089
PRT(241) = INCRF
                      PROINFO,    COMMENT CONTAINS ELBAT WORD FOR PROCEDURE BEING       01608000 T  0089
```

```
PRT(242) = PROINFO
                    G,          DECLARED;                                                   01609000 T  0089
PRT(243) = G                COMMENT GLOBAL TEMPORARY FOR BLOCK;                         01610000 T  0089
                    TYPEV,  COMMENT USED TO CARRY CLASS OF IDENTIFIER                   01611000 T  0089
PRT(244) = TYPEV
                    PROADO,         BEING DECLARED;                                     01612000 T  0089
                              COMMENT CONTAINS ADDRESS OF PROCEDURE BEING               01613000 T  0089
PRT(245) = PROADO
                    MARK ,          DECLARED;                                           01614000 T  0089
                            COMMENT CONTAINS INDEX INTO INFO WHERE FIRST WORD            01615000 T  0089
PRT(246) = MARK
                    PJ,             OF ADDITIONAL INFO FOR A PROCEDURE ENTRY;           01616000 T  0089
                            COMMENT FORMAL PARAMETER COUNTER;                           01617000 T  0089
PRT(247) = PJ
                     J,     COMMENT ARRAY COUNTER;                                      01618000 T  0089
PRT(250) = J
                    LASTINFO,   COMMENT INDEX TO LAST ENTRY IN INFO;                    01619000 T  0089
PRT(251) = LASTINFO
                    NEXTINFO,   COMMENT INDEX FOR NEXT ENTRYIN INFO;                    01620000 T  0089
PRT(252) = NEXTINFO
                    FIRSTX,     COMMENT RELATIVE ADD OF FIRST EXECUTABLE CODE           01621000 T  0089
PRT(253) = FIRSTX
                                IN BLOCK,INITIALIZED TO 4095 EACH TIME;                 01622000 T  0089
                    SAVEL;  COMMENT SAVE LOCATION FOR FIXUPS IN BLOCK;                  01623000 T  0089
PRT(254) = SAVEL
              INTEGER NCII; COMMENT THIS CONTAINS THE COUNT OF CONSTANTS                01624000 T  0089
PRT(255) = NCII
                                ENTERED IN INFO AT ANY GIVEN TIME;                      01625000 T  0089
                                                                                        01626000 T  0089
              PROCEDURE UNHOOK;FORWARD;
PRT(256) = UNHOOK                                                                       01627000 T  0093
              PROCEDURE MAKEUPACCUM;FORWARD;
PRT(257) = MAKEUPACCUM                                                                  01628000 T  0093
              DEFINE PURPT=[4:8]#,SECRET=2#;                                            01629000 T  0093
               COMMENT THESE DEFINES GIVE THE NAMES OF THE WORD MODE OPERATORS.  THE    01630000 T  0093
                  NUMBERS REFER TO THE APPROPRIATE SECTION OF THE PRODUCT SPECS.  THE   01631000 T  0093
                  FULL NAME IS ALSO GIVEN;                                              01632000 T  0093
               DEFINE                                                                   01633000 T  0093
                 ADD =  16#,    COMMENT (0101) 7.4.2.1   ADD;                           01634000 T  0093
                 BBC =  22#,    COMMENT (0131) 7.4.5.4   BRANCH BACKWARD CONDITIONAL;   01635000 T  0093
                 BBW = 534#,    COMMENT (4131) 7.4.5.2   BRANCH BACKWARD;               01636000 T  0093
                 BFC =  38#,    COMMENT (0231) 7.4.5.3   BRANCH FORWARD CONDITIONAL;    01637000 T  0093
                 BFW = 550#,    COMMENT (4231) 7.4.5.1   BRANCH FORWARD;                01638000 T  0093
                 CDC = 168#,    COMMENT (1241) 7.4.10.4  CONSTRUCT DESCRIPTOR CALL;     01639000 T  0093
                 CHS = 134#,    COMMENT (1031) 7.4.7.11  CHANGE SIGN;                   01640000 T  0093
                 COC =  40#,    COMMENT (0241) 7.4.10.3  CONSTRUCT OPERAND CALL;        01641000 T  0093
                 COM = 130#,    COMMENT (1011) 7.4.10.5  COMMUNICATION OPERATOR;        01642000 T  0093
                 DEL =  10#,    COMMENT (0045) 7.4.9.3   DELETE;                        01643000 T  0093
                 DUP = 261#,    COMMENT (2025) 7.4.9.2   DUPLICATE;                     01644000 T  0093
                 EQL = 581#,    COMMENT (4425) 7.4.4.3   EQUAL;                         01645000 T  0093
                 LBC = 278#,    COMMENT(2131) 7.4.5.9    GO BACKWARD CONDITIONAL;       01646000 T  0093
                 LBU = 790#,    COMMENT(6131) 7.4.5.7    GO BACKWARD (WORD);            01647000 T  0093
                 GEQ =  21#,    COMMENT (0125) 7.4.4.2   GREATER THAN OR EQUAL TO;      01648000 T  0093
                 LFC = 294#,    COMMENT(2231) 7.4.5.8    GO FORWARD CONDITIONAL;        01649000 T  0093
                  LFU = 806#,   COMMENT(6231) 7.4.5.6    GO FORWARD (WORD);             01650000 T  0093
                 GTR =  37#,    COMMENT (0225) 7.4.4.1   GREATER THAN;                  01651000 T  0093
                 IDV = 384#,    COMMENT (3001) 7.4.2.5   INTEGER DIVIDE;
```

```
INX =  24#,     COMMENT (0141) 7.4.10.2   INDEX;                                  01652000 T  0093
ISD = 532#,     COMMENT (4121) 7.4.6.3    INTEGER STORE DESTRUCTIVE;             01653000 T  0093
ISN = 548#,     COMMENT (4221) 7.4.6.4    INTEGER STORE NON-DESTRUCT;            01654000 T  0093
LEQ = 533#,     COMMENT (4125) 7.4.4.4    LESS THAN OR EQUAL TO;                 01655000 T  0093
LND =  67#,     COMMENT (0415) 7.4.3.1    LOGICAL AND;                           01656000 T  0093
LNG =  19#,     COMMENT (0115) 7.4.3.4    LOGICAL NEGATE;                        01657000 T  0093
LOD = 260#,     COMMENT (2021) 7.4.10.1   LOAD OPERATOR;                         01658000 T  0093
LOR =  35#,     COMMENT (0215) 7.4.3.2    LOGICAL OR;                            01659000 T  0093
LQV = 131#,     COMMENT (1015) 7.4.3.3    LOGICAL EQIVALENCE;                    01660000 T  0093
LSS = 549#,     COMMENT (4225) 7.4.4.5    LESS THAN;                             01661000 T  0093
MKS =  72#,     COMMENT (0441) 7.4.8.1    MARK STACK;                            01662000 T  0093
MUL =  64#,     COMMENT (0401) 7.4.2.3    MULTIPLY;                              01663000 T  0093
NEQ =  69#,     COMMENT (0425) 7.4.4.6    NOT EQUAL TO;                          01664000 T  0093
NOP =  11#,     COMMENT (0055) 7.4.7.1    NO OPERATION;                          01665000 T  0093
PRL =  18#,     COMMENT (0111) 7.4.10.6   PROGRAM RELEASE;                       01666000 T  0093
PRTE=  12#,     COMMENT (0061) 7.4.10.9   EXTEND PRT;                            01667000 T  0093
RDV = 896#,     COMMENT (7001) 7.4.2.6    REMAINDER DIVIDE;                      01668000 T  0093
RTN =  39#,     COMMENT (0235) 7.4.8.3    RETURN NORMAL;                         01669000 T  0093
RTS = 167#,     COMMENT (1235) 7.4.8.4    RETURN SPECIAL;                        01670000 T  0093
SND = 132#,     COMMENT (1021) 7.4.6.2    STORE NON-DESTRUCTIVE;                 01671000 T  0093
SSP = 582#,     COMMENT (4431) 7.4.7.10   SET SIGN PLUS;                         01672000 T  0093
STD =  68#,     COMMENT (0421) 7.4.6.1    STORE DESTRUCTIVE;                     01673000 T  0093
SUB =  48#,     COMMENT (0301) 7.4.2.2    SUBTRACT;                              01674000 T  0093
XCH = 133#,     COMMENT (1025) 7.4.9.1    EXCHANGE;                              01675000 T  0093
XIT =  71#,     COMMENT (0435) 7.4.8.2    EXIT;                                  01676000 T  0093
ZP1 = 322#,     COMMENT (2411) 7.4.10.8   CONDITIONAL HALT;                      01677000 T  0093
SCI =1003#,     COMMENT (7655)            SCAN OUT INITIALIZE;                   01677050 T  0093
SAN =1004#,     COMMENT (7661)            SYSTEM ATTENTION NEEDED;               01677100 T  0093
  SCS =1019#;   COMMENT (7755)            SCAN OUT STOP;                         01677150 T  0093
 COMMENT  THESE DEFINES ARE USED BY EMITD;                                       01678000 T  0093
 DEFINE                                                                          01679000 T  0093
  DIA =  45#,    COMMENT (XX55) 7.4.7.1    DIAL A;                              01680000 T  0093
  DIB =  49#,    COMMENT (XX61) 7.4.7.2    DIAL B;                              01681000 T  0093
  TRB =  53#;    COMMENT (XX65) 7.4.7.3    TRANSFER BITS;                       01682000 T  0093
 REAL MAXSTACK,STACKCTR;                                                         01683000 T  0093
PRT(260) = MAXSTACK
PRT(261) = STACKCTR
 INTEGER MAXROW;                                                                 01684000 T  0093
PRT(262) = MAXROW
          COMMENT THIS CONTAINS THE MAXIMUM ROW SIZE OF ALL NON-SAVE             01685000 T  0093
          ARRAYS DECLARED. ITS USE IS LIKE THAT OF MAXSAVE;                      01686000 T  0093
 INTEGER SEGSIZEMAX; COMMENT CONTAINS MAX SEGMENT SIZE;                          01687000 T  0093
PRT(263) = SEGSIZEMAX
 INTEGER F;                                                                      01688000 T  0093
PRT(264) = F
          REAL NLO,NHI,TLO,THI;                                                  01689000 T  0093
PRT(265) = NLO
PRT(266) = NHI
PRT(267) = TLO
PRT(270) = THI
          BOOLEAN DPTOG;                                                         01690000 T  0093
PRT(271) = DPTOG
          COMMENT  THE ABOVE THINGS ARE TEMP STORAGE FOR DOUBLE NOS;             01691000 T  0093
 BOOLEAN DOLLAR2TOG;                                                             01691500 T  0093
PRT(272) = DOLLAR2TOG
 DEFINE FZERO=896#;                                                              01692000 T  0093
 REAL T1,T2,N,K,AKKUM;                                                           01693000 T  0093
```

```
PRT(273) = T1
PRT(274) = T2
PRT(275) = N
PRT(276) = K
PRT(277) = AKKUM
          BOOLEAN STOPGSP;                                                    01694000 T  0093
PRT(300) = STOPGSP
          INTEGER BUP;                                                        01695000 T  0093
PRT(301) = BUP
          BOOLEAN INLINETOG;                                                  01695500 T  0093
PRT(302) = INLINETOG
                COMMENT UNIQUE GLOBAL TEMP FOR BLOCK;                         01696000 T  0093
          ARRAY GTA1[0:10];                                                   01697000 T  0093
PRT(303) = GTA1
           BOOLEAN ARRAY SPRT[0:31];                                          01698000 T  0095
PRT(304) = SPRT
              COMMENT  SPRT IS TO BE CONSIDERED TO BE AN ARRAY OF 32 32 BIT   01699000 T  0098
                       FIELDS.  THE 32 BITS ARE IN THE LOW ORDER PART OF EACH 01700000 T  0098
                       WORD.  THE BIT IS ON IF AND ONLY IF THE CORRESPONDING  01701000 T  0098
                       PRT CELL HAS A PERMANENT ASSIGNMENT;                   01702000 T  0098
           INTEGER PRTI,PRTIMAX;                                              01703000 T  0098
PRT(305) = PRTI
PRT(306) = PRTIMAX
              COMMENT  PRTIMAX GIVES NEXT PRT CELL AVAILABLE FOR PERMANENT ASSIGN-  01704000 T  0098
                       MENT.  PRTI GIVES NEXT PRT CELL POSSIBLY AVAILABLE FOR       01705000 T  0098
                       TEMPORARY ASSIGNMENT;                                        01706000 T  0098
          DEFINE ALPHASIZE = [12:6]#;  COMMENT ALPHASIZE IS THE DEFINE FOR THE BIT  01707000 T  0098
                                       POSITION IN THE SECOND WORD OF INFO WHICH    01708000 T  0098
                                       CONTAINS THE LENGTH OF ALPHA;                01709000 T  0098
          DEFINE EDOCINDEX = L.[36:3],L.[39:7]#;  COMMENT EDOCINDEX IS THE WORD     01710000 T  0098
                                                  PORTION OF L SPLIT INTO A ROW AND 01711000 T  0098
                                                  COLUMN INDEX FOR EDOC;            01712000 T  0098
          DEFINE CPLUS1 = 769#;  COMMENT SEE COMMENT AT CPLUS2 DEFINE;              01713000 T  0098
          DEFINE CPLUS2 = 770#;  COMMENT CPLUS1 AND CPLUS2 ARE EXPLICIT CONSTANTS   01714000 T  0098
                                 USED IN THE GENERATION OF C-RELATIVE CODE;         01715000 T  0098
           PROCEDURE FLAG(ERRNUM); VALUE ERRNUM; INTEGER ERRNUM; FORWARD;           01716000 T  0098
PRT(307) = FLAG
           ALPHA PROCEDURE B2D(B); VALUE B; REAL B; FORWARD;                      01717000 T  0098
PRT(310) = B2D
           REAL PROCEDURE TAKE(W); VALUE W; INTEGER W; FORWARD;                   01717700 T  0098
PRT(311) = TAKE
           BOOLEAN MACROID;                                                       01717800 T  0098
PRT(312) = MACROID
           REAL PROCEDURE FIXDEFINEINFO(T); VALUE T; REAL T; FORWARD;             01717900 T  0098
PRT(313) = FIXDEFINEINFO
           PROCEDURE ERR (ERRNUM); VALUE ERRNUM; INTEGER ERRNUM; FORWARD;         01718000 T  0098
PRT(314) = ERR
           INTEGER PROCEDURE GIT(L); VALUE L; REAL L; FORWARD;                    01719000 T  0098
PRT(315) = GIT
           ARRAY CALLA[0:31,0:255];                                               01720000 T  0098
PRT(316) = CALLA
           DEFINE CALL[CALL1]=CALLA[(GT3*CALL1),LINKR,GT3.LINKC]#;                01721000 T  0100
           REAL CALLX,CALLINFO,NESTCTR,NESTCUR;                                   01722000 T  0100
PRT(317) = CALLX
PRT(320) = CALLINFO
PRT(321) = NESTCTR
PRT(322) = NESTCUR
```

```
                BOOLEAN NESTOG;                                                  01723000 T  0100
PRT(323) = NESTOG
                ARRAY NESTPRT[PRTBASE:PRTOP];                                    01724000 T  0100
PRT(324) = NESTPRT
                ARRAY  SORTPRT[0:PRTOP-PRTBASE];                                 01725000 T  0103
PRT(325) = SORTPRT
                                                                                 01726000 T  0107
              COMMENT  "BLANKET" BLANKS OUT N+1 WORDS IN "THERE";                01737300 T  0107
              STREAM PROCEDURE BLANKET(N,THERE); VALUE N;                        01737350 T  0107
PRT(326) = BLANKET
                  BEGIN                                                          01737400 T  0107
                  DI:=THERE; DS:=8 LIT" "; SI:=THERE; DS:=N WDS;                 01737450 T  0108
                  END BLANKET;                                                   01737500 T  0110



              STREAM PROCEDURE CHANGESEQ(VAL,OLDSEQ); VALUE OLDSEQ;              01741200 T  0110
PRT(327) = CHANGESEQ
                    BEGIN  DI:=OLDSEQ; SI:=VAL; DS:=8 DEC  END CHANGESEQ;        01741300 T  0110



              STREAM PROCEDURE SEQUENCEERROR(L);                                 01742100 T  0112
PRT(330) = SEQUENCEERROR
                  BEGIN  DI:=L; DS:=16 LIT"SEQUENCE ERROR  ";  END SEQUENCEERROR;  01742110 T  0112
```

```
              STREAM PROCEDURE GETVOID(VP,NCR,LCR,SEQ);  VALUE NCR,LCR;          01756000 T  0114
PRT(331) = GETVOID
                  BEGIN                                                          01757000 T  0114
                  LABEL L,EXIT;                                                  01758000 T  0115
                  LOCAL N;                                                       01759000 T  0115
                  SI:=NCR; DI:=VP; DS:=8 LIT "0";                                01761000 T  0115
                  2(34(IF SC=" " THEN SI:=SI+1 ELSE JUMP OUT 2 TO L));           01762000 T  0116
                  GO TO EXIT;                    % NO VOID RANGE GIVEN, RETURN ZERO.  01763000 T  0119
              L:                                                                 01764000 T  0119
                  IF SC="%" THEN GO TO EXIT;     % STILL NO RANGE.              01764500 T  0119
                  IF SC="'" THEN                                                 01765000 T  0120
                       BEGIN                                                     01766000 T  0121
                       SI:=SI+1; DI:=LCR; DS:=1 LIT"'"; % STOPPER FOR SCAN       01767000 T  0121
                       NCR:=SI; % TEMP. STORAGE, SINCE NCR IS "LOCAL" TO GETVOID.  01768000 T  0122
                       8(IF SC="'" THEN JUMP OUT ELSE                            01769000 T  0123
                            BEGIN  TALLY:=TALLY+1; SI:=SI+1 END));               01770000 T  0124
                       END                                                       01771000 T  0126
                  ELSE BEGIN                                                     01772000 T  0126
                       NCR:=SI; % TEMP. STORAGE, SINCE NCR IS "LOCAL" TO GETVOID.  01773000 T  0126
                       DI:=LCR; DS:=1 LIT" "; % STOPPER FOR SCAN                 01774000 T  0126
                       8(IF SC=" " THEN JUMP OUT ELSE                            01775000 T  0127
                            BEGIN  TALLY:=TALLY+1; SI:=SI+1 END));               01776000 T  0128
                       END;                                                      01777000 T  0130
                  SI:=NCR; DI:=VP; DI:=DI+8; % RESTORE POINTERS.                 01780000 T  0130
                  N:=TALLY; DI:=DI-N; DS:=N CHR;                                 01781000 T  0130
              EXIT:                                                              01782000 T  0132
```

```
                    END OF GETVOID;                                                                  01784000 T  0132


                    REAL VOIDCR,VOIDPLACE,VOIDTCR,VOIDTPLACE;                                        01785000 T  0132
PRT(332) = VOIDCR
PRT(333) = VOIDPLACE
PRT(334) = VOIDTCR
PRT(335) = VOIDTPLACE
                    FORMAT                                                                           01800000 T  0132
                            BUG(X24,4(A4,X2));                                                       01802000 T  0132
                                                                                            START OF SEGMENT **********    4
PRT(336) = BUG
                                                                                               4 IS    8 LONG, NEXT SEG    3
                    PROCEDURE DATIME;                                                                01820000 T  0132
PRT(337) = DATIME
                        BEGIN                                                                        01821000 T  0132
                        INTEGER H,MIN,Q; ALPHA N1,N2;                                                01822000 T  0132
                                                                                            START OF SEGMENT **********    5
STACK(F+2) = H
STACK(F+3) = MIN
STACK(F+4) = Q
STACK(F+5) = N1
STACK(F+6) = N2
                        ALPHA STREAM PROCEDURE DATER(DATE); VALUE DATE;                              01823000 T  0000
PRT(340) = DATER
                            BEGIN                                                                    01824000 T  0000
                            DI:=LOC DATER; SI:=LOC DATE; SI:=SI+2;                                   01825000 T  0000
                            2(DS:=2 CHR; DS:=LIT"/"); DS:=2 CHR;                                     01826000 T  0000
                            END OF DATER;                                                            01827000 T  0002


                        H:=TIME1 DIV 216000; MIN:=(TIME1 DIV 3600) MOD 60;                           01828000 T  0003
                        N1:=DISK.MFID; N2:=DISK.FID;                                                 01828500 T  0007
                        WRITE(LINE,                                                                  01829000 T  0013
                              <X22,"BURROUGHS B-5700 ESPOL COMPILER MARK ",                          01830000 T  0015
PRT(341) = *FORMAT DESCRIPTOR*                                                                       01831000 T  0015
                "XVI.0.00"                                                                           01832000 T  0015
                              ,"  ",A6,"DAY, ",O,",  ",I2,"!",A2,X1,A3,                             01832500 T  0015
                             ////X45,A1,A6,"/",A1,A6,/X45,15("=")//>,                          6 IS   40 LONG, NEXT SEG    5
                              TIME(6),DATER(TIME(5)),12*REAL(Q:=H MOD 12=0)+Q,                       01833000 T  0016
PRT(342) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*                                                      01834000 T  0032
                              Q:=MIN MOD 10+(MIN DIV 10)*64,                                         01835000 T  0036
                              IF H>12 THEN "PM." ELSE "AM.",                                         01835500 T  0043
                              N1.[6:6],N1,N2.[6:6],N2);                                              01836000 T  0053
PRT(343) = OUTPUT(W)                                                                                 01837000 T  0054
                        NOHEADING:=FALSE;                                                       5 IS   58 LONG, NEXT SEG    3
                        END OF DATIME;


                    COMMENT  THIS SECTION CONTAINS ALL CODE PERTAINENT TO READING CARDS              02000000 T  0132
```

```
                        AND SCANNING THEM;                                                  02001000 T  0132
                COMMENT  OCTIZE REFORMATS ACCUM FOR OCTAL CONSTANTS;                        02001836 T  0132
                BOOLEAN STREAM PROCEDURE OCTIZE(S,D,SKP,CNT); VALUE SKP,CNT;               02001838 T  0132
PRT(344) = OCTIZE
                    BEGIN                                                                  02001840 T  0132
                    SI:=S; SI:=SI+4; DI:=D; SKP(DS:=3 RESET); % RIGHT JUSTIFY.             02001842 T  0133
                    CNT(IF SC≥"8" THEN TALLY:=1 ELSE IF SC<"0" THEN TALLY:=1; SKIP 3 SB;   02001844 T  0135
                        3(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB));                    02001846 T  0138
                    SI:=D; IF SB THEN                                                      02001848 T  0140
                         BEGIN  TALLY:=1; DI:=D; DS:=RESET   END; % PREVENT FLAG BIT.      02001850 T  0141
                    OCTIZE:=TALLY; % "1" = NON OCTAL CHARACTER OR FLAG BIT.                 02001852 T  0142
                    END OCTIZE;                                                            02001854 T  0142



                COMMENT  HEXIZE REFORMATS ACCUM FOR HEXADECIMAL CONSTANTS;                 02001856 T  0143
                BOOLEAN STREAM PROCEDURE HEXIZE(S,D,SKP,CNT); VALUE SKP,CNT;               02001858 T  0143
PRT(345) = HEXIZE
                    BEGIN  LOCAL T1,T2,TEMP2,TEMP1; LABEL AGIN;                            02001860 T  0143
                COMMENT  LOCAL VARIABLES ARE LOCATED IN REVERSE ORDER FROM THE             02001862 T  0144
                         WAY THEY ARE DECLARED IN STREAM PROCEDURES;                       02001864 T  0144
                    DI:=LOC TEMP1; CNT(DS:=LIT"1"); % IN CASE A CHAR=A,B,C,D,OR F.         02001866 T  0144
                    SI:=S; SI:=SI+3; DI:=LOC TEMP1; % WE MAY OVERFLOW INTO TEMP2.          02001868 T  0146
                    CNT(IF SC<"0" THEN IF SC≥"A" THEN IF SC≤"F" THEN % WORK HARD.          02001870 T  0146
                         BEGIN                                                             02001872 T  0150
                         T1:=SI; T2:=DI; DI:=T1; SI:=T2; % FLIP, MAN.                      02001874 T  0150
                         DS:=3 RESET; SI:=T1; DI:=T2; % FLIP BACK.                         02001876 T  0151
                         DS:=1 ADD; DI:=DI-1; SKIP 2 DB; DS:=1 SET; SKIP 3 DB;             02001878 T  0152
                         GO AGIN;                                                          02001880 T  0153
                         END;                                                              02001882 T  0154
                         IF SC<"0" THEN TALLY:=1; DS:=CHR; % < 0 = NON-HEX CHARACTER.      02001884 T  0154
                AGIN:                                                                      02001886 T  0155
                     );                                                                    02001888 T  0155
                    SI:=LOC TEMP1; DI:=D; SKP(DS:=4 RESET); % RIGHT ADJUST CONSTANT.       02001890 T  0155
                    CNT(SKIP 2 SB;                                                         02001892 T  0157
                      4(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB));% FINAL CONVERT.      02001894 T  0158
                    SI:=D; IF SB THEN                                                      02001895 T  0160
                         BEGIN  TALLY:=1; DI:=D; DS:=RESET   END; % PREVENT FLAG BIT.      02001896 T  0161
                    HEXIZE:=TALLY; % "1" IF PROGRAMMER GOOFED.                             02001897 T  0162
                    END HEXIZE;                                                            02001898 T  0163



                COMMENT  PUTSEQNO PUTS THE SEQUENCE NUMBER OF THE CARD-IMAGE               02002000 T  0164
                         CURRENTLY BEING SCANNED INTO THE INFO TABLE IN CASE               02003000 T  0164
                         IT IS NEEDED FOR FUTURE REFERENCE;                                02004000 T  0164
                STREAM PROCEDURE PUTSEQNO(INFO,LCR); VALUE LCR;                            02005000 T  0164
PRT(346) = PUTSEQNO
                    BEGIN  DI:=INFO; SI:=LCR; DS:=WDS;  END PUTSEQNO;                      02006000 T  0164



                COMMENT  TURNONSTOPLIGHT TURNS THE LIGHT "RED" ON THE "CORNER".            02007000 T  0165
                         I.E., THE PURPOSE OF THIS ROUTINE IS TO INSERT A PER-             02008000 T  0165
```

```
                    CENT SIGN IN COLUMN 73 AS AN END OF CARD SENTINEL FOR                02009000 T  0165
                    THE SCANNER;                                                         02010000 T  0165
               STREAM PROCEDURE TURNONSTOPLIGHT(RED,CORNER); VALUE RED,CORNER;           02011000 T  0165
PRT(347) = TURNONSTOPLIGHT
                  BEGIN  DI:=CORNER; SI:=LOC CORNER; SI:=SI-1; DS:=CHR  END;          02012000 T  0165



                  COMMENT WRITNEW TRANSFERS THE CARD IMAGE TO THE NEWTAPE BUFFER      02014000 T  0166
                          AND REPORTS IF THE CARD MIGHT BE CONTROL CARD;              02015000 T  0166
               BOOLEAN STREAM PROCEDURE WRITNEW(NEW,FCR); VALUE FCR;                 02016000 T  0166
PRT(350) = WRITNEW
                  BEGIN SI ← FCR; IF SC ≠ "$" THEN TALLY ← 1;                        02017000 T  0166
                  DI←NEW;DS←10 WDS;                                                  02018000 T  0168
                       WRITNEW ← TALLY END WRITNEW;                                  02020000 T  0168



                  COMMENT MKABS CONVERTS A DESCRIPTOR  TO AN ABSOLUTE ADDRESS;       02021000 T  0169
               REAL STREAM PROCEDURE MKABS(A);                                       02022000 T  0169
PRT(351) = MKABS
                        BEGIN DI ← A; MKABS ← DI END MKABS;                          02023000 T  0169



               REAL STREAM PROCEDURE CONV(ACCUM,SKP,N);VALUE SKP,N;                  02041000 T  0171
PRT(352) = CONV
                     BEGIN                                                           02042000 T  0171
                          SI← ACCUM; SI←SI+SKP;SI←SI+3;DI←LOC CONV;DS←N OCT           02043000 T  0172
                     END;                                                            02044000 T  0173



               STREAM PROCEDURE  MOVECHARACTERS(N,SORCE,SSKIP,DEST,DSKIP);           02045000 T  0174
PRT(353) = MOVECHARACTERS
                          VALUE  N,SSKIP,DSKIP;                                      02046000 T  0174
                          BEGIN                                                      02047000 T  0174
                          SI←SORCE ; DI←DEST;                                        02048000 T  0175
                          SI←SI+SSKIP; DI← DI+DSKIP ;                                02049000 T  0175
                          DS ← N CHR ;                                               02050000 T  0176
                          END ;                                                      02051000 T  0177



               COMMENT  MOVECHARACTERS MOVES  N CHARACTERS FROM THE SSKIP-TH CHAR IN 02052000 T  0177
                        "SORCE" TO THE DSKIP-TH CHAR IN "DEST".           ;        02053000 T  0177
               STREAM PROCEDURE MOVE(W)"WORDS FROM"(A)"TO"(B); VALUE W;              02054000 T  0177
PRT(354) = MOVE
                        BEGIN SI ← A; DI ← B; DS ← W WDS END;                        02055000 T  0177
```

```
                STREAM PROCEDURE RESIZE(FIEL);                                  02056000 T  0179
PRT(355) = RESIZE
                BEGIN LOCAL T;                                                  02057000 T  0179
                 SI←FIEL; DI←LOC T; DS←WDS;                                     02058000 T  0180
                SI←T;DI←FIEL;DI←DI+1; SKIP 2 DB; DS←10 SET                      02059000 T  0180
                END;                                                            02060000 T  0182



              COMMENT  EQUAL COMPARES COUNT CHARACTERS LOCATED AT A AND B FOR   02061000 T  0182
                       EQUALITY.  THIS ROUTINE IS USED IN THE LOOK-UP OF ALPHA  02061500 T  0182
                       QUANTITIES IN THE DIRECTORY;                             02062000 T  0182
              BOOLEAN STREAM PROCEDURE EQUAL(COUNT,A,B); VALUE COUNT;           02062500 T  0182
PRT(356) = EQUAL
                  BEGIN                                                         02063000 T  0182
                  TALLY:=1; SI:=A; DI:=B;                                       02063500 T  0183
                  IF COUNT SC=DC THEN EQUAL:=TALLY                              02064000 T  0183
                  END EQUAL;                                                    02064500 T  0184



                PROCEDURE READACARD;  FORWARD;                                  02065000 T  0185
PRT(357) = READACARD
                PROCEDURE DOLLARCARD; FORWARD;                                  02065500 T  0185
PRT(360) = DOLLARCARD
                BOOLEAN PROCEDURE BOOLEXP; FORWARD;                             02065600 T  0185
PRT(361) = BOOLEXP
                PROCEDURE SCANNER;                                              02066000 T  0185
PRT(362) = SCANNER
                    BEGIN                                                       02066500 T  0185
                COMMENT  "SCAN" IS THE STREAM PROCEDURE WHICH DOES THE ACTUAL SCANNING.  02067000 T  0185
                         IT IS DRIVEN BY A SMALL WORD MODE PROCEDURE CALLED "SCANNER",   02067500 T  0185
                         WHICH CHECKS FOR A QUANTITY BEING BROKEN ACROSS A CARD. "SCAN"  02068000 T  0185
                         IS CONTROLLED BY A VARIABLE CALLED "RESULT". "SCAN" ALSO        02068500 T  0185
                         INFORMS THE WORLD OF ITS ACTION BY MEANS OF THE SAME VARIABLE.  02069000 T  0185
                         HENCE THE VARIABLE "RESULT" IS PASSED BY BOTH NAME AND VALUE.   02069500 T  0185
                         THE MEANING OF "RESULT" AS INPUT IS:                           02070000 T  0185
                            VALUE     MEANING                                           02070500 T  0185
                            =====     =======================================           02071000 T  0185
                              0       INITIAL CODE - DEBLANK AND START TO FETCH THE     02071500 T  0185
                                      NEXT QUANTITY.                                    02072000 T  0185
                              1       CONTINUE BUILDING AN IDENTIFIER (INTERRUPTED BY   02072500 T  0185
                                      END-OF-CARD BREAK).                               02073000 T  0185
                              2       LAST QUANTITY BUILT WAS SPECIAL CHARACTER. HENCE, 02073500 T  0185
                                      EXIT (INTERRUPTION BY END-OF-CARD BREAK IS NOT    02074000 T  0185
                                      IMPORTANT).                                       02074500 T  0185
                              3       CONTINUE BUILDING A NUMBER (INTERRUPTED BY END-OF- 02075000 T  0185
                                      CARD BREAK).                                      02075500 T  0185
                              4       LAST THING WAS AN ERROR (COUNT EXCEEDED 63). HENCE, 02076000 T  0185
                                      EXIT (INTERRUPTION BY END-OF-CARD BREAK NOT       02076500 T  0185
                                      IMPORTANT).                                       02077000 T  0185
                              5       GET NEXT CHARACTER AND EXIT.                      02077500 T  0185
                              6       SCAN A COMMENT.                                   02078000 T  0185
                              7       DEBLANK ONLY.                                     02078500 T  0185
                         THE MEANING OF "RESULT" AS OUTPUT IS:                          02079000 T  0185
```

```
                    VALUE     MEANING                                                  02079500 T  0185
                    *****     ========================================                 02080000 T  0185
                      1       AN IDENTIFIER WAS BUILT.                                 02080500 T  0185
                      2       A SPECIAL CHARACTER WAS OBTAINED.                        02081000 T  0185
                      3       A NUMBER (INTEGER) WAS BUILT.                            02081500 T  0185
                  "SCAN" PUTS ALL STUFF SCANNED (EXCEPT FOR COMMENTS AND               02082000 T  0185
                  DISCARDED BLANKS) INTO "ACCUM" (CALLED "ACCUMULATOR"                 02082500 T  0185
                  FOR THE REST OF THIS DISCUSSION).                                    02083000 T  0185
                  "COUNT" IS THE VARIABLE THAT GIVES THE NUMBER OF CHARACTERS          02083500 T  0185
                  "SCAN" HAS PUT INTO THE "ACCUMULATOR".  SINCE "SCAN" NEEDS           02084000 T  0185
                  THE VALUE SO THAT IT CAN PUT MORE CHARACTERS INTO THE "ACCUM-        02084500 T  0185
                  ULATOR" AND NEEDS TO UPDATE "COUNT" FOR THE OUTSIDE WORLD.           02085000 T  0185
                  "COUNT" IS PASSED BY BOTH NAME AND VALUE.  IT IS ALSO                02085500 T  0185
                  CONVENIENT TO HAVE (63-COUNT). THIS IS CALLED "COMCOUNT".            02086000 T  0185
                  "NCR" (NEXT CHARACTER TO BE SCANNED) IS ALSO PASSED BY               02086500 T  0185
                  NAME AND VALUE SO THAT IT MAY BE UPDATED.                            02087000 T  0185
                  "ST1" AND "ST2" ARE TEMPORARY STORAGES WHICH ARE EXPLICITLY          02087500 T  0185
                  PASSED TO "SCAN" IN ORDER TO OBTAIN THE MOST USEFULL STACK           02088000 T  0185
                  ARRANGEMENT.                                                         02088500 T  0185
                  ;                                                                    02089000 T  0185
                STREAM PROCEDURE SCAN(NCR,COUNTV,ACCUM,COMCOUNT,RESULT,RESULTV,        02089500 T  0185
                                                                  START OF SEGMENT ********** 7
PRT(363) = SCAN
                                     COUNT,ST2,NCRV,ST1);                              02090000 T  0000
                                VALUE COUNTV, COMCOUNT,RESULTV,ST2,NCRV,ST1;           02090500 T  0000
                  BEGIN                                                                02091000 T  0000
                  LABEL DEBLANK,NUMBERS,IDBLDR,GNC,K,EXIT,FINIS,L,ERROR,               02091500 T  0000
                        COMMENTS,COMMANTS;                                             02092000 T  0000
                        DI:=RESULT; DI:=DI+7; SI:=NCRV;                                02092500 T  0000
                  COMMENT  SETUP "DI" FOR A CHANGE IN "RESULT" AND "SI" FOR A LOOK AT  02093000 T  0000
                        THE BUFFER;                                                    02093500 T  0000
                        CI:=CI+RESULTV;   % SWITCH ON VALUE OF RESULT;                 02094000 T  0000
                        GO DEBLANK;  % 0 IS INITIAL CODE.                              02094500 T  0001
                        GO IDBLDR;   % 1 IS ID CODE.                                  02095000 T  0001
                        GO FINIS;    % 2 IS SPECIAL CHARACTER CODE.                    02095500 T  0001
                        GO NUMBERS;  % 3 IS NUMBER CODE.                             02096000 T  0002
                        GO FINIS;    % 4 IS ERROR CODE.                               02096500 T  0002
                        GO GNC;      % 5 IS GET NEXT CHARACTER CODE.                  02097000 T  0002
                        GO COMMANTS; % 6 IS COMMENT CODE.                            02097500 T  0002
                                     % 7 IS DEBLANK ONLY CODE.                        02098000 T  0003
                        IF SC=" " THEN                                                 02098500 T  0003
                  K:         BEGIN  SI:=SI+1; IF SC=" " THEN GO K  END;                02099000 T  0003
                        GO FINIS;                                                      02099500 T  0005
                  DEBLANK:                                                             02100000 T  0005
                        IF SC=" " THEN                                                 02100500 T  0005
                  L:         BEGIN  SI:=SI+1; IF SC=" " THEN GO L  END;                02101000 T  0006
                  COMMENT  IF WE ARRIVE HERE WE HAVE A NON-BLANK CHARACTER;            02101500 T  0008
                        NCRV:=SI;                                                      02102000 T  0008
                        IF SC ≥ "0" THEN GO NUMBERS;                                   02102500 T  0008
                        IF SC=ALPHA THEN GO IDBLDR;                                    02103000 T  0009
                  COMMENT  IF WE ARRIVE HERE WE HAVE A SPECIAL CHARACTER (OR GNC);     02103500 T  0009
                  GNC:                                                                 02104000 T  0009
                        DS:=LIT"2"; TALLY:=1; SI:=SI+1; GO EXIT;                       02104500 T  0009
                  COMMANTS:                                                            02105000 T  0011
                        IF SC≠";" THEN                                                 02105500 T  0011
                             BEGIN                                                     02106000 T  0012
                  COMMENTS:                                                            02106500 T  0012
```

```
                    SI:=SI+1;                                                    02107000 T  0013
                    IF SC > "%" THEN GO COMMENTS;                                02107500 T  0013
                    IF SC < ";" THEN GO COMMENTS;                                02108000 T  0014
               COMMENT  CHARACTERS BETWEEN % AND SEMICOLON ARE HANDLED BY WORD-   02108500 T  0014
               MODE PART OF COMMENT ROUTINE;                                     02109000 T  0014
                    END;                                                         02109500 T  0014
               GO FINIS;                                                         02110000 T  0014
               IDBLDR:                                                           02110500 T  0015
               TALLY:=63; DS:=LIT "1";                                           02111000 T  0015
               COMCOUNT(TALLY:=TALLY+1;                                          02111500 T  0015
                        IF SC=ALPHA THEN SI:=SI+1 ELSE JUMP OUT TO EXIT);        02112000 T  0017
               TALLY:=TALLY+1;                                                   02112500 T  0018
               IF SC=ALPHA THEN                                                  02113000 T  0019
                    BEGIN                                                        02113500 T  0019
               ERROR:                                                            02114000 T  0019
                    DI:=DI-1; DS:=LIT "4"; GO EXIT;                              02114500 T  0020
                    END                                                          02115000 T  0021
               ELSE GO EXIT;                                                     02115500 T  0021
               COMMENT  IF WE ARRIVE AT ERROR WE HAVE MORE THAN 63 CHARACTERS    02116000 T  0021
               IN AN IDENTIFIER OR NUMBER;                                       02116500 T  0021
               NUMBERS:                                                          02117000 T  0021
               TALLY:=63; DS:=LIT "3";                                           02117500 T  0021
PRT(364) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               COMCOUNT(TALLY:=TALLY+1;                                          02118000 T  0022
                        IF SC <"0"THEN JUMP OUT TO EXIT; SI:=SI+1);              02118500 T  0024
               GO ERROR;                                                         02119000 T  0025
               EXIT:                                                             02119500 T  0025
               ST1:=TALLY; % "ST1" CONTAINS NUMBER OF CHARACTERS WE ARE          02120000 T  0025
                           % GOING TO MOVE INTO THE "ACCUMULATOR".               02120500 T  0026
               TALLY:=TALLY+COUNTV; ST2:=TALLY;                                  02121000 T  0026
               DI:=COUNT; SI:=LOC ST2; DS:=WDS;                                  02121500 T  0027
               COMMENT  THIS CODE UPDATED "COUNT";                               02122000 T  0027
               DI:=ACCUM; SI:=SI-3; DS:=3 CHR;                                   02122500 T  0027
               COMMENT  THIS CODE PLACES "COUNT" IN "ACCUM" AS WELL;             02123000 T  0028
               DI:=DI+COUNTV; % POSITION "DI" PAST CHARACTERS ALREADY            02123500 T  0028
                              % IN THE "ACCUMULATOR", IF ANY.                    02124000 T  0029
               SI:=NCRV; DS:=ST1 CHR;                                            02124500 T  0029
               COMMENT  MOVE CHARACTERS INTO "ACCUM";                            02125000 T  0029
               FINIS:                                                            02125500 T  0029
               DI:=NCR; ST1:=SI; SI:=LOC ST1; DS:=WDS;                           02126000 T  0029
PRT(365) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               COMMENT  RESET "NCR" TO LOCATION OF NEXT CHARACTER TO BE SCANNED; 02126500 T  0031
               END OF SCAN;                                                      02127000 T  0031
```

```
    LABEL L;%                                                                    02127500 T  0031
L:                                                                               02128000 T  0031
    SCAN(NCR,COUNT,ACCUM[1],63-COUNT,RESULT,                                     02128500 T  0032
         RESULT,COUNT,0,NCR,0);                                                  02129000 T  0034
    IF NCR=LCR THEN                                                              02129500 T  0036
         BEGIN                                                                   02130000 T  0037
         READACARD;                                                              02130500 T  0037
         GO TO L; % GO DIRECTLY TO L, DO NOT PASS GO,                            02135500 T  0038
                  % DO NOT COLLECT $200.                                         02136000 T  0038
         END;                                                                    02136500 T  0038
```

```
        END SCANNER;                                                                                    02137000 T  0038
                                                                                                      7 IS   39 LONG, NEXT SEG    3



    DEFINE WRITELINE = IF SINGLTOG THEN WRITE(LINE,15,LIN[*])                                           02181000 T  0185
                         ELSE WRITE(LINE[DBL],15,LIN[*])#,                                          02181250 T  0185
            PRINTCARD = BEGIN                                                                       02182500 T  0185
                        EDITLINE(LIN,FCR,L DIV 4,L,[46:2],MEDIUM,OMITTING);                         02182750 T  0185
                        IF NOHEADING THEN DATIME; WRITELINE;                                        02183000 T  0185
                        END #;                                                                      02183250 T  0185
    STREAM PROCEDURE EDITLINE(LINE,NCR,R,L,SYMBOL,OMIT);                                            02183500 T  0185
PRT(366) = EDITLINE
                              VALUE  NCR,R,L,SYMBOL,OMIT;                                           02183750 T  0185
                                                                                                    02184000 T  0185
        BEGIN                                                                                       02184250 T  0186
        DI := LINE; DS := 16 LIT " ";                                                               02184500 T  0188
        SI := NCR; DS := 9 WDS;                                                                     02184750 T  0189
        DS := 8 LIT " ";                                                                            02185000 T  0190
        DS := WDS; % SEQUENCE NUMBER.                                                               02185250 T  0190
        DS:=LIT" "; SI:=LOC SYMBOL; SI:=SI+6;                                                       02185500 T  0191
        DS:=2 CHR; DS:=LIT" ";                                                                      02185750 T  0192
        SI+LOC R;  DS+4 DEC;  DS+LIT ":";                                                           02186000 T  0193
        SI+LOC L;  DS+1 DEC;                                                                        02186250 T  0193
        DS+6 LIT " ";                                                                               02186750 T  0194
        OMIT(DI:=DI-12; DS:=8 LIT"    OMIT");                                                       02187000 T  0197
        END EDITLINE;

    COMMENT  COMPARE COMPARES SEQUENCE NUMBERS OF TAPE AND CARD. IF                                 02187250 T  0197
             TAPE IS SMALLER THEN RESULT = 0 ELSE IF CARD IS SMALLER                                02187500 T  0197
             RESULT = 1 ELSE RESULT = 2;                                                            02187750 T  0197
    REAL STREAM PROCEDURE COMPARE(TAPE,CARD); VALUE TAPE,CARD;                                      02188000 T  0197
PRT(367) = COMPARE
        BEGIN                                                                                       02188250 T  0197
        SI := TAPE; DI := CARD;                                                                     02188500 T  0198
        IF 8 SC ≥ DC THEN                                                                           02188750 T  0198
             BEGIN                                                                                  02189000 T  0199
             SI := SI-8; DI := DI-8; TALLY := 1;                                                    02189250 T  0199
             IF 8 SC = DC THEN TALLY := 2                                                           02189500 T  0200
             END;                                                                                   02189750 T  0200
        COMPARE := TALLY                                                                            02190000 T  0201
        END COMPARE;                                                                                02190250 T  0201



    PROCEDURE OUTPUTSOURCE;                                                                         02190500 T  0202
PRT(370) = OUTPUTSOURCE
        BEGIN                                                                                       02190750 T  0202
        LABEL LCARD,LTAPE,AWAY;                                                                     02191000 T  0202
                                                                                                  START OF SEGMENT **********     8
        SWITCH SW:=LCARD,LCARD,LTAPE,AWAY,LCARD,LTAPE;                                              02191250 T  0000
        IF SEQTOG THEN % RESEQUENCING.                                                              02191500 T  0006
             BEGIN                                                                                  02191750 T  0007
```

```
        IF TOTALNO = -10 OR NEWBASE THEN                                02192000 T  0008
             BEGIN                                                      02192250 T  0009
             NEWBASE := FALSE; GTI1:= TOTALNO:=BASENUM                  02192500 T  0010
             END                                                        02192750 T  0011
        ELSE GTI1:= TOTALNO:= TOTALNO + ADDVALUE;                       02193000 T  0012
        CHANGESEQ(GTI1,LCR);                                            02193250 T  0014
        END;                                                            02193500 T  0015
    IF NEWTOG THEN                                                      02193750 T  0015
      IF WRITNEW(LIN,FCR) THEN WRITE(NEWTAPE,10,LIN[*]);                02194000 T  0016
    IF OMITTING THEN IF NOT LISTATOG THEN GO AWAY;                      02194250 T  0023
    GO SW[LASTUSED];                                                    02194500 T  0025
LCARD:                                                                  02194750 T  0027
    IF LISTER OR LISTPTOG THEN PRINTCARD;                               02195000 T  0028
    GO AWAY;                                                            02195250 T  0045
LTAPE:                                                                  02195500 T  0045
    IF LISTER THEN PRINTCARD;                                           02195750 T  0046
%   GO AWAY;                                                            02196000 T  0062
AWAY:                                                                   02196250 T  0062
    END OUTPUTSOURCE;                                                   02196500 T  0062
                                                                  8 IS   63 LONG, NEXT SEG     3


PROCEDURE READACARD;                                                    02196750 T  0202
COMMENT  READACARD READS CARDS FROM EITHER THE CARD READER OR THE       02197000 T  0202
         TAPE MERGING AS REQUESTED AND CREATING A NEW TAPE AND          02197250 T  0202
         LISTING IF REQUESTED.  READACARD ALSO INSERTS A PERCENT        02197500 T  0202
             SIGN AS AN END OF CARD SENTINEL IN COLUMN 73 AND SETS       02197750 T  0202
             FCR,NCR,LCR,TLCR, AND CLCR;                                02198000 T  0202
                 BEGIN                                                  02198250 T  0202
                 PROCEDURE READTAPE;                                    02198500 T  0202
                                                                START OF SEGMENT **********     9
PRT(371) = READTAPE
                          BEGIN                                         02201500 T  0000
             LABEL ENDREADTAPE, EOFT;                                   02201510 C  0000
                                                                START OF SEGMENT **********    10
             READ (TAPE, 10, TBUFF[*])[EOFT];                          02201750 P  0000
PRT(372) = EOFT
PRT(373) = GO TO SOLVER
                            LCR:=MKABS(TBUFF[9]);                       02202000 P  0005
             GO TO ENDREADTAPE;                                         02202010 C  0007
             EOFT:                                                      02202020 C  0007
             DEFINEARRAY[35]:="ND;END."& "E"[1:43:5];                   02202030 C  0008
             DEFINEARRAY[34]:="9999" & "9999"[1:25:23];                 02202040 C  0010
             TLCR:= MKABS(DEFINEARRAY[34]);                             02202050 C  0012
             PUTSEQNO (DEFINEARRAY[33],TLCR-8);                         02202060 C  0014
             TURNONSTOPLIGHT("%", TLCR-8);                              02202070 C  0016
             ENDREADTAPE:                                               02202080 C  0017
                          END READTAPE;                                 02202250 T  0018
                                                                 10 IS   24 LONG, NEXT SEG     9


                 PROCEDURE SEQCOMPARE(TLCR,CLCR, LIB); VALUE LIB; BOOLEAN LIB;  02202500 T  0000
PRT(374) = SEQCOMPARE
```

```
                                        REAL TLCR, CLCR ;                                                02202750 T  0000
                                                                                                         02203000 T  0000
                                                                                                         02203250 T  0000
     BEGIN                                                                                               02203500 T  0000
     MEDIUM:="C "; % CARD READER.                                                                        02203750 T  0003
     IF GT1:=COMPARE(TLCR,CLCR)=0 THEN  % TAPE HAS LOW SEQUENCE NUMB                                     02204000 T  0003
          BEGIN                                                                                          02204250 T  0005
          LCR:=TLCR; LASTUSED:=3;                                                                        02204500 T  0006
          MEDIUM:="T "; % TAPE INPUT.                                                                    02204750 T  0006
          END                                                                                            02205000 T  0009
     ELSE BEGIN                                                                                          02205250 T  0009
          IF GT1 ≠ 1 THEN % TAPE AND CARD HAVE SAME SEQ                                                  02205500 T  0010
               BEGIN                                                                                     02208500 T  0011
               MEDIUM:="P "; % CARD PATCHES TAPE.                                                        02208750 T  0011
               READTAPE;                                                                                 02209000 T  0011
               END;                                                                                      02209250 T  0012
          LCR:=CLCR;                                                                                     02209500 T  0013
          LASTUSED:=2;                                                                                   02209750 T  0013
          END;
     END OF SEQCOMPARE;



     LABEL CARDONLY, CARDLAST, TAPELAST,  EXIT, FIRSTTIME,                                               02210000 T  0015
           EOF, USETHESWITCH,                                                                            02210250 T  0015
           COMPAR, TESTVOID, XIT;                                                                        02210500 T  0015
     SWITCH USESWITCH:=CARDONLY,CARDLAST,TAPELAST,FIRSTTIME;                                             02210750 T  0015
     IF ERRORCOUNT≥ERRMAX THEN ERR(611); % ERR LIMIT EXCEEDED - STOP.                                    02211500 T  0020
USETHESWITCH:                                                                                            02211750 T  0023
     DOLLAR2TOG:=FALSE;                                                                                  02211800 T  0023
     GO TO USESWITCH[LASTUSED];                                                                          02212000 T  0023
     MOVE(1,INFO[LASTUSED,LINKR,LASTUSED,LINKC],                                                         02212250 T  0025
          DEFINEARRAY[DEFINEINDEX*2]);                                                                   02212500 T  0028
     LASTUSED := LASTUSED + 1;                                                                           02212750 T  0029
     NCR := LCR*1;                                                                                       02213000 T  0031
     GO TO XIT;                                                                                          02213250 T  0032
FIRSTTIME:                                                                                               02213500 T  0032
     READ(CARD,10,CBUFF[*]);                                                                             02213750 T  0033
     FCR:=NCR:=(LCR:=MKABS(CBUFF[9]))*9;                                                                 02214000 T  0037
     MEDIUM:="C ";                                                                                       02214100 T  0040
     IF EXAMIN(FCR)≠"$" AND LISTER THEN PRINTCARD;                                                       02214200 T  0041
     PUTSEQNO(INFO[LASTSEQROW,LASTSEQUENCE],LCR);                                                        02214250 T  0061
     TURNONSTOPLIGHT("%",LCR);                                                                           02214500 T  0063
     GO XIT;                                                                                             02214750 T  0064
COMMENT  WE HAVE JUST INITIALIZED CARD INPUT;                                                            02215000 T  0064
CARDONLY:                                                                                                02215250 T  0064
     READ(CARD,10,CBUFF[*]);                                                                             02215500 T  0065
     LCR := MKABS(CBUFF[9]); GO EXIT;                                                                    02215750 T  0069
CARDLAST:                                                                                                02216000 T  0071
     READ(CARD,10,CBUFF[*])[EOF];                                                                        02216250 T  0072
PRT(375) = EOF
     CLCR := MKABS(CBUFF[9]);                                                                            02216500 T  0077
     GO COMPAR;                                                                                          02216750 T  0079
EOF:                                                                                                     02217000 T  0079
     DEFINEARRAY[25]:="ND;END."&"E"[1:43:5];                                                             02217250 T  0080
     DEFINEARRAY[34]:="9999"&"9999"[1:25:23];                                                            02217500 T  0082
     CLCR:=MKABS(DEFINEARRAY[34]);                                                                       02217750 T  0084
     PUTSEQNO(DEFINEARRAY[33],CLCR-8);                                                                   02218000 T  0086
```

```
    TURNONSTOPLIGHT("%",CLCR=8);                                          02218250 T  0088
%                                                                         02218400 T  0089
    GO COMPAR;                                                            02218500 T  0089
COMMENT  THIS RELEASES THE PREVIOUS CARD FROM CARD READER AND             02218750 T  0093
         SETS UP CLCR;                                                    02219000 T  0093
TAPELAST:                                                                 02219250 T  0093
    READTAPE;                                                             02219500 T  0093
COMMENT  THIS RELEASES THE PREVIOUS CARD FROM TAPE AND SETS UP TLCR;      02219750 T  0093
COMPAR:                                                                   02224250 T  0093
    SEQCOMPARE(TLCR,CLCR,FALSE);                                          02224500 T  0094
EXIT:                                                                     02225000 T  0095
    NCR := FCR := LCR + 9;                                                02225250 T  0096
COMMENT  SETS UP NCR AND FCR;                                             02225500 T  0097
    IF EXAMIN(FCR)#"$" THEN % $=CARDS DON"T COUNT.                        02225750 T  0097
      IF COMPARE(MKABS(INFO[LASTSEQROW,LASTSEQUENCE]),LCR)=1 THEN         02226000 T  0099
        BEGIN                                                             02226250 T  0104
        FLAG(610); % SEQUENCE ERROR.                                      02226500 T  0104
        SEQUENCEERROR(LIN);                                               02226750 T  0105
        END;                                                              02227000 T  0106
    CARDNUMBER:=CONV(INFO[LASTSEQROW,LASTSEQUENCE-1],5,8);                02228000 T  0106
    IF LASTUSED=3 THEN                                                    02228050 T  0110
         BEGIN                                                            02228075 T  0110
         IF VOIDTAPE THEN GO USETHESWITCH;                                02228100 T  0111
         IF VOIDTCR#0 THEN                                                02228125 T  0112
           IF COMPARE(LCR,VOIDTCR)=0 THEN GO USETHESWITCH;                02228150 T  0113
         END;                                                             02228175 T  0116
    IF EXAMIN(FCR)="$" THEN                                               02228250 T  0116
         BEGIN                                                            02228500 T  0118
         IF LISTPTOG OR PRINTDOLLARTOG THEN PRINTCARD;                    02228750 T  0118
         NCR:=NCR+32768; DOLLARCARD;                                      02229000 T  0136
COMMENT  DONT FORGET THAT NCR IS NOT WORD MODE, BUT CHAR. MODE POINTER;   02229250 T  0138
         GO USETHESWITCH;                                                 02229500 T  0138
         END;                                                             02229750 T  0140
    IF EXAMIN(FCR)=" " THEN                                               02230000 T  0140
      IF DOLLAR2TOG:=EXAMIN(FCR+32768)="$" THEN                           02230100 T  0141
         BEGIN                                                            02230250 T  0145
         OUTPUTSOURCE;                                                    02230500 T  0145
         NCR:=NCR+65536; % SCAN PAST " $" (CHARACTER MODE).               02230750 T  0146
         DOLLARCARD;                                                      02231000 T  0147
         END;                                                             02231250 T  0147
    IF VOIDING THEN GO USETHESWITCH;                                      02231500 T  0147
    IF VOIDCR#0 THEN                                                      02231750 T  0149
         IF COMPARE(LCR,VOIDCR)>0 THEN VOIDCR:=VOIDPLACE:=0               02232000 T  0150
           ELSE GO USETHESWITCH;                                          02232250 T  0153
    IF VOIDTAPE THEN GO TESTVOID;                                         02232500 T  0154
    IF VOIDTCR#0 THEN                                                     02233000 T  0155
         IF COMPARE(LCR,VOIDTCR)>0 THEN VOIDTCR:=VOIDTPLACE:=0 ELSE       02233500 T  0156
TESTVOID:  IF LASTUSED=3 THEN GO USETHESWITCH;                            02234000 T  0160
    CARDCOUNT:=CARDCOUNT+1;                                               02234500 T  0165
    IF DOLLAR2TOG THEN GO USETHESWITCH;                                   02234600 T  0166
    PUTSEQNO(INFO[LASTSEQROW,LASTSEQUENCE],LCR);                          02234750 T  0167
    OUTPUTSOURCE;                                                         02235000 T  0169
    IF OMITTING THEN GO USETHESWITCH;                                     02235250 T  0170
%                                                                         02235500 T  0171
    TURNONSTOPLIGHT("%",LCR);                                             02235750 T  0171
XIT:                                                                      02237750 T  0172
    END READACARD;                                                        02238000 T  0173
```

```
                                                                                9 IS  177 LONG, NEXT SEG     3


            REAL PROCEDURE CONVERT;                                                     02248000 T  0202
PRT(376) = CONVERT
                        BEGIN   REAL  T;  INTEGER  N;                                   02249000 T  0202
                                                                                START OF SEGMENT **********   11
STACK(F+3) = T
STACK(F+4) = N
                        TLO←0;   THI←                                                   02250000 T  0000
                             T← CONV(ACCUM[1],TCOUNT,N←(COUNT-TCOUNT)MOD 8);            02251000 T  0000
                             FOR N← TCOUNT+N STEP 8 UNTIL COUNT= 1 DO                   02252000 T  0005
                        IF DPTOG THEN                                                   02253000 T  0010
                        BEGIN                                                           02254000 T  0010
                        DOUBLE(THI,TLO,100000000.0,0,×,CONV(ACCUM[1],N,8),0,+,←,        02255000 T  0010
                                  THI,TLO);                                             02256000 T  0015
                        T←THI;                                                          02257000 T  0016
                        END  ELSE                                                       02258000 T  0016
                             T←  T×100000000+ CONV(ACCUM[1],N,8);                       02259000 T  0016
                             CONVERT←T;                                                 02260000 T  0024
                        END;                                                            02261000 T  0024
                                                                                 11 IS   28 LONG, NEXT SEG     3


            REAL STREAM PROCEDURE FETCH(F); VALUE F;                                    02262000 T  0202
PRT(377) = FETCH
                        BEGIN  SI:=F; SI:=SI+8; DI:=LOC FETCH; DS:=WDS  END FETCH;      02263000 T  0202



               PROCEDURE DUMPINFO;                                                      02264000 T  0205
PRT(400) = DUMPINFO
                        BEGIN                                                           02264050 T  0205
                        ARRAY A[0:14]; INTEGER JEDEN,DWA;                               02264100 T  0205
                                                                                START OF SEGMENT **********   12
STACK(F+2) = A
STACK(F+3) = JEDEN
STACK(F+4) = DWA
                        STREAM PROCEDURE OCTALWORDS(S,D,N); VALUE N;                    02264400 T  0001
PRT(401) = OCTALWORDS
                             BEGIN                                                      02264450 T  0001
                             SI:=S; DI:=D;                                              02264500 T  0003
                             N(2(8(DS:=3 RESET; 3(IF SB THEN DS:=1 SET ELSE             02264550 T  0003
                                  DS:=1 RESET; SKIP 1 SB)); DS:=1 LIT " ");DS:=2 LIT" ");02264600 T  0006
                             END OF OCTALWORDS;                                         02264650 T  0009



                        STREAM PROCEDURE ALPHAWORDS(S,D,N); VALUE N;                    02264700 T  0009
PRT(402) = ALPHAWORDS
                             BEGIN                                                      02264750 T  0009
```

```
                    SI:=S; DI:=D;                                                    02264800 T  0010
                    N(2(4(DS:=1 LIT" "; DS:=1 CHR); DS:=1 LIT" "); DS:=2 LIT" ");    02264850 T  0010
                    END OF ALPHAWORDS;                                               02264900 T  0014



               IF NOHEADING THEN DATIME;WRITE(LINE[DBL],<//"ELBAT">);                02264950 T  0014
PRT(403) = *FORMAT DESCRIPTOR*
                                                                          13 IS    6 LONG, NEXT SEG   12
               FOR JEDEN:=0 STEP 6 UNTIL 71 DO                                    02265000 T  0019
                    BEGIN                                                         02265050 T  0020
                    BLANKET(14,A); OCTALWORDS(ELBAT[JEDEN],A,6);                  02265100 T  0020
                    WRITE(LINE[DBL],15,A[*]);                                     02265150 T  0023
                    END;                                                          02265200 T  0027
               BLANKET(14,A); OCTALWORDS(ELBAT[72],A,4);                          02265250 T  0029
               WRITE(LINE[DBL],15,A[*]);                                          02265300 T  0032
               FOR JEDEN:=0 STEP 1 UNTIL NEXTINFO DIV 256 DO                      02265350 T  0036
                    BEGIN                                                         02265400 T  0041
                    WRITE(LINE[DBL],<//"INFO[",I2,",*]">,JEDEN);                  02265450 T  0041
PRT(404) = *FORMAT DESCRIPTOR*
                                                                          14 IS    8 LONG, NEXT SEG   12
PRT(405) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*                                  02265500 T  0048
               FOR DWA:=0 STEP 6 UNTIL 251 DO                                    02265550 T  0049
                    BEGIN                                                        02265600 T  0049
                    BLANKET(14,A); ALPHAWORDS(INFO[JEDEN,DWA],A,6);              02265650 T  0052
                    WRITE(LINE,15,A[*]);                                         02265700 T  0057
                    BLANKET(14,A); OCTALWORDS(INFO[JEDEN,DWA],A,6);              02265750 T  0060
                    WRITE(LINE[DBL],15,A[*]);                                    02265800 T  0065
                    END;                                                         02265850 T  0067
               BLANKET(14,A); ALPHAWORDS(INFO[JEDEN,252],A,4);                   02265900 T  0071
               WRITE(LINE,15,A[*]);                                              02265950 T  0075
               BLANKET(14,A); OCTALWORDS(INFO[JEDEN,252],A,4);                   02266000 T  0079
               WRITE(LINE[DBL],15,A[*]);                                         02266050 T  0083
               END;                                                              02266100 T  0083
          END OF DUMPINFO;
                                                                          12 IS   88 LONG, NEXT SEG    3



DEFINE SKAN = BEGIN                                                              02277000 T  0205
               COUNT:=RESULT:=ACCUM[1]:=0;                                       02278000 T  0205
               SCANNER;                                                          02279000 T  0205
               Q:=ACCUM[1];                                                      02280000 T  0205
               END #;                                                            02281000 T  0205
COMMENT  DOLLARCARD HANDLES THE COMPILER CONTROL CARDS,                          02282000 T  0205
         ALL COMPILER- AND USER-DEFINED OPTIONS ARE KEPT                         02283000 T  0205
         IN THE ARRAY "OPTIONS".                                                 02284000 T  0205
         EACH OPTION HAS A TWO-WORD ENTRY:                                       02285000 T  0205
                                                                                 02286000 T  0205
           WORD          CONTAINS                                                02287000 T  0205
           ----          --------                                                02288000 T  0205
             1           ENTRY FROM ACCUM[1]: 00XZZZZZ, WHERE                    02289000 T  0205
                         X IS THE SIZE OF THE ID AND                             02290000 T  0205
                         ZZZZZ IS THE FIRST FIVE CHARS OF THE ID.                02291000 T  0205
             2           PUSH-DOWN, 47-BIT STACK CONTAINING THE                  02292000 T  0205
```

```
                                        HISTORY OF THE SETTINGS OF THIS OPTION.

                                IN "FINDOPTION", ALL COMPILER-DEFINED OPTIONS ARE USUALLY
                                LOCATED BASED UPON A UNIQUE NUMBER ASSIGNED TO EACH.
                                FOR ALL USER-DEFINED OPTIONS, A SEQUENTIAL TABLE SEARCH IS
                                INITIATED USING "USEROPINX" AS THE INITIAL INDEX INTO THE
                                "OPTIONS" ARRAY. IF THE NUMBER OF COMPILER-DEFINED OPTIONS
                                IS CHANGED, THEN "USEROPINX" MUST BE ACCORDINGLY CHANGED.
                                THE NUMBER OF USER DEFINED OPTIONS ALLOWED CAN BE
                                CHANGED BY CHANGING THE DEFINE "OPARSIZE".
                                THE VARIABLE "OPTIONWORD" CONTAINS THE CURRENT TRUE OR FALSE
                                SETTING OF ALL OF THE COMPILER-DEFINED OPTIONS, ONE BIT PER
                                OPTION.
                        ;
                    BOOLEAN PROCEDURE FINDOPTION(BIT); VALUE BIT; INTEGER BIT;
PRT(406) = FINDOPTION
                        BEGIN
                        LABEL FOUND;

                        REAL ID;
STACK(F+3) = ID
                        OPINX:=2*BIT-4;
                        WHILE ID:=OPTIONS[OPINX:=OPINX+2]#0 DO
                             IF Q=ID THEN GO FOUND;
                        OPTIONS[OPINX]:=Q; % NEW USER-DEFINED OPTION.
                    FOUND:
                         IF OPINX +1>OPARSIZE THEN FLAG(602) ELSE % TOO MANY USER OPTIONS
                        FINDOPTION:=BOOLEAN(OPTIONS[OPINX+1]);
                        END FINDOPTION;
```

```
                    PROCEDURE DOLLARCARD;
                        BEGIN
                        STREAM PROCEDURE RESTORESEQNUM(LCR,INFO);  VALUE LCR;
PRT(407) = RESTORESEQNUM
                            BEGIN
                            DI:=LCR; SI:=INFO; DS:=WDS;
                            END;



                        PROCEDURE SWITCHIT(XBIT); VALUE XBIT; INTEGER XBIT;
PRT(410) = SWITCHIT
                            BEGIN
                            BOOLEAN B,T;

STACK(F+2) = B
STACK(F+3) = T
                            INTEGER SAVEINX;
STACK(F+4) = SAVEINX
                            LABEL XMODE0,XMODE1,XMODE2,XMODE3,XMODE4,ALONG;
                            SWITCH SW:=XMODE0,XMODE1,XMODE2,XMODE3,XMODE4;
```

```
02293000 T  0205
02294000 T  0205
02295000 T  0205
02296000 T  0205
02297000 T  0205
02298000 T  0205
02299000 T  0205
02300000 T  0205
02301000 T  0205
02302000 T  0205
02303000 T  0205
02304000 T  0205
02305000 T  0205
02306000 T  0205
02307000 T  0205

02308000 T  0205
02309000 T  0205
START OF SEGMENT **********    15
02310000 T  0000

02311000 T  0000
02312000 T  0001
02313000 T  0005
02314000 T  0006
02315000 P  0008
02316000 P  0008
02317000 T  0010
02318000 T  0012
 15 IS   15 LONG, NEXT SEG     3

02319000 T  0205
02320000 T  0205
02320200 T  0205
START OF SEGMENT **********    16

02320400 T  0000
02320600 T  0000
02320800 T  0000



02321000 T  0001

02322000 T  0001
02323000 T  0001
START OF SEGMENT **********    17

02324000 T  0000

02325000 T  0000
02326000 T  0000
```

```
                    SETTING:=FINDOPTION(XBIT); SKAN;                                                    02327000 T  0005
                    GO SW[XMODE+1];                                                                     02328000 T  0011
               XMODE0:  % FIRST OPTION ON CARD, BUT NOT SET, RESET, OR POP.                             02329000 T  0013
                    OPTIONWORD:=BOOLEAN(0);                                                             02330000 T  0014
                    FOR SAVEINX:=1 STEP 2 UNTIL OPARSIZE DO OPTIONS[SAVEINX]:=0;                        02331000 T  0014
                    XMODE:=LASTUSED:=1; % CARD INPUT ONLY.                                              02332000 T  0019
               XMODE1:  % NOT FIRST OPTION AND NOT BEING SET, RESET, OR POPPED.                          02333000 T  0020
                    OPTIONS[OPINX+1]:=REAL(TRUE);                                                       02334000 T  0021
                    IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & TRUE[XBIT:1];                       02335000 T  0022
PRT(411) = DYNAMIC DIALS
                    GO ALONG;                                                                           02336000 T  0026
               XMODE2:  % RESET.                                                                        02337000 T  0027
                    OPTIONS[OPINX+1]:=REAL(FALSE & SETTING[1:2:46]);                                    02338000 T  0028
                    IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & FALSE[XBIT:1];                      02339000 T  0030
                    GO ALONG;                                                                           02340000 T  0034
               XMODE3:  % SET.                                                                          02341000 T  0035
                    SAVEINX:=OPINX; % REMEMBER OPTION WE ARE SETTING.                                   02342000 T  0036
                    B:=IF Q="1=0000" THEN BOOLEXP ELSE TRUE;                                            02343000 T  0036
                    OPTIONS[SAVEINX+1]:=REAL(B & SETTING[1:46]);                                        02352000 T  0039
                    IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & B [XBIT:1];                         02353000 T  0042
                    GO ALONG;                                                                           02354000 T  0046
               XMODE4:  % POP.                                                                          02355000 T  0048
                    OPTIONS[OPINX+1]:=REAL(B:=SETTING.[1:46]);                                          02356000 T  0048
                    IF XBIT<USEROPINX THEN OPTIONWORD:=OPTIONWORD & B [XBIT:1];                         02357000 T  0051
               ALONG:                                                                                   02358000 T  0055
                    END SWITCHIT;                                                                       02359000 T  0056
                                                                                    17 IS   59 LONG, NEXT SEG   16
```

```
                LABEL EXIT,AGAIN,SKANAGAIN,LENGTH1,LENGTH2,LENGTH3,LENGTH4,                             02360000 T  0001
                      LENGTH5,LENGTH6,LENGTH7,LENGTH8,LENGTH9,                                          02361000 T  0001
                      WHATISIT,                                                                         02362000 T  0001
                      CARDOPTION,MERGEOPTION;                                                           02363000 T  0001
                SWITCH OPTIONLENGTH:=LENGTH1,WHATISIT,LENGTH3,LENGTH4,LENGTH5,                          02364000 T  0001
                                     LENGTH6,LENGTH7,WHATISIT,LENGTH9,WHATISIT;                         02365000 T  0003
                INTEGER SRESULT,SCOUNT;                                                                 02365100 T  0009
STACK(F+2) = SRESULT
STACK(F+3) = SCOUNT
                ALPHA VOIDRANGE;                                                                        02365200 T  0009
STACK(F+4) = VOIDRANGE
                DOLLARTOG:=TRUE;                                                                        02366000 T  0009
                MOVE(10,ACCUM[0],DEFINEARRAY[0]); % SAVE INFORMATION FOR                                02366100 T  0010
                SCOUNT:=COUNT; SRESULT:=RESULT;   % "TABLE" TO RESUME SCAN.                             02366200 T  0012
                XMODE:=0;                                                                               02367000 T  0014
                PUTSEQNO(INFO[LASTSEQROW,LASTSEQUENCE],LCR);                                            02368000 T  0014
                TURNONSTOPLIGHT("%",LCR);                                                               02369000 T  0016
            SKANAGAIN:                                                                                  02370000 T  0017
                SKAN;                                                                                   02371000 T  0018
            AGAIN:                                                                                      02372000 T  0021
                GO OPTIONLENGTH[MIN(COUNT,10)];                                                         02373000 T  0022
            LENGTH1:                                                                                    02374000 T  0026
                IF Q = "1%0000" THEN GO EXIT;                                                           02375000 T  0027
                IF Q = "1$0000" THEN                                                                    02376000 T  0028
                    BEGIN SWITCHIT(PRINTDOLLARBIT); GO AGAIN END;                                       02377000 T  0029
                IF Q = "1,0000" THEN GO SKANAGAIN;                                                      02378000 T  0033
```

```
    GO WHATISIT;                                                                02379000 T  0034
LENGTH2: % NO OPTIONS OF THIS LENGTH ARE CURRENTLY IMPLEMENTED.                 02380000 T  0036
LENGTH3:                                                                        02381000 T  0036
    IF Q = "3SETOO" THEN                                                        02382000 T  0036
        BEGIN XMODE:=3; GO SKANAGAIN END;                                       02383000 T  0036
    IF Q = "3POPOO" THEN                                                        02384000 T  0040
        BEGIN XMODE:=4; GO SKANAGAIN END;                                       02385000 T  0040
    IF Q = "3NEWOO" THEN                                                        02386000 T  0044
        BEGIN                                                                   02387000 T  0044
        SWITCHIT(NEWBIT);                                                       02388000 T  0045
        IF Q = "4TAPEO" THEN GO SKANAGAIN;                                      02389000 T  0046
        GO AGAIN;                                                               02390000 T  0047
        END;                                                                    02391000 T  0050
    IF Q = "3SEQOO" THEN                                                        02392000 T  0050
        BEGIN SWITCHIT(SEQBIT); GO AGAIN END;                                   02393000 T  0050
    IF Q = "3PRTOO" THEN                                                        02394000 T  0054
        BEGIN SWITCHIT(PRTBIT); GO AGAIN END;                                   02395000 T  0054
    IF Q = "3MCPOO" THEN                                                        02396000 T  0058
        BEGIN SWITCHIT(MCPBIT); GO AGAIN END;                                   02397000 T  0058
    GO WHATISIT;                                                                02398000 T  0062
LENGTH4:                                                                        02399000 T  0062
    IF Q = "4LISTO" THEN                                                        02400000 T  0063
        BEGIN                                                                   02401000 T  0063
        SWITCHIT(LISTBIT);                                                      02402000 T  0064
        GO AGAIN;                                                               02404000 T  0065
        END;                                                                    02405000 T  0067
    IF Q = "4VOIDO" THEN                                                        02406000 T  0067
        BEGIN                                                                   02407000 T  0067
        IF XMODE=0 THEN                                                         02408000 T  0068
            BEGIN                                                               02409000 T  0069
            GETVOID(VOIDRANGE,NCR,LCR,INFO[LASTSEQROW,LASTSEQUENCE]);           02410000 T  0069
            IF VOIDCR=0 THEN VOIDCR:=MKABS(VOIDPLACE) ELSE                      02410500 T  0072
              IF COMPARE(MKABS(VOIDRANGE),VOIDCR)#1 THEN GO TO EXIT;            02411000 T  0075
            MOVE(1,VOIDRANGE,VOIDPLACE);                                        02412000 T  0080
            GO EXIT;                                                            02413000 T  0081
            END;                                                                02414000 T  0082
        SWITCHIT(VOIDBIT);                                                      02415000 T  0082
        GO AGAIN;                                                               02418000 T  0083
        END;                                                                    02419000 T  0083
    IF Q = "4OMITO" THEN                                                        02420000 T  0083
        BEGIN IF NOT DOLLAR2TOG THEN BEGIN PRINTCARD; FLAG(605); END;           02421000 T  0084
            SWITCHIT(OMITBIT); GO AGAIN END;                                    02421100 T  0103
    IF Q = "4CARDO" THEN                                                        02422000 T  0104
        BEGIN                                                                   02423000 T  0105
        Q:="4TAPEO"; % FAKE OUT SWITCHIT.                                       02424000 T  0105
        SWITCHIT(MERGEBIT);                                                     02425000 T  0106
        IF XMODE#2 THEN MERGETOG:=NOT MERGETOG;                                 02425500 T  0107
        OPTIONS[2*MERGEBIT-1]:=                 % CARD IS                      02426000 T  0110
          REAL(SETTING & (MERGETOG)[47:1]);  % INVERSE OF MERGE.                02427000 T  0112
        IF MERGETOG THEN GO MERGEOPTION;                                        02428000 T  0114
CARDOPTION:                                                                     02429000 T  0116
        LASTUSED:=1;                                                            02430000 T  0116
        GO AGAIN;                                                               02431000 T  0116
        END;                                                                    02432000 T  0120
    IF Q = "4TAPEO" THEN                                                        02433000 T  0120
        BEGIN                                                                   02434000 T  0120
        SWITCHIT(MERGEBIT);                                                     02435000 T  0121
```

```
                    IF NOT MERGETOG THEN GO CARDOPTION;                                  02436000 T  0122
               MERGEOPTION:                                                              02437000 T  0123
                    LASTUSED:=2; % NEXT CARD IS READ FROM READER.                        02438000 T  0124
                    IF MAXTLCR=0 THEN                                                    02439000 T  0124
                         BEGIN                                                           02440000 T  0125
                         INTEGER STREAM PROCEDURE FEJ(F,T); VALUE T;                     02441000 T  0126
PRT(412) = *SEGMENT DESCRIPTOR*
                                                                  START OF SEGMENT **********   18
PRT(413) = FEJ
                              BEGIN                                                      02442000 T  0000
                              SI:=F; DI:=LOC T; DS:=WDS;                                 02443000 T  0000
                              SI:=T; SI:=SI"16; DI:=LOC FEJ; DS:=WDS;                    02444000 T  0000
                              END FEJ;                                                   02445000 T  0001



                         STREAM PROCEDURE FIX(F,T); VALUE T;                             02446000 T  0002
PRT(414) = FIX
                              BEGIN                                                      02447000 T  0002
                              SI:=F; SI:=SI"24; DI:=LOC T; DS:=WDS;                      02448000 T  0003
                              DI:=T; DI:=DI+47; SKIP 4 DB; DS:=2 RESET;                  02449000 T  0004
                              2(DI:=DI+48); DS:=8 LIT"00#01+0#";                         02450000 T  0005
                              END FIX;                                                   02451000 T  0007



                         IF GT1:=FEJ(TAPE,0)=10 THEN                                     02452000 T  0007
                              BEGIN                                                      02453000 T  0010
                              REWIND(TAPE); FIX(TAPE,0);                                 02454000 T  0011
                              END;                                                       02455000 T  0014
                         MAXTLCR:=GT1+TLCR:=9+MKABS(TBUFF[0]);                           02456000 T  0014
                         READ(TAPE,10,TBUFF[*]); % INITIALIZE TAPE INPUT.                02457000 T  0017
                         LASTUSED:=2;                                                    02458000 T  0022
                         END;                                                            02459000 T  0022
PRT(415) = *SEGMENT DESCRIPTOR*
                                                                  18 IS   24 LONG, NEXT SEG   16
                    GO AGAIN;                                                            02460000 T  0128
                    END;                                                                 02461000 T  0128
               IF Q = "4PAGEO" THEN                                                      02462000 T  0128
                    BEGIN                                                                02463000 T  0129
                    IF LISTER THEN WRITE(LINE[PAGE]);                                    02464000 T  0129
                    GO SKANAGAIN;                                                        02465000 T  0134
                    END;                                                                 02466000 T  0136
               IF Q = "4INFOO" THEN                                                      02467000 T  0136
                    BEGIN DUMPINFO; GO SKANAGAIN END;                                    02468000 T  0136
               IF Q = "4SEGSO" THEN                                                      02469000 T  0140
                    BEGIN SWITCHIT(SEGSBIT); GO AGAIN END;                               02470000 T  0140
               IF Q = "4NESTO" THEN                                                      02471000 T  0144
                    BEGIN SWITCHIT(NESTBIT); GO AGAIN END;                               02472000 T  0144
               IF Q = "4DECKO" THEN                                                      02473000 T  0148
                    BEGIN SWITCHIT(DECKBIT); GO AGAIN END;                               02474000 T  0148
               GO WHATISIT;                                                              02475000 T  0152
          LENGTH5:                                                                       02476000 T  0152
               IF Q = "5RESET" THEN                                                      02477000 T  0153
                    BEGIN XMODE:=2; GO SKANAGAIN END;                                    02478000 T  0153
```

```
IF Q = "5LISTP" THEN
     BEGIN SWITCHIT(LISTPBIT); GO AGAIN; END;
IF Q = "5VOIDT" THEN
     BEGIN
     IF XMODE=0 THEN
          BEGIN
          GETVOID(VOIDRANGE,NCR,LCR,INFO[LASTSEQROW,LASTSEQUENCE]);
          IF VOIDTCR=0 THEN VOIDTCR:=MKABS(VOIDTPLACE) ELSE
            IF COMPARE(MKABS(VOIDRANGE),VOIDTCR)#1 THEN GO TO EXIT;
          MOVE(1,VOIDRANGE,VOIDTPLACE);
          GO EXIT;
          END;
     SWITCHIT(VOIDTBIT);
     GO AGAIN;
     END;
IF Q = "5CHECK" THEN
     BEGIN SWITCHIT(CHECKBIT); GO AGAIN END;
IF Q = "5LIMIT" THEN
     BEGIN
     SKAN;
     IF RESULT#3 THEN % SHOULD BE NUMBER.
          BEGIN FLAG(600); GO AGAIN END;
     ERRMAX:=CONV(ACCUM[1],0,ACCUM[1].[12:6]);
     GO SKANAGAIN;
     END;
IF Q = "5PUNCH" THEN
     BEGIN SWITCHIT(PUNCHBIT); GO AGAIN; END;
IF Q = "5PURGE" THEN
     BEGIN SWITCHIT(PURGEBIT); GO AGAIN; END;
IF Q = "5LISTA" THEN
     BEGIN
     SWITCHIT(LISTABIT);
     GO AGAIN;
     END;
IF Q = "5STUFF" THEN
     BEGIN SWITCHIT(STUFFBIT); GO AGAIN; END;
GO WHATISIT;
LENGTH6:
IF Q = "6SEQER" THEN
     BEGIN SWITCHIT(SEQERRBIT); GO AGAIN END;
IF Q = "6SINGL" THEN
     BEGIN SWITCHIT(SINGLBIT); GO AGAIN  END;
IF Q = "6SEQXE" THEN
     BEGIN
     SEQXEQTOG:=XMODE#2 AND XMODE# 4 OR SEQXEQTOG;%NEVER RESET.
     IF BUILDLINE.[45:1] THEN BUILDLINE:=TRUE;
     GO SKANAGAIN;
     END;
IF Q = "6DEBUG" THEN
     BEGIN
     SWITCHIT(DEBUGBIT);
     IF DEBUGTOG THEN
          IF WOP[0]=0 THEN
               BEGIN
               FILL WOP[*] WITH
     "LITC","    ",
```

```
02479000 T  0157
02480000 T  0157
02481000 T  0161
02482000 T  0161
02483000 T  0162
02484000 T  0163
02485000 T  0163
02485500 T  0166
02486000 T  0169
02487000 T  0174
02488000 T  0175
02489000 T  0176
02490000 T  0176
02493000 T  0177
02494000 T  0177
02495000 T  0177
02496000 T  0178
02497000 T  0181
02498000 T  0181
02499000 T  0182
02500000 T  0186
02501000 T  0186
02502000 T  0190
02503000 T  0193
02504000 T  0193
02505000 T  0193
02506000 T  0194
02507000 T  0198
02508000 T  0198
02509000 T  0202
02510000 T  0202
02511000 T  0203
02513000 T  0204
02514000 T  0206
02515000 T  0206
02516000 T  0206
02517000 T  0210
02518000 T  0210
02519000 T  0211
02520000 T  0211
02521000 T  0215
02522000 T  0215
02523000 T  0219
02524000 T  0219
02525000 T  0220
02526000 T  0223
02527000 T  0225
02528000 T  0227
02529000 T  0227
02530000 T  0227
02531000 T  0228
02533000 T  0229
02534000 T  0229
02535000 T  0231
02536000 T  0231
02537000 T  0232
```

```
        "OPDC","DESC",
 11, "NOP ", 12, "PRT ", 13, "DEL ", 16, "ADD ", 18, "PRL ", 19, "LNG ",
 21, "GEQ ", 22, "BBC ", 24, "INX ", 35, "LOR ", 37, "GTR ", 38, "BFC ",
 39, "BTN ", 40, "COC ", 48, "SUB ", 64, "MUL ", 67, "LND ", 68, "STD ",
 69, "NEQ ", 71, "XIT ", 72, "MKS ",128, "DIV ",130, "COM ",131, "LQV ",
132, "SND ",133, "XCH ",134, "CHS ",167, "RTS ",168, "CDC ",260, "LOD ",
261, "DUP ",278, "LBC ",294,"LFC ",322,"ZP1 ",384, "IDV ",532, "ISD ",
533, "LEQ ",534, "BBW ",548, "ISN ",549, "LSS ",550, "BFW ",581, "EQL ",
582, "SSP ",790, "LBU ",806, "LFU ",896, "RDV ",
1003,"SCI ",1004,"SAN ",1019,"SCS ",
 1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,
        1023,1023,1023,1023,1023,1023,1023,1023,1023,1023,1023, 1023;

                FILL COP[*] WITH  % CHARACTER MODE MNEMONICS
  0,"   ",0,0,

  0,"EXC ", 2,"BSD ", 3,"BSS ", 4,"RDA ", 5,"TRW ",6,"SED ", 7,"TDA ",
 12,"SDA ",13,"SSA ",14,"SFD ",15,"SRD ",18,"SES ",20,"TEQ ",21, "TNE ",
 22,"TEG ",23,"TGR ",24,"SRS ",25,"SFS ",28,"TEL ",29,"TLS ",30,"TAN ",
 31,"BIT ",32,"INC ",33,"STC ",34,"SEC ",35,"CRF ",36,"JNC ",37,"JFC ",
 38,"JNS ",39,"JFW ",40,"RCA ",41,"ENS ",42,"BNS ",43,"RSA ",44,"SCA ",
 45,"JRC ",46,"TSA ",47,"JRV ",48,"CEQ ",49,"CNE ",50,"CEG ",51,"CGR ",
52,"BIT ",53,"BIR ",54,"DCV ",55,"ICV ",56,"CEL ",57,"CLS ",58,"FSU ",
 59,"FAD ",60,"TRP ",61,"TRN ",62,"TRZ ",63,"TRS ",64,0,64,0,64,0,64,0,
 64,0,64,0,64,0,64,0;

                END;
        GO AGAIN;
        END;
    IF Q = "6FORMA" THEN
        BEGIN SWITCHIT(FORMATBIT); GO AGAIN; END;
    GO WHATISIT;
LENGTH7:
%   IF Q = "7INCLU" THEN
%       BEGIN DOLLARCARD:=STARTINCLUDING;  GO EXIT;  END;
%   IF Q = "7INCLN" THEN
%       BEGIN SWITCHIT(NEWINCLBIT);  GO AGAIN; END;
LENGTH8: % NO OPTIONS OF THIS LENGTH ARE CURRENTLY IMPLEMENTED.
LENGTH9:
    IF Q = "9INTRI" THEN
        BEGIN
        INTOG:=XMODE#2 AND XMODE# 4 OR INTOG; % NEVER RESET.
        GO SKANAGAIN;
        END;
WHATISIT:
    IF RESULT=3 THEN
        BEGIN
        BASENUM:=CONV(ACCUM[1],0,ACCUM[1].[12:6]);
        TOTALNO:=-10;
        NEWBASE:=TRUE;
        GO SKANAGAIN;
        END;
    IF RESULT=2 THEN
        BEGIN
        IF Q = "1+0000" THEN
            BEGIN
            SKAN;
```

```
02538000 T  0233
02539000 T  0233
02540000 T  0233
02541000 T  0233
02542000 T  0233
02543000 T  0233
02544000 T  0233
02545000 T  0233
02546000 T  0233
02547000 T  0233
02548000 T  0233
02549000 T  0233
19 IS  128 LONG, NEXT SEG   16
02550000 T  0233
02551000 T  0234
START OF SEGMENT **********  20
02552000 T  0235
02553000 T  0235
02554000 T  0235
02555000 T  0235
02556000 T  0235
02557000 T  0235
02558000 T  0235
02559000 T  0235
02560000 T  0235
20 IS  128 LONG, NEXT SEG   16
02563000 T  0235
02564000 T  0235
02565000 T  0237
02566000 T  0237
02567000 T  0237
02568000 T  0241
02569000 T  0241
02570000 T  0242
02571000 T  0242
02572000 T  0242
02573000 T  0242
02574000 T  0242
02575000 T  0242
02576000 T  0242
02576500 T  0242
02577000 T  0243
02577250 T  0247
02577500 T  0249
02578000 T  0249
02579000 T  0249
02580000 T  0249
02581000 T  0250
02582000 T  0253
02583000 T  0254
02584000 T  0255
02585000 T  0255
02586000 T  0255
02587000 T  0256
02588000 T  0257
02589000 T  0257
02590000 T  0258
```

```
            IF RESULT=3 THEN                                                    02591000 T  0262
             ADDVALUE:=CONV(ACCUM[1],0,ACCUM[1].[12:6])                         02592000 T  0262
         ELSE FLAG(600); % NUMBER EXPECTED.                                     02593000 T  0265
            END;                                                                02594000 T  0268
        GO SKANAGAIN;                                                           02595000 T  0268
        END;                                                                    02596000 T  0269
COMMENT  DID NOT RECOGNIZE OPTION;                                              02597000 T  0269
    IF RESULT#1 THEN % NOT AN IDENTIFIER.                                       02598000 T  0269
        BEGIN FLAG(601); GO SKANAGAIN END;                                      02599000 T  0270
    SWITCHIT(USEROPINX); % USEROPINX MEANS A USER-DEFINED OPTION.               02600000 T  0271
    GO AGAIN;                                                                   02601000 T  0272
 EXIT:                                                                          02602000 T  0273
    LISTER:=DEBUGTOG OR LISTOG OR LISTATOG;                                     02602500 T  0273
    MOVE(10,DEFINEARRAY[0],ACCUM[0]); % RESTORE INFORMATION FOR                 02602600 T  0276
    COUNT:=SCOUNT; RESULT:=SRESULT;    % "TABLE" TO RESUME SCAN.                02602700 T  0278
    RESTORESEQNUM(LCR,INFO[LASTSEQROW,LASTSEQUENCE]); % FOR VOID TESTS         02602800 T  0279
    DOLLARTOG:=FALSE;                                                           02603000 T  0281
    END DOLLARCARD;                                                             02604000 T  0282
                                                                  16 IS  285 LONG, NEXT SEG     3


COMMENT  TABLE IS THE ROUTINE THAT MOST CODE IN THE COMPILER                    02605000 T  0205
       USES WHEN IT IS DESIRED TO SCAN ANOTHER LOGICAL QUANTITY.                02606000 T  0205
       THE RESULT RETURNED IS THE CLASS OF THE ITEM DESIRED.                    02607000 T  0205
       TABLE MAINTAINS THE VARIABLES I AND NXTELBT AND THE ARRAY                02608000 T  0205
       ELBAT.  ELBAT AND I ARE PRINCIPAL VARIABLES USED FOR                     02609000 T  0205
       COMUNICATION BETWEEN TABLE AND THE OUTSIDE WORLD.  NXTELBT               02610000 T  0205
       IS ALMOST EXCLUSIVELY USED BY TABLE, ALTHOUGH AN OCCASION-               02611000 T  0205
       AL OTHER USE IS MADE IN ORDER TO FORGET THAT SOMETHING WAS               02612000 T  0205
       SCANNED.  (SEE, FOR EXAMPLE, COMPOUNDTAIL). FOR FURTHER                  02613000 T  0205
       GENERAL DISCUSSION SEE THE DECLARATION OF THESE VARIABLES.               02614000 T  0205
         THE PARAMETER P IS THE ACTUAL INDEX OF THE QUANTITY                    02615000 T  0205
       DESIRED (USUALLY I-1,I, OR I+1).                                         02616000 T  0205
         THE GENERAL PLAN OF TABLE IS THIS:                                     02617000 T  0205
           I) IF P < NXTELBAT GO ON TO III).                                    02618000 T  0205
          II) PROCESS ONE QUANTITY.                                             02619000 T  0205
            A) SCAN.                                                            02620000 T  0205
            B) TEST FOR IDENTIFIER, NUMBER, OR SPECIAL CHARACTER.               02621000 T  0205
             1) IDENTIFIER = LOOKUP IN DIRECTORY AND PROCESS                    02622000 T  0205
                  IN SPECIAL MANNER IF COMMENT OR DEFINED ID.                   02623000 T  0205
             2) NUMBER = PROCESS INTEGER PART, FRACTIONAL PART,                 02624000 T  0205
                  AND EXPONENT PART.                                            02625000 T  0205
             3) TEST IF SPECIAL CHARACTER REQUIRES SPECIAL                      02626000 T  0205
                  PROCESSING = OTHERWISE GET ELBAT WORD FROM                    02627000 T  0205
                  SPECIAL.                                                      02628000 T  0205
            C) LOAD ELBAT AND INCREMENT NXTELBT.                                02629000 T  0205
            D) IF ELBAT IS FULL ADJUST ELBAT, NXTELBT, I, AND P.                02630000 T  0205
            E) GO BACK TO I).                                                   02631000 T  0205
         III) RETURN WITH CLASS OF ELBAT[P].                                    02632000 T  0205
       FURTHER DETAILS ARE GIVEN IN BODY OF TABLE.                              02633000 T  0205
                                                                                02634000 T  0205
    ;                                                                           02635000 T  0205
  INTEGER PROCEDURE TABLE(P); VALUE P; INTEGER P;
PRT(416) = TABLE                                                                02636000 T  0205
          BEGIN                                                                 02637000 T  0205
          LABEL PERCENT,SPECIALCHAR,COMPLETE,COLON,DOT,ATSIGN,QUOTE,
```

```
          STRNGXT,MOVEIT,ARGH,FINISHNUMBER,                                    02638000 T  0000
          SCANAGAIN,FPART,EPART,IPART,IDENT,ROSE,COMPOST,DOLLAR,RTPAREN,       02639000 T  0000
          CROSSHATCH,NUMBEREND;                                                02640000 T  0000
     SWITCH SPECIALSWITCH:=PERCENT,DOLLAR,DOT,ATSIGN,COLON,QUOTE,              02641000 T  0000
                          RTPAREN,CROSSHATCH;                                  02642000 T  0002
     SWITCH RESULTSWITCH:=IDENT,SPECIALCHAR,IPART;                             02643000 T  0007
     WHILE P ≥ NXTELBT                                                         02644000 T  0012
     DO   BEGIN                                                                02645000 T  0013
SCANAGAIN:                                                                     02646000 T  0014
          COUNT:=RESULT:=ACCUM[1]:=0; SCANNER;                                 02647000 T  0015
          GO RESULTSWITCH[RESULT];                                             02648000 T  0017
ARGH:                                                                          02649000 T  0019
          Q:=ACCUM[1]; FLAG(141); GO SCANAGAIN;                                02650000 T  0020
SPECIALCHAR:                                                                   02651000 T  0022
          GT1:=ACCUM[1],[18:6] - 2;                                            02652000 T  0023
          ENDTOG:=GT1 = 57 AND ENDTOG;                                         02653000 T  0025
COMMENT   OBTAIN ACTUAL CHARACTER FROM ACCUM;                                  02654000 T  0026
          T:=SPECIAL[GT1&GT1[42:41:3]];                                        02655000 T  0026
COMMENT   NOTICE COMPRESSION TECHNIQUE USED TO SHORTEN TABLE OF                02656000 T  0028
          ELBAT WORDS FOR SPECIAL CHARACTERS;                                  02657000 T  0028
          IF GT1:=T.INCR = 0 THEN GO COMPLETE;                                 02658000 T  0028
          GO SPECIALSWITCH[GT1];                                               02659000 T  0031
COMMENT   INCR FIELD OF SPECIAL CHARACTER IS NON-ZERO FOR SPECIAL               02660000 T  0033
          CHARACTERS REQUIRING SPECIAL HANDLING.  INCR IS SWITCHED             02661000 T  0033
          ON TO OBTAIN DISCRIMINATION;                                         02662000 T  0033
COLON:    RESULT:=7; SCANNER; COMMENT ELIMINATE BLANKS - CHECKING              02663000 T  0033
                                      FOR := IN PLACE OF + ;                   02664000 T  0034
          IF EXAMIN (NCR) = "=" THEN                                           02665000 T  0034
               BEGIN RESULT:=0; SCANNER; T:=SPECIAL[13]  END;                  02666000 T  0036
          RESULT:=2; GO COMPLETE;                                              02667000 T  0038
DOT:                                                                           02668000 T  0040
          IF EXAMIN(NCR)>9 OR ENDTOG THEN GO COMPLETE;                         02680000 T  0040
          NHI:=NLO:=0;                                                         02681000 T  0043
          C:=0; GO FPART;                                                      02682000 T  0044
ATSIGN:                                                                        02683000 T  0045
          RESULT:=0; SCANNER;          % SCAN PAST "@",                       02684000 T  0046
          IF COUNT>17 THEN GO ARGH;    % 16 CHARS, + "@",                     02685000 T  0047
          IF OCTIZE(ACCUM[1],C,17-COUNT,COUNT-1) THEN                          02686000 T  0048
               BEGIN  Q:=ACCUM[1]; FLAG(521); GO SCANAGAIN  END;               02686500 T  0051
          GO NUMBEREND;                                                        02687000 T  0054
COMMENT   DOT AND ATSIGN ENTER NUMBER CONVERSION AT CORRECT SPOT;              02689000 T  0055
QUOTE:                                                                         02690000 T  0055
          COUNT:=0;                                                            02691000 T  0055
          T:=IF STREAMTOG THEN 63                                              02692000 T  0055
     ELSE IF REAL(STREAMTOG)>1 THEN 8 ELSE 7;                                  02692500 T  0056
          DO   BEGIN                                                           02693000 T  0060
               RESULT:=5; SCANNER;                                             02694000 T  0060
               IF COUNT>T THEN                                                 02695000 T  0061
                    BEGIN  Q:=ACCUM[1]; FLAG(520); GO SCANAGAIN  END;          02696000 T  0062
               END UNTIL EXAMIN(NCR) = """;                                    02697000 T  0064
          Q:=ACCUM[1]; RESULT:=5; SCANNER; COUNT:=COUNT-1;                     02698000 T  0067
          IF COUNT<0 THEN COUNT:=COUNT+64;                                     02699000 T  0070
          ACCUM[1]:=Q; RESULT:=4;                                              02700000 T  0073
STRNGXT:  T:=C:=0;                                                             02701000 T  0075
          IF COUNT < 8 THEN                                                    02703000 T  0076
MOVEIT:                                                                        02704000 T  0077
```

```
          MOVECHARACTERS(COUNT,ACCUM[1],3,C,8-COUNT);                                  02705000 T  0078
          T.CLASS:=STRNGCON;                                                           02705100 T  0080
          GO COMPLETE;                                                                 02705200 T  0082
COMMENT   CROSSHATCH HANDLES TWO SITUATIONS:                                           02707000 T  0082
          THE CROSSHATCH AT END OF DEFINE DECLARATIONS AND                             02708000 T  0082
          THE CROSSHATCH AT END OF ALPHA REPRESENTING DEFINED IDS.                     02709000 T  0082
          THE TWO CASES ARE PROCESSED DIFFERENTLY.  THE FIRST CASE                     02710000 T  0082
          MERELY PLACES THE CROSSHATCH IN ELBAT.  THE SECOND CASE                      02711000 T  0082
          CAUSES AN EXIT FROM SCANNING THE ALPHA FOR THE DEFINED ID.                   02712000 T  0082
          FOR A FULL DISCUSSION SEE DEFINEGEN;                                         02713000 T  0082
CROSSHATCH:                                                                            02714000 T  0082
          IF DEFINECTR#0 THEN GO COMPLETE;                                             02715000 T  0083
          PUTSEQNO(GT1,LCR);                                                           02716000 T  0084
          TURNONSTOPLIGHT(0,LCR);                                                      02717000 T  0085
          IF DEFINEINDEX = 0 THEN GO ARGH;                                             02718000 T  0086
          LCR:=(GT1:=DEFINEARRAY[DEFINEINDEX-1]) DIV 262144;                           02719000 T  0087
          NCR:=GT1 MOD 262144;                                                         02720000 T  0090
          GT2:=0&(T:=DEFINEARRAY[DEFINEINDEX:=DEFINEINDEX-3])[33:18:15];               02721000 T  0091
          LASTUSED:=T.[33:15];                                                         02722000 T  0094
          FOR GT1:=1 STEP 1 UNTIL GT2 DO                                               02723000 T  0095
               BEGIN                                                                   02723500 T  0098
               STACKHEAD[(T:=TAKE(LASTINFO+1)),[12:36] MOD 125]:=                      02724000 T  0098
                         TAKE(LASTINFO).LINK;                                          02725000 T  0101
               LASTINFO:=(NEXTINFO:=LASTINFO)-T.PURPT;                                 02726000 T  0102
               END;                                                                    02727000 T  0105
          GO SCANAGAIN;                                                                02728000 T  0107
DOLLAR:   COMMENT THIS CODE HANDLES CONTROL CARDS;                                     02729000 T  0107
          DOLLARCARD;                                                                  02730000 T  0108
PERCENT:  IF NCR # FCR THEN READACARD;                                                 02731000 T  0108
          GO SCANAGAIN;                                                                02737000 T  0110
COMMENT   MOST PERCENT SIGNS ACTING AS END OF CARD SENTINELS GET TO                    02738000 T  0111
          PERCENT.  PERCENT READS THE NEXT CARD AND STARTS OVER.  A                    02739000 T  0111
          SIDE EFFECT IS THAT ALL CHARACTERS ON A CARD ARE IGNORED                     02740000 T  0111
          AFTER A FREE PERCENT SIGN (ONE NOT IMBEDDED IN A STRING OR                   02741000 T  0111
          COMMENT);                                                                    02742000 T  0111
COMMENT   MIGHT BE FUNNY COMMA - HANDLE HERE;                                          02743000 T  0111
RTPAREN:  RESULT:=7; SCANNER;                                                          02744000 T  0111
          IF EXAMIN(NCR) = """ THEN                                                    02745000 T  0113
               BEGIN                                                                   02746000 T  0115
               RESULT:=0; SCANNER;                                                     02747000 T  0115
               DO   BEGIN                                                              02748000 T  0116
                    RESULT:=5; SCANNER                                                 02749000 T  0117
                    END UNTIL EXAMIN(NCR) = """;                                       02750000 T  0117
               RESULT:=0; SCANNER;                                                     02751000 T  0120
               RESULT:=7; SCANNER;                                                     02752000 T  0121
               IF EXAMIN(NCR) # "(" THEN GO ARGH;                                      02753000 T  0123
               RESULT:=0; SCANNER;  Q:=ACCUM[1];                                       02754000 T  0125
               T:=SPECIAL[24]                                                          02755000 T  0127
               END;                                                                    02756000 T  0127
          RESULT:=2; GO COMPLETE;                                                      02757000 T  0128
IPART:    TCOUNT:=0; C:=CONVERT;                                                       02758000 T  0129
%         RESULT:=7; SCANNER; % DEBLANK.                                               02759000 T  0131
%         IF DEFINECTR=0 THEN                                                          02760000 T  0131
%          IF (C=3 OR C=4) AND EXAMIN(NCR)="""" THEN %OCTAL OR HEX STRING.             02761000 T  0131
%               BEGIN    INTEGER SIZ;                                                02762000 T  0131
%               RESULT:=5; SCANNER; % SKIP QUOTE.                                    02763000 T  0131
%               COUNT:=0;                                                            02764000 T  0131
```

```
%               DO   BEGIN                                                    02765000 T  0131
%                    RESULT:=5; SCANNER;                                      02766000 T  0131
%                    IF COUNT > SIZ:=48 DIV C THEN % > 1 WORD LONG.           02767000 T  0131
%                         BEGIN  ERR(520); GO SCANAGAIN  END;                 02768000 T  0131
%                    END UNTIL EXAMIN(NCR)="""";                              02769000 T  0131
%               Q:=ACCUM[1]; RESULT:=5; SCANNER; COUNT:=COUNT-1;              02770000 T  0131
%               IF C=3 THEN % OCTAL STRING.                                  02771000 T  0131
%                 IF OCTIZE(ACCUM[1],ACCUM[4],16-COUNT,COUNT) THEN           02772000 T  0131
%                   FLAG(521) % NON OCTAL CHARACTER IN STRING.               02773000 T  0131
%          ELSE ELSE IF HEXIZE(ACCUM[1],ACCUM[4],12-COUNT,COUNT) THEN        02774000 T  0131
%                    FLAG(521); % NON CHARACTER IN HEX STRING.               02775000 T  0131
%               IF COUNT < SIZ THEN                                          02776000 T  0131
%                    BEGIN                                                   02777000 T  0131
%                    C:=ACCUM[4]; GO FINISHNUMBER;                           02778000 T  0131
%                    END;                                                    02779000 T  0131
%               T.INCR:=COUNT:=8; T.CLASS:=STRING;                           02780000 T  0131
%               MOVECHARACTERS(8,ACCUM[4],0,ACCUM[1],3);                     02781000 T  0131
%               GO COMPLETE;                                                 02782000 T  0131
%               END OCTAL OR HEX STRING;                                     02783000 T  0131
          IF OPTOG THEN                                                     02784000 T  0131
               BEGIN NHI:=THI; NLO:=TLO; END;                               02785000 T  0132
          IF EXAMIN(NCR)="." THEN                                           02786000 T  0134
               BEGIN                                                        02787000 T  0135
               RESULT:=0; SCANNER;                                          02788000 T  0136
               C:=1.0* C;                                                   02789000 T  0137
FPART:         TCOUNT:=COUNT;                                               02790000 T  0138
               IF EXAMIN(NCR)≤9 THEN                                        02791000 T  0139
                    BEGIN                                                   02792000 T  0141
                    RESULT:=0; SCANNER;                                     02793000 T  0142
                    IF DPTOG THEN                                           02794000 T  0143
                         BEGIN                                              02795000 T  0143
                         DOUBLE(CONVERT,TLO,TEN[(COUNT-TCOUNT)MOD 12],      02796000 T  0144
                                   0,/,:=,THI,TLO);                         02797000 T  0147
                         FOR T:=12 STEP 12 UNTIL COUNT - TCOUNT DO          02798000 T  0148
                           DOUBLE(THI,TLO,TEN[12],0,/,:=,THI,TLO);          02799000 T  0153
                         DOUBLE(THI,TLO,NHI,NLO,+,:=,NHI,NLO);              02800000 T  0158
                         C:=NHI                                             02801000 T  0160
                         END                                                02802000 T  0160
                    ELSE C:=TEN[TCOUNT-COUNT]*CONVERT+C;                    02803000 T  0161
                    END                                                     02804000 T  0164
               END;                                                         02805000 T  0164
          RESULT:=7; SCANNER;                                               02806000 T  0164
          IF EXAMIN(NCR)="@" THEN                                           02807000 T  0166
               BEGIN                                                        02808000 T  0167
               RESULT:=0; SCANNER;                                          02809000 T  0168
EPART:         TCOUNT:=COUNT;                                               02810000 T  0169
               C:=C*1.0;                                                    02811000 T  0170
               RESULT:=7; SCANNER;                                          02812000 T  0172
               IF T:=EXAMIN(NCR)>9 THEN                                     02813000 T  0173
                    BEGIN                                                   02815000 T  0175
                    RESULT:=0; SCANNER;                                     02816000 T  0176
                    TCOUNT:=COUNT;                                          02817000 T  0177
                    END;                                                    02818000 T  0178
               RESULT:=0; SCANNER;                                          02820000 T  0178
               Q:=ACCUM[1];                                                 02822000 T  0179
               IF GT1:=T:=(IF T=""""THEN -CONVERT ELSE CONVERT)<=46 OR      02823000 T  0180
                 T>69 THEN FLAG(269)                                        02824000 T  0185
```

```
                ELSE BEGIN                                                      02825000 T  0186
                     T:=TEN[T];                                                 02826000 T  0189
                     IF ABS(O&C[42:3:6]&C[1:2:1]+O&T[42:3:6]&T[1:2:1]           02827000 T  0190
                       + 12) >63 THEN FLAG(269)                                 02828000 T  0195
                ELSE IF DPTOG THEN                                              02829000 T  0197
                       IF GT1<0 THEN                                            02830000 T  0198
                          BEGIN                                                 02831000 T  0199
                          GT1:=-GT1;                                            02832000 T  0200
                          DOUBLE(NHI,NLO,TEN[GT1 MOD 12],0,/,:=,NHI,NLO);       02833000 T  0201
                          FOR GT2:=12 STEP 12 UNTIL GT1 DO                      02834000 T  0205
                            DOUBLE(NHI,NLO,TEN[12],0,/,:=,NHI,NLO);             02835000 T  0206
                          END                                                   02836000 T  0211
                     ELSE BEGIN                                                 02837000 T  0211
                          DOUBLE(NHI,NLO,TEN[GT1 MOD 12],0,X,:=,NHI,NLO);       02838000 T  0212
                          FOR GT2:=12 STEP 12 UNTIL GT1 DO                      02839000 T  0215
                            DOUBLE( NHI,NLO,TEN[12],0,X,:=,NHI,NLO);            02840000 T  0217
                          END                                                   02841000 T  0222
                ELSE C:=C*T;                                                    02842000 T  0222
                     END;                                                       02843000 T  0224
                END;                                                            02844000 T  0224
NUMBEREND:                                                                      02845000 T  0224
          Q:=ACCUM[1]; RESULT:=3;                                               02846000 T  0225
FINISHNUMBER:                                                                   02847000 T  0226
          T:=0;                                                                 02848000 T  0227
          IF C.[1:37]=0 THEN                                                    02849000 T  0227
               BEGIN   T.CLASS:=LITNO ; T.ADDRESS:=C  END                       02850000 T  0229
          ELSE T.CLASS:=NONLITNO ;                                              02851000 T  0233
          GO COMPLETE;                                                          02852000 T  0235
COMMENT   THE CODE BETWEEN IDENT AND COMPOST DOES A LOOKUP IN INFO,             02853000 T  0236
          IF QUANTITY IS NOT FOUND THE ELBAT WORD EXPECTS TO BE                 02854000 T  0236
          ZERO,  THE SCRAMBLE FOR APPROPRIATE STACK IS FIRST THING              02855000 T  0236
          TO BE DONE, THEN THE LOUP BETWEEN COMPOST AND                         02859000 T  0236
          ROSE IS ENTERED.  THE LAST THING DONE FOR ANY                         02860000 T  0236
          IDENTIFIER WHICH IS FOUND IS TO STUFF THE LOCATION                    02861000 T  0236
          OF THE ELBATWORD IN INFO INTO THE LINK FIELD.  THIS                   02862000 T  0236
          ALLOWS REFERENCE BACK TO INFO FOR ADDITIONAL DATA,                    02863000 T  0236
          SHOULD THIS BE REQUIRED.                                 ;            02864000 T  0236
IDENT:    T:=STACKHEAD[SCRAM:=(Q:=ACCUM[1])MOD 125];                            02865000 T  0236
ROSE:     GT1:=T.LINKR;                                                         02875000 T  0239
          IF(GT2:=T.LINKC)+GT1= 0 THEN                                          02876000 T  0241
               BEGIN T:=0; GO COMPLETE END;                                     02877000 T  0243
        IF T = INFO[GT1, GT2] THEN BEGIN                                        02877010 C  0245
           T:= 0; GO TO COMPLETE END;                                           02877020 C  0247
          T:=INFO[GT1,GT2];                                                     02878000 T  0248
          IF INFO[GT1,GT2+1]&O[1:1:11] ≠ Q THEN GO ROSE;                        02879000 T  0250
          IF COUNT ≤ 5 THEN GO COMPOST ;                                        02880000 T  0254
          IF NOT EQUAL(COUNT-5,ACCUM[2],INFO[GT1,GT2+2])THEN GO ROSE;           02881000 T  0255
COMPOST:  T:=T&GT1[35:43:5]&GT2[40:40:8];                                       02882000 T  0260
COMMENT   CHECK HERE FOR COMMENTS AND DEFINED IDS;                              02883000 T  0262
          IF NOT ENDTOG THEN                                                    02884000 T  0262
               BEGIN                                                            02885000 T  0263
               IF GT1:=T.CLASS = COMMENTV THEN                                  02886000 T  0263
                    BEGIN                                                       02887000 T  0265
                    WHILE EXAMIN(NCR) ≠ "}" DO                                  02888000 T  0266
                         BEGIN RESULT:=6; COUNT:=0; SCANNER END;                02889000 T  0268
                    RESULT:=0;SCANNER;GO SCANAGAIN                              02890000 T  0270
                    END                                                         02891000 T  0272
```

```
                END;                                                                    02892000 T  0272
           IF STOPDEFINE THEN GO COMPLETE;                                              02893000 T  0272
           IF GT1 ≠ DEFINEDID THEN GO COMPLETE;                                         02894000 T  0273
COMMENT    SETUP FOR DEFINED IDS - SEE DEFINEGEN FOR MORE DETAILS;                      02895000 T  0274
           IF T.ADDRESS≠0 THEN T:=FIXDEFINEINFO(T);                                     02896000 T  0274
           IF DEFINEINDEX = 24 THEN                                                     02898000 T  0277
                BEGIN FLAG(139);GO ARGH END;                                            02899000 T  0278
           DEFINEARRAY[DEFINEINDEX]:=LASTUSED&T.ADDRESS [18:33:15];                     02900000 T  0280
           LASTUSED:=GIT(T);                                                            02901000 T  0282
           DEFINEARRAY[DEFINEINDEX+2]:=262144×LCR+NCR;                                  02902000 T  0283
           LCR:=(NCR:=MKABS(DEFINEARRAY[DEFINEINDEX+1]))+1;                             02903000 T  0286
           PUTSEQNO(GT4,LCR);                                                           02904000 T  0290
           TURNONSTOPLIGHT("%",LCR); DEFINEINDEX:=DEFINEINDEX+3;                        02905000 T  0291
           GO PERCENT;                                                                  02906000 T  0293
COMPLETE:                                                                               02909000 T  0295
           ELBAT[NXTELBT]:=T;                                                           02910000 T  0295
           STOPDEFINE:=FALSE; COMMENT ALLOW DEFINES AGAIN;                              02911000 T  0296
           IF NXTELBT:=NXTELBT+1 > 74 THEN                                              02912000 T  0297
             IF NOT MACROID THEN                                                        02913000 T  0298
                BEGIN                                                                   02914000 T  0299
COMMENT    ELBAT IS FULL:  ADJUST IT;                                                   02915000 T  0300
                MOVE(10,ELBAT[65],ELBAT);                                               02916000 T  0300
                I:=I-65; P:=P-65; NXTELBT:=10;                                          02917000 T  0302
                END                                                                     02918000 T  0305
           END;    % OF DO WHILE  P≥ NXTELBT                                            02919000 T  0305
      IF TABLE:=ELBAT[P].CLASS = COMMENTV THEN                                          02920000 T  0305
           BEGIN                                                                        02921000 T  0309
COMMENT    SPECIAL HANDLING OF CONSTANTS FOR SAKE OF FOR STATEMENTS;                    02922000 T  0309
           C:=INFO[0,ELBAT[P].ADDRESS];                                                 02923000 T  0309
           ELBAT[P].CLASS:=TABLE:=NONLITNO                                              02924000 T  0312
           END;                                                                         02925000 T  0313
      STOPDEFINE:=FALSE;  COMMENT ALLOW DEFINE;                                         02926000 T  0315
      END TABLE ;                                                                       02927000 T  0315
                                                                                  21 IS  321 LONG, NEXT SEG     3
```

apparently redundant code.

```
                BOOLEAN PROCEDURE BOOLPRIM;        FORWARD;                             02955000 T  0205
PRT(417) = BOOLPRIM
                PROCEDURE BOOLCOMP(B); BOOLEAN B; FORWARD;                          02955500 T  0205
PRT(420) = BOOLCOMP
                INTEGER PROCEDURE NEXT;                                             02956000 T  0205
PRT(421) = NEXT
                     BEGIN                                                          02956500 T  0205
                     LABEL EXIT;                                                    02957000 T  0205
                                                                          START OF SEGMENT **********    22
                     INTEGER T;                                                     02957500 T  0000
STACK(F+3) = T
                     DEFINE ERROR = BEGIN  FLAG(603); GO EXIT  END#;                02958000 T  0000
                     SKAN;                                                          02958500 T  0000
                     IF RESULT=3 THEN ERROR; % NUMBERS NOT ALLOWED.                 02959000 T  0003
                     IF RESULT=2 THEN % SPECIAL CHARACTER.                          02959500 T  0006
                          BEGIN                                                     02960000 T  0007
                          T:=IF Q="1,0000" OR Q="1%0000" THEN 20 % FAKE OUT BOOLEXP. 02960500 T  0007
                             ELSE ((T:=Q.[18:6]-2) & T[42:41:3]);                   02961000 T  0010
                          IF T=11 OR T=19 OR T=20 THEN BATMAN:=SPECIAL[T] % (,),OR ; 02961500 T  0013
```

```
                    ELSE FLAG(603);                                                                  02962000 T  0017
                         GO EXIT                                                                     02962500 T  0021
                         END SPECIAL CHARACTERS;                                                     02963000 T  0022
               COMMENT  LOOK FOR BOOLEAN OPERATORS, THEN OPTIONS;                                    02963500 T  0022
                    T:=      IF Q="3NOT00" THEN NOTOP                                              02964000 T  0022
                        ELSE IF Q="3AND00" THEN ANDOP                                              02964500 T  0023
                        ELSE IF Q="2OR000" THEN OROP                                               02965000 T  0025
                        ELSE IF Q="3EQV00" THEN EQVOP                                              02965500 T  0027
                        ELSE 0;                                                                    02966000 T  0029
                    IF T#0 THEN BATMAN.CLASS:=T                                                    02966500 T  0031
                ELSE BATMAN:=1 & BOOID[2:7] & REAL(FINDOPTION(1))[1:1]; % OPTION.                  02967000 T  0032
               EXIT:                                                                               02967500 T  0042
                    NEXT:=MYCLASS:=BATMAN.CLASS;                                                   02968000 T  0043
                END NEXT;                                                                          02968500 T  0044
                                                                                                 22 IS   48 LONG, NEXT SEG    3


                BOOLEAN PROCEDURE BOOLEXP;                                                         02969000 T  0205
                     BEGIN                                                                         02969500 T  0205
                     BOOLEAN B;                                                                    02970000 T  0205
                                                                                                 START OF SEGMENT **********   23
STACK(F+3) = B
                     B:=BOOLPRIM;                                                                  02970500 T  0000
                     WHILE MYCLASS≥EQVOP AND MYCLASS≤ANDOP DO BOOLCOMP(B);                         02971000 T  0001
                     BOOLEXP:=B                                                                    02971500 T  0004
                     END BOOLEXP;                                                                  02972000 T  0004
                                                                                                  23 IS    8 LONG, NEXT SEG    3


               BOOLEAN PROCEDURE BOOLPRIM;                                                         02972500 T  0205
                    BEGIN                                                                          02973000 T  0205
                    BOOLEAN B,KNOT;                                                                02973500 T  0205
                                                                                                 START OF SEGMENT **********   24
STACK(F+3) = B
STACK(F+4) = KNOT
                    DEFINE SKIPIT = MYCLASS:=NEXT #;                                               02974000 T  0000
                    IF KNOT:=(NEXT=NOTOP) THEN SKIPIT;                                             02974500 T  0000
                    IF MYCLASS=LEFTPAREN THEN                                                      02975000 T  0003
                         BEGIN                                                                     02975500 T  0003
                         B:=BOOLEXP;                                                               02976000 T  0004
                         IF MYCLASS≠RTPAREN THEN FLAG(604);                                        02976500 T  0005
                         END                                                                       02977000 T  0007
               ELSE IF MYCLASS≠BOOID THEN FLAG(601)                                                02977500 T  0007
               ELSE B:=BATMAN<0;                                                                   02978000 T  0009
                    IF KNOT THEN B:=NOT B; SKIPIT;                                                 02978500 T  0011
                    BOOLPRIM:=B                                                                    02979000 T  0014
                    END BOOLPRIM;                                                                  02979500 T  0014
                                                                                                  24 IS   18 LONG, NEXT SEG    3


               PROCEDURE BOOLCOMP(B); BOOLEAN B;                                                   02980000 T  0205
```

```
                        BEGIN                                                                  02980500 T  0205
                        REAL OPCLASS;                                                          02981000 T  0205
                                                                                    START OF SEGMENT **********    25
STACK(F+2) = OPCLASS
                        BOOLEAN T;                                                             02981500 T  0000
STACK(F+3) = T
                        OPCLASS:=MYCLASS;                                                      02982000 T  0000
                        T:=BOOLPRIM;                                                           02982500 T  0000
                        WHILE OPCLASS<MYCLASS DO BOOLCOMP(T);                                  02983000 T  0001
                        B:=     IF OPCLASS=ANDOP THEN (B AND T)                                02983500 T  0004
                           ELSE IF OPCLASS=OROP  THEN (B OR T)                                 02984000 T  0006
                           ELSE                       (B EQV T);                               02984500 T  0008
                        END BOOLCOMP;                                                          02985000 T  0011
                                                                                      25 IS   14 LONG, NEXT SEG    3


                    %                                                                          02985500 T  0205
                    COMMENT##################################################################  02986000 T  0205
                                  FORWARD DECLARATIONS                                         02986500 T  0205
                    ##########################################################################;  02987000 T  0205
                    %                                                                          02987500 T  0205
                     PROCEDURE AEXP; FORWARD;                                                  03001000 T  0205
PRT(422) = AEXP
                     PROCEDURE ARITHSEC; FORWARD;                                              03002000 T  0205
PRT(423) = ARITHSEC
                     PROCEDURE SIMPARITH; FORWARD;                                             03003000 T  0205
PRT(424) = SIMPARITH
                     PROCEDURE ARITHCOMP; FORWARD;                                             03004000 T  0205
PRT(425) = ARITHCOMP
                     PROCEDURE PRIMARY; FORWARD;                                               03005000 T  0205
PRT(426) = PRIMARY
                     DEFINE BEXP = AEXP#;                                                      03006000 T  0205
                     INTEGER PROCEDURE EXPRSS; FORWARD;                                        03007000 T  0205
PRT(427) = EXPRSS
                     PROCEDURE POLISHER(EXPECT); VALUE EXPECT; REAL EXPECT; FORWARD;           03009000 T  0205
PRT(430) = POLISHER
                     PROCEDURE INLINE; FORWARD;                                                03010000 T  0205
PRT(431) = INLINE
                     PROCEDURE SUBHAND(FROM); VALUE FROM; BOOLEAN FROM; FORWARD;               03011000 T  0205
PRT(432) = SUBHAND
                     PROCEDURE IOSTMT; FORWARD;                                                03012000 T  0205
PRT(433) = IOSTMT
                     INTEGER PROCEDURE IFEXP; FORWARD;                                         03013000 T  0205
PRT(434) = IFEXP
                     PROCEDURE PARSE; FORWARD;                                                 03014000 T  0205
PRT(435) = PARSE
                     PROCEDURE DOT; FORWARD;                                                   03015000 T  0205
PRT(436) = DOT
                     PROCEDURE  IFCLAUSE; FORWARD;                                             03018000 T  0205
PRT(437) = IFCLAUSE
                    INTEGER PROCEDURE GET(SYLLABLE);VALUE SYLLABLE; REAL SYLLABLE; FORWARD;    03019000 T  0205
PRT(440) = GET
                    INTEGER PROCEDURE GNAT(L); VALUE L; REAL L; FORWARD;                       03020000 T  0205
PRT(441) = GNAT
                     PROCEDURE PANA; FORWARD;                                                  03021000 T  0205
```

```
PRT(442) = PANA
                PROCEDURE IFSTMT; FORWARD;                                                                                  03022000 T  0205
PRT(443) = IFSTMT
                PROCEDURE GOGEN(LABELBAT,BRANCHTYPE);                                                                       03023000 T  0205
PRT(444) = GOGEN
                           VALUE LABELBAT,BRANCHTYPE;                                                                       03024000 T  0205
                           REAL LABELBAT,BRANCHTYPE; FORWARD;                                                               03025000 T  0205
                BOOLEAN PROCEDURE SIMPGO; FORWARD;                                                                          03026000 T  0205
PRT(445) = SIMPGO
                PROCEDURE STMT; FORWARD;                                                                                    03027000 T  0205
PRT(446) = STMT
                PROCEDURE EMIT(SYLLABLE); VALUE SYLLABLE; REAL SYLLABLE; FORWARD;                                           03028000 T  0205
PRT(447) = EMIT
                PROCEDURE PROCSTMT(FROM); VALUE FROM; BOOLEAN FROM; FORWARD;                                                03029000 T  0205
PRT(450) = PROCSTMT
                PROCEDURE STRMPROCSTMT; FORWARD;                                                                            03030000 T  0205
PRT(451) = STRMPROCSTMT
                PROCEDURE CONSTANTCLEAN; FORWARD;                                                                           03034000 T  0205
PRT(452) = CONSTANTCLEAN
                PROCEDURE SCATTERELBAT; FORWARD;                                                                            03035000 T  0205
PRT(453) = SCATTERELBAT
                PROCEDURE EMITB(BRANCH,FROM,TOWARDS); VALUE BRANCH,FROM,TOWARDS;                                            03036000 T  0205
PRT(454) = EMITB
                           INTEGER BRANCH,FROM,TOWARDS; FORWARD;                                                            03037000 T  0205
                PROCEDURE VARIABLE(FROM); INTEGER FROM; FORWARD;                                                            03038000 T  0205
PRT(455) = VARIABLE
                PROCEDURE IMPFUN; FORWARD;                                                                                  03039000 T  0205
PRT(456) = IMPFUN
                PROCEDURE RIGHT(L); VALUE L; INTEGER L; FORWARD;                                                            03039500 T  0205
PRT(457) = RIGHT
                PROCEDURE STREAMSTMT; FORWARD;                                                                              03040000 T  0205
PRT(460) = STREAMSTMT
                PROCEDURE SEGMENTSTART(B);VALUE B;BOOLEAN B;FORWARD;                                                        03041000 T  0205
PRT(461) = SEGMENTSTART
                PROCEDURE SEGMENT(SIZE,FR); VALUE SIZE,FR; INTEGER SIZE,FR; FORWARD;                                        03042000 T  0205
PRT(462) = SEGMENT
                                                                                                                            03043000 T  0205
                                                                                                                            03044000 T  0205
                                                                                                                            03045000 T  0205
                                                                                                                            03046000 T  0205
                INTEGER PROCEDURE BAE; FORWARD;
PRT(463) = BAE
                     PROCEDURE PROGDESCBLDR(A,B,C,D);VALUE A,B,C,D;                                                         03047000 T  0205
PRT(464) = PROGDESCBLDR
                               INTEGER A,C,D; BOOLEAN B; FORWARD;                                                           03047100 T  0205
                PROCEDURE BANA; FORWARD;                                                                                    03048000 T  0205
PRT(465) = BANA
                PROCEDURE EMITNUM(A); VALUE A; REAL A; FORWARD;                                                             03049000 T  0205
PRT(466) = EMITNUM
                PROCEDURE EMITD(A,B,T); VALUE A,B,T; INTEGER A,B,T; FORWARD;                                                03050000 T  0205
PRT(467) = EMITD
                INTEGER PROCEDURE GETSPACE(S,L); VALUE S,L;                                                                 03051000 T  0205
PRT(470) = GETSPACE
                                                    INTEGER L; BOOLEAN S; FORWARD;                                          03051001 T  0205
                PROCEDURE FORSTMT; FORWARD;                                                                                 03052000 T  0205
PRT(471) = FORSTMT
                REAL PROCEDURE TAKE(INDEX); VALUE INDEX; INTEGER INDEX; FORWARD;                                            03053000 T  0205
                PROCEDURE E;  FORWARD;                                                                                      03054000 T  0205
```

```
PRT(472) = E
          PROCEDURE ENTRY(TYPE); VALUE TYPE;REAL TYPE; FORWARD;                                  03055000 T  0205
PRT(473) = ENTRY
          PROCEDURE PUTNBUMP(P1); VALUE P1;  REAL P1;  FORWARD;                                  03057000 T  0205
PRT(474) = PUTNBUMP
          PROCEDURE JUMPCHKNX;  FORWARD;                                                         03058000 T  0205
PRT(475) = JUMPCHKNX
          PROCEDURE JUMPCHKX;  FORWARD;                                                          03059000 T  0205
PRT(476) = JUMPCHKX
           PROCEDURE DBLSTMT; FORWARD;                                                           03060000 T  0205
PRT(477) = DBLSTMT
           PROCEDURE BLOCK(S); VALUE S; BOOLEAN S; FORWARD;                                      03067000 T  0205
PRT(500) = BLOCK
          PROCEDURE PURGE(STOPPER);VALUE STOPPER;REAL STOPPER;FORWARD;                           03068000 T  0205
PRT(501) = PURGE
          PROCEDURE ENTER(TYPEV);                                                                03069000 T  0205
PRT(502) = ENTER
                    VALUE TYPEV;                                                                 03070000 T  0205
          REAL TYPEV;FORWARD;                                                                    03071000 T  0205
                                                                                                 03072000 T  0205
                                                                                                 03073000 T  0205
                                                                                                 03074000 T  0205
                                                                                                 03075000 T  0205
           COMMENT THIS SECTION CONTAINS THE EMITTERS.  THEY ARE THE AGENTS WHICH                 04000000 T  0205
                    ACTUALLY PRODUCE CODE AND DEBUGING OUTPUT;                                   04001000 T  0205
           COMMENT EMITL EMITS A LIT CALL;                                                       04002000 T  0205
           PROCEDURE EMITL(LITERAL); VALUE LITERAL; INTEGER LITERAL;                             04003000 T  0205
PRT(503) = EMITL
                    EMIT(0&LITERAL[36:38:10]);                                                   04004000 T  0205

           COMMENT EMITO EMIT AN OPERATOR;                                                       04005000 T  0207
           PROCEDURE EMITO(OPERATOR); VALUE OPERATOR; INTEGER OPERATOR;                          04006000 T  0207
PRT(504) = EMITO
                    EMIT(1&OPERATOR[36:38:10]);                                                  04007000 T  0207

           COMMENT EMITC IS PRIMARILY FOR USE BY STRMSTMT TO EMIT CHARACTOR MODE                 04008000 T  0209
                    OPERATORS.  HOWEVER IT ALSO HANDLES DIA, DIB, AND TRB;                       04009000 T  0209
          PROCEDURE EMITC(REPEAT,OPERATOR); VALUE REPEAT,OPERATOR;                               04010000 T  0209
PRT(505) = EMITC
                    INTEGER REPEAT,OPERATOR;                                                     04011000 T  0209
                    BEGIN                                                                        04012000 T  0209
                      IF REPEAT≥64 THEN FLAG(268);                                               04013000 T  0209
                      EMIT(OPERATOR&REPEAT[36:42:6]) END EMITC;                                  04014000 T  0211

           COMMENT EMITV EMITS AN OPERAND CALL.  IF THE ADDRESS IS FOR THE SECOND                04015000 T  0213
                    HALF OF THE PRT, THEN IT ALSO EMITS A PRTE;                                  04016000 T  0213
           PROCEDURE EMITV(ADDRESS); VALUE ADDRESS; INTEGER ADDRESS;                             04017000 T  0213
PRT(506) = EMITV
```

```
                    BEGIN IF ADDRESS > 1023 THEN EMITO(PRTE);                    04018000 T  0213
                         EMIT(2 & ADDRESS [36:38:10]) END EMITV;                 04019000 T  0215



               COMMENT EMITN EMITS A DESCRIPTOR CALL.  IF THE ADDRESS IS FOR THE 04020000 T  0217
                         SECOND HALF OF THE PRT, THEN IT ALSO EMITS A PRTE;      04021000 T  0217
               PROCEDURE EMITN(ADDRESS); VALUE ADDRESS; INTEGER ADDRESS;         04022000 T  0217
PRT(507) = EMITN
                    BEGIN IF ADDRESS > 1023 THEN EMITO(PRTE);                    04023000 T  0217
                         EMIT(3 & ADDRESS [36:38:10]) END EMITN;                 04024000 T  0219



               COMMENT EMITPAIR EMITS A LITC ADDRESS FOLLOWED BY OPERATOR.  IF THE 04025000 T  0221
                         ADDRESS IS FOR THE SECOND HALF OF THE PRT, THEN IT ALSO 04026000 T  0221
                         EMITS PRTE;                                             04027000 T  0221
               PROCEDURE EMITPAIR(ADDRESS,OPERATOR);                             04028000 T  0221
PRT(510) = EMITPAIR
                    VALUE ADDRESS,OPERATOR;                                      04029000 T  0221
                    INTEGER ADDRESS,OPERATOR;                                    04030000 T  0221
                    BEGIN                                                        04031000 T  0221
                         EMITL(ADDRESS);                                         04032000 T  0221
                         IF ADDRESS > 1023 THEN EMITO(PRTE);                     04033000 T  0221
                         EMITO(OPERATOR) END EMITPAIR;                           04034000 T  0223
```

```
               COMMENT ADJUST ADJUST L TO THE BEGINING OF A WORD AND FILLS IN THE 04080000 T  0224
                         INERVENING SPACE WITH NOPS. IT CHECKS STREAMTOG TO DECIDE 04081000 T  0224
                         WHICH SORT OF NOP TO USE;                               04082000 T  0224
               PROCEDURE ADJUST;                                                 04083000 T  0224
PRT(511) = ADJUST
                    BEGIN                                                        04084000 T  0224
                                                                                 04085000 T  0224
              WHILE L.[46:2]≠0 DO EMIT(45);                                      04086000 T  0224
                    END ADJUST;                                                  04087000 T  0228



               PROCEDURE EMITLNG;                                                04098000 T  0228
PRT(512) = EMITLNG
                    BEGIN LABEL E;                                               04099000 T  0228
                                                                    START OF SEGMENT **********   26
                      IF NOT LINKTOG THEN GO TO E;                               04100000 T  0000
          COMMENT  GO TO E IF LAST THING IS A LINK;                              04101000 T  0000
                      IF GET(L) ≠ 0 THEN GO TO E;                                04102000 T  0000
          COMMENT  EITHER LAST EXPRESSION WAS CONDITIONAL OR THERE IS NO         04103000 T  0002
                    LNG OR RELATIONAL OPERATOR;                                  04104000 T  0002
                      IF GT1 ← GET(L-1) = 77 THEN L ← L-1                        04105000 T  0002
          COMMENT LAST THING WAS AN LNG - SO CANCEL IT;                          04106000 T  0005
               ELSE IF GT1.[42:6]=21 AND GT1.[37:2]=0 THEN % AHA                 04107000 T  0005
          COMMENT LAST THING WAS A RELATIONAL;                                   04108000 T  0009
```

```
                    BEGIN L←L-1; EMITO(REAL(BOOLEAN(GT1,[36:10]) EQV                  04109000 T  0009
                       BOOLEAN(IF GT1,[40:2] = 0 THEN 511 ELSE 463)))                 04110000 T  0012
      COMMENT NEGATE THE RELATIONAL; END ELSE                                         04111000 T  0015
E:          EMITO(LNG) END EMITLNG;                                                   04112000 T  0015
                                                                                  26 IS   18 LONG, NEXT SEG    3



      COMMENT   EMITB EMITS A BRANCH OPERATOR AND ITS ASSOCIATED NUMBER;              04113000 T  0228
    PROCEDURE EMITB(BRANCH,FROM,TOWARDS);                                             04114000 T  0228
              VALUE BRANCH,FROM,TOWARDS;                                              04115000 T  0228
              INTEGER BRANCH,FROM,TOWARDS;                                            04116000 T  0228
              BEGIN                                                                   04117000 T  0228
                INTEGER TL;                                                           04118000 T  0228
                                                                            START OF SEGMENT **********   27
STACK(F+2) = TL
                TL ← L;                                                               04119000 T  0000
              IF TOWARDS > FOULED THEN FOULED ← TOWARDS;                              04119500 T  0000
                L ← FROM-2;                                                           04120000 T  0002
                GT1 ← TOWARDS-FROM;                                                   04120100 T  0004
                IF TOWARDS,[46:2] = 0                                                 04120200 T  0005
                  THEN BEGIN                                                          04120300 T  0006
                   BRANCH ← BRANCH&1[39:47:1];                                        04120400 T  0007
                   GT1 ← TOWARDS DIV 4 - (FROM-1) DIV 4 END;                          04120500 T  0008
                EMITNUM(ABS(GT1));                                                    04121000 T  0011
                EMITO(BRANCH&(REAL(GT1≥ 0)+1)[42:46:2]);                              04122000 T  0012
                                                                                      04123000 T  0015
                L ← TL                                                                04124000 T  0015
              END EMITB;                                                              04125000 T  0015
                                                                                  27 IS   19 LONG, NEXT SEG    3



          COMMENT   DEBUGWORD FORMATS TWO FIELDS FOR DEBUGGING OUTPUT IN              04126000 T  0228
                    OCTAL, NAMELY :                                                   04127000 T  0228
                         1, 4 CHARACTERS FOR THE L REGISTER,                          04128000 T  0228
                         2,16 CHARACTERS FOR THE WORD BEING EMITTED.   ;              04129000 T  0228
                STREAM  PROCEDURE DEBUGWORD( SEQ,CODE,FEIL); VALUE SEQ,CODE ;         04130000 T  0228
PRT(513) = DEBUGWORD
                                                                                      04131000 T  0228
                    BEGIN                                                             04132000 T  0229
                      DI←FEIL; SI← LOC SEQ; SI← SI+4; DS ← 4 CHR;                     04133000 T  0230
                      DS ← 2 LIT"  ";                                                 04134000 T  0230
                      SI ← LOC CODE ;                                                 04135000 T  0230
                      16(  DS ← 3 RESET; 3( IF SB THEN DS←SET ELSE                    04136000 T  0232
                           DS ← RESET ; SKIP 1 SB));                                  04137000 T  0233
                    29(DS ← 2 LIT"  " );                                              04138000 T  0234
                     END ;



          COMMENT   EMITWORD PLACES THE PARAMETER,"WORD",INTO EDOC, IF                04139000 T  0234
                    DEBUGGING IS REQUIRED, "L" AND "WORD" ARE OUTPUT ON               04140000 T  0234
                    THE  PRINTER FILE IN OCTAL FORMAT,     ;                          04141000 T  0234
              PROCEDURE  EMITWORD (WORD); VALUE WORD; REAL WORD;                      04142000 T  0234
```

```
PRT(514) = EMITWORD
                        BEGIN                                                    04143000 T  0234
                        ADJUST;                                                  04144000 T  0234
                        IF L≥ 4088 THEN BEGIN ERR(200); L←0; END                 04145000 T  0235
                            ELSE BEGIN                                           04146000 T  0238
                    MOVE(1,WORD, CODE(L DIV 4+1));                               04147000 T  0240
                            IF DEBUGTOG THEN                                     04148000 T  0245
                            BEGIN DEBUGWORD(B2D(L),WORD,LIN);                    04149000 T  0245
                                 WRITELINE  END;                                 04150000 T  0248
                            FOULED ← L ← L+4; END                                04151000 T  0258
                        END  EMITWORD;                                           04152000 T  0260


                   COMMENT CONSTANTCLEAN IS CALLED AFTER AN UNCONDITIONAL BRANCH HAS  04153000 T  0260
                            BEEN EMITTED. IF ANY CONSTANTS HAVE BEEN ACCUMULATED BY   04154000 T  0260
                            EMITNUM IN INFO[0,*], CONSTANTCLEAN WILL FIX  THE CHAIN   04155000 T  0260
                            OF  C-RELATIVE  OPDC S LEFT BY EMITNUM. IF  C-RELATIVE    04156000 T  0260
                            ADDRESSING IS IMPOSSIBLE (I.E. THE ADDRESS                04157000 T  0260
                            IF GREATER THAN 127 WORDS) THEN THE CONSTANT ALONG WITH   04158000 T  0260
                            THE  1ST  LINK OF THE OPDC CHAIN IS ENTERED IN INFO.      04159000 T  0260
                            AT  PURGE  TIME THE REMAINING OPDC S ARE EMITTED WITH     04160000 T  0260
                            F -RELATIVE ADDRESSING AND CODE EMITTED TO STORE THE      04161000 T  0260
                            CONSTANTS INTO THE PROPER F-RELATIVE CELLS.  ;            04162000 T  0260
                    PROCEDURE CONSTANTCLEAN ;                                        04163000 T  0260
                            IF MRCLEAN THEN                                          04164000 T  0260
                              BEGIN                                                  04165000 T  0261
                                 INTEGER J,TEMPL,D,LINK;                             04166000 T  0261
PRT(515) = *SEGMENT DESCRIPTOR*
                                                                  START OF SEGMENT **********   28
STACK(F+2) = J
STACK(F+3) = TEMPL
STACK(F+4) = D
STACK(F+5) = LINK
                                 BOOLEAN CREL;                                   04167000 T  0000
STACK(F+6) = CREL
                                 LABEL  ALLTHU ;                                 04168000 T  0000
                                                                                 04169000 T  0000
                                 FOR J ← 1 STEP 2 UNTIL LASTENTRY DO             04170000 T  0000
                                   BEGIN                                         04171000 T  0001
                                  ADJUST; TEMPL←L; L←INFO[0,255-J+1];            04172000 T  0001
                                     CREL ← FALSE;                               04173000 T  0005
                                     DO BEGIN                                    04174000 T  0005
                                         IF D←(TEMPL-L+3)DIV 42128 THEN          04175000 T  0006
                                  IF MODE ≠ 0 THEN                               04175500 T  0008
                            BEGIN  FLAG(50);       GO TO ALLTHU END;            04176000 T  0010
                                                                                 04177000 T  0011
                                                                                 04178000 T  0011
                                                                                 04179000 T  0011
                                                                                 04180000 T  0011
                                                                                 04181000 T  0011
                                       LINK←GET(L);                              04182000 T  0011
                                       CREL ← TRUE;                              04183000 T  0013
                                         IF MODE ≠ 0 THEN EMITV(D+768) ELSE      04184000 T  0013
                                         EMITV(REAL(TEMPL≥2048)×1024+TEMPL DIV 4);  04184500 T  0016
                                    END  UNTIL L←   LINK    = 4095  ;            04185000 T  0019
```

```
                    ALLTHU:       L ← TEMPL;                                                    04186000 T  0021
                                  IF  CREL  THEN  EMITWORD( INFO[0,255-J ]);                    04187000 T  0022
                                 END;                                                           04188000 T  0025
                                LASTENTRY ← 0;                                                  04189000 T  0028
                               END ;                                                            04190000 T  0028
PRT(516) = *SEGMENT DESCRIPTOR*
                                                                                  28 IS   33 LONG, NEXT SEG    3



                         COMMENT   EMITNUM HANDLES THE EMISSION OF CODE FOR CONSTANTS,BOTH      04191000 T  0266
                                   EXPLICIT AND IMPLICIT. IN EVERY CASE,EMITNUM WILL            04192000 T  0266
                                   PRODUCE CODE TO GET THE DESIRED CONSTANT ON TOP OF           04193000 T  0266
                                   THE STACK. IF THE NUMBER IS A LITERAL A SIMPLE LITC          04194000 T  0266
                                   SYLLABLE IS PRODUCED, HOWEVER,NON-LITERALS ARE KEPT          04195000 T  0266
                                   IN THE ZERO-TH ROW OF INFO WITH THE SYLLABLE                 04196000 T  0266
                                   POSITION,L. THE FIRST EMITNUM ON A PARTICULAR                04197000 T  0266
                                   CONSTANT CAUSES THE VALUES OF L AND THE CONSTANT             04198000 T  0266
                                   TO BE STORED IN INFO[0,*] (NOTE:ITEMS ARE STORED             04199000 T  0266
                                   IN REVERSE STARTING WITH INFO[0,255],ETC.). THEN             04200000 T  0266
                                   ITS THE JOB OF CONSTANTCLEAN TO EMIT THE ACTUAL              04201000 T  0266
                                   OPDC (SEE CONSTANTCLEAN PROCEDURE FOR DETAILS) ;             04202000 T  0266
                         PROCEDURE EMITNUM( C ); VALUE C; REAL C;                               04203000 T  0266
                                   BEGIN LABEL FINISHED,FOUND ; REAL N;                         04204000 T  0266
                                                                                  START OF SEGMENT **********   29
STACK(F+2) = N
                                   IF C,[1:37]=0 THEN EMITL(C)                                  04205000 T  0000
                                   ELSE                                                         04206000 T  0002
                                        BEGIN                                                   04207000 T  0003
                                        FOULED ← L;                                             04207500 T  0003
                                        FOR  N ← 1 STEP 2 UNTIL LASTENTRY DO                    04208000 T  0004
                                     IF INFO[0,255-N] = C THEN  GO TO FOUND ;                   04209000 T  0005
                                        INFO[0,255 -LASTENTRY] ← L;                             04210000 T  0010
                                        INFO[0,255 -LASTENTRY-1]← C ;                           04211000 T  0012
                                        EMITN(1023);                                            04212000 T  0015
                                        IF MODE=0 THEN EMITO(NOP);                              04212100 T  0016
                                        LINKTOG←FALSE;                                          04213000 T  0018
                                        IF LASTENTRY ← LASTENTRY+2 ≥ 128  THEN                  04214000 T  0019
                                             BEGIN                                              04215000 T  0020
                                             C ← BUMPL;                                         04216000 T  0021
                                             CONSTANTCLEAN;                                     04217000 T  0023
                                             EMITB(BFW,C,L);                                    04218000 T  0023
                                             END;                                               04219000 T  0024
                                        GO TO FINISHED;                                         04220000 T  0024
                          FOUND:        EMIT(INFO[0,255 -N+1]);                                 04221000 T  0025
                                        LINKTOG←FALSE;                                          04222000 T  0028
                                        INFO[0,255-N+1]← L-1;                                   04223000 T  0029
                                        IF MODE=0 THEN EMITO(NOP);                              04223100 T  0033
                                        END;                                                    04224000 T  0035
                          FINISHED:END  EMITNUM  ;                                              04225000 T  0035
                                                                                  29 IS   88 LONG, NEXT SEG    3



                         COMMENT   SEARCH PERFORMS A BINARY SEARCH ON THE COP AND WOP           04226000 T  0266
```

```
                        ARRAYS, GIVEN THE OPERATOR BITS SEARCH  YIELDS THE BCD          04227000 T  0266
                        MNEUMONIC FOR THAT OPERATOR. IF THE OPERATOR CANNOT              04228000 T  0266
                        BE FOUND SEARCH YIELDS BLANKS.                                   04229000 T  0266
                        NOTE: DIA,DIB,TRB ARE RETURNED AS BLANKS.       ;                04230000 T  0266
                ALPHA PROCEDURE   SEARCH (Q,KEY); VALUE KEY; ARRAY Q[0]; REAL KEY ;      04231000 T  0266
PRT(517) = SEARCH
                        BEGIN  LABEL L;                                                  04232000 T  0266
                                                                        START OF SEGMENT **********    30
                    COMMENT   GT1 AND GT2 ARE INITIALIZED ASSUMMING THAT Q IS ORDERED    04233000 T  0000
                              BY PAIRS (ARGUMENT,FUNCTION,ARGUMENT,FUNCTION,ETC.)        04234000 T  0000
                              AND THAT THE FIRST ARGUMENT IS IN Q[4].  FURTHERMORE       04235000 T  0000
                              THE LENGTH OF Q IS 128,                           ;        04236000 T  0000
                              INTEGER N,I ;                                              04237000 T  0000
STACK(F+3) = N
STACK(F+4) = I
                        N ← 64 ;                                                         04238000 T  0000
                        FOR I ← 66 STEP IF Q[I]<KEY THEN N ELSE - N                      04239000 T  0000
                                   WHILE N←N DIV 2 ≥ 1 DO                                04240000 T  0005
                        IF Q[ I ]= KEY THEN GO TO L ;                                    04241000 T  0009
                         I ←0;   COMMENT ARGUMENT NOT FOUND,SEARCH=Q[1] ;                04242000 T  0011
                    L:  SEARCH←Q[  I +1] ;                                               04243000 T  0011
                        END       SEARCH  ;                                              04244000 T  0013
                                                                        30 IS   16 LONG, NEXT SEG     3


                    COMMENT   B2D  CONVERTS THE FOUR LOW ORDER OCTAL DIGITS TO BCD       04245000 T  0266
                              CODE ;                                                     04246000 T  0266
                ALPHA   PROCEDURE   B2D(B); VALUE B; REAL B;                             04247000 T  0266
                              B2D←0&B[45:45:3]&B[39:42:3]&B[33:39:3]&B[27:36:3] ;        04248000 T  0266


                    COMMENT   PACK IS A STREAM PROCEDURE WHICH INSERTS THE SYLLABLE       04265000 T  0274
                              INTO THE EDOC ARRAY.   THE SPECIFIC ELEMENT OF EDOC        04266000 T  0274
                              IS  PRECISILY = EDOC[(L DIV 4) DIV 128,(< DIV 4)MOD 128]   04267000 T  0274
                              SYLLABLE  POSITION=(L MOD 4 ), WHERE L IS THE SYLLABLE     04268000 T  0274
                              NUMBER RELATIVE TO THE BEGINNING OF THE SEGMENT;           04269000 T  0274
                STREAM PROCEDURE PACK(WORD,POSITION,SYLLABLE);                           04270000 T  0274
PRT(520) = PACK
                        VALUE POSITION,SYLLABLE;                                         04271000 T  0274
                        BEGIN                                                            04272000 T  0274
                        DI←WORD ; DI ← DI+POSITION ; DI ← DI+POSITION;                   04273000 T  0275
                        SI←LOC SYLLABLE ; SI←SI+6;                                       04274000 T  0276
                        DS←2 CHR ;                                                       04275000 T  0276
                        END  PACK  ;                                                     04276000 T  0277


                COMMENT DEBUG PRINTS OUT OBJECT CODE IF "DEBUGN" IS SET;                 04277000 T  0277
                PROCEDURE DEBUG(S); VALUE S; REAL S;                                     04277500 T  0277
PRT(521) = DEBUG
                    BEGIN  REAL T1;                                                      04278000 T  0277
                                                                        START OF SEGMENT **********    31
```

```
STACK(F+2) = T1                                                                  04278500 T  0000
                    IF SINGLTOG THEN                                             04279000 T  0000
                      WRITE(LINE,BUG,B2D(L),
PRT(522) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*                                  04279500 T  0008
                         IF STREAMTOG THEN                                       04280000 T  0009
                            SEARCH(COP,S,[42:6])                                 04280500 T  0010
                       ELSE IF T1:=S,[46:2]=1 THEN SEARCH(WOP,S,[36:10])         04281000 T  0015
                       ELSE WOP[T1] , IF STREAMTOG THEN                          04281500 T  0019
                      B2D(S,[36:6]) ELSE IF T1=1 THEN WOP[1]                     04282000 T  0023
                  ELSE B2D(S,[36:10]),B2D(S))                                    04282500 T  0027
                ELSE WRITE(LINE,BUG,B2D(L),
PRT(523) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*                                  04283000 T  0038
                         IF STREAMTOG THEN                                       04283500 T  0039
                            SEARCH(COP,S,[42:6])                                 04284000 T  0040
                       ELSE IF T1:=S,[46:2]=1 THEN SEARCH(WOP,S,[36:10])         04284500 T  0045
                       ELSE WOP[T1] , IF STREAMTOG THEN                          04285000 T  0049
                      B2D(S,[36:6]) ELSE IF T1=1 THEN WOP[1]                     04285500 T  0053
                  ELSE B2D(S,[36:10]),B2D(S));                                   04286000 T  0061
                   END DEBUG;                                                31 IS   64 LONG, NEXT SEG     3



                     COMMENT   EMIT PLACES SYLLABLES INTO EDOC, CALLS DEBUG FOR  04288000 T  0277
                               DEBUGGING OUTPUT ON THE PRINTER, AND CHECKS FOR SEGMENTS 04289000 T  0277
                               GREATER THAN 4093 SYLLABLES.        ;             04290000 T  0277
                PROCEDURE EMIT ( S ) ; VALUE S; REAL   S  ;                      04291000 T  0277
                               BEGIN                                             04292000 T  0277
                               IF L < 4088 THEN                                  04293000 T  0277
                               BEGIN                                             04294000 T  0278
                               LINKTOG ← TRUE;                                   04295000 T  0279
                          PACK(CODE(L DIV 4+1),L,[46:2],S);                      04296000 T  0280
                               IF DEBUGTOG THEN DEBUG(S);                        04297000 T  0285
                               L←L+1;                                            04298000 T  0287
                               END  ELSE                                         04299000 T  0288
                               BEGIN ERR(200); L←1; END;                         04300000 T  0288
                     COMMENT   200 EMIT - SEGMENT GREATER THAN 4093 SYLLABLES        *;  04301000 T  0292
                               END  EMIT ;                                       04302000 T  0292



                     COMMENT   EMITD EMITS THE DIA,DIB,TRB SEQUENCE OF CODE. THE 04305000 T  0292
                               PREVIOUS SETTING OF THE G-H AND K-V REGISTERS IS COMPARED 04306000 T  0292
                               THE CURRENT . IF THE G-H,K-V OR BOTH ARE ALREADY SET THEN 04307000 T  0292
                               THE APPROIATE SYLLABLE(S) ARE OMITTED             04308000 T  0292
                               IF 0 BITS ARE TO BE TRANSFERED THEN NO SYLLABLES ARE 04309000 T  0292
                               EMITTED     ;                                     04310000 T  0292
                PROCEDURE      EMITD(A,B,T); VALUE A,B,T ; INTEGER A,B,T;        04311000 T  0292
                        BEGIN LABEL EXIT,NORMAL;                                 04311010 T  0292
                                                                    START OF SEGMENT ********** 32
                               REAL Q;                                           04311020 T  0000
STACK(F+2) = Q
                               IF T = 15 THEN                                    04311030 T  0000
                               BEGIN                                             04311040 T  0000
                               IF A = 33 THEN Q ← 512                            04311050 T  0001
```

```
                    ELSE IF A ≠ 18 THEN GO TO NORMAL;                                                    04311060 T  0002
                    IF B = 18 THEN Q ← Q+256                                                             04311070 T  0005
                    ELSE IF B ≠ 33 THEN GO TO NORMAL;                                                    04311080 T  0006
                    EMITO(Q+197);  COMMENT -- THIS GETS OUT FIXED FIELD;                                 04311090 T  0009
                    GO TO EXIT;                                                                          04311100 T  0010
                    END;                                                                                 04311110 T  0011
NORMAL;                                                                                                  04311120 T  0011
                    IF  T≠ 0 THEN                                                                        04312000 T  0011
                    BEGIN                                                                                04313000 T  0011
                                                                                                         04314000 T  0012
                     EMIT(((DIALA+A) DIV 6)×512 + ( A← A MOD 6)× 64 + DIA);                              04315000 T  0012
                                                                                                         04316000 T  0017
                     EMIT(((DIALB+B) DIV 6)×512 + ( B← B MOD 6)× 64 + DIB);                              04317000 T  0017
                     EMIT(TRB+64×T);                                                                     04318000 T  0021
                                                                                                         04319000 T  0023
                                                                                                         04320000 T  0023
                                                                                                         04321000 T  0023
                    END       EMITD  ;                                                                   04322000 T  0023
EXIT;               END;                                                                                 04322100 T  0023
                                                                                                  32 IS   27 LONG, NEXT SEG    3


                PROCEDURE EMITI(E,A,B); VALUE E,A,B; REAL E,A,B;                                         04500000 T  0292

PRT(524) = EMITI
                                                                                                         04501000 T  0292
                    BEGIN LABEL EXIT,IS;                                                          START OF SEGMENT **********   33
                    INTEGER S,T1,T2;                                                                     04502000 T  0000

STACK(F+2) = S
STACK(F+3) = T1
STACK(F+4) = T2

                    PROCEDURE EMIT21(E,B); VALUE E,B;                                                    04503000 T  0000

PRT(525) = EMIT21
                         REAL E;                                                                         04504000 T  0000
                         BOOLEAN B;                                                                      04505000 T  0000
                         BEGIN IF E = 0 THEN                                                             04506000 T  0000
                                   BEGIN IF B THEN EMITO(XCH); END                                       04507000 T  0000
                              ELSE BEGIN GT1 ← E.ADDRESS;                                                04508000 T  0002
                                        IF E ← E.CLASS ≤ INTID THEN                                      04509000 T  0004
                                             EMITV(GT1)                                                  04521000 T  0006
                                        ELSE IF E ≤ INTARRAYID THEN                                      04522000 T  0007
                                             EMITPAIR(GT1,LOD) ELSE                                      04523000 T  0008
                                        EMITN(GT1)                                                       04524000 T  0010
                                   END                                                                   04525000 T  0011
                         END;                                                                            04526000 T  0011


                    IF B = 0 THEN                                                                        04526100 T  0011
                         BEGIN EMIT21(E,FALSE); GO TO EXIT END;                                          04526200 T  0012
                    IF STACKCT ≠ 0 THEN GO TO IS;                                                        04527000 T  0014
                    IF B = 15 THEN                                                                       04528000 T  0016
                         BEGIN IF A = 33 THEN                                                            04529000 T  0016
                                   BEGIN EMIT21(E,FALSE);                                                04530000 T  0018
                                        EMIT(0); EMITO(INX);                                             04531000 T  0019
```

```
                                        GO TO EXIT;                                                 04532000 T  0021
                                   END;                                                             04533000 T  0021
                              IF A = 18 THEN                                                        04534000 T  0021
                                   BEGIN EMIT(0);                                                   04535000 T  0022
                                        EMIT21(E,TRUE);                                             04536000 T  0023
                                        EMIT0(197);                                                 04537000 T  0024
                                        GO TO EXIT;                                                 04538000 T  0025
                                   END;                                                             04539000 T  0025
                              GO TO IS;                                                             04540000 T  0025
                         END;                                                                       04541000 T  0026
                    IF B ≤ 10 AND A+B = 48 THEN                                                     04542000 T  0026
                         BEGIN EMIT21(E,FALSE);                                                     04543000 T  0028
                              EMITL(2*B-1);                                                         04544000 T  0030
PRT(526) = LN
PRT(527) = EXP
PRT(530) = X TO THE I
                              EMIT0(LND);                                                           04545000 T  0033
                              GO TO EXIT;                                                           04546000 T  0034
                         END;                                                                       04547000 T  0034
               IS;       IF (S + (48-A-B) MOD 6)+B ≤ 39 THEN                                      04548000 T  0034
                         BEGIN EMIT21(E,FALSE);                                                     04549000 T  0038
                              EMIT(T2+(T1+A DIV 6)×512+(A MOD 6)×64+DIA);                           04550000 T  0039
                              EMIT(((A+B-1) DIV 6 -T1+1)×512+64×S+37);                              04551000 T  0044
                              GO TO EXIT;                                                           04552000 T  0049
                         END;                                                                       04553000 T  0050
                    EMIT(0);                                                                        04554000 T  0050
                    EMIT21(E,TRUE);                                                                 04555000 T  0051
                    EMITD(A,48-B,B);                                                                04556000 T  0052
                                                                                                    04557000 T  0053
                       EXIT: END;                                                                   04558000 T  0053
                                                                                                  33 IS   97 LONG, NEXT SEG    3


                    COMMENT THIS SECTION CONTAINS MISCELLANEOUS SERVICE ROUTINES;                   05000000 T  0292
                    COMMENT STEPI AND STEPIT ARE SHORT CALLS ON TABLE;                              05001000 T  0292
                    PROCEDURE STEPIT; ELCLASS ← TABLE(I+I+1);                                       05002000 T  0292
PRT(531) = STEPIT


                    INTEGER PROCEDURE STEPI; STEPI←ELCLASS←TABLE(I+I+1);                            05003000 T  0295
PRT(532) = STEPI


                    COMMENT TAKE FETCHS A WORD FROM INFO;                                           05004000 T  0301
                    REAL PROCEDURE TAKE(INDEX); VALUE INDEX; INTEGER INDEX;                         05005000 T  0301
                                   TAKE ← INFO[INDEX.LINKR,INDEX.LINKC];                            05006000 T  0301


                    COMMENT PUT PLACES A WORD INTO INFO;                                            05007000 T  0307
```

```
                PROCEDURE PUT(WORD,INDEX); VALUE WORD,INDEX; REAL WORD,INDEX;                    05008000 T  0307
PRT(533) = PUT
                        INFO[INDEX,LINKR,INDEX,LINKC] ← WORD;                                    05009000 T  0307



                COMMENT FLAG FLAGS ERROR MESSAGES, COUNTS THEM AND SUPPRESS FUTURE               05010000 T  0311
                        ERROR MESSAGES UNTIL THE COMPILER THINKS IT HAS RECOVERED;               05011000 T  0311
                PROCEDURE FLAG(ERRNUM); VALUE ERRNUM; INTEGER ERRNUM;                            05012000 T  0311
                        BEGIN                                                                    05013000 T  0311
                   COMMENT WRITERROR IS THE STREAM PROCEDURE WHICH ACTUALLY PRODUCES              05014000 T  0311
                        THE ERROR MESSAGE ON THE PRINTER;                                        05015000 T  0311
                     STREAM PROCEDURE WRITERROR(ERRNUM,ACCUM,LINE,COUNT,LSTSEQ);                 05016000 T  0311
                                                                          START OF SEGMENT ********** 34
PRT(534) = WRITERROR
                        VALUE ERRNUM,COUNT;                                                      05017000 T  0000
                        BEGIN                                                                    05018000 T  0000
                             DI ← LINE; 44(DS←2LIT" "); COMMENT CLEAR BUFFER;                    05019000 T  0000
                             SI ←LSTSEQ; SI ← SI-8; DS ←WDS;                                     05020000 T  0001
                             SI ← LINE; DS ← 2 WDS;                                              05021000 T  0002
                             4(DS ← 2 LIT "XX"); COMMENT  SET RIGHT MARGIN FLAG;                 05022000 T  0002
                             SI ← LSTSEQ; DI ← LSTSEQ; DI ← DI-8; DS ← WDS;                      05023000 T  0003
                             DI ← LINE; DI ← DI+8; COMMENT INDENT MESSAGE;                       05024000 T  0004
                             DS ← 13 LIT "ERROR NUMBER ";                                        05025000 T  0005
                             SI ← LOC ERRNUM; DS ← 3 DEC; COMMENT CONVERT ERRNUM;                05026000 T  0007
                             DS ← 4 LIT " -- ";                                                  05027000 T  0007
                             SI ← ACCUM; SI ← SI+3; DS ← COUNT CHR;                              05028000 T  0008
                                COMMENT PLACE ALPHA IN BUFFER;                                   05029000 T  0009
                             DS ← LIT ","                                                        05030000 T  0009
                        END WRITERROR;                                                           05031000 T  0009

                        IF ERRORTOG THEN % DO NOTHING IF WE SUPPRESS MSSGS.                      05032000 T  0010
                             BEGIN                                                               05033000 T  0010
                             SPECTOG := FALSE;                                                   05034000 T  0010
                             ERRORCOUNT := ERRORCOUNT+1; COMMENT COUNT ERRORS;                   05035000 T  0011
                             IF NOT LISTER THEN                                                  05036000 T  0012
                                  BEGIN                                                          05037000 T  0013
                                  EDITLINE(LIN,FCR,L DIV 4,L.[46:2],MEDIUM,0);                   05038000 T  0013
                                 MOVE(1,INFO[LASTSEQROW,LASTSEQUENCE],LIN[12]);                  05039500 T  0017
                                 IF NOHEADING THEN DATIME; WRITELINE;                            05039600 T  0019
                                 END;                                                            05041000 T  0031
                        COMMENT   PRINT CARDIMAGE IF WE ARE NOT LISTING;                         05042000 T  0031
                             ACCUM[1] ← Q;    COMMENT RESTORE ACCUMULATOR;                       05043000 T  0031
                             WRITERROR(ERRNUM,ACCUM[1],LIN[*],Q,[12:6],                          05044000 T  0032
                                  INFO[LASTSEQROW,LASTSEQUENCE]);                                05045000 T  0034
                        IF NOT NOHEADING THEN WRITELINE ;                                        05046000 T  0036
                             ERRORTOG ← FALSE; COMMENT INHIBIT MESSAGES;                         05047000 T  0047
                   IF PUNCHTOG THEN                                                              05048000 T  0048
                        BEGIN REAL T1,T2,T3,T4; LABEL L,L1,EXIT;                                 05049000 T  0048
PRT(535) = *SEGMENT DESCRIPTOR*
                                                                          START OF SEGMENT ********** 35
STACK(F+2) = T1
STACK(F+3) = T2
```

```
STACK(F+4) = T3
STACK(F+5) = T4
                STREAM PROCEDURE P1(L,P); VALUE P;                                  05050000 T  0000
PRT(536) = P1
                      BEGIN DI ← L; 15(DS←8LIT" ");                                 05051000 T  0000
                SI ← LOC P; DI←L; SI←SI+5; SKIP 3 SB;                               05052000 T  0002
                5                                                                   05053000 T  0003
                (DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB));          05054000 T  0003
                SI←P;                                                               05055000 T  0005
                DI←DI+2; 2(8                                                        05056000 T  0006
                (DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB));          05057000 T  0006
                DS←LIT" ") END;                                                     05058000 T  0009



                STREAM PROCEDURE P2(L,P); VALUE P;                                  05059000 T  0010
PRT(537) = P2
                    BEGIN DI←L; DI ←DI+26;DS←LIT"0";                                05060000 T  0010
                       SI←P; SI←SI-2; SKIP 1SB;                                     05061000 T  0012
                3                                                                   05062000 T  0012
                (DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB));          05063000 T  0012
                END;                                                                05064000 T  0015



                REAL STREAM PROCEDURE ABS(A); VALUE A;                              05065000 T  0015
PRT(540) = ABS
                      BEGIN DI ← LOC ABS; SI ←A; DS← WDS;DI←DI+8; DS←RESET END;      05066000 T  0015



              STREAM PROCEDURE  BITEDUST(X,N,ID,LINE,COUNT); VALUE ID,N,COUNT;      05067000 T  0018
PRT(541) = BITEDUST
               BEGIN  LOCAL  T,F,H;                                                 05068000 T  0018
                       DI←LOC F; SI←LINE; DS←WDS;DI←F;                              05069000 T  0019
                SI←LOC ID;SI←SI+2;DS← 6CHR;  57(DS←2 LIT"  ");                      05070000 T  0020
                           SI ← LOC COUNT ;SI←SI+8 ;                                05071000 T  0021
                           DI ←LOC H ; DS ←WDS ;                                    05071100 T  0022
                           DI ← LOC LINE ; SI ← H;                                  05071200 T  0022
                      SI←LOC X;SKIP 2 SB;                                           05072000 T  0023
                     IF SB THEN                                                     05073000 T  0023
                       BEGIN   SI←X;  T←SI;                                         05074000 T  0024
                           N(                                                       05075000 T  0025
                       DI←LOC F; SI←LINE; DS←WDS;DI←F;                              05076000 T  0026
                           SI← LOC COUNT; SI←SI+6;                                  05077000 T  0027
                4                                                                   05078000 T  0027
                (DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB));          05079000 T  0027
                           DS←2 LIT" "; SI←COUNT; SI←SI+48;COUNT←SI;                05080000 T  0030
                SI←T;                                                               05081000 T  0031
                       6(2(8                                                        05082000 T  0032
                (DS←3 RESET;3(IF SB THEN DS←SET ELSE DS←RESET;SKIP 1 SB));          05083000 T  0032
                DS← LIT" "); DS←LIT" " );                                           05084000 T  0035
                           DI ← LOC LINE ; SI ← H;                                  05085000 T  0036
                SI←T;SI←SI+48;T←SI);END END;                                        05086000 T  0037
```

```
                BITEDUST(ERRORCOUNT, 63 ,"PRT   ",LINE, 21);                                  05087000 T  0038
            FOR  T1← 0 STEP 1 UNTIL 3100 BITEDUST(INFO[ T1,*],43,"INFO  ",LINE,0);            05088000 T  0041
            BITEDUST(      ELBAT[*],13,"ELBAT ",LINE,0);                                      05089000 T  0049
                BITEDUST(STACKHEAD[*],21,"STHEAD",LINE,0);                                    05090000 T  0052
                   T1←MKABS(ERRNUM)=1; T3←T1;                                                 05091000 T  0055
            L:     T2←ABS(T3);                                                                05092000 T  0058
                   IF T2.[33:15]=0 THEN BEGIN T3←T2.[18:15];                                  05093000 T  0060
                      IF T3=0 THEN GO TO EXIT ELSE GO TO L END ;                              05094000 T  0063
              T4 ← IF T2.[33:15] < 512 THEN 0 ELSE T2.[33:15]&T2[30:10:2];                    05095000 T  0065
             L1:P1(LIN[0],T1);IF T1=T3 THEN BEGIN IF T≠0 THEN P2(LIN[0],T4);                  05096000 T  0070
                T1←T1-1;                                                                      05097000 T  0076
                 WRITELINE;                                                                   05097500 T  0077
                                  T3←T2.[18:15];GO TO L  END;                                 05098000 T  0091
                 T1:=T1-1; WRITELINE;                                                         05099000 T  0093
                 GO TO L1;                                                                    05099100 T  0104
            EXIT: END ;                                                                       05100000 T  0105
PRT(542) = *SEGMENT DESCRIPTOR*

                                                                                    35 IS  106 LONG, NEXT SEG   34
                                END END FLAG;                                                 05101000 T  0050
                                                                                    34 IS   93 LONG, NEXT SEG    3


                                LABEL ENDOFITALL;                                             05101100 T  0311
            COMMENT ERR, IS THE SAME AS FLAG EXCEPT THAT IT MAKES AN ATTEMPT TO               05102000 T  0311
                        RECOVER FROM ERROR SITUATIONS BY SEARCHING FOR A                      05103000 T  0311
                        SEMICOLON, END, OR BEGIN;                                             05104000 T  0311
            PROCEDURE ERR(ERRNUM); VALUE ERRNUM; INTEGER ERRNUM;                              05105000 T  0311
                        BEGIN FLAG(ERRNUM);                                                   05106000 T  0311
                             I ← I-1;                                                         05107000 T  0312
                             IF ERRNUM = 200 THEN I:=I/0;% SEGMENT TOO LARGE                  05107100 P  0314
                             IF ERRNUM = 611 THEN I:=I/0;% ERRMAX EXCEEDED                    05107200 P  0316
                             DO IF STEPI = BEGINV THEN STMT UNTIL                             05108000 T  0319
                                ELCLASS = ENDV OR ELCLASS = SEMICOLON END ERR;                05109000 T  0321



            DEFINE ERROR = ERR#;  COMMENT ERROR IS A SYNONM FOR ERR;                          05110000 T  0323
            COMMENT CHECKER IS A SMALL PROCEDURE THAT CHECKS TO SEE THAT THE                  05111000 T  0323
                        UPLEVEL ADDRESSING CONVENTIONS ARE OBEYED;                            05112000 T  0323
            PROCEDURE CHECKER(ELBATWORD); VALUE ELBATWORD; REAL ELBATWORD;                    05113000 T  0323
PRT(543) = CHECKER
                        BEGIN                                                                 05114000 T  0323
                             IF MODE ≥ 2 THEN                                                 05115000 T  0323
                             IF GT1 ← ELBATWORD.LVL ≥ FRSTLEVEL THEN                          05116000 T  0324
                             IF GT1 < SUBLEVEL THEN                                           05117000 T  0327
                        IF ELBATWORD.[9:2] ≠ 1                                                05118000 T  0328
                              THEN BEGIN FLAG(101); ERRORTOG ← TRUE END                       05119000 T  0329
                        END CHECKER;                                                          05120000 T  0332
```

```
                    COMMENT GIT IS USED TO OBTAIN THE INDEX TO ADDITIONAL INFORMATION          05121000 T  0332
                            GIVEN THE LINK TO THE ELBAT WORD;                                  05122000 T  0332
                    INTEGER PROCEDURE GIT(L); VALUE L; REAL L;                                 05123000 T  0332
                            GIT ← TAKE(L).INCR+L.LINK;                                         05124000 T  0332



                    COMMENT GNAT IS USED TO OBTAIN THE PRT ADDRESS OF A GIVEN DESCRIPTOR.      05125000 T  0338
                            IF THE ADDRESS HAS NOT BEEN ASSIGNED, THEN IT USES                 05126000 T  0338
                            GETSPACE TO OBTAIN THE PRT ADDRESS;                                05127000 T  0338
                    INTEGER PROCEDURE GNAT(L); VALUE L; REAL L;                                05128000 T  0338
                            BEGIN                                                              05129000 T  0338
                            REAL A;                                                            05130000 T  0338
                                                                                  START OF SEGMENT **********    36
STACK(F+3) = A
                            IF GNAT ←(A←TAKE(L)).ADDRESS=0                                     05131000 T  0000
                            THEN PUT(A&(GNAT:=GETSPACE(TRUE,L.LINK+1))[16:37:11],L)            05132000 T  0002
                            END GNAT;                                                          05133000 T  0007
                                                                                    36 IS   10 LONG, NEXT SEG     3



                    REAL PROCEDURE TAKEFRST;                                                   05188000 T  0338
PRT(544) = TAKEFRST
                            TAKEFRST ← TAKE(ELBAT[I].LINK+ELBAT[I].INCR);                      05189000 T  0338
```

```
                    COMMENT STUFFF DIALS THE F-REGISTER INTO THE F-REGISTER FIELD OF A         05196000 T  0344
                            DESCRIPTOR.  THE DESCRIPTOR REMAINS ON THE TOP OF THE              05197000 T  0344
                            STACK;                                                             05198000 T  0344
                    PROCEDURE STUFFF(ADDRESS); VALUE ADDRESS; INTEGER ADDRESS;                 05199000 T  0344
PRT(545) = STUFFF
                            BEGIN                                                              05200000 T  0344
                                EMITPAIR(ADDRESS,LOD);                                         05201000 T  0344
                                    EMITN(512);                                            05202000 T  0346
                                    EMITD(33,18,15) END STUFFF;                            05203000 T  0346



                    COMMENT    LOCAL IS USED TO SEE WHETHER OR NOT A LABEL IS LOCAL TO OUR      05204000 T  0348
                               PRESENT CODE;                                                   05205000 T  0348
                    BOOLEAN PROCEDURE    LOCAL(ELBATWORD);                                     05206000 T  0348
PRT(546) = LOCAL
                            VALUE ELBATWORD; REAL ELBATWORD;                                   05207000 T  0348
                            BEGIN IF ELBATWORD.LVL = LEVEL AND                                 05208000 T  0348
                                  NOT BOOLEAN(ELBATWORD,FORMAL) THEN                           05209000 T  0350
                                      LOCAL ← TRUE END    LOCAL;                             05210000 T  0351



                    COMMENT PASSFORMAT COMPILES CODE THAT PASSES A FORMAT.  TWO ITEMS ARE      05211000 T  0355
```

```
                    PASSED = THE ARRAY REFERENCING FORMAT TABLE AND THE                05212000 T  0355
                    STARTING INDEX.  THE ROUTINE HANDLES SUPERFORMATS ALSO;            05213000 T  0355
                                                                                       05214000 T  0355
               PROCEDURE PASSFORMAT;
PRT(547) = PASSFORMAT
                    BEGIN INTEGER ADRES;                                               05215000 T  0355
                                                                          START OF SEGMENT **********    37

STACK(F+2) = ADRES
                        CHECKER(ELBAT[I]);                                             05216000 T  0000
                        ADRES ← ELBAT[I].ADDRESS;                                      05217000 T  0001
                        IF BOOLEAN(ELBAT[I].FORMAL)                                    05218000 T  0002
                           THEN BEGIN EMITV(ADRES); ADRES ← ADRES+1 END                05219000 T  0002
                           ELSE BEGIN                                                  05220000 T  0006
                             IF TABLE(I) = SUPERFRMTID                                 05221000 T  0006
                               THEN EMITL(TAKEFRST) ELSE EMITL(ELBAT[I].INCR)          05222000 T  0007
                           END;                                                        05223000 T  0010
                        IF TABLE(I) = SUPERFRMTID                                      05224000 T  0011
                           THEN BEGIN  BANA; I ← I+1;                                  05225000 T  0012
                             EMITO(SSP); EMITO(ADD); EMITV(ADRES) END;                 05226000 T  0014
                        EMITPAIR(ADRES,LOD) END PASSFORMAT;                            05227000 T  0017
                                                                          37 IS   21 LONG, NEXT SEG    3


               COMMENT  STREAMWORDS EITHER RESERVES OR UNRESERVES STREAM RESERVED       05228000 T  0355
                        WORDS - IT COMPLEMENTS THEIR STATE;                            05229000 T  0355
               PROCEDURE STREAMWORDS;                                                  05230000 T  0355
PRT(550) = STREAMWORDS
                    BEGIN GT1 ← 0;                                                     05231000 T  0355
                     DO BEGIN                                                          05232000 T  0356
                      INFO[1,GT1].LINK←STACKHEAD[GT2←(T←INFO[1,GT1]).ADDRESS];         05233000 T  0357
                      STACKHEAD[GT2] ← T.LINK;                                         05234000 T  0361
                      GT1 ← GT1+2;                                                     05235000 T  0364
                     END UNTIL BOOLEAN(T.FORMAL) END STREAMWORDS;                      05236000 T  0366


                                                                                       05237000 T  0367
                                                                                       05238000 T  0367
                                                                                       05239000 T  0367
                                                                                       05240000 T  0367
                                                                                       05241000 T  0367
                                                                                       05242000 T  0367
                                                                                       05243000 T  0367
                                                                                       05244000 T  0367
                                                                                       05245000 T  0367
                                                                                       05246000 T  0367
                                                                                       05247000 T  0367
                                                                                       05247500 T  0367
                                                                                       05248000 T  0369
                                                                                       05248500 T  0371
                                                                                       05249000 T  0373
                                                                                       05250000 T  0376
                                                                                       05251000 T  0379
                                                                                       05252000 T  0379
                                                                                       05253000 T  0382
```

```
                        DISKADR←(SIZE+29) DIV 30+DISKADR;                                05254000 T  0385
                    END;                                                                 05254500 T  0387
                    END PROGDESCBLDR;                                                    05255000 T  0387



                COMMENT  DOTSYNTAX ANALYSES THE SYNTAX OF A PARTIAL WORD DESIGNATOR,     05267000 T  0387
                         IT REPORTS IF AN ERROR IS FOUND.  IT RETURNS WITH THE           05268000 T  0387
                         LITERALS INVOLVED;                                              05269000 T  0387
                BOOLEAN PROCEDURE DOTSYNTAX(FIRST,SECOND);                               05270000 T  0387
PRT(551) = DOTSYNTAX
                         INTEGER FIRST,SECOND;                                           05271000 T  0387
                         BEGIN                                                           05272000 T  0387
                           LABEL EXIT;                                                   05273000 T  0387
                                                                        START OF SEGMENT **********    38
                             IF STEPI = LFTBRKET THEN                                    05274000 T  0000
                             IF STEPI = LITNO THEN                                       05275000 T  0001
                             IF STEPI = COLON THEN                                       05276000 T  0002
                             IF STEPI = LITNO THEN                                       05277000 T  0004
                             IF STEPI = RTBRKET THEN                                     05278000 T  0005
                    COMMENT IF TESTS ARE PASSED THEN SYNTAX IS CORRECT;                  05279000 T  0007
                             IF (FIRST ← ELBAT[I-3].ADDRESS) ×                           05280000 T  0007
                               (SECOND ← ELBAT[I-1].ADDRESS)≠0 THEN                      05281000 T  0009
                             IF FIRST + SECOND ≤ 48 THEN                                 05282000 T  0012
                    COMMENT IF TESTS ARE PASSED THEN RANGES OF LITERALS ARE O.K.;        05283000 T  0014
                               GO TO EXIT;                                               05284000 T  0014
                             ERR(114);  COMMENT ERROR IF SYNTAX OR RANGE FAILS;          05285000 T  0015
                             DOTSYNTAX ← TRUE; EXIT: END DOTSYNTAX;                      05286000 T  0015
                                                                        38 IS   20 LONG, NEXT SEG    3



                                                                                         05287000 T  0387
                                                                                         05288000 T  0387
                                                                                         05289000 T  0387
                                                                                         05290000 T  0387
                                                                                         05291000 T  0387
                                                                                         05292000 T  0387
                                                                                         05293000 T  0387
                                                                                         05294000 T  0387
                                                                                         05295000 T  0387
                                                                                         05296000 T  0387
                BOOLEAN PROCEDURE RANGE(LOWER,UPPER);                                    05297000 T  0387
PRT(552) = RANGE
                             VALUE LOWER,UPPER;                                          05298000 T  0387
                              REAL LOWER,UPPER;                                          05299000 T  0387
                     COMMENT RANGE TESTS THE CLASS OF THE ITEM IN ELBAT[I] TO SEE IF     05300000 T  0387
                    IT IS GREATER THAN OR EQUAL TO LOWER OR LESS THAN OR EQUAL TO        05301000 T  0387
                    UPPER AND SETS RANGE TO TRUE OR FALSE ACCORDINGLY.  THE ITEMS        05302000 T  0387
                    CLASS MUST BE IN ELCLASS;                                            05303000 T  0387
                    RANGE←ELCLASS ≥ LOWER AND ELCLASS ≤ UPPER;                           05304000 T  0387



                COMMENT GET OBTAINS A SYLLABLE FROM EDOC, THE ARRAY INTO WHICH CODE IS   05305000 T  0392
```

```
                    EMITTED;                                                                    05306000 T  0392
                INTEGER PROCEDURE GET(L); VALUE L; REAL L;                                      05307000 T  0392
                        BEGIN                                                                   05308000 T  0392
                          INTEGER STREAM PROCEDURE GETSYL(W,S); VALUE S;                        05309000 T  0392
                                                                                    START OF SEGMENT **********    39
PRT(553) = GETSYL
                              BEGIN DI ← LOC GETSYL; DI ← DI+6;                                 05310000 T  0000
                                    SI ← W; SI ← SI+S; SI ← SI+S; DS ← 2 CHR END;               05311000 T  0000



                        GET←GETSYL(CODE(    L DIV 4+1),L,[4612]); END  GET;                     05312000 T  0003
                                                                                      39 IS   12 LONG, NEXT SEG     3



                COMMENT  CALL SWITCH PERFORMS THE FINAL MESS OF GETTING A PROPER DE-            05313000 T  0392
                         SCRIPTOR TO THE TOP OF THE STACK;                                      05314000 T  0392
                PROCEDURE CALLSWITCH(H); VALUE H; REAL H;                                       05315000 T  0392
PRT(554) = CALLSWITCH
                        BEGIN EMITV(GNAT(H)); EMITO(PRTE); EMITO(LOD) END CALLSWITCH;           05316000 T  0392



                                                                                                05317000 T  0396
                                                                                                05318000 T  0396
                                                                                                05319000 T  0396
                                                                                                05320000 T  0396
                                                                                                05321000 T  0396
                                                                                                05322000 T  0396
                                                                                                05323000 T  0396
                                                                                                05324000 T  0396
                                                                                                05325000 T  0396
                                                                                                05325010 T  0396

                    PROCEDURE WRITEPRT(PORS,N,GS); VALUE PORS,N,GS; INTEGER PORS,N,GS;           05325020 T  0396
PRT(555) = WRITEPRT                                                                             05325030 T  0396
                    BEGIN                                                           START OF SEGMENT **********    40
                    LABEL EXIT;                                                                 05325040 T  0000

                    STREAM PROCEDURE FILLIT(LIN,PORS,CELL,N,ID);                                05325050 T  0000
PRT(556) = FILLIT                                                                               05325060 T  0000
                                        VALUE PORS,CELL,N;                                      05325070 T  0000
                         BEGIN                                                                  05325080 T  0000
                         LOCAL COUNT;                                                           05325090 T  0000
                         LABEL M0,M1,M2,M3,M4,M5,M6,M7,XIT;                                     05325100 T  0000
                         SI:=LOC PORS; SI:=SI+3; DI:=LIN; % "PRT" OR "STACK".                   05325110 T  0001
                         IF SC="P" THEN                                                         05325120 T  0002
                              BEGIN  DS:=3 CHR; DS:=LIT"(";  END                                05325130 T  0002
                         ELSE BEGIN                                                             05325140 T  0004
                              DS:=5 CHR; DS:=LIT"("; SI:=LOC CELL; SI:=SI+5;                    05325150 T  0005
                              IF SC≥"6" THEN DS:=2 LIT"F-" ELSE DS:=2 LIT"F+";                  05325160 T  0006
                              COUNT:=DI; DI:=LOC CELL; DI:=DI+4;                                05325170 T  0007
                              DS:=11 RESET; DI:=COUNT;                                          05325180 T  0007
                              END;
                         SI:=LOC CELL; SI:=SI+4; TALLY:=4; % LOCATION.
```

```
                    3(IF SC="0" THEN % DONT PRINT LEADING ZEROES.                                  05325190 T  0007
                         BEGIN SI:=SI+1; TALLY:=TALLY+63  END ELSE JUMP OUT);                      05325200 T  0008
                    COUNT:=TALLY; DS:=COUNT CHR; TALLY:=0; COUNT:=TALLY;                           05325210 T  0011
                    DS:=4 LIT") = "; CELL:=DI; % SAVE OUR PLACE.                                   05325220 T  0012
                    CI:=CI+N;                                                                      05325230 T  0013
                    GO M0;                                                                         05325240 T  0013
                    GO M1;                                                                         05325250 T  0014
                    GO M2;                                                                         05325260 T  0014
                    GO M3;                                                                         05325270 T  0014
                    GO M4;                                                                         05325280 T  0014
                    GO M5;                                                                         05325290 T  0015
                    GO M6;                                                                         05325300 T  0015
                    GO M7;                                                                         05325310 T  0015
               M0:  SI:=ID; SI:=SI+2; DI:=LOC COUNT;                                               05325320 T  0015
                    DI:=DI+7; DS:=CHR; DI:=CELL;  DS:=COUNT CHR;                                   05325330 T  0016
                    GO XIT;                                                                        05325340 T  0018
               M1:  DI:=CELL; DS:=19 LIT"*TEMPORARY STORAGE*"; GO XIT;                              05325350 T  0018
               M2:  DI:=CELL;                                                                      05325360 T  0022
                      DS:=36 LIT"*LIST, LABEL, OR SEGMENT DESCRIPTOR*"; GO XIT;                    05325370 T  0023
               M3:  DI:=CELL; DS:=27 LIT"*CASE STATEMENT DESCRIPTOR*"; GO XIT;                      05325380 T  0028
               M4:  DI:=CELL; DS:=19 LIT"*FORMAT DESCRIPTOR*"; GO XIT;                             05325390 T  0033
PRT(557) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               M5:  DI:=CELL; DS:=24 LIT"*OUTER BLOCK DESCRIPTOR*"; GO XIT;                        05325400 T  0037
PRT(560) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               M6:  DI:=CELL; DS:=20 LIT"*SEGMENT DESCRIPTOR*"; GO XIT;                            05325410 T  0041
PRT(561) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               M7:  DI:=CELL; DS:=18 LIT"*LABEL DESCRIPTOR*";                                      05325420 T  0045
PRT(562) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
               XIT:                                                                                05325430 T  0048
                    END FILLIT;                                                                    05325440 T  0048
PRT(563) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
```

```
                     BLANKET(14,LIN);                                                              05325450 T  0049
                     IF N=1 THEN FILLIT(LIN,PORS,GS,0,ACCUM[1])                                    05325460 T  0051
                ELSE IF N>1 THEN FILLIT(LIN,PORS,GS,0,INFO[N,LINKR,N,LINKC])                       05325470 T  0055
                ELSE FILLIT(LIN,PORS,GS,ABS(N),N);                                                 05325480 T  0063
                    IF NOHEADING THEN DATIME; WRITELINE;                                           05325490 T  0067
                    END WRITEPRT;                                                                  05325500 T  0079
                                                                                          40 IS   80 LONG, NEXT SEG     3


                 COMMENT  GETSPACE MAKES ASSIGNMENTS TO VARIABLES AND DESCRIPTORS IN               05326000 T  0396
                          THE STACK AND PRT.  PERMANENT TELLS WHETHER IT IS A                      05327000 T  0396
                          PERMANENTLY ASSIGNED CELL (ALWAYS IN PRT) OR NOT.  NON                   05328000 T  0396
                          PERMANENT CELLS ARE EITHER IN STACK OR PRT ACCORDING TO                  05329000 T  0396
                          MODE.  CARE IS TAKEN TO REUSE NON PERMANENT PRT CELLS;                   05330000 T  0396
                INTEGER PROCEDURE GETSPACE(PERMANENT,L); VALUE PERMANENT,L;                        05331000 T  0396
                                                         BOOLEAN PERMANENT; INTEGER L;             05333000 T  0396
                        BEGIN LABEL L1,L2,EXIT;                                                    05334000 T  0396
                                                                                         START OF SEGMENT **********   41
                        STREAM PROCEDURE DOIT(C,A,I,S); VALUE C,A;                                 05334100 T  0000
PRT(564) = DOIT
```

```
                    BEGIN LOCAL N;                                                  05334200 T  0000
                         DI←S; DS←8 LIT" "; SI←S; DS←9 WDS;                         05334300 T  0000
                         SI←I; SI←SI+2; DI←LOC N; DI←DI+7; DS←CHR;                  05334400 T  0002
                         DI←S; SI←LOC C; 2(DS←4 DEC);                               05334500 T  0003
                         SI←I; SI←SI+3; DS←N CHR;                                   05334600 T  0004
                    END;                                                            05334700 T  0005



                      BOOLEAN M,Q;                                                  05343000 T  0005
STACK(F+3) = M
STACK(F+4) = Q
                      INTEGER ROW,COL,GS;                                           05344000 T  0005
STACK(F+5) = ROW
STACK(F+6) = COL
STACK(F+7) = GS
               IF NOT(STREAMTOG AND (LEVEL>2))THEN                                  05344400 T  0005
                    IF STEPI=RELOP THEN                                             05344500 T  0007
                         BEGIN                                                      05344510 T  0009
                              IF STEPI>IDMAX                                        05344520 T  0009
                                THEN                                                05344530 T  0010
                                    BEGIN                                           05344540 T  0010
                                         IF ELCLASS=ADOP                            05344550 T  0011
                                              THEN                                  05344560 T  0011
                                                   IF ELBAT[I].ADDRESS=SUBOP        05344570 T  0011
                                                        THEN GS←FZERO ELSE GS←512   0534458= T  0013
                                    ELSE                                            05344590 T  0015
                               BEGIN GS←0;I←I-1 END;                                05344600 T  0016
                              IF STEPI≠LITNO THEN FLAG(51);                         05344610 T  0018
                              IF ELBAT[I].ADDRESS≥512 THEN GS←1024;                 05344615 T  0021
                              GS←GS+ELBAT[I].ADDRESS                                05344620 T  0023
                         END                                                        05344630 T  0024
                           ELSE                                                     05344640 T  0025
                              BEGIN                                                 05344650 T  0025
                                   GS←ELBAT[I].ADDRESS;                             05344660 T  0028
                                   IF GS=0 THEN FLAG(51);                           05344661 T  0029
                                   IF GS≥FZERO AND GS≤1023 THEN GS←-GS;              05344662 T  0031
                                   IF STEPI≠ADOP THEN I←I-1 ELSE                    05344670 T  0034
                                        BEGIN                                       05344680 T  0037
                                        STEPIT;                                     05344690 T  0038
                                        GS←ELBAT[I].ADDRESS+                        05344700 T  0038
                                             (IF ELBAT[I-1].ADDRESS=SUBOP           05344710 T  0039
                                                  THEN -GS ELSE +GS);               05344720 T  0041
                                        END;                                        05344730 T  0044
                                   GS←ABS(GS);                                      05344740 T  0044
                              END; Q←GS<512 OR GS>1023;                             05344750 T  0045
          GO TO EXIT                                                                05344760 T  0047
          END ELSE I←I-1;                                                           05344770 T  0047
IF MODE = 0 OR PERMANENT                                                            05345000 T  0049
     THEN BEGIN                                                                     05346000 T  0050
          IF PRTIMAX > 1023 THEN FLAG(148);                                         05347000 T  0051
          IF ASTOG THEN FLAG(505);                                                  05348000 T  0053
          PRTI ←                                                                    05349000 T  0054
          PRTIMAX←(GS←PRTIMAX)+1;                                                   05350000 T  0054
          IF STUFFTOG THEN IF (M←(LEVEL=1 AND KLASSF>19)) OR                        05350100 T  0057
             (LEVEL≥3 AND ELBAT[I].CLASS=LABELID) THEN BEGIN                        05350120 T  0060
```

```
                    IF NOT M THEN                                                    05350140 T  0063
                    DOIT(LABELID,GS,INFO[(ELBAT[I]).LINKR,                           05350160 T  0064
                    (ELBAT[I].LINKC+1)],TWXA[0])  ELSE                               05350180 T  0067
                    DOIT(KLASSF,GS,INFO[(LASTINFO+1).LINKR,(LASTINFO+1).LINKC]       05350200 T  0069
                              ,TWXA[0]);  WRITE(STUFF,10,TWXA[*]) END; END           05350300 T  0074
                 ELSE BEGIN                                                          05369000 T  0079
                    IF STACKCTR > 767 THEN FLAG(149);                                05370000 T  0080
                    STACKCTR ← (GS ← STACKCTR)+1; Q ← FALSE;                         05371000 T  0082
                    GO TO EXIT END;                                                  05372000 T  0084
           L2:      IF GS ≥ 512 THEN GS ← GS+1024;                                   05373000 T  0085
                    Q ← TRUE;                                                        05374000 T  0087
           EXIT:    GETSPACE ← GS;                                                   05375000 T  0088
                    IF GS≥NESTCTR AND GS<FZERO THEN NESTCTR←GS+1;                    05375100 T  0089
                    IF GS > 1023 THEN GS ← GS-1024;                                  05376000 T  0093
           IF PRTOG THEN WRITEPRT(IF Q THEN "PRT   " ELSE "STACK",L,B2D(GS));        05376100 T  0095
                END GETSPACE;                                                        05378000 T  0100
                                                                          41 IS  107 LONG, NEXT SEG     3


          REAL PROCEDURE DEPTH(I); VALUE I; REAL I;                                  05400000 T  0396
PRT(565) = DEPTH
                    BEGIN REAL   J,K,T,S,M;                                          05401000 T  0396
                                                                        START OF SEGMENT **********    42

STACK(F+3) = J
STACK(F+4) = K
STACK(F+5) = T
STACK(F+6) = S
STACK(F+7) = M
                    IF T←NESTPRT[I]<0 THEN                                           05402000 T  0000
                         BEGIN DEPTH←CALL[T.[22:13]-1].[35:13];                      05402100 T  0002
                              IF NESTPRT[I].[2:1]=0 THEN NESTCUR←NESTCUR+1;          05402200 T  0007
                              NESTPRT[I].[2:1]←1;                                    05402300 T  0011
                         END                                                         05402400 T  0014
                    ELSE IF T.[9:13]≠0 THEN DEPTH←T.[9:13]                           05403000 T  0014
                    ELSE BEGIN M←0; NESTPRT[I]←-T;                                   05404000 T  0016
                         J←T.[22:13]; K←CALL[J-1].[22:13];                           05405000 T  0020
                         FOR J←J STEP 1 UNTIL K DO                                   05406000 T  0026
                         IF S←DEPTH(CALL[J])>M THEN M←S;                             05407000 T  0027
                          M←DEPTH←M←CALL[T.[22:13]-1].[35:13];                       05409000 T  0034
                         IF NESTCUR≠0 THEN                                           05409100 T  0040
                         IF NESTPRT[I].[2:1]=0 THEN ELSE                             05409200 T  0041
                         BEGIN T←T&M[9:35:13]; NESTCUR←NESTCUR-1 END                 05409300 T  0044
                         ELSE T←T&M[9:35:13];                                        05409400 T  0047
                         NESTPRT[I]←T;                                               05409500 T  0050
                         END;                                                        05410000 T  0051
                END;                                                                 05411000 T  0051
                                                                          42 IS   56 LONG, NEXT SEG     3


          PROCEDURE NESTSORT(L,U); VALUE L,U; REAL L,U; FORWARD;                     05411100 T  0396
PRT(566) = NESTSORT
          PROCEDURE SORTNEST;                                                        05412000 T  0396
PRT(567) = SORTNEST
```

```
                    BEGIN ARRAY A[0:14];                                                    05413000 T  0396
                                                                                  START OF SEGMENT **********   43
STACK(F+2) = A
                         REAL I,J,K,T;                                              05414000 T  0001
STACK(F+3) = I
STACK(F+4) = J
STACK(F+5) = K
STACK(F+6) = T
                         REAL P,Q;                                                  05414100 T  0001
STACK(F+7) = P
STACK(F+10) = Q
                STREAM PROCEDURE NESTFORM(I,N,L,A); VALUE I,N;                      05415000 T  0001
PRT(570) = NESTFORM
                    BEGIN LOCAL S;                                                  05416000 T  0001
                         DI←A; 15(DS←8 LIT " ");                                    05417000 T  0003
                         DI←LOC S; DI←DI+7; SI←L; SI←SI+10; DS←CHR;                  05418000 T  0005
                         DI←A; DI←DI+I; A←DI;                                       05419000 T  0006
                         DI←DI+6; DS← S CHR;                                        05420000 T  0007
                         DI←A; SI←LOC N; DS←4 DEC;                                  05421000 T  0008
                         DI←A; DS←3 FILL;                                           05422000 T  0008
                    END;                                                            05423000 T  0009


                    FOR I←PRTBASE STEP 1 UNTIL PRTOP DO                             05424000 T  0009
                    IF   NESTPRT[I]≠0 THEN                                          05425000 T  0011
                         BEGIN SORTPRT[Q]←I; Q←Q+1 END;                             05425100 T  0012
                    NESTSORT(0,Q←Q-1);                                              05425200 T  0017
                    FOR P←0 STEP 1 UNTIL Q DO                                       05425300 T  0019
                    BEGIN I←SORTPRT[P]; T←NESTPRT[I];                               05425400 T  0021
                         NESTFORM(0,DEPTH(I),INFO[T.LINKR,T.LINKC],A);              05426000 T  0023
                         WRITE(LINE[DBL],15,A[*]);                                  05427000 T  0028
                         J←T.[22:13]; K←CALL[J-1].[22:13];                          05428000 T  0032
                         FOR J←J STEP 1 UNTIL K DO                                  05429000 T  0037
                         BEGIN I←CALL[J];                                           05430000 T  0039
                              T←NESTPRT[I];                                         05430500 T  0042
                              NESTFORM(32,DEPTH(I),INFO[T.LINKR,T.LINKC],A);        05431000 T  0043
                              WRITE(LINE,15,A[*]);                                  05432000 T  0048
                         END;                                                       05433000 T  0052
                         WRITE(LINE[DBL]);                                          05434000 T  0054
                    END;                                                            05435000 T  0058
                    END;                                                            05436000 T  0061
                                                                                  43 IS   66 LONG, NEXT SEG     3


                PROCEDURE NESTSORT(L,U); VALUE L,U; REAL L,U;                       05437000 T  0396
                    BEGIN REAL I,J,K,M;                                             05438000 T  0396
                                                                                  START OF SEGMENT **********   44
STACK(F+2) = I
STACK(F+3) = J
STACK(F+4) = K
STACK(F+5) = M
                         LABEL AGAIN,TOP,BOTTOM,EXIT;                               05439000 T  0000
                         IF L≠U THEN                                                05440000 T  0000
```

```
          BEGIN M← (U+L) DIV 2;                                                  05441000 T  0000
               NESTSORT(L,M);                                                    05442000 T  0003
               NESTSORT(M+1,U);                                                  05443000 T  0004
               I←K←L; J←M+1;                                                     05444000 T  0005
AGAIN:         IF I>M THEN GO TO TOP;                                            05445000 T  0008
               IF J>U THEN GO TO BOTTOM;                                         05446000 T  0009
               GT1←NESTPRT[SORTPRT[I],[33:15]].LINK;                             05447000 T  0010
               GT2←NESTPRT[SORTPRT[J],[33:15]].LINK;                             05448000 T  0013
               IF INFO[GT1,LINKR,(GT1+1),LINKC],[18:30]≤                         05449000 T  0016
                  INFO[GT2,LINKR,(GT2+1),LINKC],[18:30] THEN                     05450000 T  0019
                    GO TO BOTTOM;                                                05451000 T  0022
TOP:           SORTPRT[K],[18:15]←SORTPRT[J];                                    05452000 T  0023
               J←J+1;                                                            05453000 T  0026
               IF K←K+1≤U THEN GO TO AGAIN ELSE GO TO EXIT;                      05454000 T  0027
BOTTOM:        SORTPRT[K],[18:15]←SORTPRT[I];                                    05455000 T  0030
               I←I+1;                                                            05456000 T  0033
               IF K←K+1≤U THEN GO TO AGAIN ELSE GO TO EXIT;                      05457000 T  0034
EXIT:          FOR I←L STEP 1 UNTIL U DO                                         05458000 T  0037
               SORTPRT[I]←SORTPRT[I],[18:15];                                    05459000 T  0039
          END;                                                                   05460000 T  0043
     END;                                                                        05461000 T  0043
                                                                  44 IS   46 LONG, NEXT SEG    3


COMMENT      ROUTINES IN THIS SECTION COMPILE CODE FOR ALL EXPRESSIONS;          06000000 T  0396
COMMENT   AEXP IS THE ARITHMETIC EXPRESSION ROUTINE;                            06001000 T  0396
PROCEDURE AEXP;                                                                 06002000 T  0396
          BEGIN                                                                 06003000 T  0396
               IF ELCLASS = IFV                                                 06004000 T  0396
                 THEN BEGIN IF IFEXP ≠ ATYPE THEN ERR(102) END                  06005000 T  0396
                 ELSE BEGIN ARITHSEC; SIMPARITH END                             06006000 T  0399
          END AEXP;                                                             06007000 T  0401


COMMENT ARITHSEC COMPILES FIRST PRIMARY IN AN ARITHMETIC EXPRESSION.            06008000 T  0401
          IN PARTICULAR IT HANDLES P, +P, -P, AND -P*Q WHERE P                  06009000 T  0401
          AND Q ARE PRIMARIES;                                                  06010000 T  0401
PROCEDURE ARITHSEC;                                                             06011000 T  0401
          BEGIN                                                                 06012000 T  0401
            IF ELCLASS = ADOP                                                   06013000 T  0401
               THEN BEGIN                                                       06014000 T  0402
                 STEPIT;                                                        06015000 T  0403
                 IF ELBAT[I-1].ADDRESS ≠ SUB THEN PRIMARY                       06016000 T  0403
                   ELSE BEGIN                                                   06017000 T  0406
                     PRIMARY;                                                   06018000 T  0407
                     ENDTOG ← LINKTOG; EMITO(CHS);                              06021000 T  0407
                     LINKTOG ← ENDTOG; ENDTOG ← FALSE END END                   06022000 T  0409
               ELSE PRIMARY END ARITHSEC;                                       06023000 T  0410


COMMENT SIMPARITH COMPILES SIMPLE ARITHMETIC EXPRESSIONS ON THE                 06024000 T  0412
```

```
                    ASSUMPTION THAT AN ARITHMETIC PRIMARY HAS ALREADY BEEN          06025000 T  0412
                    COMPILED.  IT ALSO HANDLES THE CASE OF A CONCATENATE            06026000 T  0412
                    WHERE ACTUALPARAPART CAUSED THE VARIABLE ROUTINE TO             06027000 T  0412
                    COMPILE ONLY PART OF A PRIMARY.  MOST OF THE WORK OF            06028000 T  0412
                    SIMPARITH IS DONE BY ARITHCOMP; AN ARTIFIAL ROUTINE             06029000 T  0412
                    WHICH DOES THE HIERARCHY ANALYSIS USING RECURSION.              06030000 T  0412
                    ARITHCOMP IS A SUBROUTINE ONLY TO GET THIS RECURSION;           06031000 T  0412
                PROCEDURE SIMPARITH;                                                06032000 T  0412
                    BEGIN                                                           06033000 T  0412
                      WHILE ELCLASS = AMPERSAND                                     06034000 T  0412
                         DO BEGIN STEPIT; PRIMARY; PARSE END;                       06035000 T  0412
                      WHILE ELCLASS ≥ EQVOP DO ARITHCOMP END;                       06036000 T  0415



                COMMENT ARITHCOMP IS THE GUTS OF THE ARITHMETIC EXPRESSION ROUTINE  06037000 T  0418
                    ANALYSIS.  IT CALLS PRIMARY AT APPROPRIATE TIMES AND            06038000 T  0418
                    EMITS THE ARITHMETIC OPERATORS.  THE HIERARCHY ANALYSIS         06039000 T  0418
                    IS OBTAINED BY RECURSION;                                       06040000 T  0418
                PROCEDURE ARITHCOMP;                                                06041000 T  0418
                    BEGIN INTEGER OPERATOR, OPCLASS;                                06042000 T  0418
                                                                  START OF SEGMENT **********    45
STACK(F+2) = OPERATOR
STACK(F+3) = OPCLASS
                     DO BEGIN                                                       06043000 T  0000
                      OPERATOR ← 1 & ELBAT[I] [36:17:10];                           06044000 T  0000
                COMMENT THIS SETS UP THE OPERATOR WHICH WILL BE EMITTED.  THE HIGH   06045000 T  0002
                    ORDER TEN BITS OF THE OPERATOR ARE LOCATED IN [17:10]           06046000 T  0002
                    OF THE ELBAT WORD;                                              06047000 T  0002
                      OPCLASS ← ELCLASS;                                            06048000 T  0002
                      STEPIT; PRIMARY;                                              06049000 T  0002
                         BEGIN                                                      06051000 T  0003
                           WHILE OPCLASS < ELCLASS DO ARITHCOMP;                    06052000 T  0003
                COMMENT THE CLASSES ARE ARRANGED IN ORDER OF HIERARCHY;             06053000 T  0006
                           EMIT(OPERATOR);                                          06054000 T  0006
                           EMIT(0); L ← L-1;                                        06054100 T  0007
                      STACKCT ← 1;                                                  06054150 T  0009
                         END;                                                       06054200 T  0009
                     END UNTIL OPCLASS ≠ ELCLASS END ARITHCOMP;                     06055000 T  0009
                                                                  45 IS   14 LONG, NEXT SEG     3



                INTEGER PROCEDURE EXPRSS; BEGIN AEXP; EXPRSS ← ATYPE END;           06057000 T  0418



                PROCEDURE POLISHER(EXPECT); VALUE EXPECT; REAL EXPECT;              06060000 T  0422
                    BEGIN LABEL EXIT;                                               06061000 T  0422
                                                                  START OF SEGMENT **********    46
                      LABEL EL;                                                     06061900 T  0000
                      REAL COUNT,T1, T2;                                            06062000 T  0000
STACK(F+2) = COUNT
STACK(F+3) = T1
```

```
STACK(F+4) = T2

STACK(F+5) = S

STACK(F+6) = SSS
STACK(F+7) = Z

PRT(571) = WRITEOUT
```

```
BOOLEAN S;                                                          06063000 T  0000

REAL SSS;    INTEGER Z;                                             06063500 T  0000


STREAM PROCEDURE WRITEOUT(C,N,L); VALUE C,N;                        06064000 T  0000

     BEGIN DI ← L; DS ← 2 LIT "S=";                                 06065000 T  0000
          SI ← LOC C; SI ← SI+7; DS ← CHR;                          06066000 T  0000
          SI ← LOC N; DS ← DEC;                                     06067000 T  0001
          58(DS←2LIT " ");                                          06067500 T  0002
     END;                                                           06068000 T  0003



SSS← STACKCTR;                                                      06068500 T  0003
IF STEPI ≠ LEFTPAREN THEN GO TO EXIT;                               06069000 T  0004
DO BEGIN                                                            06070000 T  0006
     IF STEPI ≥ OPERATORS THEN                                      06071000 T  0007
          BEGIN T1 ← (T2 ← ELBAT[I]).ADDRESS;                       06072000 T  0008
               S ← S OR COUNT ≠ T2.[11:3] < 0;                      06074000 T  0010
               COUNT ← T2.[14:2]+COUNT-2;                           06075000 T  0013
               IF ELCLASS ≥ OPERATOR THEN                           06076000 T  0015
               BEGIN IF T1 ≠ 0 THEN EMITO(T1)                       06077000 T  0016
               ELSE BEGIN                                           06078000 T  0018
                    T1 ← T2.LINK+2;                                 06079000 T  0019
                    T2 ← T2.INCR+T1;                                06080000 T  0021
                    FOR T1 ← T1 STEP 1 UNTIL T2 DO                  06081000 T  0022
                      EMIT(TAKE(T1));                               06082000 T  0024
                    END;                                            06083000 T  0027
      END ELSE BEGIN T2 ← ELCLASS;                                  06084000 T  0027
               IF STEPI ≠ LITNO THEN                                06085000 T  0028
                    BEGIN ERR(500); GO TO EXIT END;                 06086000 T  0029
               IF T2 = BITOP THEN EMIT(T1&C                         06087000 T  0031
                    [36:42:6]) ELSE                                 06088000 T  0033
               IF T2 = HEXOP THEN EMIT(T1&                          06089000 T  0034
                    (T2←C DIV 6)[36:45:3]&(C-T2×6)                  06090000 T  0036
                    [39:45:3]) ELSE                                 06091000 T  0039
              IF T2 = ISOLATE THEN                                  06092000 T  0041
                    BEGIN T2 ← C;                                   06093000 T  0042
                         IF STEPI ≠ LITNO                           06094000 T  0043
                         THEN BEGIN ERR(500);                       06095000 T  0044
                         GO TO EXIT END;                            06096000 T  0045
                                                                    06097000 T  0046
                                                                    06098000 T  0046
                                                                    06099000 T  0046
                         EMIT(Z←((T2←C-1)DIV 6-C DIV                06099100 T  0046
                        6+1)×512←(48-T2-C)MOD 6×64←                 06099200 T  0048
                         37);                                       06100000 T  0052
                    END END;                                        06101000 T  0054
      STEPIT;                                                       06102000 T  0054
      S ← S OR COUNT < 0;                                           06103000 T  0054
END ELSE BEGIN                                                      06104000 T  0056
     IF ELCLASS = LABELID THEN                                      06104100 T  0056
          BEGIN T1:=2;                                              06104200 T  0057
```

```
EL:                     GT4 ← TAKE(T2←GIT(ELBAT[I]));                    06104300 T  0058
                        PUT(L,T2);                                       06104400 T  0061
                        IF GT4 = 0 THEN GT4 ← L;                         06104500 T  0062
                        IF (GT4:=L-GT4)DIV 4 ≥ 128 THEN                  06104510 T  0064
                           BEGIN GT4:=0;FLAG(50);END;                    06104520 T  0066
                        EMIT(GT4×4+T1);                                  06104600 T  0068
                        STEPIT;                                          06104700 T  0070
                     END ELSE                                            06104800 T  0071
                IF ELCLASS ≠ PERIOD THEN AEXP ELSE BEGIN                 06105000 T  0071
                T2←0;                                                    06106000 T  0073
                IF STEPI=PERIOD THEN                                     06106100 T  0074
                BEGIN T2←1; STEPIT END;                                  06106200 T  0075
                IF ELCLASS>IDMAX THEN                                    06106300 T  0077
                     BEGIN ERR(500); GO TO EXIT END;                     06107000 T  0078
                IF ELCLASS = LABELID THEN                                06107100 T  0079
                     BEGIN T1 ← 0; GO TO EL END;                         06107200 T  0080
                IF T1 ← ELBAT[I].ADDRESS = 0 THEN                        06108000 T  0082
                     BEGIN ERR(100); GO TO EXIT END;                     06109000 T  0084
                 EMITL(T1);                                              06110000 T  0086
                IF T1>1023 THEN                                          06110100 T  0086
                IF T2=0THEN FLAG(500)                                    06110200 T  0087
                ELSE EMITO(PRTE);                                        06110300 T  0089
                 STEPIT;                                                 06111000 T  0091
                END; COUNT ← COUNT+1;                                    06112000 T  0091
                END;                                                     06113000 T  0093
END UNTIL ELCLASS ≠ COMMA;                                               06114000 T  0093
IF ELCLASS ≠ RTPAREN THEN                                                06115000 T  0094
     BEGIN ERR(104); GO TO EXIT END;                                     06116000 T  0095
STEPIT;                                                                  06117000 T  0096
IF FALSE THEN                                                            06118000 T  0097
     BEGIN COUNT ← COUNT-EXPECT;                                         06119000 T  0097
          WRITEOUT(IF COUNT < 0 THEN "-" ELSE                            06120000 T  0099
                   IF COUNT = 0 THEN " " ELSE "+",                       06121000 T  0101
                   ABS(COUNT),LIN[0]);                                   06122000 T  0103
          WRITELINE;                                                     06123000 T  0105
     END;                                                                06124000 T  0115
EXIT: STACKCTR ← SSS; END;                                               06125000 T  0115
                                                                  46 IS  120 LONG, NEXT SEG     3
```

```
PROCEDURE PRIMARY;                                                       06126000 T  0422
        BEGIN LABEL                                                      06127000 T  0422
                                                                  START OF SEGMENT **********    47
             L1,  L2,  L3,  L4,  L5,  L6,  L7,  L8,  L9,  L10,          06128000 T  0000
             L11, L12, L13, L14, L15, L16, L17, L18, L19, L20,           06129000 T  0000
             L21, L22, L23, L24, L25, L26, L27, L28, L29, L30,           06130000 T  0000
             L31, L32, L33, L34, L35, L36, L37, L38, L39;                06131000 T  0000
             SWITCH S ←                                                  06132000 T  0000
             L1,  L2,  L3,  L4,  L5,  L6,  L7,  L8,  L9,  L10,          06133000 T  0002
             L11, L12, L13, L14, L15, L16, L17, L18, L19, L20,           06134000 T  0002
             L21, L22, L23, L24, L25, L26, L27, L28, L29, L30,           06135000 T  0002
             L31, L32, L33, L34, L35, L36, L37, L38, L39;                06136000 T  0002
             LABEL EXIT,RP,LDOT,LAMPER;                                  06137000 T  0022
             GO TO S[ELCLASS];                                           06138000 T  0022
             IF ELCLASS = LFTBRKET THEN                                  06139000 T  0025
```

```
                    BEGIN STEPIT; VARIABLE(FL);                                   06140000 T  0025
                         IF ELCLASS ≠ RTBRKET THEN                                06141000 T  0027
                              BEGIN ERR(118); GO TO EXIT END;                     06142000 T  0028
                         STEPIT;                                                  06143000 T  0030
                         GO TO LDOT;                                              06144000 T  0030
                    END;                                                          06145000 T  0031
               IF ELCLASS = NOTOP THEN                                            06146000 T  0031
                    BEGIN STEPIT; PRIMARY;                                        06147000 T  0031
                         EMITLNG; EMIT(0); L←L-1;                                 06148000 T  0033
                         GO TO EXIT;                                              06149000 T  0035
                    END;                                                          06150000 T  0036
               IF ELCLASS = UNKNOWNID THEN ERR(100);                              06151000 T  0036
L1:L2:L3:L4:L5:L6:L8:L9:L10:L12:L13:L16:L17:L20:L21:L24:L25:L28:L29:               06152000 T  0038
L32:                                                                              06153000 T  0039
               ERR(103); GO TO EXIT;                                              06154000 T  0039
          L7:                                                                     06155000 T  0040
               SUBHAND(FALSE); GO TO LDOT;                                        06156000 T  0041
          L11:                                                                    06157000 T  0042
               IMPFUN; STACKCT ← STACKCT-1; GO TO LDOT;                           06158000 T  0043
          L14:L15:                                                                06159000 T  0045
               STRMPROCSTMT; GO TO LDOT;                                          06160000 T  0046
          L18:L19:                                                                06161000 T  0047
               PROCSTMT(FALSE); GO TO LDOT;                                       06162000 T  0047
          L22:L23:L26:L27:L30:L31:                                                06163000 T  0048
               VARIABLE(FP); GO TO LAMPER;                                        06164000 T  0049
          L33:L35:                                                                06165000 T  0050
               EMIT(0&ELBAT[I] [36:17:10]); STEPIT; GO TO LAMPER;                 06166000 T  0051
          L34:L36:                                                                06167000 T  0054
               EMITNUM(C); STEPIT; GO TO LAMPER;                                  06168000 T  0054
          L38:                                                                    06169000 T  0055
               POLISHER(1); GO TO LDOT;                                           06170000 T  0056
          L39:                                                                    06171000 T  0057
               STEPIT; PRIMARY; STACKCT ← STACKCT-1;                              06172000 T  0058
               EMITO(LOD); GO TO LDOT;                                            06172500 T  0060
          L37:                                                                    06173000 T  0061
               STEPIT; AEXP;                                                      06174000 T  0062
               STACKCT ← STACKCT-1;                                               06174500 T  0063
               IF ELCLASS ≠ RTPAREN THEN                                          06175000 T  0064
                    BEGIN ERR(104); GO TO EXIT END;                               06176000 T  0065
               STEPIT;                                                            06177000 T  0066
          LDOT:DOT;                                                               06178000 T  0067
          LAMPER:                                                                 06179000 T  0068
               STACKCT ← STACKCT +1;                                              06179500 T  0069
               WHILE ELCLASS = AMPERSAND DO                                       06180000 T  0070
                    BEGIN STEPIT; PRIMARY; PARSE END;                             06181000 T  0072
EXIT: END PRIMARY;                                                                06182000 T  0074
                                                                        47 IS   76 LONG, NEXT SEG    3


          PROCEDURE IMPFUN;                                                       06183000 T  0422
               BEGIN REAL T1,T2;                                                  06184000 T  0422
                                                                        START OF SEGMENT **********   48
STACK(F+2) = T1
STACK(F+3) = T2
               T1 ← (T2 ← ELBAT[I]).ADDRESS;                                      06185000 T  0000
```

```
                    PANA;                                                                    06186000 T  0002
                    IF T1 # 0 THEN EMITO(T1)                                                 06187000 T  0002
                         ELSE BEGIN                                                          06188000 T  0004
                         T1 ← T2.LINK+T2.INCR+1;                                             06189000 T  0005
                         T2 ← T2.LINK+2;                                                     06190000 T  0007
                         FOR T2 ← T2 STEP 1 UNTIL T1 DO EMIT(TAKE(T2));                      06191000 T  0009
                         END;                                                                06192000 T  0014
               END;                                                                          06193000 T  0014
                                                                                       48 IS    17 LONG, NEXT SEG     3


          PROCEDURE SUBHAND(FROM); VALUE FROM; BOOLEAN FROM;                                 06194000 T  0422
               BEGIN LABEL EXIT;                                                             06195000 T  0422
                                                                                       START OF SEGMENT **********    49
                    REAL T1;                                                                 06196000 T  0000
STACK(F+2) ← T1
                    T1 ← TAKEFRST;                                                           06197000 T  0000
                    IF ELCLASS # SUBID AND FROM THEN                                         06198000 T  0001
                         BEGIN IF STEPI # ASSIGNOP THEN                                      06199000 T  0002
                              BEGIN FLAG(503); GO TO EXIT END;                               06200000 T  0003
                              STEPIT;                                                        06201000 T  0005
                              AEXP;                                                          06202000 T  0006
                              EMITO(XCH);                                                    06203000 T  0006
                              GO TO EXIT;                                                    06204000 T  0007
                         END;                                                                06205000 T  0007
                    EMITL((L+6) DIV 4-(T1.[24:12]-1) DIV 4);                                 06206000 T  0007
                    EMITB(BBW,BUMPL,T1.[36:12]);                                             06207000 T  0011
                    STEPIT;                                                                  06208000 T  0014
                    ADJUST;                                                                  06208500 T  0014
               EXIT:  END SUBHAND;                                                           06209000 T  0015
                                                                                       49 IS    19 LONG, NEXT SEG     3


          COMMENT IFEXP COMPILES CONDITIONAL EXPRESSIONS.  IT REPORTS THE TYPE               06292000 T  0422
                    OF EXPRESSIONS AS EXPRSS REPORTS;                                        06293000 T  0422
          INTEGER PROCEDURE IFEXP;                                                           06294000 T  0422
                    BEGIN INTEGER TYPE,THENBRANCH,ELSEBRANCH;                                06295000 T  0422
                                                                                       START OF SEGMENT **********    50
STACK(F+3) ← TYPE
STACK(F+4) ← THENBRANCH
STACK(F+5) ← ELSEBRANCH
                         IFCLAUSE;                                                           06296000 T  0000
                    STACKCT ← 0;                                                             06296500 T  0000
                         THENBRANCH ← BUMPL;                                                 06297000 T  0001
               COMMENT SAVE L FOR LATER FIXUP;                                               06298000 T  0003
                         IFEXP ← TYPE ← EXPRSS; COMMENT COMPILE 1ST EXPRSS;                   06299000 T  0003
                    STACKCT ← 0;                                                             06299500 T  0004
                         ELSEBRANCH ← BUMPL;                                                 06300000 T  0005
                         EMITB(BFC,THENBRANCH,L);                                            06301000 T  0007
                         IF ELCLASS # ELSEV THEN ERR(155) ELSE BEGIN                         06302000 T  0008
                         STEPIT;                                                             06303000 T  0010
                    AEXP; STACKCT ← 1;                                                       06305000 T  0011
                         COMMENT THIS COMPILES PROPER TYPE SECOND EXPRSS;                     06306000 T  0012
```

```
                    EMITB(BFW,ELSEBRANCH,L);                                    06307000 T  0012
                    EMIT(1); L = L-1;                                           06308000 T  0013
          COMMENT THIS IS USED BY EMITLNG TO CLEANUP CODE.  COMPARE WITH        06309000 T  0015
                BOOSEC, BOOCOMP, AND RELATION;                                  06310000 T  0015
                END END IFEXP;                                                  06311000 T  0015
                                                                      50 IS   19 LONG, NEXT SEG    3


          COMMENT PARSE COMPILES CODE FOR THE CONCATENATE;                       06312000 T  0422
          PROCEDURE PARSE;                                                      06313000 T  0422
                BEGIN INTEGER FIRST,SECOND,THIRD;                               06314000 T  0422
                                                                    START OF SEGMENT **********   51
STACK(F+2) = FIRST
STACK(F+3) = SECOND
STACK(F+4) = THIRD
                    LABEL EXIT;                                                 06315000 T  0000
                    IF ELCLASS = LFTBRKET THEN                                  06316000 T  0000
                    IF STEPI = LITNO THEN                                       06317000 T  0000
                    IF STEPI = COLON THEN                                       06318000 T  0002
                    IF STEPI = LITNO THEN                                       06319000 T  0003
                    IF STEPI = COLON THEN                                       06320000 T  0005
                    IF STEPI = LITNO THEN                                       06321000 T  0006
                    IF STEPI = RTBRKET THEN                                     06322000 T  0008
          COMMENT IF TEST ARE PASSED THEN SYNTAX IS CORRECT;                     06323000 T  0009
                    IF (FIRST  = ELBAT[I-5],ADDRESS) ×                          06324000 T  0009
                       (SECOND = ELBAT[I-3],ADDRESS) ×                          06325000 T  0012
                       (THIRD  = ELBAT[I-1],ADDRESS) ≠ 0 THEN                   06326000 T  0014
                      IF FIRST + THIRD ≤ 48 THEN                                06327000 T  0017
                      IF SECOND+ THIRD ≤ 48 THEN                                06328000 T  0019
          COMMENT IF TEST ARE PASSED THEN RANGES OF LITERALS ARE O.K.;          06329000 T  0020
                       BEGIN                                                    06330000 T  0020
                         STEPIT;                                                06331000 T  0021
                          EMITD(SECOND,FIRST,THIRD);                            06332000 T  0021
                STACKCT = 1;                                                    06332500 T  0023
                          GO TO EXIT END;                                       06333000 T  0023
                     ERR(113); COMMENT ERROR IF SYNTAX OR RANGE FAILS;          06334000 T  0024
          EXIT:     END PARSE;                                                  06335000 T  0025
                                                                      51 IS   28 LONG, NEXT SEG    3


          COMMENT DOT COMPILES CODE FOR PARTIAL WORD DESIGNATORS, EXCEPT FOR     06336000 T  0422
                THOSE CASES HANDLED BY THE VARIABLE ROUTINE;                    06337000 T  0422
          PROCEDURE DOT;                                                        06338000 T  0422
                BEGIN INTEGER FIRST,SECOND; LABEL EXIT;                         06339000 T  0422
                                                                    START OF SEGMENT **********   52
STACK(F+2) = FIRST
STACK(F+3) = SECOND
                  IF ELCLASS = PERIOD THEN BEGIN                                06340000 T  0000
                        IF DOTSYNTAX(FIRST,SECOND) THEN GO TO EXIT;             06341000 T  0001
                                                                                06342000 T  0003
                                                                                06343000 T  0003
                    EMITI(0,FIRST,SECOND);                                      06344000 T  0003
                        STEPIT;                                                 06345000 T  0004
```

```
   EXIT; END END DOT;                                                                  06346000 T  0004
                                                                                 52 IS     8 LONG, NEXT SEG     3


PROCEDURE IFCLAUSE;                                                                    06409000 T  0422
            BEGIN STEPIT; BEXP;                                                        06410000 T  0422
              IF ELCLASS # THENV THEN ERR(116)ELSE STEPIT END IFCLAUS;                 06411000 T  0424
COMMENT  PANA COMPILES THE CONSTRUCT: (<ARIT. EXP.>);                                  06412000 T  0427


PROCEDURE PANA;                                                                        06413000 T  0427
            BEGIN                                                                      06414000 T  0427
             IF STEPI # LEFTPAREN THEN ERR(105)                                        06415000 T  0427
             ELSE BEGIN STEPIT; AEXP; IF ELCLASS # RTPAREN THEN                        06416000 T  0429
                 ERR(104) ELSE STEPIT END END PANA;                                    06417000 T  0432


COMMENT  BANA COMPILES THE CONSTRUCT:  [<ARITH. EXP.>];                                06418000 T  0435
PROCEDURE BANA;                                                                        06419000 T  0435
            BEGIN                                                                      06420000 T  0435
             IF STEPI # LFTBRKET THEN ERR(117)                                         06421000 T  0435
                 ELSE BEGIN STEPIT; AEXP; IF ELCLASS # RTBRKET THEN                    06422000 T  0436
                   ERR(118) ELSE STEPIT END END BANA      ;                           06423000 T  0439


          COMMENT THIS SECTION CONTAINS THE STATEMENT ROUTINES;                        07000000 T  0442
          COMMENT COMPOUNDTAIL COMPILES COMPOUNDTAILS.  IT ALSO ELIMINATES             07001000 T  0442
                  COMMENTS FOLLOWING ENDS.  AFTER ANY ERROR, ERROR MESSAGES            07002000 T  0442
                  ARE SUPPRESSED.  COMPOUNDTAIL IS PARTIALLY RESPONSIBLE               07003000 T  0442
                  FOR RESTORING THE ABILITY TO WRITE ERROR MESSAGES.  SOME             07004000 T  0442
                  CARE IS ALSO TAKEN TO PREVENT READING BEYOND THE "END.";             07005000 T  0442
        PROCEDURE COMPOUNDTAIL;                                                        07006000 T  0442
PRT(572) = COMPOUNDTAIL
                  BEGIN LABEL ANOTHER;                                                 07007000 T  0442
                                                                              START OF SEGMENT **********   53
                    I ← I-1; BEGINCTR ← BEGINCTR+1;                                    07008000 T  0000
           ANOTHER:   ERRORTOG ← TRUE;   COMMENT  ALLOW ERROR MESSAGES;                07009000 T  0002
                    STEPIT;                                                            07010000 T  0003
                    IF STREAMTOG THEN STREAMSTMT ELSE STMT;                            07011000 T  0004
                    IF ELCLASS = SEMICOLON THEN GO TO ANOTHER;                         07012000 T  0006
                    IF ELCLASS # ENDV                                                  07013000 T  0007
                      THEN BEGIN                                                       07014000 T  0008
                       ERR(119); GO TO ANOTHER END;                                    07015000 T  0009
                  ENDTOG←TRUE;                                                         07016000 T  0010
                  DO STUPDEFINE←TRUE UNTIL                                             07017000 T  0011
                       STEPI≤ENDV AND ELCLASS≥UNTILV                                   07018000 T  0011
                       OR NOT ENDTOG;                                                  07019000 T  0013
                  ENDTOG←FALSE;                                                        07020000 T  0015
                       IF BEGINCTR ← BEGINCTR-1 # 0 EQV ELCLASS = PERIOD               07021000 T  0015
```

```
                           THEN BEGIN                                              07022000 T  0017
                              IF BEGINCTR = 0 THEN                                 07023000 T  0019
                               BEGIN FLAG(143); BEGINCTR ← 1; GO ANOTHER END;      07024000 T  0019
            FLAG (120);                                                            07025000 P  0022
            FCR:= (LCR:=MKABS(CBUFF[9]))-9;                                        07025010 C  0023
                IF LISTER THEN PRINTCARD;                                          07025020 C  0026
            FCR:= (LCR:=MKABS(TBUFF[9]))-9  END;                                   07025030 C  0042
                         IF ELCLASS = PERIOD THEN                                  07026000 T  0045
                           BEGIN                                                   07027000 T  0045
                              GT5 ← "ND;END."&"E"[1:43:5];                         07028000 T  0046
                              MOVE(1,GT5,CBUFF[0]);                                07029000 T  0048
                              LASTUSED←4;                                          07030000 T  0049
                              ELBAT[I←I-2] ←SPECIAL[20];                           07031000 T  0050
                              ELCLASS ← SEMICOLON END                              07032000 T  0052
                       END COMPOUNDTAIL;                                           07033000 T  0053
                                                                     53 IS    55 LONG, NEXT SEG    3


             REAL AXNUM;                                                           07034000 T  0442
PRT(573) = AXNUM
             PROCEDURE ACTUALPARAPART(SBIT,INDEX); VALUE SBIT,INDEX;               07035000 T  0442
PRT(574) = ACTUALPARAPART
                   BOOLEAN SBIT; REAL INDEX;                                       07036000 T  0442
                   BEGIN LABEL EXIT,COMMON,ANOTHER,POL;                            07037000 T  0442
                                                                   START OF SEGMENT **********    54
                       REAL PCTR,SCLASS,ACLASS;                                    07038000 T  0000
STACK(F+2) = PCTR
STACK(F+3) = SCLASS
STACK(F+4) = ACLASS
             STREAM PROCEDURE WRITEAX(LINE,ACCUM,N,SEQ); VALUE N;                  07038100 T  0000
PRT(575) = WRITEAX
                   BEGIN DI ← LINE; 15(DS ← 8 LIT " ");                            07038200 T  0000
                       DI ← LINE; SI ← SEQ; SI ← SI-16; DS ← WDS;                  07038300 T  0002
                       DI ← DI+4; DS ← 20 LIT "ACCIDENTAL ENTRY AT ";              07038400 T  0003
                              SI ← ACCUM; SI ← SI+3; DS ← N CHR;                   07038500 T  0006
                       SI ← SEQ; DI ← SEQ; DI ← DI-16; DS ← WDS;                   07038600 T  0007
                   END;                                                            07038700 T  0008


                       BOOLEAN VBIT,IDBIT;                                         07039000 T  0008
STACK(F+5) = VBIT
STACK(F+6) = IDBIT
                       PCTR ← 1;                                                   07040000 T  0008
              ANOTHER: ACLASS ← STEPI&0[47:47:1];                                  07041000 T  0009
                       STACKCT ← 0;                                                07041200 T  0012
                       GT1 ← TAKE(INDEX+PCTR);                                     07042000 T  0012
                       VBIT ← BOOLEAN(GT1.VO);                                     07043000 T  0014
                       SCLASS ← GT1.CLASS&0[47:47:1];                              07044000 T  0015
                       IF VBIT THEN BEGIN AEXP; GO TO COMMON END;                  07045000 T  0018
                       IF SBIT THEN SCLASS ← NAMEID;                               07046000 T  0019
                       IDBIT ← BOOID < ACLASS AND ACLASS < LABELID;                07047000 T  0021
                       IF SCLASS = NAMEID THEN                                     07048000 T  0023
                           BEGIN                                                   07049000 T  0024
```

```
                    IF IDBIT THEN VARIABLE(FL)                                  07050000 T  0024
                    ELSE                                                        07051000 T  0025
POL:                     IF ELCLASS = POLISHV THEN POLISHER(1)                  07052000 T  0026
                    ELSE      ERR(IF ELCLASS=0 THEN 0 ELSE 123);                07053000 T  0028
                    GO TO COMMON;                                               07054000 T  0032
               END;                                                             07055000 T  0032
          IF SCLASS = REALARRAYID THEN                                          07056000 T  0032
               IF ACLASS = REALARRAYID THEN                                     07057000 T  0033
                    BEGIN VARIABLE(FL); GO TO COMMON END                        07058000 T  0034
               ELSE GO TO POL;                                                  07059000 T  0036
          IF SCLASS # REALID THEN                                               07060000 T  0036
               BEGIN FLAG(503);                                                 07061000 T  0037
                    AEXP;                                                       07062000 T  0038
                    ERRORTOG ← TRUE;                                            07063000 T  0039
                    GO TO COMMON;                                               07064000 T  0039
               END;                                                             07065000 T  0040
          GT1 ← TABLE(I+1);                                                     07066000 T  0040
          IF GT1 = COMMA OR GT1 = RTPAREN THEN                                  07067000 T  0042
               BEGIN IF IDBIT THEN                                              07068000 T  0043
                         BEGIN IF ACLASS = REALID AND                           07069000 T  0044
                                   BOOLEAN(ELBAT[I].FORMAL)THEN BEGIN           07070000 T  0045
                                   CHECKER (ELBAT[I]);                          07070500 T  0047
                                   EMITPAIR(ELBAT[I].ADDRESS,LOD);              07071000 T  0048
                                   STEPIT; END                                  07072000 T  0050
                                   ELSE VARIABLE(FL);                           07073000 T  0050
                                   GO TO COMMON END;                            07074000 T  0052
                         IF ELCLASS ≤ STRNGCON AND ELCLASS > LABELID             07075000 T  0052
                              THEN BEGIN PRIMARY; GO TO COMMON END;             07076000 T  0053
                    END;                                                        07077000 T  0055
               EMITO(NOP); EMITO(NOP);                                          07078000 T  0055
               SCLASS ← L;                                                      07079000 T  0057
               ADJUST;                                                          07080000 T  0058
               ACLASS ← L.[36:10];                                              07081000 T  0058
               IF IDBIT THEN                                                    07082000 T  0059
                    BEGIN VARIABLE(FL);                                         07083000 T  0060
                         IF ELCLASS < AMPERSAND THEN GO TO COMMON;              07084000 T  0061
                                                                                07084500 T  0062
                         SIMPARITH;                                             07085000 T  0062
                    END ELSE AEXP;                                              07086000 T  0063
               IF LISTER THEN                                                   07086100 T  0064
                    BEGIN ACCUM[1] ← Q;                                         07086200 T  0064
                         WRITEAX(LIN[0],ACCUM[1],Q,[12:6],                      07086300 T  0066
                              INFO[LASTSEQROW,LASTSEQUENCE]);                   07086400 T  0068
                         WRITELINE;                                             07086500 T  0069
                    END;                                                        07086600 T  0079
               AXNUM ← AXNUM+1;                                                 07086700 T  0079
               EMITO(RTS);                                                      07087000 T  0081
               EMITB(BFW,SCLASS,L);                                             07088000 T  0081
               EMITNUM(ACLASS);                                                 07089000 T  0083
               EMITPAIR(TAKE(PROINFO).ADDRESS,LOD);                             07090000 T  0083
               EMITO(INX);                                                      07091000 T  0085
               EMITN(512);                                                      07092000 T  0086
               EMITD(33,18,15);                                                 07093000 T  0087
               EMIT(0);                                                         07093100 T  0088
               EMITD(5,5,1);                                                    07093200 T  0089
COMMON:        PCTR ← PCTR+1;                                                   07094000 T  0090
               IF ELCLASS = COMMA THEN GO TO ANOTHER;                           07095000 T  0092
```

```
                    IF ELCLASS ≠ RTPAREN THEN                                   07096000 T  0093
                        BEGIN ERR(129); GO TO EXIT END;                         07097000 T  0094
                    IF TAKE(INDEX).NODIMPART+1 ≠ PCTR THEN                      07098000 T  0096
                        BEGIN ERR(128); GO TO EXIT END;                         07099000 T  0098
                    STEPIT;                                                     07100000 T  0100
                    STACKCT ← 0;                                                07100500 T  0100
EXIT:   END ACTUAL PARAPART;                                                    07101000 T  0101
                                                                      54 IS  106 LONG, NEXT SEG     3



    PROCEDURE PROCSTMT(FROM); VALUE FROM; BOOLEAN FROM;                         07391000 T  0442
        BEGIN                                                                   07392000 T  0442
            REAL HOLE,ADDRESS;                                                  07393000 T  0442
                                                                     START OF SEGMENT **********    55
STACK(F+2) = HOLE
STACK(F+3) = ADDRESS
            REAL J; LABEL OK;                                                   07393100 T  0000
STACK(F+4) = J
            LABEL EXIT;                                                         07394000 T  0000
            SCATTERELBAT;                                                       07395000 T  0000
            HOLE← ELBAT[I];                                                     07396000 T  0000
            ADDRESS ← ADDRSF;                                                   07397000 T  0001
            IF NESTOG THEN                                                      07397100 T  0002
            IF MODE≠0 THEN                                                      07397200 T  0002
            IF TABLE(I+1)≠ASSIGNOP THEN                                         07397210 T  0003
            BEGIN FOR J←CALLINFO STEP 1 UNTIL CALLX DO                          07397300 T  0006
                    IF CALL[J]=ADDRESS THEN GO TO OK;                           07397400 T  0008
                CALL[CALLX←CALLX+1]←ADDRESS;                                    07397500 T  0014
OK:         END;                                                                07397600 T  0018
            CHECKER(HOLE);                                                      07398000 T  0019
            IF ELCLASS ≠ PROCID THEN                                            07399000 T  0019
            IF NOT FORMALF THEN                                                 07400000 T  0020
            IF TABLE(I+1) = ASSIGNOP THEN                                       07401000 T  0021
                BEGIN VARIABLE(2←REAL(FROM)); GO TO EXIT END;                   07402000 T  0023
PRT(576) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
            COMMENT  CALL VARIABLE TO HANDLE THIS ASSIGNMENT OPERATION;         07403000 T  0028
            IF ELCLASS ≠ PROCID EQV FROM                                        07404000 T  0028
                THEN BEGIN ERR(159); GO TO EXIT END;                            07405000 T  0029
            COMMENT   IT IS PROCEDURE IF AND ONLY WE COME FROM STMT;            07406000 T  0031
            STEPIT;                                                             07407000 T  0031
            EMITO(MKS);                                                         07408000 T  0032
            IF ELCLASS = LEFTPAREN                                              07409000 T  0032
                THEN ACTUALPARAPART(FALSE,GIT(HOLE))                            07410000 T  0033
                ELSE IF FORMALF THEN L ← L-1                                    07411000 T  0035
                  ELSE IF TAKE(GIT(HOLE)).NODIMPART≠0 THEN ERR(128);            07412000 T  0037
            EMITV(ADDRESS);                                                     07413000 T  0042
EXIT:   END  PROCSTMT;                                                          07425000 T  0042
                                                                      55 IS   46 LONG, NEXT SEG     3



    PROCEDURE STRMPROCSTMT;                                                     07426000 T  0442
        BEGIN REAL WHOLE,FIX,T1;                                                07427000 T  0442
                                                                     START OF SEGMENT **********    56
```

```
STACK(F+2) = WHOLE
STACK(F+3) = FIX
STACK(F+4) = T1
```

```
                    WHOLE ← ELBAT[I]; FIX ← -1;
                    IF ELCLASS ≠ STRPROCID THEN EMIT(0);
                    IF WHOLE.LVL  ≠ 1 THEN
                         BEGIN FIX ← L; L ← L+1 END;
                    EMITO(MKS);
                    T1 ← TAKEFRST.[1:6];
                    FOR GT1 ← 1 STEP 1 UNTIL T1 DO EMIT(0);
                    IF STEPI ≠ LEFTPAREN THEN ERR(128)
                    ELSE BEGIN ACTUALPARAPART(TRUE,GIT(WHOLE));
                         IF FIX < 0 THEN EMITV(WHOLE,ADDRESS)
                         ELSE BEGIN T1 ← L; L ← FIX;
                              WHOLE ← TAKE(GIT(WHOLE));
                              EMITNUM(T1+2-WHOLE.[16:12]);
                              L ← T1;
                              EMITB(BBW,BUMPL,WHOLE.[28:12]);
                         END;
               END END STRMPROCSTMT;
```

```
07428000 T  0000
07429000 T  0000
07430000 T  0000
07431000 T  0002
07432000 T  0004
07433000 T  0005
07434000 T  0007
07435000 T  0008
07436000 T  0010
07437000 T  0014
07438000 T  0015
07439000 T  0018
07440000 T  0019
07441000 T  0022
07442000 T  0024
07443000 T  0026
07444000 T  0027
07445000 T  0030
07446000 T  0030
56 IS   33 LONG, NEXT SEG     3
```

```
          INTEGER PROCEDURE BAE;
                    BEGIN BAE ← BUMPL; CONSTANTCLEAN; ADJUST END BAE;
```

```
07458000 T  0442
07459000 T  0442
```

```
          COMMENT RELSESTMT COMPILES THE RELEASE STATEMENT;
          COMMENT  DOSTMT HANDLES THE DO STATEMENT;
          PROCEDURE DOSTMT;
PRT(577) = DOSTMT
                    BEGIN INTEGER TL;

STACK(F+2) = TL
                    FOULED ← L;

                         STEPIT; TL←L; STMT; IF ELCLASS ≠ UNTILV THEN ERR(131)
                         ELSE BEGIN
                           STEPIT; BEXP; EMITB(BBC,BUMPL,TL) END
                    END DOSTMT;
```

```
07460000 T  0447
07481000 T  0447
07482000 T  0447
07483000 T  0447
START OF SEGMENT **********    57
07483500 T  0000
07484000 T  0000
07485000 T  0000
07486000 T  0004
07487000 T  0005
07488000 T  0008
57 IS   11 LONG, NEXT SEG     3
```

```
          COMMENT  WHILESTMT COMPILES THE WHILE STATEMENT;
          PROCEDURE WHILESTMT;
PRT(600) = WHILESTMT
                    BEGIN INTEGER BACK,FRONT;

STACK(F+2) = BACK
```

```
07489000 T  0447
07490000 T  0447
07491000 T  0447
START OF SEGMENT **********    58
```

```
STACK(F+3) = FRONT
                        FOULED ← L;                                                    07491500 T  0000
                                                                                       07492000 T  0000
                        STEPIT; BACK ← L; BEXP; FRONT ← BUMPL;                         07493000 T  0000
                        IF ELCLASS ≠ DOV THEN ERR(132) ELSE                            07494000 T  0004
                          BEGIN STEPIT; STMT; EMITB(BBW,BUMPL,BACK);                   07495000 T  0006
                             CONSTANTCLEAN; EMITB(BFC,FRONT,L) END END WHILESTMT;      07496000 T  0010
                                                                                  58 IS   14 LONG, NEXT SEG    3



                COMMENT  GOSTMT COMPILES GO TO STATEMENTS.  GOSTMT LOOKS AT THE        07497000 T  0447
                         EXPRESSION.  IF IT IS SIMPLE ENOUGH WE GO DIRECTLY,           07498000 T  0447
                         OTHERWISE A CALL ON THE MCP IS GENERATED IN ORDER TO GET      07499000 T  0447
                         STORAGE RETURNED.  SEE DEXP AND GENGO;                        07500000 T  0447
                PROCEDURE GOSTMT;                                                      07501000 T  0447
PRT(601) = GOSTMT
                    BEGIN                                                              07502000 T  0447
                        REAL ELBW;                                                     07503000 T  0447
                                                                                START OF SEGMENT **********   59
STACK(F+2) = ELBW
                        LABEL GOMCP,EXIT;                                              07504000 T  0000
                        IF STEPI = TOV THEN STEPIT;                                    07505000 T  0000
                        IF ELCLASS = LABELID THEN TB1 ← TRUE                           07506000 T  0002
                        ELSE IF ELCLASS = SWITCHID THEN TB1 ← FALSE                    07507000 T  0003
                        ELSE BEGIN IF ELCLASS = POLISHV THEN                           07511000 T  0006
                                BEGIN POLISHER(1); EMITO(BFW) END                      07512000 T  0007
                             ELSE ERR(501);                                            07513000 T  0009
                             GO TO EXIT                                                07514000 T  0011
                        END;                                                           07515000 T  0011
                    IF NOT LOCAL(ELBAT[I]) THEN                                        07516000 T  0011
                        BEGIN                                                          07516100 T  0012
                            IF TB1 THEN                                                07516200 T  0013
                              BEGIN EMITV(GNAT(ELBAT[I]));                             07516300 T  0013
                                    EMITO(BFW);                                        07516400 T  0015
                                    STEPIT;                                            07516500 T  0016
                                    GO TO EXIT END;                                    07516600 T  0016
                        BEGIN ERR(501); GO TO EXIT END;                                07517000 T  0017
                        END;                                                           07517500 T  0018
                    IF TB1 THEN BEGIN GOGEN(ELBAT[I],BFW); STEPIT;                     07518000 T  0018
                        CONSTANTCLEAN; GO EXIT END                                     07519000 T  0021
                    ELSE BEGIN                                                         07520000 T  0022
                        ELBW ← ELBAT[I];                                               07521000 T  0022
                                                                                       07522000 T  0023
                        BANA;                                                          07523000 T  0023
                        EMITO(DUP);                                                    07524000 T  0024
                        EMITO(ADD);                                                    07525000 T  0024
                            EMITO(BFW);                                                07526000 T  0025
                        GT3 ← TAKE(GT4←GIT(ELBW))←GT4;                                 07527000 T  0026
                        FOR GT4 ← GT4+1 STEP 1 UNTIL GT3 DO                            07528000 T  0029
                            GOGEN(TAKE(GT4),BFW);                                      07529000 T  0033
                    END;                                                               07530000 T  0035
EXIT: END GOSTMT;                                                                      07531000 T  0035
                                                                                  59 IS   39 LONG, NEXT SEG    3
```

```
PROCEDURE GOGEN(LABELBAT,BRANCHTYPE);                                  07535000 T  0447
          VALUE LABELBAT,BRANCHTYPE;                                   07536000 T  0447
          REAL LABELBAT,BRANCHTYPE;                                    07537000 T  0447
          BEGIN                                                        07538000 T  0447
            IF BOOLEAN(GT1←TAKE(GT2←GIT(LABELBAT))).[1:1]              07539000 T  0447
              THEN EMITB(BRANCHTYPE,BUMPL,GT1,[36:12])                 07540000 T  0450
   COMMENT  LABELR SETS THE SIGN OF THE ADDITIONAL INFO FOR A LABEL     07541000 T  0453
            NEGATIVE WHEN THE LABEL IS ENCOUNTERED.  SO THIS MEANS     07542000 T  0453
            THAT WE NOW KNOW WHERE TO GO;                              07543000 T  0453
                ELSE BEGIN EMIT(GT1); EMIT(BRANCHTYPE);                07544000 T  0453
                PUT(GT1&L[36:36:12],GT2) END END GOGEN;                07545000 T  0456


COMMENT SIMPGO IS USED ONLY BY THE IF STMT ROUTINE.  IT DETERMINES IF  07546000 T  0458
        A STATEMENT IS A SIMPLE GO TO STATEMENT;                       07547000 T  0458
BOOLEAN PROCEDURE SIMPGO;                                              07548000 T  0458
        BEGIN LABEL EXIT;                                              07549000 T  0458
                                                                  START OF SEGMENT **********   60
            IF ELCLASS = GOV                                           07550000 T  0000
              THEN BEGIN                                               07551000 T  0000
                 IF STEPI = TOV THEN STEPIT;                           07552000 T  0001
                 IF ELCLASS = LABELID THEN                             07553000 T  0003
                 IF LOCAL(ELBAT[I]) THEN                               07554000 T  0004
                    BEGIN SIMPGO ← TRUE; GO EXIT END;                  07555000 T  0005
                      I ← I-1; ELCLASS ← GOV END;                      07556000 T  0007
 EXIT:  END SIMPGO;                                                    07557000 T  0009
                                                                   60 IS    13 LONG, NEXT SEG     3

                COMMENT IFSTMT COMPILES IF STATEMENTS.  SPECIAL CARE IS TAKEN TO   07558000 T  0458
                        OPTIMIZE CODE IN THE NEIGHBORHOOD OF THE JUMPS.  TO SOME   07559000 T  0458
                        EXTENT SUPPERFULOUS BRANCHING IS AVOIDED;                  07560000 T  0458
                PROCEDURE IFSTMT;                                                  07561000 T  0458
                        BEGIN REAL T1,T2; LABEL EXIT;                              07562000 T  0458
                                                                          START OF SEGMENT **********   61
STACK(F+2) = T1
STACK(F+3) = T2
                        IFCLAUSE;                                                  07563000 T  0000
                        IF SIMPGO                                                  07564000 T  0000
                          THEN BEGIN                                               07565000 T  0000
                             T1 ← ELBAT[I];                                        07566000 T  0001
                        IF STEPI = ELSEV                                           07567000 T  0002
                              THEN BEGIN                                           07568000 T  0003
                                 STEPIT;                                           07569000 T  0004
                                 IF SIMPGO                                         07570000 T  0004
                                   THEN BEGIN                                      07571000 T  0004
                                     GOGEN(ELBAT[I],BFC); GOGEN(T1,BFW);           07572000 T  0005
                        STEPIT; GO TO EXIT END ELSE BEGIN EMITLNG;GOGEN(T1,BFC);   07573000 T  0007
                             STMT ; GO TO EXIT  END END ;                          07574000 T  0010
                             EMITLNG; GOGEN(T1,BFC);                               07575000 T  0011
                             GO EXIT END;                                          07576000 T  0013
                        T1 ← BUMPL; STMT;                                          07577000 T  0013
                        IF ELCLASS ≠ ELSEV THEN                                    07578000 T  0016
```

```
                    BEGIN IF L-T1>1023 THEN ADJUST; EMITB(BFC,T1,L);                07579000 T  0016
                                GO EXIT END;                                        07579100 T  0020
                  STEPIT;                                                           07580000 T  0021
                  IF SIMPGO                                                         07581000 T  0021
                    THEN BEGIN                                                      07582000 T  0021
                       T2 ← L; L ←T1-2;GOGEN(ELBAT[I],BFC); L ← T2;                 07583000 T  0022
                       STEPIT; GO EXIT END;                                         07584000 T  0026
                  T2 ← BUMPL; CONSTANTCLEAN;                                        07585000 T  0027
                  IF L-T1>1023 THEN ADJUST; EMITB(BFC,T1,L); STMT;                  07585100 T  0030
                  IF L-T2>1023 THEN ADJUST; EMITB(BFW,T2,L);                        07586000 T  0034
EXIT: END   IFSTMT;                                                                 07587000 T  0037
                                                                          61 IS   41 LONG, NEXT SEG    3


              COMMENT  LABELR HANDLES LABELED STATEMENTS.  IT PUTS L INTO THE       07588000 T  0458
                       ADDITIONAL INFO AND MAKES ITS SIGN NEGATIVE.  IT COMPILES    07589000 T  0458
                       AT THE SAME TIME ALL THE PREVIOUS FORWARD REFERENCES SET     07590000 T  0458
                       UP FOR IT BY GOGEN.  (THE ADDITIONAL INFO LINKS TO A LIST    07591000 T  0458
                       IN THE CODE ARRAY OF ALL FORWARD REFERENCES);                07592000 T  0458
              PROCEDURE LABELR;                                                     07593000 T  0458
PRT(602) = LABELR
                    BEGIN LABEL EXIT, ROUND;                                        07594000 T  0458
                                                                        START OF SEGMENT **********  62
              DEFINE ELBATWORD=RR9#,LINK=GT2#,INDEX=GT3#,ADDITIONAL                 07595000 T  0000
                          =GT4#,NEXTLINK=GT5#;                                      07596000 T  0000
                     REAL OLDL;                                                     07596500 T  0000
STACK(F+2) = OLDL
                   DO BEGIN OLDL ← L;                                               07597000 T  0000
                        IF STEPI ≠ COLON THEN                                       07597500 T  0000
                          BEGIN ERR(133); GO TO EXIT END;                           07598000 T  0001
                        IF NOT LOCAL(ELBATWORD ← ELBAT[I-1])                        07599000 T  0003
                          THEN BEGIN FLAG(134); GO TO ROUND END;                    07600000 T  0004
                        IF STEPI = COLON THEN                                       07600100 T  0007
                          BEGIN I ← I-1; ADJUST END ELSE                            07600200 T  0008
                        IF ELCLASS = LITNO THEN L ← 4×C ELSE                        07600300 T  0010
                        IF ELCLASS=ASTRISK THEN                                     07600400 T  0013
                          BEGIN IF MODE ≠ 0 OR ASTOG THEN                           07600410 T  0015
                                FLAG(505);                                          07600420 T  0016
                                ASTOG ← TRUE;                                       07600430 T  0018
                                L ← 4×PRTI;                                         07600440 T  0018
                          END ELSE                                                  07600450 T  0020
                        I ← I-2;                                                    07600500 T  0020
                        IF STEPI ≠ COLON THEN                                       07600600 T  0021
                          BEGIN ERR(133); GO TO EXIT END;                           07600700 T  0022
                        IF L < OLDL THEN                                            07600800 T  0024
                          BEGIN FLAG(504); GO TO ROUND END;                         07600900 T  0025
                   GT1 ← TABLE(I+1);                                                07600950 T  0027
                        LINK ← (ADDITIONAL ← TAKE(INDEX ← GIT(ELBATWORD)))          07601000 T  0028
                             .[36:12];                                              07602000 T  0030
                        IF ADDITIONAL < 0 THEN                                      07603000 T  0032
                           BEGIN FLAG(135); GO TO ROUND END;                        07604000 T  0032
                        FOULED ← L;                                                 07604010 T  0034
                        IF TABLE(I+1) = COLON THEN                                  07604020 T  0035
                          BEGIN                                                     07604030 T  0037
                                IF LINK≠0 THEN BEGIN OLDL ← L;                      07604040 T  0037
```

```
                        DO BEGIN  NEXTLINK ← GET(LINK);                    07604050 T  0039
                                  L ← LINK;                                07604060 T  0041
                                  IF OLDL.[36:10]=L.[36:10]≥128            07604067 T  0042
                                  THEN FLAG(50) ELSE                       07604068 T  0043
                                 EMIT(OLDL=LINK&0[46:46:2]+                07604070 T  0045
                                       0&NEXTLINK[46:46:2]+3072);           07604080 T  0048
                                  L ← L=1;                                 07604085 T  0050
                             END UNTIL LINK←LINK=NEXTLINK DIV 4=L;          07604090 T  0051
                        L ← OLDL; END; STEPIT;                            07604100 T  0054
                        DO IF STEPI ≤ STRNGCON AND ELCLASS ≥               07604110 T  0055
                             NONLITNO THEN EMITWORD(C)                    07604120 T  0057
                             ELSE BEGIN ERR(500); I ← I=1 END             07604130 T  0058
                        UNTIL STEPI ≠ COMMA;                              07604140 T  0063
                        I ← I=1 ;                                         07604150 T  0064
                    END ELSE                                              07604160 T  0065
                WHILE LINK ≠ 0                                            07605000 T  0065
                  DO BEGIN                                                07606000 T  0067
                     NEXTLINK ← GET(LINK=2);                              07607000 T  0068
                     IF L=LINK>1023 THEN ADJUST;                          07607100 T  0070
                    EMITB(GET(LINK=1),LINK,L);                            07608000 T  0072
                     LINK ← NEXTLINK END;                                 07609000 T  0074
                PUT(=ADDITIONAL&L[36:36:12],INDEX);                       07610000 T  0075
ROUND:      ERRORTOG ← TRUE END UNTIL STEPI ≠ LABELID;                    07645000 T  0078
  EXIT: END LABELR;                                                       07646000 T  0080
                                                          62 IS   84 LONG, NEXT SEG    3

            PROCEDURE FILLSTMT(SIZE); VALUE SIZE; INTEGER SIZE;           07647000 T  0458
PRT(603) = FILLSTMT
                BEGIN                                                     07647500 T  0458
            COMMENT  "COCT" PERFORMS THE OCTAL CONVERT FOR THE FILL STATEMENT,   07648000 T  0458
                     IF THERE ARE ANY NON=OCTAL DIGITS, THIS PROCEDURE RETURNS   07648500 T  0458
                     A ZERO AND THEN THE 3 LOW=ORDER BITS OF THE BAD DIGIT ARE   07649000 T  0458
                     RESET AND IGNORED AND ERROR NUMBER 303 IS PRINTED. "COCT"   07649500 T  0458
                     ALLOWS FLAG BITS TO BE SET, WHEREAS "OCTIZE" DOES NOT.      07650000 T  0458
                       N       NUMBER OF CHARACTERS TO BE CONVERTED.             07650500 T  0458
                       SKBIT   NUMBER OF BITS TO SKIP BEFORE STARTING CONVERSION.  07651000 T  0458
                               THIS IS BECAUSE THE NO. OF CHARS. MAY BE LESS THAN  07651500 T  0458
                               8 AND IT MUST BE RIGHT=JUSTIFIED IN CD(CODEFILE).   07652000 T  0458
                       ACC     ADDRESS OF THE ACCUM WHERE ALPHA INFO IS KEPT.     07652500 T  0458
                ;                                                         07653000 T  0458
                   REAL STREAM PROCEDURE COCT(N,SKBIT,ACC,CD);VALUE N,SKBIT;  07653500 T  0458
                                                          START OF SEGMENT **********   63
PRT(604) = COCT
                   BEGIN                                                  07654000 T  0000
                   SI:=ACC; SI:=SI+6;  DI:=CD;  DS:=8 LIT"00000000";      07654500 T  0000
                   DI:=CD ; SKIP SKBIT DB;TALLY:=1;                       07655000 T  0002
                   N(IF SC>"7"THEN TALLY:=0; SKIP 3 SB;                   07655500 T  0003
                     3(IF SB   THEN DS:=1 SET ELSE SKIP 1 DB; SKIP 1 SB));  07656000 T  0005
                   COCT:=TALLY                                            07656500 T  0007
                   END COCT;                                              07657000 T  0007



                   REAL T2;                                               07657500 T  0008
```

```
STACK(F+2) = T2
                    LABEL L1;                                                                07658000 T  0008
                    STREAM PROCEDURE ZEERO(D);                                               07658500 T  0008
PRT(605) = ZEERO
                         BEGIN                                                               07659000 T  0008
                         DI:=D;DS:=8 LIT"00000000";                                          07659500 T  0009
                         SI:=D;31(32(DS:=WDS));DS:=30 WDS;                                   07660000 T  0010
                         END ZEERO;                                                          07660500 T  0012



                         STREAMTOG:=BOOLEAN(2);                                              07661000 T  0012
                         SEGMENTSTART(TRUE);                                                 07661500 T  0013
                         IF STEPI≠ASSIGNOP THEN ZEERO(CODE(1))                               07662000 T  0014
                    ELSE      BEGIN                                                          07662500 T  0019
                              FOR T2:=1 STEP 1 UNTIL SIZE DO                                 07663000 T  0020
                                   BEGIN                                                     07663500 T  0021
                                   IF STEPI>IDMAX THEN                                       07664000 T  0021
                                        BEGIN                                                07664500 T  0022
                                        IF ELCLASS≠LITNO AND ELCLASS≠NONLITNO THEN           07665000 T  0022
                                          IF ELCLASS≠STRNGCON THEN                           07665500 T  0024
                                            IF ELCLASS=ADOP AND                              07666000 T  0025
                                              (STEPI=NONLITNO OR ELCLASS=LITNO) THEN         07666500 T  0026
                                                C:=C & ELBAT[I-1][1:21:1]                    07667000 T  0029
                                        ELSE       BEGIN  ERROR(302); GO TO L1  END;         07667500 T  0031
                                        IF ELCLASS=STRNGCON AND COUNT=8 THEN                 07668000 T  0033
                                          MOVECHARACTERS(8,ACCUM[1],3,CODE(T2),0)            07668500 T  0035
                                   ELSE MOVE(1,C,CODE(T2))                                   07669000 T  0040
                                        END                                                  07669500 T  0044
                              ELSE IF COUNT≤19 AND ACCUM[1].[18:18]="OCT" THEN               07670000 T  0045
                                        BEGIN                                                07670500 T  0048
                                        IF COCT(COUNT-3,48-(COUNT-3)×3,ACCUM[1],             07671000 T  0048
                                          CODE(T2))=0 THEN FLAG(303)                         07671500 T  0052
                                        END                                                  07672000 T  0057
                                   ELSE BEGIN  ERROR(302); GO TO L1  END ;                   07672500 T  0057
                                   IF STEPI≠COMMA THEN GO TO L1                              07673000 T  0060
                                   END;                                                      07673500 T  0061
                              ERROR(54);                                                     07674000 T  0064
                              END;                                                           07674500 T  0064
L1:                                                                                          07675000 T  0064
                    RIGHT(SIZE×4);                                                           07675500 T  0065
                    STREAMTOG:=FALSE;                                                        07676000 T  0066
                    SEGMENT(SIZE,0);                                                         07676500 T  0067
                    PROGDESCBLDR(ADDRSF,TRUE,SIZE,DDES);                                     07677000 T  0068
                    END FILLSTMT;                                                            07677500 T  0069
                                                                                       63 IS   72 LONG, NEXT SEG     3



PROCEDURE STMT;                                                                              07711000 T  0458
        BEGIN LABEL                                                                          07712000 T  0458
                                                                                   START OF SEGMENT **********   64
              L1,  L2,  L3,  L4,  L5,  L6,  L7,  L8,  L9,  L10,                              07713000 T  0000
              L11, L12, L13, L14, L15, L16, L17, L18, L19, L20,                              07714000 T  0000
              L21, L22, L23, L24, L25, L26, L27, L28, L29, L30,                              07715000 T  0000
```

```
                L31, L32, L33, L34, L35, L36, L37, L38, L39, L40,                                07716000 T  0000
                L41, L42, L43, L44, L45, L46, L47, L48, L49, L50,                                07717000 T  0000
                L51, L52, L53, L54;                                                              07718000 T  0000
                SWITCH S ← L1, L2, L3, L4, L5, L6, L7, L8, L9, L10,                              07719000 T  0000
                L11, L12, L13, L14, L15, L16, L17, L18, L19, L20,                                07720000 T  0002
                L21, L22, L23, L24, L25, L26, L27, L28, L29, L30,                                07721000 T  0002
                L31, L32, L33, L34, L35, L36, L37, L38, L39, L40,                                07722000 T  0002
                L41, L42, L43, L44, L45, L46, L47, L48, L49, L50,                                07723000 T  0002
                L51, L52, L53, L54;                                                              07724000 T  0002
                LABEL AGAIN,EXIT;                                                                07725000 T  0002
                STACKCT ← 0;                                                                     07726000 T  0030
                                                                                                 07726990 T  0030
AGAIN:          GO TO S[ELCLASS];                                                                07727000 T  0031
                IF ELCLASS = COLON THEN                                                          07727010 T  0034
                     BEGIN STEPIT; GT1 ← L;                                                      07727020 T  0034
                          IF ELCLASS = COLON THEN                                                07727030 T  0036
                               BEGIN ADJUST; I ← I-1 END                                         07727040 T  0037
                          ELSE IF ELCLASS = LITNO THEN L ← 4×C                                   07727050 T  0039
                          ELSE I ← I-1;                                                          07727060 T  0041
                          IF L < GT1 OR STEPI ≠ COLON THEN                                       07727070 T  0044
                               BEGIN ERR(504); GO TO EXIT END;                                   07727080 T  0046
                          STEPIT;                                                                07727090 T  0048
                          GO TO AGAIN;                                                           07727100 T  0048
                     END;                                                                        07727110 T  0049
                IF ELCLASS = 0 THEN FLAG(100); FLAG(145);                                        07728000 T  0049
L1:L2:L3:L4:L5:L6:L9:L11:L13:L14:L15:L16:L17:L20:L21:L25:L28:L29:L24:                            07729000 T  0051
L33:L34:L35:L36:L37:L39:                                                                         07730000 T  0052
                ERR(144); GO TO EXIT;                                                            07732000 T  0052
   L7:L8:                                                                                        07732000 T  0053
                SUBHAND(TRUE); GO TO EXIT;                                                       07733000 T  0054
L10:L18:L19:                                                                                     07734000 T  0055
                PROCSTMT(TRUE); GO TO EXIT;                                                      07735000 T  0056
L12:                                                                                             07736000 T  0057
                STRMPROCSTMT; GO TO EXIT;                                                        07737000 T  0058
L22:L23:L26:L27:L30:L31:                                                                         07738000 T  0059
                VARIABLE(FS); GO TO EXIT;                                                        07739000 T  0059
L32:                                                                                             07740000 T  0060
                LABELR; GO TO AGAIN;                                                             07741000 T  0061
L38:                                                                                             07742000 T  0062
                POLISHER(0); GO TO EXIT;                                                         07743000 T  0062
L40:                                                                                             07744000 T  0063
                IF ELBAT[I].ADDRESS = STREAMV THEN                                               07745000 T  0064
                     BEGIN INLINE; GO TO EXIT END;                                               07746000 T  0065
                FLAG(146);                                                                       07747000 T  0067
                IF TABLE(I-2) = ENDV AND MODE > 0 THEN                                           07748000 T  0067
                     BEGIN I ← I-2; ELCLASS ← ENDV; GO TO EXIT END;                              07749000 T  0070
                I ← I-1; ERRORTOG ← TRUE; BLOCK(FALSE);                                          07750000 T  0073
                ELCLASS ← TABLE(I+I-1); GO TO EXIT;                                              07751000 T  0076
L42:                                                                                             07752000 T  0079
                DBLSTMT; GO TO EXIT;                                                             07753000 T  0079
L43:                                                                                             07754000 T  0080
                FORSTMT; GO TO EXIT;                                                             07755000 T  0080
L44:                                                                                             07756000 T  0081
                WHILESTMT; GO TO EXIT;                                                           07757000 T  0081
L45:                                                                                             07758000 T  0082
                DOSTMT; GO TO EXIT;                                                              07759000 T  0082
L51:                                                                                             07760000 T  0083
```

```
            IFSTMT; GO TO EXIT;                                                    07761000 T  0083
   L52:                                                                            07762000 T  0084
            GOSTMT; GO TO EXIT;                                                    07763000 T  0084
   L53:                                                                            07764000 T  0085
            IOSTMT; GO TO EXIT;                                                    07765000 T  0085
   L54:                                                                            07766000 T  0086
            IF STEPI = DECLARATORS THEN                                            07767000 T  0086
         BEGIN                                                                     07768000 T  0087
              IF ELBAT[I].ADDRESS = STREAMV THEN IF STEPI =          % 6           07768100 T  0087
                LEFTPAREN THEN                                      % 6           07768110 T  0090
         BEGIN                                                      % 6           07768120 T  0090
              ELCLASS←TABLE(I←I-1) ;                                             07768130 T  0091
              COMPOUNDTAIL ;                                                     07768140 T  0093
              GO TO EXIT ;                                                       07768160 T  0093
         END  ELSE I ← I - 1;                                       % 6           07768170 T  0094
              I ← I - 1;                                            % 6           07768180 T  0096
              BLOCK(FALSE);     END ELSE COMPOUNDTAIL;                           07768200 T  0097
L46:L47:L48:L50:                                                                   07769000 T  0099
L49:L41:                                                                           07770000 T  0099
EXIT: END STMT;                                                                    07771000 T  0099
                                                                        64 IS  100 LONG, NEXT SEG     3
```

```
                                                                                   07991000 T  0458
PROCEDURE IOSTMT;                                                                  07993000 T  0458
        IF STEPI ≠ LITNO OR (GT1←ELBAT[I].ADDRESS)>15 THEN ERR(98)ELSE             07994000 T  0458
        BEGIN EMIT(ELBAT[I-1].ADDRESS&GT1[41:47:1]&GT1[36:44:3]);                  07995000 T  0463
            STEPIT                                                                 07996000 T  0468
        END SCOPE STATEMENT;                                                       07997000 T  0468
```

```
                                                                                   08008000 T  0468
                PROCEDURE FORSTMT;                                                 08009000 T  0468
                        BEGIN                                                      08010000 T  0468
                         OWN REAL B,STMTSTART,REGO,RETURNSTORE,ADDRES,V,VRET,                  
                                                                        START OF SEGMENT **********    65
PRT(606) = B
PRT(607) = STMTSTART
PRT(610) = REGO
PRT(611) = RETURNSTORE
PRT(612) = ADDRES
PRT(613) = V
PRT(614) = VRET
                                BRET;                                              08011000 T  0000
PRT(615) = BRET
                        OWN BOOLEAN SIGNA,SIGNB,SIGNC, INT,                        08012000 T  0000
PRT(616) = SIGNA
PRT(617) = SIGNB
PRT(620) = SIGNC
PRT(621) = INT
                                CONSTANA,CONSTANB,CONSTANC;                        08013000 T  0000
PRT(622) = CONSTANA
PRT(623) = CONSTANB
PRT(624) = CONSTANC
```

```
                    DEFINE SIMPLEB = SIGNC#,          FORMALV = SIGNA#,          08014000 T  0000
                           SIMPLEV = CONSTANA#, A = V#, Q = REGO#,               08015000 T  0000
                           OPDC = TRUE#, DESC = FALSE#, K = BRET#;               08016000 T  0000
                    LABEL EXIT;                                                  08017000 T  0000
               COMMENT PLUG EMITS EITHER AN OPERAND CALL ON A VARIABLE OR A CALL ON A  08018000 T  0000
                    CONSTANT DEPENDING ON THE REQUIREMENTS;                      08019000 T  0000
               PROCEDURE PLUG(C,A); VALUE C,A; REAL A; BOOLEAN C;                08020000 T  0000
PRT(625) = PLUG
                    IF C THEN EMITNUM(A) ELSE EMITV(A,ADDRESS);                  08021000 T  0000


               COMMENT SIMPLE DETERMINES IF AN ARITHMETIC EXPRESSION IS + OR - A  08022000 T  0003
                    CONSTANT OR A SIMPLE VARIABLE.  IT MAKES A THROUGH REPORT    08023000 T  0003
                    ON ITS ACTIVITY.  IT ALSO MAKES PROVISION FOR THE RESCAN     08024000 T  0003
                    OF ELBAT (THIS IS THE ACTION WITH K = SEE CODE IN THE        08025000 T  0003
                    TABLE ROUTINE FOR FURTHER DETAILS);                          08026000 T  0003
               BOOLEAN PROCEDURE SIMPLE(B,A,S); BOOLEAN B,S; REAL A;             08027000 T  0003
PRT(626) = SIMPLE
                    BEGIN                                                        08028000 T  0003
                         S + IF STEPI ≠ ADOP THEN FALSE ELSE ELBAT[I].ADDRESS    08029000 T  0003
                                   = SUB;                                        08030000 T  0006
                         IF ELCLASS = ADOP THEN STEPIT;                          08031000 T  0008
                         IF ELCLASS ≥ NONLITNO AND ELCLASS ≤ STRNGCON             08032000 T  0010
                           THEN BEGIN K + K+1; SIMPLE + TRUE;                    08033000 T  0011
                              ELBAT[I] + 0&COMMENTV[2:41:7]&K[16:37:11];         08034000 T  0014
                              INFO[0,K] + A + C; B + TRUE END                    08035000 T  0017
                         ELSE BEGIN                                              08036000 T  0021
                          B + FALSE; A + ELBAT[I];                               08037000 T  0022
                          SIMPLE + REALID ≤ ELCLASS AND ELCLASS ≤ INTID END;     08038000 T  0024
                    STEPIT END SIMPLE;                                           08039000 T  0026


               COMMENT TEST EMITS THE STEP-UNTIL ELEMENT TEST;                   08040000 T  0029
               PROCEDURE TEST;                                                   08041000 T  0029
PRT(627) = TEST
                    BEGIN                                                        08042000 T  0029
                         IF NOT CONSTANB THEN                                    08043000 T  0029
                           BEGIN EMITO(SUB); IF SIMPLEB THEN EMITV(B,ADDRESS)    08044000 T  0030
                              ELSE BEGIN                                         08045000 T  0032
                                EMITL(2+L-BRET);                                 08046000 T  0034
                                EMITB(BBW,BUMPL,B);                              08047000 T  0036
                              END;                                               08048000 T  0038
                              EMITO(MUL); EMIT(0) END;                           08049000 T  0038
                              EMITO(IF SIGNB THEN GEQ ELSE LEQ); EMIT (0); L+L-1 08050000 T  0039
                    END TEST;                                                    08051000 T  0043


               BOOLEAN PROCEDURE SIMPI(ALL); VALUE ALL; REAL ALL;                08052000 T  0044
PRT(630) = SIMPI
                    BEGIN                                                        08053000 T  0044
                         CHECKER(VRET+ALL);                                      08054000 T  0044
```

```
                        ADDRES ← ALL.ADDRESS;                                          08055000 T  0046
                        FORMALV ← ALL.[9:2] = 2;                                       08056000 T  0047
                        IF T ← ALL.CLASS > INTARRAYID OR T < BOOID OR                  08057000 T  0049
                           GT1 ← (T-BOOID) MOD 4 < 1 THEN                              08058000 T  0052
                             ERR(REAL(T ≠ 0) × 51 + 100);                              08059000 T  0054
                        INT ← GT1 = 2;                                                 08060000 T  0057
                        SIMPI ← T ≤ INTID END SIMPI;                                   08061000 T  0058



                    COMMENT  STORE EMITS THE CODE FOR THE STORE INTO THE FOR INDEX;    08062000 T  0062
                    PROCEDURE STORE(S); VALUE S; BOOLEAN S;                            08063000 T  0062
PRT(631) = STORE
                        BEGIN                                                          08064000 T  0062
                          IF FORMALV THEN BEGIN EMITO(XCH); S ← FALSE END              08065000 T  0062
                           ELSE BEGIN                                                  08066000 T  0065
                             EMITL(ADDRES);                                            08067000 T  0065
                             IF ADDRES  > 1023 THEN EMITO(PRTE) END;                   08068000 T  0066
                              T  ← (REAL(S)+1)×16;                                     08069000 T  0068
                             EMITO((IF INT THEN T+512 ELSE 4×T)+4) END STORE;          08070000 T  0070



                    COMMENT CALL EFFECTS A CALL ON THE INDEX;                          08071000 T  0074
                    PROCEDURE CALL(S); VALUE S; BOOLEAN S;                             08072000 T  0074
PRT(632) = CALL
                        BEGIN                                                          08073000 T  0074
                         IF SIMPLEV                                                    08074000 T  0074
                           THEN IF S THEN EMITV(ADDRES) ELSE EMITN(ADDRES)             08075000 T  0075
                           ELSE BEGIN                                                  08076000 T  0078
                             EMITL(2+L-VRET);                                          08077000 T  0079
                             EMITB(BBW,BUMPL,V);                                       08078000 T  0080
                             IF S THEN EMITO(LOD) END END CALL;                        08079000 T  0083



                PROCEDURE FORLIST(NUMLE); VALUE NUMLE; BOOLEAN NUMLE;                  08080000 T  0084
PRT(633) = FORLIST
                    BEGIN                                                              08081000 T  0084
                PROCEDURE FIX(STORE,BACK,FORWART,START);                               08082000 T  0084
                                                                                  START OF SEGMENT **********   66
PRT(634) = FIX
                    VALUE STORE,BACK,FORWART,START;                                    08083000 T  0000
                    REAL  STORE,BACK,FORWART,START;                                    08084000 T  0000
                    BEGIN                                                              08085000 T  0000
                         EMITB(GET(FORWART-1),FORWART,START);                          08086000 T  0000
                         IF RETURNSTORE ≠ 0                                            08087000 T  0002
                           THEN BEGIN                                                  08088000 T  0002
                              L ← STORE; EMITNUM(B-BACK);                              08089000 T  0003
                              EMITPAIR(RETURNSTORE,STO) END END FIX;                   08090000 T  0005



                         INTEGER BACKFIX, FORWARDBRANCH, FOOT, STOREFIX;               08091000 T  0006
```

```
STACK(F+2) = BACKFIX
STACK(F+3) = FORWARDBRANCH
STACK(F+4) = FOOT
STACK(F+5) = STOREFIX
                        LABEL BRNCH,EXIT;                                          08092000 T  0006
                        STOREFIX ← L; Q ← REAL(MODE=0)+3;                          08093000 T  0006
                        FOR K ← 1 STEP 1 UNTIL Q DO EMITO(NOP);                    08094000 T  0009
                        IF NUMLE                                                   08095000 T  0014
                         THEN BEGIN                                                08096000 T  0014
                              BACKFIX ← L;                                         08097000 T  0014
                              IF FORMALV THEN CALL(DESC) END                       08098000 T  0015
                         ELSE BACKFIX ← V + REAL(SIMPLEV)-1;                       08099000 T  0017
                                                                                   08100000 T  0019
                        AEXP;                                                      08101000 T  0019
                COMMENT PICK UP FIRST ARITHMETIC EXPRESSION;                       08102000 T  0019
                        IF  ELCLASS = STEPV                                        08103000 T  0019
                           THEN BEGIN                                              08104000 T  0020
                COMMENT HERE WE HAVE A STEP ELEMENT;                               08105000 T  0021
                              BACKFIX ← BUMPL;                                     08106000 T  0021
                COMMENT LEAVE ROOM FOR FORWARD JUMP;                               08107000 T  0022
                              IF FORMALV THEN CALL(DESC); CALL(OPDC);              08108000 T  0022
                COMMENT FETCH INDEX;                                               08109000 T  0025
                              IF I > 70 THEN BEGIN NXTELBT ← 1; I ← 0 END          08110000 T  0025
                                  ELSE REGO ← I;                                   08111000 T  0027
                              IF SIMPLEB ← SIMPLE(CONSTANB,B,SIGNB) AND            08112000 T  0029
                               (ELCLASS = UNTILV OR ELCLASS = WHILEV)              08113000 T  0030
                                THEN BEGIN                                         08114000 T  0031
                COMMENT WE HAVE A SIMPLE STEP FUNCTION;                            08115000 T  0033
                                  PLUG(CONSTANB ,B);                               08116000 T  0033
                                END ELSE BEGIN                                     08117000 T  0034
                COMMENT THE STEP FUNCTION IS NOT SIMPLE;  WE CONSTRUCT A           08118000 T  0034
                              SUBROUTINE;                                          08119000 T  0034
                                  I ← IF I < 4 THEN 0 ELSE REGO; STEPIT;           08120000 T  0034
                                  SIGNB ← CONSTANB ← FALSE;                        08121000 T  0038
                                  EMIT(0); B ← L;                                  08122000 T  0039
                                  AEXP; EMITO(XCH);                                08123000 T  0040
                                  BRET ← L;                                        08124000 T  0042
                                  EMITO(BFW) END;                                  08125000 T  0042
                              EMITO(REAL(SIGNB)×32+ADD);                           08126000 T  0043
                              EMITB(BFW,BACKFIX,L);                                08127000 T  0045
                              IF ELCLASS = UNTILV                                  08128000 T  0046
                                THEN BEGIN  COMMENT  STEP=UNTIL ELEMENT;           08129000 T  0046
                                   STORE(TRUE); IF FORMALV THEN CALL(OPDC);        08130000 T  0047
                                   STEPIT; AEXP; TEST END                          08131000 T  0050
                                ELSE BEGIN  COMMENT  STEP=WHILE ELEMENT;           08132000 T  0051
                                   IF ELCLASS ≠ WHILEV THEN                        08133000 T  0052
                                     BEGIN ERR(153); GO TO EXIT END;               08134000 T  0052
                                   STEPIT; STORE(FALSE); BEXP END END              08135000 T  0054
                           ELSE BEGIN                                              08136000 T  0056
                COMMENT WE DO NOT HAVE A STEP ELEMENT;                             08137000 T  0056
                             STORE(FALSE);                                         08138000 T  0056
                             IF ELCLASS = WHILEV                                   08139000 T  0057
                               THEN BEGIN                                          08140000 T  0057
                COMMENT WE HAVE A WHILE ELEMENT;                                   08141000 T  0058
                                 STEPIT; BEXP END                                  08142000 T  0058
                               ELSE BEGIN                                          08143000 T  0059
                COMMENT ONE EXPRESSION ELEMENT;                                    08144000 T  0060
```

```
                    IF ELCLASS ≠ COMMA THEN BEGIN                                  08145000 T  0060
                         EMITB(BFW,BUMPL,L+2); BACKFIX ← L END                     08146000 T  0061
                     ELSE BACKFIX ← L + 2;                                         08147000 T  0065
                    L ← L+1; EMIT(BFW); GO TO BRNCH END END;                       08148000 T  0066
   COMMENT    THIS IS THE COMMON POINT;                                            08149000 T  0069
              IF ELCLASS = COMMA THEN EMITLNG; L ← L+1;                            08150000 T  0069
              EMIT(BFC);                                                           08151000 T  0072
BRNCH:        FORWARDBRANCH ← L; DIALA ← DIALB ← 0;                                08152000 T  0073
              IF ELCLASS = COMMA                                                   08153000 T  0075
                THEN BEGIN                                                         08154000 T  0075
                   STEPIT;                                                         08155000 T  0076
                   FORLIST(TRUE);                                                  08156000 T  0076
                   FIX(STOREFIX,BACKFIX,FORWARDBRANCH,STMTSTART) END               08157000 T  0077
                ELSE BEGIN                                                         08158000 T  0079
                   IF ELCLASS ≠ DOV                                                08159000 T  0079
                        THEN BEGIN ERR(154); REGO←L; GO EXIT END;                  08160000 T  0079
                   STEPIT;                                                         08161000 T  0082
                   IF NUMLE THEN FOOT := GETSPACE(FALSE,-1); % TEMP,               08162000 T  0083
              STMT;                                                                08163000 T  0085
                                                                                   08164000 T  0086
                   IF NUMLE THEN BEGIN                                             08165000 T  0086
                     EMITV(RETURNSTORE ← FOOT); EMITO(BBW) END                     08166000 T  0087
                   ELSE BEGIN                                                      08167000 T  0089
                     EMITB(BBW,BUMPL,BACKFIX); RETURNSTORE ← 0 END;                08168000 T  0089
                   STMTSTART ← FORWARDBRANCH; B ← L;                               08169000 T  0092
                   CONSTANTCLEAN; REGO ← L;                                        08170000 T  0094
                   FIX(STOREFIX,BACKFIX,FORWARDBRANCH,L) END;                      08171000 T  0095
EXIT:         END FORLIST;                                                         08172000 T  0096
                                                                    66 IS  100 LONG, NEXT SEG   65
```

```
              REAL T1,T2,T3,T4;                                                    08173000 T  0084
STACK(F+2) = T1
STACK(F+3) = T2
STACK(F+4) = T3
STACK(F+5) = T4
                   NXTELBT ← 1; I ← 0;                                             08174000 T  0084
                   STEPIT;                                                         08175000 T  0086
                   IF SIMPI(VRET←ELBAT[I])                                         08176000 T  0087
                     THEN BEGIN                                                    08177000 T  0087
                        IF STEPI ≠ ASSIGNOP THEN BEGIN ERR(152); GO EXIT END;      08178000 T  0089
                        T1 ← L; IF FORMALV THEN EMITN(ADDRES);                     08179000 T  0091
                        K ← 0;                                                     08180000 T  0094
                        IF SIMPLE(CONSTANA,A,SIGNA) THEN                           08181000 T  0094
                        IF ELCLASS = STEPV THEN                                    08182000 T  0096
                        IF SIMPLE(CONSTANB,B,SIGNB) THEN                           08183000 T  0097
                        IF ELCLASS = UNTILV THEN                                   08184000 T  0099
                        IF SIMPLE(CONSTANC,Q,SIGNC) THEN                           08185000 T  0100
                        IF ELCLASS = DOV THEN                                      08186000 T  0102
                          BEGIN                                                    08187000 T  0103
                             PLUG(CONSTANA,A);                                     08188000 T  0103
                             IF SIGNA THEN EMITO(CHS);                             08189000 T  0104
                             RETURNSTORE  ← BUMPL; ADJUST; CONSTANTCLEAN;          08190000 T  0106
                             STMTSTART ← L;                                        08191000 T  0109
                   STEPIT;                                                         08192000 T  0109
```

```
T1 ← ((((4096 × RETURNSTORE+STMTSTART)×2+                                    08193000 T  0110
     REAL(CONSTANB))×2+                                                      08194000 T  0112
     REAL(CONSTANC))×2+                                                      08195000 T  0113
     REAL(SIGNB))×2+                                                         08196000 T  0114
     REAL(SIGNC);                                                            08197000 T  0115
T2 ← VRET;                                                                   08198000 T  0116
T3 ← B;                                                                      08199000 T  0116
T4 ← Q;                                                                      08200000 T  0117
STMT;                                                                        08201000 T  0118
SIGNC ← BOOLEAN(T1.[47:1]);                                                  08202000 T  0118
SIGNB ← BOOLEAN(T1.[46:1]);                                                  08203000 T  0120
CONSTANC ← BOOLEAN(T1.[45:1]);                                               08204000 T  0121
CONSTANB ← BOOLEAN(T1.[44:1]);                                               08205000 T  0122
STMTSTART ← T1.[32:12];                                                      08206000 T  0123
RETURNSTORE ← T1.[20:12];                                                    08207000 T  0125
VRET ← T2;                                                                   08208000 T  0126
B ← T3;                                                                      08209000 T  0127
Q ← T4;                                                                      08210000 T  0127
SIMPLEV← SIMPI(VRET);                                                        08211000 T  0128
          IF FORMALV THEN EMITN(ADDRES); EMITV(ADDRES);                      08212000 T  0129
          PLUG(CONSTANB,B);                                                  08213000 T  0132
          EMITO(IF SIGNB THEN SUB ELSE ADD);                                 08214000 T  0133
          EMITB(BFW,RETURNSTORE,L);                                          08215000 T  0135
          STORE(TRUE);                                                       08216000 T  0136
          IF FORMALV THEN CALL(OPDC);                                        08217000 T  0137
          PLUG(CONSTANC,Q);                                                  08218000 T  0138
          IF SIGNC THEN EMITO(CHS);                                          08219000 T  0139
         SIMPLEB ← TRUE; TEST; EMITLNG;                                      08220000 T  0141
         EMITB(BBC,BUMPL,STMTSTART);                                         08221000 T  0143
         GO TO EXIT END;                                                     08222000 T  0145
      I ← 2; K ← 0;                                                          08223000 T  0147
      SIMPLEV ← SIMPI(VRET);                                                 08224000 T  0148
      V ← T1 END                                                             08225000 T  0149
   ELSE BEGIN                                                                08226000 T  0150
      EMIT(0); V ← L; SIMPLEV ← FALSE; FORMALV ← TRUE;                       08227000 T  0151
      VARIABLE(FR); EMITO(XCH); VRET ← L; EMITO(BFW);                        08228000 T  0154
      IF ELCLASS≠ASSIGNOP THEN BEGIN ERR(152); GO EXIT END;                  08229000 T  0157
      END;                                                                   08230000 T  0159
   STEPIT; FORLIST(FALSE); L ← REGO;                                         08231000 T  0159
EXIT:  K ← 0 END FORSTMT;                                                    08232000 T  0161
                                                                  65 IS  165 LONG, NEXT SEG    3


          REAL PROCEDURE REED;                                               08999000 T  0468
PRT(635) = REED
               BEGIN                                                         08999025 T  0468
               LABEL EOF; INTEGER I,J,K;                                     08999050 T  0468
                                                                  START OF SEGMENT **********   67

STACK(F+3) = I
STACK(F+4) = J
STACK(F+5) = K
               STREAM PROCEDURE MOVE(N,F,T); VALUE N,T;                      08999075 T  0000
PRT(636) = MOVE
                    BEGIN  SI:=F; DI:=T; DS:=N WDS  END MOVE;                08999100 T  0000
```

```
                    J:=-1;
                    READ(CODISK[NO])[EOF];
PRT(637) = EOF
                    REED:=I:=FETCH(MKABS(CODISK(1)));
                    K:=MKABS(CODE(0))-1;
                    WHILE I-J>30 DO
                         BEGIN
                         MOVE(30,CODISK(0),K); K:=K+30; J:=J+30;
                         READ(CODISK);
                         END;
                    MOVE(I-J,CODISK(0),K);
                    READ(CODISK)[EOF];
               EOF:
               END REED;
```

```
08999125 T  0001
08999150 T  0003
08999175 T  0008
08999200 T  0014
08999225 T  0019
08999250 T  0020
08999275 T  0020
08999300 T  0027
08999325 T  0031
08999350 T  0031
08999375 T  0036
08999400 T  0041
08999425 T  0042
67 IS   47 LONG, NEXT SEG    3
```

```
               PROCEDURE RIGHT(L); VALUE L; INTEGER L;
                    BEGIN
                    INTEGER I,J;
STACK(F+2) = I
STACK(F+3) = J
                    I:=(L+7) DIV 4;
                    MOVE(1,I,CODISK(0));
                    MOVE(29,CODE(0),CODISK(1));
                    WRITE(CODISK);
                    J:=29;
                    WHILE I-J>0 DO
                         BEGIN
                         MOVE(30,CODE(J),CODISK(0));
                         WRITE(CODISK);
                         J:=J+30;
                         END;
               END RIGHT;
```

```
08999450 T  0468
08999475 T  0468
08999500 T  0468
START OF SEGMENT **********   68
08999525 T  0000
08999550 T  0001
08999575 T  0005
08999600 T  0012
08999625 T  0016
08999650 T  0017
08999675 T  0019
08999700 T  0019
08999725 T  0026
08999750 T  0030
08999775 T  0031
08999800 T  0032
68 IS   85 LONG, NEXT SEG    3
```

```
          COMMENT THE PROGRAM ROUTINE DOES THE INITIALIZATION AND THE WRAPUP
                  FOR THE REST OF THE COMPILER.  THE MAIN PROGRAM OF THE COMPILER
                  IS SIMPLY A CALL ON THE PROGRAM ROUTINE;
          PROCEDURE PROGRAM;
PRT(640) = PROGRAM
                  BEGIN
          STREAM PROCEDURE MDESC(WD,TOLOC);VALUE WD;
PRT(641) = MDESC
            BEGIN DI←LOC WD; DS←  SET;SI← LOC WD; DI←TOLOC;DS←WDS END;
```

```
09000000 T  0468
09001000 T  0468
09002000 T  0468
09003000 T  0468
09004000 T  0468
09005000 T  0468
START OF SEGMENT **********   69
09006000 T  0000
```

```
            DEFINE STARTINTRSC=426#;                                                    09024000 T  0001
            LABEL L1;                                                                   09025000 T  0001
            LISTOG:=LISTER:=BOOLEAN(1=ERRORCOUNT.[46:1]);                               09028000 T  0001
COMMENT  LISTOG IS NOT SET BY DEFAULT ON TIMESHARING;                                   09028010 T  0005
            NOHEADING := TRUE;                                                          09028050 T  0005
            ERRORCOUNT := 0;                                                            09028900 T  0006
            ERRMAX:=999; % MAY BE CHANGED IN DOLLARCARD.                                09028910 T  0006
            BASENUM:=10000; ADDVALUE:=1000; NEWBASE:=TRUE;                              09028920 T  0007
COMMENT  DEFAULT VALUES FOR "$SEQ" OPTION;                                              09028930 T  0009
            LASTUSED := 4; % FOR INITIALIZATION.                                        09029000 T  0009
            NEXTINFO ← LASTINFO ← LASTSEQROW*256+LASTSEQUENCE+1;                        09033000 T  0010
            PUTNBUMP(0);                                                                09034000 T  0013
            GT1 ← "       ";                                                            09034100 T  0014
            MDESC(GT1,INFO[LASTSEQROW,LASTSEQUENCE]);                                   09034200 T  0015
            BLANKET(0,INFO[LASTSEQROW,LASTSEQUENCE]); % FOR "$ CHECK".                  09034500 T  0017
            READACARD; % INITIALIZATION OF NCR,FCR, AND LCR, AND                        09035000 T  0019
                       % READS FIRST CARD INTO CARD BUFFER.                            09036000 T  0019
            LASTUSED := 1; % ASSUMES CARD ONLY UNTIL TOLD DIFFERENTLY.                  09037000 T  0019
            NXTELBT ← 1;                                                                09038000 T  0020
            PRTI←PRTIMAX←PRTBASE;                                                       09039000 T  0021
            MRCLEAN ← TRUE;                                                             09040000 T  0022
COMMENT  START FILLING TABLES NEEDED TO COMPILE A PROGRAM;                              09040100 T  0023
            FILL TEN[*] WITH                                                            09041000 T  0023
   OCT1771110463422054,     OCT1761332600326467,     OCT1751621340414205,              09042000 T  0023
                                                                          START OF SEGMENT **********   70
   OCT1742165630517247,     OCT1732623176643120,     OCT1723370036413744,              09043000 T  0024
   OCT1714266046116735,     OCT1705343457542525,     OCT1676634373473252,              09044000 T  0024
   OCT1651040347241213,     OCT1641250441111455,     OCT1631522551333770,              09045000 T  0024
   OCT1622047303622767,     OCT1612461164567564,     OCT1603175421725521,              09046000 T  0024
   OCT1574034726313046,     OCT1565044113775657,     OCT1556255136775233,              09047000 T  0024
   OCT1547730366574502,     OCT1521171646433362,     OCT1511430220142257,              09048000 T  0024
   OCT1501736264172732,     OCT1472325741231521,     OCT1463013331500045,              09049000 T  0024
   OCT1453616220020057,     OCT1444561664024072,     OCT1435716241031111,              09050000 T  0024
   OCT1427301711237333,     OCT1401116227350722,     OCT1371341675243107,              09051000 T  0024
   OCT1361632254513731,     OCT1352200727636717,     OCT1342641115606502,              09052000 T  0024
   OCT1333411341150223,     OCT1324313631402270,     OCT1315376577702746,              09053000 T  0024
   OCT1306676337663537,     OCT1261045602764047,     OCT1251257143561061,              09054000 T  0024
   OCT1241532774515275,     OCT1232061573640554,     OCT1222476132610706,              09055000 T  0024
   OCT1213215561353071,     OCT1204061115645707,     OCT1175075341217270,              09056000 T  0024
   OCT1166314631463146,     OCT1141000000000000,     OCT1131200000000000,              09057000 T  0024
   OCT1121440000000000,     OCT1111750000000000,     OCT1102342000000000,              09058000 T  0024
   OCT1073032400000000,     OCT1063641100000000,     OCT1054611320000000,              09059000 T  0024
   OCT1045753604000000,     OCT1037346545000000,     OCT1011124027620000,              09060000 T  0024
    OCT0001351035564000,     OCT0011643245121000,     OCT0022214116345200,             09061000 T  0024
    OCT0032657142036440,     OCT0043432772446150,     OCT0054341571157602,             09062000 T  0024
    OCT0065432127413543,     OCT0076740555316473,     OCT0111053071060221,             09063000 T  0024
   OCT0121265707274266,     OCT0131543271153343,     OCT0142074147406234,              09064000 T  0024
   OCT0152513201307703,     OCT0163236041571663,     OCT0174105452130240,              09065000 T  0024
   OCT0205126764556310,     OCT0216354561711772,     OCT0231004771627437,              09066000 T  0024
   OCT0241206170175347,     OCT0251447626234641,     OCT0261761573704011,              09067000 T  0024
   OCT0272356132665013,     OCT0303051561442216,     OCT0313664115752661,              09068000 T  0024
   OCT0324641141345435,     OCT0336011371636745,     OCT0347413670206536,              09069000 T  0024
   OCT0361131664625027,     OCT0371360241772234,     OCT0401654312370703,              09070000 T  0024
   OCT0412227375067064,     OCT0422675274304701,     OCT0433454553366062,              09071000 T  0024
   OCT0444367706263476,     OCT0455465667740415,     OCT0467003245730521,              09072000 T  0024
   OCT0501060411731665,     OCT0511274514320242,     OCT0521553637404312,              09073000 T  0024
   OCT0532106607305375,     OCT0542530351166674,     OCT0553256443424453,              09074000 T  0024
```

```
    OCT0564132154331566,     OCT0575160607420123,     OCT0606414751324150,          09075000 T  0024
    OCT0621012014361120,     OCT0631214417455344,     OCT0641457523370635,          09076000 T  0024
    OCT0651773450267005,     OCT0662372362344606,     OCT0673071057035747,          09077000 T  0024
    OCT0703707272645341,     OCT0714671151416632,     OCT0726047403722400,          09078000 T  0024
    OCT0737461304707100,     OCT0751137556607072,     OCT0761367512350710,          09079000 T  0024
    OCT0771665435043072;                                                           09080000 T  0024
                                                                            70 IS  115 LONG, NEXT SEG   69
COMMENT  THIS IS THE FILL FOR THE SECOND ROW OF INFO;                              09081000 T  0024
         THE FIRST ITEMS ARE STREAM RESERVED WORDS,                                09082000 T  0024
         THEN ORDINARY RESERVED WORDS,                                             09083000 T  0024
         THEN INTRINSIC FUNCTIONS;                                                 09084000 T  0024
 FILL INFO[1,*] WITH                                                               09085000 T  0024
    OCT0670000600000002, "2SI000",                          %256                   09086000 T  0025
                                                                            START OF SEGMENT ********** 71
    OCT0700001040000002, "2DI000",                          %258                   09087000 T  0026
    OCT0710001460000002, "2CI000",                          %260                   09088000 T  0026
    OCT0720001630000002, "5TALLY",                          %262                   09089000 T  0026
    OCT0730000530000002, "2DS000",                          %264                   09090000 T  0026
    OCT0740000150000002, "4SKIP0",                          %266                   09091000 T  0026
    OCT0750001620000002, "4JUMP0",                          %268                   09092000 T  0026
    OCT0760000740000002, "2DB000",                          %270                   09093000 T  0026
    OCT0770000500000002, "2SB000",                          %272                   09094000 T  0026
    OCT1010000730000002, "2SC000",                          %274                   09095000 T  0026
    OCT1020001160000002, "3LOC00",                          %276                   09096000 T  0026
    OCT1030001170000002, "2DC000",                          %278                   09097000 T  0026
    OCT1040001430000002, "5LOCAL",                          %280                   09098000 T  0026
    OCT1050000340000002, "3LIT00",                          %282                   09099000 T  0026
    OCT1060001036400002, "3SET00",                          %284                   09100000 T  0026
    OCT1060001066500002, "5RESET",                          %286                   09101000 T  0026
    OCT1060001020500002, "3WDS00",                          %288                   09102000 T  0026
    OCT1060001357700002, "3CHR00",                          %290                   09103000 T  0026
    OCT1060001057300002, "3ADD00",                          %292                   09104000 T  0026
    OCT1060001617200002, "3SUB00",                          %294                   09105000 T  0026
    OCT1060000727600002, "3ZON00",                          %296                   09106000 T  0026
    OCT1060000417500002, "3NUM00",                          %298                   09107000 T  0026
    OCT1060000766700002, "3OCT00",                          %300                   09108000 T  0026
    OCT1060000176600002, "3DEC00",                          %302                   09109000 T  0026
    OCT1004000260000003, "6TOGGL", "E0000000",              %304                   09110000 T  0026
    OCT0130311060000002, "3ABS00",                          %307                   09110001 T  0026
    OCT1360441030000002, "3AND00",                          %309                   09112000 T  0026
    OCT0500000170000002, "5ARRAY",                          %311                   09112100 T  0026
    OCT0660000000000002, "5BEGIN",                          %313                   09112200 T  0026
    OCT0500000040000003, "7BOOLE", "AN000000",              %315                   09112300 T  0026
    OCT1070000000000003, "7COMME", "NT000000",              %318                   09112400 T  0026
    OCT0500000230000003, "6DEFIN", "E0000000",              %321                   09112500 T  0026
    OCT1410446000000002, "3DIV00",                          %324                   09112600 T  0026
    OCT0550000000000002, "2D0000",                          %326                   09112700 T  0026
    OCT0520000000000003, "6DOUBL", "E0000000",              %328                   09112800 T  0026
    OCT0570000000000002, "4ELSE0",                          %331                   09112900 T  0026
    OCT0600000000000002, "3END00",                          %333                   09113000 T  0026
    OCT1340442030000002, "3EQV00",                          %335                   09113100 T  0026
    OCT0410000000000002, "5FALSE",                          %337                   09113200 T  0026
    OCT0130310030000002, "4FLAG0",                          %339                   09113300 T  0026
    OCT0530000000000002, "3FOR00",                          %341                   09113400 T  0026
    OCT1100000000000003, "7FORWA", "RD      ",              %343                   09113500 T  0026
    OCT0640000000000002, "2G0000",                          %346                   09113600 T  0026
    OCT0130316060320002, "4HUNT0",                          %348                   09113700 T  0026
```

```
OCT0630000000000002, "2IF000",                                  %350    09113800 T  0026
OCT0500000040000002, "4REAL0",                                  %352    09113900 T  0026
OCT0500000050000003, "7INTEG", "ER000000",                      %354    09114000 T  0026
OCT0500000070000002, "5LABEL",                                  %357    09114100 T  0026
OCT0360002000020003, "6MEMOR", "Y       ",                      %359    09114200 T  0026
OCT1410456000000002, "3MOD00",                                  %362    09114300 T  0026
OCT0500000140000003, "7MONIT", "OR      ",                      %364    09114400 T  0026
OCT0130301060000002, "4NABS0",                                  %367    09114500 T  0026
OCT0500000200000002, "4NAME0",                                  %369    09114600 T  0026
OCT0130304030000002, "5NFLAG",                                  %371    09114700 T  0026
OCT1320300230000002, "3NOT00",                                  %373    09114800 T  0026
OCT1350440430000002, "2OR000",                                  %375    09114900 T  0026
OCT0500000020000002, "4SAVE0",                                  %377    09115000 T  0026
OCT0500000010000002, "3OWN00",                                  %379    09115100 T  0026
OCT0460000000000003, "6POLIS", "H       ",                      %381    09115200 T  0026
OCT0500000160000003, "9PROCE", "DURE    ",                      %384    09115300 T  0026
OCT0130300000160011, "4SIGNO",                                  %387    09115400 T  0026
          OCT2025,   COMMENT  DUP   ;                                   09115500 T  0026
          OCT0000,   COMMENT  LITC 0;                                   09115600 T  0026
          OCT0425,   COMMENT  NEQ   ;                                   09115700 T  0026
          OCT1025,   COMMENT  XCH   ;                                   09115800 T  0026
          OCT0155,   COMMENT  DIA  1;                                   09115900 T  0026
          OCT0161,   COMMENT  DIB  1;                                   09116000 T  0026
          OCT0165,   COMMENT  TRB  1;                                   09116100 T  0026
OCT1110000000000002, "4STEP0",                                  %396    09116200 T  0026
OCT0500000220000003, "6STREA", "M       ",                      %398    09116300 T  0026
OCT0500000110000003, "#SUBRO", "UTINE   ",                      %401    09116400 T  0026
OCT0500000150000003, "6SWITC", "H       ",                      %404    09116500 T  0026
OCT1120000000000002, "4THEN0",                                  %407    09116600 T  0026
OCT1130000000000002, "2T0000",                                  %409    09116700 T  0026
OCT0410000010000002, "4TRUE0",                                  %411    09116800 T  0026
OCT0560000000000002, "5UNTIL",                                  %413    09116900 T  0026
OCT1140000000000002, "5VALUE",                                  %415    09117000 T  0026
OCT0540000000000002, "5WHILE",                                  %417    09117100 T  0026
OCT1310440200000002, "3ADD00",                                  %419    09117200 T  0026
OCT1310240270000002, "3BRT00",                                  %421    09117300 T  0026
OCT1310453050000002, "3CCX00",                                  %423    09117400 T  0026
OCT1310442500000002, "3CDC00",                                  %425    09117500 T  0026
OCT1310457050000002, "3CFX00",                                  %427    09117600 T  0026
OCT1310302060000002, "3CHS00",                                  %429    09117700 T  0026
OCT1310440500000002, "3CDC00",                                  %431    09117800 T  0026
OCT1310242020000002, "3COM00",                                  %433    09117900 T  0026
OCT1310302060000002, "3CSB00",                                  %435    09118000 T  0026
OCT1310240120000002, "3DEL00",                                  %437    09118100 T  0026
OCT1260100550000002, "3DIA00",                                  %439    09118200 T  0026
OCT1260100610000002, "3DIB00",                                  %441    09118300 T  0026
OCT1310344050000002, "3DUP00",                                  %443    09118400 T  0026
OCT1310451050000002, "3EQL00",                                  %445    09118500 T  0026
OCT1310443050000002, "3FCX00",                                  %447    09118600 T  0026
OCT1310447050000002, "3FFX00",                                  %449    09118700 T  0026
OCT1310440250000002, "3GEQ00",                                  %451    09118800 T  0026
OCT1310440450000002, "3GTR00",                                  %453    09118900 T  0026
OCT1310104420000002, "3HLB00",                                  %455    09119000 T  0026
OCT1310104420000002, "3HP200",                                  %457    09119050 T  0026
OCT1310446000000002, "3IDV00",                                  %459    09119100 T  0026
OCT1310251020000002, "3IID00",                                  %461    09119200 T  0026
OCT1310250220000002, "3INA00",                                  %463    09119300 T  0026
```

```
    OCT1310250420000002, "3INB00",                                  %465                09119400 T  0026
    OCT1310100420000002, "3INI00",                                  %467                09119500 T  0026
    OCT1400440300000002, "3INX00",                                  %469                09119600 T  0026
    OCT1310244220000002, "3IOR00",                                  %471                09119700 T  0026
    OCT1310250220000002, "3IP100",                                  %473                09119800 T  0026
    OCT1310250420000002, "3IP200",                                  %475                09119900 T  0026
    OCT1310145060000002, "3IPS00",                                  %477                09120000 T  0026
    OCT1310410240000002, "3ISD00",                                  %479                09120100 T  0026
    OCT1310450440000002, "3ISN00",                                  %481                09120200 T  0026
    OCT1310100420000002, "3ITI00",                                  %483                09120300 T  0026
    OCT1310450250000002, "3LEQ00",                                  %485                09120400 T  0026
    OCT1310505300000002, "3LLL00",                                  %487                09120500 T  0026
    OCT1310441030000002, "3LND00",                                  %489                09120600 T  0026
    OCT1310300230000002, "3LNG00",                                  %491                09120700 T  0026
    OCT1310304040000002, "3LOD00",                                  %493                09120800 T  0026
    OCT1310440430000002, "3LOR00",                                  %495                09120900 T  0026
    OCT1310442030000002, "3LQV00",                                  %497                09121000 T  0026
    OCT1310450450000002, "3LSS00",                                  %499                09121100 T  0026
    OCT1310101100000002, "3MKS00",                                  %501                09121200 T  0026
    OCT1310441000000002, "3MUL00",                                  %503                09121300 T  0026
    OCT1310441050000002, "3NEQ00",                                  %505                09121400 T  0026
    OCT1310100130000002, "3NOP00",                                  %507                09121500 T  0026
    OCT0650006550000003, "6SCOPO", "N.......";                      %509                09121600 T  0026
                                                                                      71 IS  256 LONG, NEXT SEG    69
FILL INFO[2,*] WITH                                                                   09121650 T  0026
    OCT1310300000020004, "3RDF00",                                  %512                09121700 T  0027
                                                                                    START OF SEGMENT **********    72
            OCT0000,    COMMENT  LITC 0;                                          09121800 T  0028
            OCT2141,    COMMENT  FXS   ;                                          09121900 T  0028
    OCT1310300000020004, "3RDS00",                                  %516                09122000 T  0028
            OCT0004,    COMMENT  LITC 1;                                          09122100 T  0028
            OCT2141,    COMMENT  FXS   ;                                          09122200 T  0028
    OCT1310456000000002, "3RDV00",                                  %520                09122300 T  0028
    OCT1310304030000002, "3RFB00",                                  %522                09122400 T  0028
    OCT1310240470000002, "3RN000",                                  %524                09122500 T  0028
    OCT1310145060000002, "3RRR00",                                  %526                09122600 T  0028
    OCT1310311060000002, "3RSB00",                                  %528                09122700 T  0028
    OCT1310242470000002, "3RSP00",                                  %530                09122800 T  0028
    OCT1310141020000002, "3RTM00",                                  %532                09122900 T  0028
    OCT1310240470000002, "3RTN00",                                  %534                09123000 T  0028
    OCT1310141020000002, "3RTR00",                                  %536                09123100 T  0028
    OCT1310242470000002, "3RTS00",                                  %538                09123200 T  0028
    OCT1310310030000002, "3SFB00",                                  %540                09123300 T  0028
    OCT1310442040000002, "3SND00",                                  %542                09123400 T  0028
    OCT1310301060000002, "3SSB00",                                  %544                09123500 T  0028
    OCT1310316060000002, "3SSF00",                                  %546                09123600 T  0028
    OCT1310301060000002, "3SSN00",                                  %548                09123700 T  0028
    OCT1310311060000002, "3SSP00",                                  %550                09123800 T  0028
    OCT1310401040000002, "3STD00",                                  %552                09123900 T  0028
    OCT1310240000020004, "3STF00",                                  %554                09124000 T  0028
            OCT0010,    COMMENT  LITC 2;                                          09124100 T  0028
            OCT2141,    COMMENT  FXS   ;                                          09124200 T  0028
    OCT1310442040000002, "3STN00",                                  %558                09124300 T  0028
    OCT1310240000020004, "3STS00",                                  %560                09124400 T  0028
            OCT0014,    COMMENT LITC 3;                                           09124500 T  0028
            OCT2141,    COMMENT FXS   ;                                           09124600 T  0028
    OCT1310440600000002, "3SUB00",                                  %564                09124700 T  0028
```

```
OCT1310344060000002, "3TFB00",                          %566    09124800 T  0028
OCT1270440650000002, "3TFR00",                          %568    09124900 T  0028
OCT1310155060000002, "3TI000",                          %570    09125000 T  0028
OCT1310344060000002, "3TOP00",                          %572    09125050 T  0028
OCT1270440650000002, "3TRB00",                          %574    09125100 T  0028
OCT1300300000000002, "3VFI00",                          %576    09125200 T  0028
OCT1310502050000002, "3XCH00",                          %578    09125300 T  0028
OCT1310101070000002, "3XIT00",                          %580    09125400 T  0028
OCT1310105020000002, "3ZIP00",                          %582    09125500 T  0028
OCT1310105020000002, "3ZP100",                          %584    09125600 T  0028
OCT1270500750000002, "3CFE00",                          %586    09125700 T  0028
OCT1270500750000002, "3FCE00",                          %588    09125800 T  0028
OCT1270500710000002, "3CFL00",                          %590    09125900 T  0028
OCT1270500710000002, "3FCL00",                          %592    09126000 T  0028
OCT1310440210000002, "3DLA00",                          %594    09126100 T  0028
OCT1310440210000002, "3ADL00",                          %596    09126200 T  0028
OCT1310440610000002, "3DLS00",                          %598    09126300 T  0028
OCT1310440610000002, "3SDL00",                          %600    09126400 T  0028
OCT1310441010000002, "3DLM00",                          %602    09126500 T  0028
OCT1310441010000002, "3MDL00",                          %604    09126600 T  0028
OCT1310442010000002, "3DLD00",                          %606    09126700 T  0028
OCT1310442010000002, "3DDL00",                          %608    09126800 T  0028
OCT0460000000000002, "1P0000",                          %610    09126900 T  0028
OCT0360002000020002, "1M0000",                          %612    09127000 T  0028
OCT1310240000020004, "3PRL00",                          %614    09127100 T  0028
          OCT0111,   COMMENT  PRL;                              09127200 T  0028
          OCT0055,   COMMENT  NOP;                              09127300 T  0028
OCT0650006610000003, "7SCOPO", "FF......",              %618    09127400 T  0028
OCT0030000000040003, "2LB000", "[#      ",              %621    09127500 T  0028
OCT0030000000040003, "2RB000", "]#      ",              %624    09127600 T  0028
OCT0030000000040003, "3GTR00", ">#      ",              %627    09127700 T  0028
OCT0030000000040003, "3GEQ00", "≥#      ",              %630    09127800 T  0028
OCT0030000000040003, "3EQL00", "=#      ",              %633    09127900 T  0028
OCT0030000000040003, "3NEQ00", "≠#      ",              %636    09128000 T  0028
OCT0030000000040003, "3LEQ00", "≤#      ",              %639    09128100 T  0028
OCT0030000000040003, "3LSS00", "<#      ",              %642    09128200 T  0028
OCT0030000000040003, "5TIMES", "×#      ",              %645    09128300 T  0028
OCT1310117530000002, "3SCI00",                          %648    09128400 T  0028
OCT1310117540000002, "3SAN00",                          %650    09128500 T  0028
OCT1310157730000000, "3SCS00",                          %652    09128600 T  0028
                                                                09128700 T  0028
                                                                09128800 T  0028
                                                                09128900 T  0028
                                                                09129000 T  0028
                                                                09129100 T  0028
                                                                09129200 T  0028
                                                                09129300 T  0028
                                                                09129400 T  0028
                                                                09129500 T  0028
                                                                09129600 T  0028
                                                                09129700 T  0028
                                                                09129800 T  0028
                                                                09129900 T  0028
                                                                09130000 T  0028
                                                                09130100 T  0028
                                                                09130200 T  0028
                                                                09130300 T  0028
```

```
09130400 T  0028
09130500 T  0028
09130600 T  0028
09130700 T  0028
09130800 T  0028
09130900 T  0028
09131000 T  0028
09131100 T  0028
09131200 T  0028
09131300 T  0028
09131400 T  0028
09131500 T  0028
09131600 T  0028
09131700 T  0028
09131800 T  0028
09131900 T  0028
09132000 T  0028
09132100 T  0028
09132200 T  0028
09132300 T  0028
09132400 T  0028
09132500 T  0028
09132600 T  0028
09132700 T  0028
09132800 T  0028
09132900 T  0028
09133000 T  0028
09133100 T  0028
09133200 T  0028
09133300 T  0028
09133400 T  0028
09133450 T  0028
09133500 T  0028
09133600 T  0028
09133700 T  0028
                                                                    72 IS  143 LONG, NEXT SEG   69
09133800 T  0028
09133900 T  0034
09134000 T  0040
09134100 T  0045
09134200 T  0049
09197000 T  0053
09198000 T  0053
09199000 T  0053
                                                                START OF SEGMENT **********   73
09200000 T  0054
09201000 T  0054
09202000 T  0054
09203000 T  0054
09204000 T  0054
09205000 T  0054
09206000 T  0054
09207000 T  0054
09208000 T  0054
09209000 T  0054
09210000 T  0054
09211000 T  0054
```

```
0;   % END OF INFO FILL.

       FOR GT2←256 STEP GT1,LINK WHILE NOT BOOLEAN(GT1,FORMAL) DO
            PUT((GT1←TAKE(GT2))&GT2[35:35:13],GT2);
       FOR GT1←GT2 STEP GT2,LINK WHILE GT2.LINK≠0 DO
            PUT((GT2←TAKE(GT1))&STACKHEAD[GT3←TAKE(GT1+1).[12:36]
                 MOD 125][35:35:13],STACKHEAD[GT3]←GT1);
COMMENT THIS IS THE FILL FOR THE SPECIAL CHARACTORS;
FILL SPECIAL[*] WITH
 OCT1200000000200000,  COMMENT  #;   OCT0000000000100000,  COMMENT  @;
 OCT0000000000000000,                OCT1160000000120000,  COMMENT  !;
 OCT1370440450002763,  COMMENT  >;   OCT1370440250002662,  COMMENT  ≥;
 OCT1400440200000000,  COMMENT  +;   OCT0000000000000000,
 OCT1220000000060000,  COMMENT  ,;   OCT1210000000000000,  COMMENT  [;
 OCT1250000000000000,  COMMENT  &;   OCT0450000000000000,  COMMENT  (;
 OCT1370450450003571,  COMMENT  <;   OCT1330401040000000,  COMMENT  ←;
 OCT1410441000000000,  COMMENT  ×;   OCT0000000000000000,
 OCT0000000000040000,  COMMENT  $;   OCT0470000000000000,  COMMENT  *;
 OCT1400440600000000,  COMMENT  -;   OCT1240000000160000,  COMMENT  );
 OCT0620000000000000,  COMMENT  .;   OCT1370450250003470,  COMMENT  ≤;
 OCT0000000000000000,                OCT1410442000000000,  COMMENT  /;
 OCT1170000000000000,  COMMENT  ,;   OCT0000000000020000,  COMMENT  %;
```

```
OCT1370441050002561,  COMMENT  #;    OCT1370451050002460,  COMMENT  =;          09212000 T  0054
OCT1230000000000000,  COMMENT  ];    OCT0000000000140000,  COMMENT  ";          09213000 T  0054
 0,0;                                                                          09214000 T  0054
                                                                  73 IS   32 LONG, NEXT SEG   69
  FILL MACRO[*] WITH                                                          09215000 T  0054
       OCT0131,          COMMENT SFS A       00              ;                09216000 T  0055
                                                                START OF SEGMENT **********   74
       OCT0116,          COMMENT SFD A       01              ;                09217000 T  0056
       OCT0000,          COMMENT SYNTAX ERROR02              ;                09218000 T  0056
       OCT0140,          COMMENT INC A       03              ;                09219000 T  0056
       OCT0130,          COMMENT SRS A       04              ;                09220000 T  0056
       OCT0117,          COMMENT SRD A       05              ;                09221000 T  0056
       OCT0000,          COMMENT SYNTAX ERROR06              ;                09222000 T  0056
       OCT0000,          COMMENT SYNTAX ERROR07              ;                09223000 T  0056
       OCT00310143,      COMMENT CRF A, SFS 008              ;                09224000 T  0056
       OCT00160143,      COMMENT CRF A, SFD 009              ;                09225000 T  0056
       OCT00470143,        COMMENT CRF A,  JFW 0  10          ;               09226000 T  0056
       OCT00400143,      COMMENT CRF A, INC 011              ;                09227000 T  0056
       OCT00300143,      COMMENT CRF A, SRS 012              ;                09228000 T  0056
       OCT00170143,      COMMENT CRF A, SRD 013              ;                09229000 T  0056
       OCT0000,          COMMENT SYNTAX ERROR14              ;                09230000 T  0056
       OCT0000,          COMMENT SYNTAX ERROR15              ;                09231000 T  0056
       OCT0153,          COMMENT RSA A       16              ;                09232000 T  0056
       OCT0104,          COMMENT RDA A       17              ;                09233000 T  0056
       OCT0150,          COMMENT RCA A       18              ;                09234000 T  0056
  OCT004201430042,       COMMENT SEC 0, CRF A, SEC 0 19      ;                09235000 T  0056
       OCT0122,          COMMENT SES A       20              ;                09236000 T  0056
       OCT0106,          COMMENT SED A       21              ;                09237000 T  0056
       OCT0000,          COMMENT SYNTAX ERROR22              ;                09238000 T  0056
       OCT0000,          COMMENT SYNTAX ERROR23              ;                09239000 T  0056
       OCT0056,          COMMENT TSA 0       24              ;                09240000 T  0056
       OCT0000,          COMMENT SYNTAX ERROR25              ;                09241000 T  0056
       OCT0000,          COMMENT SYNTAX ERROR26              ;                09242000 T  0056
       OCT0000,          COMMENT SYNTAX ERROR27              ;                09243000 T  0056
       OCT0000,          COMMENT SYNTAX ERROR28              ;                09244000 T  0056
       OCT0007,          COMMENT TDA 0       29              ;                09245000 T  0056
       OCT0000,          COMMENT SYNTAX ERROR30              ;                09246000 T  0056
       OCT0000,          COMMENT SYNTAX ERROR31              ;                09247000 T  0056
       OCT0115,          COMMENT SSA A       32              ;                09248000 T  0056
       OCT0114,          COMMENT SDA A       33              ;                09249000 T  0056
       OCT0154,          COMMENT SCA A       34              ;                09250000 T  0056
       OCT0141;          COMMENT STC A       35              ;                09251000 T  0056
                                                                  74 IS   36 LONG, NEXT SEG   69
  FILL OPTIONS[*] WITH "5CHECK",0,          % 0,1               09251208 T  0056
                                                                START OF SEGMENT **********   75
                       "6DEBUG",0,          % 2,3               09251212 T  0058
                       "4DECKO",0,          % 4,5               09251214 T  0058
                       "6FORMA",0,          % 6,7               09251216 T  0058
                       "9INTRI",0,          % 8,9               09251218 T  0058
                       "5LISTA",0,          % 10,11             09251220 T  0058
                       "4LISTO",0,          % 12,13             09251224 T  0058
                       "5LISTP",0,          % 14,15             09251228 T  0058
                       "3MCPOO",0,          % 16,17             09251230 T  0058
                       "4TAPEO",0,          % 18,19             09251232 T  0058
                       "4NESTO",0,          % 20,21             09251234 T  0058
                       "3NEWOO",0,          % 22,23             09251236 T  0058
                       "7NEWIN",0,          % 24,25             09251240 T  0058
```

```
                "4OMITO",0,          % 26,27                                  09251244 T  0058
                "1$0000",0,          % 28,29                                  09251248 T  0058
                "3PRTOO",0,          % 30,31                                  09251252 T  0058
                "5PUNCH",0,          % 32,33                                  09251256 T  0058
                "5PURGE",0,          % 34,35                                  09251260 T  0058
                "4SEGSO",0,          % 36,37                                  09251264 T  0058
                "3SEQOO",0,          % 38,39                                  09251268 T  0058
                "6SEQER",0,          % 40,41                                  09251272 T  0058
                "6SINGL",0,          % 42,43                                  09251276 T  0058
                "5STUFF",0,          % 44,45                                  09251278 T  0058
                "4VOIDO",0,          % 46,47                                  09251280 T  0058
                "5VOIDT",0,          % 48,49                                  09251284 T  0058
0;                                                                              09251288 T  0058
                                                                        75 IS   51 LONG, NEXT SEG   69
 DO UNTIL STEPI = BEGINV;                                                       09252000 T  0058
 GT1 ←"       ";                                                              09253000 T  0060
 INTOG ← INTOG AND TRUE; %                                                      09253050 T  0061
 DISKADR ← IF INTOG THEN INTRINSICADR ELSE 2;                                   09253100 T  0064
 MDESC(GT1,INFO[LASTSEQROW,LASTSEQUENCE]);                                      09253500 T  0067
 MDESC(GT1,INFO[LASTSEQROW,LASTSEQUENCE-1]);                                    09254000 T  0069
 MDESC(GT1,INFO[LASTSEQROW,LASTSEQUENCE-2]);                                    09255000 T  0072
STMT;                                                                           09275000 T  0074
 LOCK(STUFF);                                                                   09281000 T  0075
 CLOSE(CARD,RELEASE);                                                           09281500 T  0076
 IF LASTUSED ≠ 1 THEN CLOSE(TAPE,RELEASE);                                      09282000 T  0078
 IF NEWTOG THEN LOCK(NEWTAPE,*);                                                09282500 T  0081
 IF T←((L+3)DIV 4) + CORADR > 4080 THEN FLAG(040);                              09282600 T  0084
 IF NOT NOHEADING THEN % PRINT THESE THINGS IF ANY                              09362000 T  0088
                   BEGIN          % LISTING HAS BEEN DONE.                     09363000 T  0089
          STREAM PROCEDURE PAN(T,FIEL,NER,LSQ); VALUE NER,T;                    09364000 T  0089
PRT(642) = *SEGMENT DESCRIPTOR*
                                                                  START OF SEGMENT **********   76
PRT(643) = PAN
                BEGIN DI ← FIEL; 44(DS←2LIT" ");                                09365000 T  0000
                SI ← LSQ; DS ← WDS; SI ←FIEL; DS ← 3 WDS;                       09366000 T  0001
                DI ← FIEL; DS← 28 LIT"NUMBER OF ERRORS DETECTED = ";            09367000 T  0002
                SI ← LOC NER; DS ← 3 DEC; DS ← 22 LIT ",  COMPILATION TIME = "; 09368000 T  0006
                SI ← LOC T; DS ← 4 DEC; DS ← 9 LIT " SECONDS."; END;            09369000 T  0009



          STREAM PROCEDURE PEN(FIL,PRTSIZ,BASE,CORE,DISK);                      09370000 T  0012
PRT(644) = PEN
                    VALUE PRTSIZ,BASE,CORE,DISK;                                09371000 T  0012
                BEGIN DI←FIL; DS ← 9 LIT"PRT SIZE="; SI←LOC PRTSIZ;              09372000 T  0012
                    DS ← 3 DEC; DS←14 LIT" BASE ADDRESS=";                      09373000 T  0014
                    SI←LOC BASE; DS←4 DEC; DS←10 LIT" CORE REQ=";               09374000 T  0016
                    SI←LOC CORE; DS←4 DEC; DS←10 LIT" DISK REQ=";               09375000 T  0018
                    SI←LOC DISK; DS←5 DEC; DS←61 LIT " ";                       09376000 T  0020
                END  PEN;                                                       09377000 T  0028



          STREAM PROCEDURE FINALAX(LINE,N,SEQ); VALUE N;                        09378000 T  0029
PRT(645) = FINALAX
```

```
                    BEGIN DI ← LINE; 15(DS ← 8 LIT " ");                                          09379000 T  0029
                          DI ← LINE; DS ← 31 LIT "NUMBER OF ACCIDENTAL ENTRIES = ";               09380000 T  0031
                          SI ← LOC N; DS ← 3 DEC; DI ← DI+8;                                      09381000 T  0035
                          SI ← SEQ; SI ← SI-16; DS ← 8 CHR;                                       09382000 T  0036
                    END;                                                                          09383000 T  0037



                    IF AXNUM ≠0 THEN                                                              09384000 T  0037
                          BEGIN                                                                   09384050 T  0038
                          FINALAX(LIN[0],AXNUM,INFO[LASTSEQROW,LASTSEQUENCE]);                    09384100 T  0039
                          WRITELINE;                                                              09384500 T  0041
                          END;                                                                    09384600 T  0052
                    SCRAM := (TIME(1)-TIME1)/60;                                                  09385000 T  0052
                    PAN(SCRAM,LIN[0],ERRORCOUNT,INFO[LASTSEQROW,LASTSEQUENCE-1])                  09386000 T  0054
                         ;                                                                        09386500 T  0057
                         WRITELINE;                                                               09387000 T  0057
                               PEN(LIN[0],PRTIMAX,T1:=(L+3)DIV 4,T1:=CORADR+T,                    09388000 T  0067
                      ((T+29)DIV 30+DISKADR)×30);                                                 09389000 T  0071
                               WRITELINE;                                                         09389500 T  0074
                               LOCK(LINE,RELEASE); END;                                           09390000 T  0084
PRT(646) = *SEGMENT DESCRIPTOR*

                                                                                      76 IS   87 LONG, NEXT SEG   69
                    IF ERRORCOUNT ≠ 0 THEN I←0/0 ELSE                                             09391000 T  0092
                        BEGIN                                                                     09392000 T  0094
                        ARRAY SAVINFO[0:31,0:255],                                                09392300 T  0095
PRT(647) = *SEGMENT DESCRIPTOR*

                                                                                     START OF SEGMENT **********  77
STACK(F+2) = SAVINFO
                    INFO[0:200,0:255]; % FOR LARGE MCP'S,                                         09392500 T  0003
STACK(F+3) = INFO
                                  INTEGER SAVNDX,NONSAVNDX,N;                                     09393000 T  0005
STACK(F+4) = SAVNDX
STACK(F+5) = NONSAVNDX
STACK(F+6) = N
                    INTEGER Q,J,K,M;                                                              09393010 T  0005
STACK(F+7) = Q
STACK(F+10) = J
STACK(F+11) = K
STACK(F+12) = M
                    BOOLEAN TSSTOG; REAL T;                                                       09393020 T  0005
STACK(F+13) = TSSTOG
STACK(F+14) = T
                    REAL PROCEDURE PUSHER(GRINCH,GOT,XMAS); VALUE XMAS; REAL XMAS;                09393050 T  0005
PRT(650) = PUSHER
                                       ARRAY GOT[0]; ARRAY GRINCH [0,0];                          09393060 T  0005
                                                                                                  09393070 T  0005
                         BEGIN                                                                    09393080 T  0005
                         REAL WHO,WHAT;                                                       START OF SEGMENT **********  78

STACK(F+3) = WHO
STACK(F+4) = WHAT
                         DEFINE LINKR = [32:8]#;                                                  09393090 T  0000
                    %                                                                             09393100 T  0000
                         IF WHO:=XMAS.LINKC ≤ 225 THEN                                            09393110 T  0000
                              BEGIN                                                               09393120 T  0001
                              MOVE(30,GRINCH[XMAS.LINKR,WHO],GOT[0]);                             09393130 T  0002
```

```
                    PUSHER:=XMAS + 30;                                          09393140 T  0005
                    END                                                         09393150 T  0006
               ELSE BEGIN                                                       09393160 T  0006
                    MOVE(WHAT:=256-WHO,GRINCH[XMAS,LINKR,WHO],GOT[0]);          09393170 T  0007
                    XMAS:=XMAS + WHAT;                                          09393180 T  0011
                    MOVE(WHO:=30-WHAT, GRINCH[XMAS,LINKR,0], GOT[WHAT]);        09393190 T  0012
                    PUSHER:=XMAS + WHO;                                         09393200 T  0016
                    END;                                                        09393220 T  0017
               END PUSHER;                                                      09393230 T  0017
                                                                          78 IS   20 LONG, NEXT SEG   77


          PROCEDURE PUSHEE(GRINCH,N,B,Y); VALUE N,B,Y; REAL N,B,Y;              09393240 T  0005
PRT(651) = PUSHEE
               ARRAY GRINCH[0,0];                                               09393250 T  0005
               BEGIN                                                            09393260 T  0005
               REAL I,J,X;                                                      09393270 T  0005
                                                                     START OF SEGMENT **********   79
STACK(F+2) = I
STACK(F+3) = J
STACK(F+4) = X
               DEFINE LINKR = [32:8]#;                                          09393280 T  0000
               J:=Y;                                                            09393290 T  0000
               I:=B + N;                                                        09393300 T  0000
               WHILE B < I DO                                                   09393310 T  0002
                    BEGIN                                                       09393320 T  0003
                    IF Y:=B.LINKC ≤ 225 THEN                                    09393330 T  0003
                    BEGIN                                                       09393340 T  0005
                    MOVE(30,CODE(J),GRINCH[B,LINKR,Y]);                         09393350 T  0005
                    J:=J + 30;                                                  09393360 T  0011
                    B:=B + 30;                                                  09393370 T  0012
                    END                                                         09393380 T  0013
               ELSE BEGIN                                                       09393390 T  0013
                    MOVE(X:=256-Y,CODE(J),GRINCH[B,LINKR,Y]);                   09393400 T  0014
                    B:=B + X;                                                   09393410 T  0020
                    J:=J + X;                                                   09393420 T  0021
                    MOVE(Y:=30-X,CODE(J),GRINCH[B,LINKR,0]);                    09393430 T  0023
                    B:=B + Y;                                                   09393440 T  0029
                    J:=J + Y;                                                   09393450 T  0030
                    END;                                                        09393460 T  0032
               END;                                                             09393470 T  0032
          END PUSHEE;                                                           09393480 T  0032
                                                                          79 IS   35 LONG, NEXT SEG   77


     STREAM PROCEDURE FIXHDR(F,N); VALUE N;                                     09393700 T  0005
PRT(652) = FIXHDR
          BEGIN SI←F; SI←SI-24; DI←LOC F; DS←WDS;                               09393710 T  0005
                SI←F; 14(SI←SI+8); DI←LOC F; DS←WDS;                            09393720 T  0007
                DI←F; DI←DI+38; SI← LOC N;                                      09393730 T  0008
                SI←SI+7; DS←CHR;                                                09393740 T  0009
          END FIXHDR;                                                           09393750 T  0009
```

```
          LABEL EOF;                                                                                  09394000 T  0010
     IF NOT INTOG THEN                                                                                09394100 T  0010
BEGIN                                                                                                 09394200 T  0011
          L←(L+3)DIV 4; COMMENT L←NUM. OF WORDS IN OUTER BLOCK;                                       09395000 T  0011
          FILL SAVINFO[0,*] WITH                                                                      09395100 T  0013
               OCT7700000000000015,                                                                   09395200 T  0014
                                                                                        START OF SEGMENT **********    80
               OCT0253010477527705,                                                                   09395300 T  0015
               OCT0051000000000000,                                                                   09395400 T  0015
               OCT0441070001000062;                                                                   09395500 T  0015
                                                                                          80 IS    4 LONG, NEXT SEG   77
          Q ← -1;                                                                                     09395700 T  0015
     PUSHEE(SAVINFO,L,4,5);                                                                           09396000 T  0016
     SAVNDX:=L;                                                                                       09397000 T  0018
END;                                                                                                  09397100 T  0019
          REWIND(CODISK);                                                                             09398000 T  0019
 DO BEGIN IF   REED=0 THEN GO TO EOF;                                                                 09399000 T  0021
           N←FETCH(MKABS(CODE(0)))-1;                                                                 09400000 T  0022
           IF BOOLEAN(FETCH(MKABS(CODE(1)))) THEN                                                     09401000 T  0028
BEGIN                                                                                                 09402000 T  0033
     PUSHEE(SAVINFO,N,SAVNDX,1);                                                                      09402100 T  0034
     SAVNDX:=SAVNDX +N;                                                                               09403000 T  0036
END ELSE BEGIN                                                                                        09404000 T  0037
     IF DECKTOG THEN                                                                                  09405000 T  0038
               STACKHEAD[Q←Q+1] ← 1024×NONSAVNDX←N;                                                    09405500 T  0038
          PUSHEE(INFO,N,NONSAVNDX,1);                                                                 09406000 T  0042
          NONSAVNDX:=((NONSAVNDX + N + 29)DIV 30)×30;                                                 09407000 T  0044
          END;                                                                                        09408000 T  0047
          END UNTIL FALSE;                                                                            09412000 T  0047
     EOF:      N←(SAVNDX+29) DIV 30; COMMENT NUMBER OF DISK SEGMENTS                                  09413000 T  0048
               OCCUPIED BY SAVE PROCEDURES AND ARRAYS;                                                09414000 T  0050
     IF INTOG AND NOT DECKTOG THEN                                                                    09414010 T  0050
BEGIN    % INTRINSIC FUNCTION OPTION                                                                  09414020 T  0052
     FOR J:=USEROPINX STEP 2 UNTIL OPARSIZE DO % IS TIMESHARING SET                                   09414022 T  0053
     IF OPTIONS[J] = "@TIMES" THEN                                                                    09414024 T  0055
     BEGIN TSSTOG:=BOOLEAN(OPTIONS[J+1]); J:=OPARSIZE END;                                            09414026 T  0056
     I ← PRTBASE + 1; J ← 0;                                                                          09414030 T  0061
     DO IF GT1 ← PRT[I] ≠0 THEN                                                                       09414040 T  0063
     BEGIN                                                                                            09414050 T  0065
          J ← J+1;                                                                                    09414060 T  0065
           SAVINFO[J,LINKR,J,LINKC] ←                                                                 09414070 T  0066
            0&GT1[8:8:10]                                                                             09414080 T  0069
            &GT1[33:18:15];                                                                           09414090 T  0070
     END UNTIL I:=I +1 ≥ PRTIMAX;                                                                     09414100 T  0071
      SAVINFO[0,0] ← J;      % # OF INTRINSICS                                                        09414110 T  0073
     SAVNDX ← MAXINTRINSIC;                                                                           09414120 T  0075
END ELSE BEGIN                                                                                        09414130 T  0076
               I←PRTBASE; DO IF GT1←PRT[I]≠0 THEN                                                     09415000 T  0078
           BEGIN IF GT1.[1:5]≠LDES THEN                                                               09415500 T  0081
          BEGIN IF (GT1←GT1&(GT1.[33:15]+L)[33:33:15]).[6:2]≠3 THEN                                   09416000 T  0082
                    GT1←GT1&(GT1.[18:15]+N)[18:33:15];                                                09417000 T  0086
           END;                                                                                       09417500 T  0089
               MDESC(GT1,SAVINFO[I,LINKR,I,LINKC]);                                                   09418000 T  0089
           END ELSE SAVINFO[I,LINKR,I,LINKC]:=0 UNTIL I:=I+12PRTIMAX;                                 09419000 T  0092
               MDESC(0&1[2:47:1],SAVINFO[0,PRTBASE-1]);                                               09419100 T  0097
```

```
        SAVNDX ← 30 × N;                                                                09420000 T  0101
END;                                                                                    09420010 T  0102
        I ← 0; J ← -1;                                                                  09420020 T  0102
                                                                                        09420100 T  0104
        IF NOT DECKTOG THEN                                                             09421000 T  0104
  BEGIN                                                                                 09421500 T  0105
      DO                                                                                09422000 T  0106
  BEGIN                                                                                 09423000 T  0106
        I:=PUSHER(SAVINFO,ELBAT,I);                                                     09424000 T  0106
        J:=J + 1;                                                                       09425000 T  0109
        WRITE(DISK,30,ELBAT[*]);                                                        09425900 T  0110
  END UNTIL I ≥ SAVNDX;                                                                 09426000 T  0114
        I:=0;                                                                           09427000 T  0115
        WHILE I < NONSAVNDX DO                                                          09427100 T  0116
  BEGIN                                                                                 09427200 T  0118
        I:=PUSHER(INFO,ELBAT,I);                                                        09427500 T  0118
        J:=J + 1;                                                                       09428000 T  0121
        WRITE(DISK,30,ELBAT[*]);                                                        09429000 T  0122
  END;                                                                                  09430000 T  0126
        N←IF INTOG THEN IF TSSTOG THEN                                                  09430050 T  0127
             TSSINTYPE ELSE DCINTYPE ELSE MCPTYPE;                                      09430060 T  0128
        FIXHDR(DISK,N);                                                                 09430075 T  0131
        LOCK(DISK,*);                                                                   09430100 T  0132
  END  ELSE                                                                             09431000 T  0134
  BEGIN ELBAT[0]←0; I←16;                                                               09432000 T  0134
         DO BEGIN MOVE(8,SAVINFO[I.LINKR,I.LINKC],ELBAT[1]);                            09433000 T  0137
             ELBAT[9]←B2D(I←96)&1[11:47:1]&(I←96)[23:35:1];                             09434000 T  0140
                  WRITE(DECK,10,ELBAT[*]);                                              09435000 T  0145
             END UNTIL I←I←8≥SAVNDX;                                                    09436000 T  0149
         FILL ELBAT[*] WITH 0,                                                          09437000 T  0151
                                                                                START OF SEGMENT **********   81
                  OCT7500000000000012,                                                  09438000 T  0153
                  OCT0004535530611765,                                                  09439000 T  0153
                  OCT7006000404210435,                                                  09440000 T  0153
                  OCT7700000000000015,                                                  09441000 T  0153
                  OCT0253010477527705,                                                  09442000 T  0153
                  OCT0051000004410046,                                                  09443000 T  0153
                  OCT0441070001000062,                                                  09444000 T  0153
                  OCT0040413100000000,                                                  09445000 T  0153
                  OCT0001000000000101;                                                  09446000 T  0153
                                                                                81 IS   10 LONG, NEXT SEG   77
   WRITE(DECK,10,ELBAT[*]);                                                             09447000 T  0153
  ELBAT[0] ←0&REAL(DECKTOG)[1:19:17];                                                   09447010 T  0157
  FOR I ← 0 STEP 1 UNTIL Q DO                                                           09447020 T  0160
       BEGIN K ← STACKHEAD[I].[23:15];                                                  09447030 T  0162
             M ← STACKHEAD[I].[38:10];                                                  09447040 T  0163
             FOR J ← 0 STEP 8 UNTIL M DO BEGIN                                          09447050 T  0165
                  MOVE(8,INFO[(J+K).LINKR,(J+K).LINKC],                                 09447060 T  0166
                       ELBAT [1]);                                                      09447070 T  0169
                  ELBAT[9] ← B2D(J)&"310"[1:31:17];                                     09447080 T  0170
                  WRITE(DECK,10,ELBAT[*]) END;                                          09447090 T  0173
       END;                                                                             09447100 T  0179
END END END PROGRAM;                                                                    09448000 T  0182
PRT(653) = *SEGMENT DESCRIPTOR*

                                                                                77 IS  186 LONG, NEXT SEG   69
                                                                                69 IS  101 LONG, NEXT SEG    3
```

```
                COMMENT THIS SECTION CONTAINS GENERATORS USED BY THE BLOCK ROUTINE;          10000000 T  0468
                PROCEDURE DEFINEGEN(MACRO,J); VALUE MACRO,J; BOOLEAN MACRO; REAL J;         10228000 T  0468
PRT(654) = DEFINEGEN
                    BEGIN                                                                   10229000 T  0468
                       OWN INTEGER CHARCOUNT, REMCOUNT;                                     10230000 T  0468
                                                                    START OF SEGMENT **********   82
PRT(655) = CHARCOUNT
PRT(656) = REMCOUNT
                COMMENT CHARCOUNT CONTAINS NUMBER OFCHARACTORS OF THE DEFINE THAT WE        10231000 T  0000
                      HAVE PUT INTO INFO.  REMCOUNT CONTAINS NUMBER OF CHARACT-             10232000 T  0000
                      ORS REMAINING IN THIS ROW OF INFO;                                    10233000 T  0000
                PROCEDURE PUTOGETHER(CHAR); REAL CHAR;                                      10234000 T  0000
PRT(657) = PUTOGETHER
                    BEGIN                                                                   10235000 T  0000
                STREAM PROCEDURE PACKINFO(INFO,ISKIP,COUNT,ASKIP,ACCUM);                    10236000 T  0000
                                                                    START OF SEGMENT **********   83
PRT(660) = PACKINFO
                      VALUE ISKIP,COUNT,ASKIP;                                              10237000 T  0000
                      BEGIN DI ← INFO; DI ← DI+ISKIP;                                       10238000 T  0000
                            SI ← ACCUM;SI ← SI+ASKIP; SI ← SI+3;                            10239000 T  0000
                            DS ← COUNT CHR END PACKINFO;                                    10240000 T  0001



                    INTEGER COUNT,SKIPCOUNT;                                                10241000 T  0002
STACK(F+2) = COUNT
STACK(F+3) = SKIPCOUNT
                  IF (COUNT ← CHAR.[12:6]) + CHARCOUNT > 2047                               10242000 T  0002
                    THEN BEGIN FLAG(142);  TB1← TRUE  END                                   10243000 T  0004
                    ELSE BEGIN                                                              10244000 T  0007
                       IF COUNT > REMCOUNT                                                  10245000 T  0009
                         THEN BEGIN                                                         10246000 T  0009
                            SKIPCOUNT ← COUNT-(COUNT+REMCOUNT);                             10247000 T  0010
                            REMCOUNT ← 2047 END                                             10248000 T  0012
                         ELSE REMCOUNT ← REMCOUNT-COUNT;                                    10249000 T  0012
                       GT1 ← CHARCOUNT DIV 8 + NEXTINFO;                                    10250000 T  0016
                       PACKINFO(INFO[GT1.LINKR,GT1.LINKC],CHARCOUNT.[45:3],                 10251000 T  0018
                       COUNT,0,CHAR);                                                       10252000 T  0021
                       IF SKIPCOUNT ≠ 0 THEN                                                10253000 T  0022
                           PACKINFO(INFO[NEXTINFO.LINKR+1,0],0,SKIPCOUNT,                   10254000 T  0023
                             COUNT,CHAR);                                                   10255000 T  0026
                       CHARCOUNT ← CHARCOUNT+SKIPCOUNT+COUNT END                            10256000 T  0027
                END PUTOGETHER;                                                             10257000 T  0029
                                                                    83 IS   32 LONG, NEXT SEG   82



                STREAM PROCEDURE SCAN(D,S,Q,N,J); VALUE J,N,Q;                              10257100 T  0000
PRT(661) = SCAN
                    BEGIN DI←D;DI←DI+11;SI←S;SI←SI+3;                                       10257200 T  0000
                       IF N SC=DC THEN                                                      10257300 T  0001
                       IF SC>"0" THEN                                                       10257400 T  0001
                       BEGIN DI←LOC J; DI←DI+7;                                             10257500 T  0002
```

```
                                IF SC≤DC THEN                                                    10257600 T  0003
                                BEGIN J←SI;DI←J;SI←LOC Q;SI←SI+6;DS←CHR;                          10257700 T  0004
                                    DI←S;DI←DI+2;DS←CHR;                                         10257800 T  0006
                          END   END   END;                                                       10257900 T  0006



                             INTEGER LASTRESULT;                                                 10258000 T  0007
STACK(F+2) = LASTRESULT
                               REAL K,N,ELCLASS;                                                 10258100 T  0007
STACK(F+3) = K
STACK(F+4) = N
STACK(F+5) = ELCLASS
                               DEFINE I=NXTELBT#;                                                10258200 T  0007
                               LABEL FINAL,PACKIN;                                               10258300 T  0007
                               LABEL BACK,SKSC,EXIT;                                             10259000 T  0007
                               TB1← FALSE;                                                       10260000 T  0007
                               CHARCOUNT←(NEXTINFO-LASTINFO)×8;                                  10261000 T  0007
                               DEFINECTR ← 1; LASTRESULT ← 2;                                    10262000 T  0009
                               REMCOUNT ← (256 - NEXTINFO MOD 256) × 8;                          10263000 T  0011
                               NEXTINFO←LASTINFO;                                                10263100 T  0013
                               IF J≠0 THEN N←TAKE(LASTINFO+1),[12:6];                            10263110 T  0014
                               K←0;                                                              10263200 T  0017
                    BACK:      STOPDEFINE←TRUE;                                                  10263300 T  0018
                               ELCLASS←TABLE(NXTELBT);                                           10263400 T  0019
                    SKSC:      NXTELBT←NXTELBT-1;                                                10263500 T  0021
                               IF MACRO THEN                                                     10263600 T  0022
                               BEGIN IF ELCLASS=COMMA THEN                                       10263700 T  0022
                                 IF K=0 THEN                                                     10263800 T  0023
                    FINAL:       BEGIN PUTOGETHER("1#0000"); GO TO EXIT END                    10263900 T  0025
                                 ELSE GO PACKIN;                                                 10264000 T  0029
                                 IF ELCLASS=LEFTPAREN OR ELCLASS=LFTBRKET THEN                   10264100 T  0029
                                 BEGIN K←K+1; GO TO PACKIN END;                                  10264200 T  0030
                                 IF ELCLASS=RTPAREN OR ELCLASS=RTBRKET THEN                      10264300 T  0033
                                 IF K←K-1<0 THEN GO FINAL ELSE GO PACKIN;                        10264400 T  0034
                                 IF ELCLASS=SEMICOLON THEN                                       10264410 T  0038
                                   BEGIN FLAG(142); GO TO FINAL END ELSE GO PACKIN               10264420 T  0038
                               END;                                                              10264500 T  0040
                               IF J≠0 THEN                                                       10264600 T  0040
                               IF ACCUM[1].[12:6]-1=N THEN                                       10264700 T  0041
                                 SCAN(INFO[LASTINFO. LINKR ,LASTINFO. LINKC],                    10264800 T  0043
                                      ACCUM[1],N+770,N,J);                                       10264900 T  0046
                    PACKIN:                                                                      10264910 T  0048
                               IF RESULT = 4                                                     10265000 T  0049
                                 THEN BEGIN                                                      10266000 T  0049
                      COMMENT  INSERT " MARKS = 2130706432 IS DECIMAL FOR 1"0000;                10267000 T  0050
                                  PUTOGETHER(2130706432);                                        10268000 T  0050
                                  PUTOGETHER(ACCUM[1]);                                          10269000 T  0051
                                  PUTOGETHER(2130706432) END                                     10270000 T  0052
                               ELSE BEGIN                                                        10271000 T  0052
                                 IF BOOLEAN(RESULT) AND BOOLEAN(LASTRESULT)                      10272000 T  0055
                                   THEN PUTOGETHER("1 0000"); COMMENT INSERT BLANK;              10273000 T  0055
                                 PUTOGETHER(ACCUM[1]) END;                                       10274000 T  0057
                               IF TB1 THEN GO TO EXIT;                                           10275000 T  0058
                               LASTRESULT ← RESULT;                                              10276000 T  0059
                               IF MACRO THEN GO BACK;                                            10276500 T  0059
```

```
                        IF ELCLASS=DECLARATORS AND ELBAT[I].ADDRESS = DEFINEV      10277000 T  0060
                          THEN BEGIN DEFINECTR ← DEFINECTR+1; GO BACK END;          10278000 T  0062
                        IF ELCLASS ≠ CROSSHATCH THEN GO BACK;                       10279000 T  0067
                        IF DEFINECTR ≠ 1                                            10280000 T  0068
                          THEN BEGIN STOPDEFINE ← TRUE;                             10281000 T  0068
                             IF ELCLASS←TABLE(I)≠COMMA THEN                         10282000 T  0070
                             DEFINECTR←DEFINECTR-1; GO SKSC END;                    10283000 T  0072
EXIT:           DEFINECTR← 0;                                                       10284000 T  0074
                NEXTINFO ←(CHARCOUNT+7) DIV 8←NEXTINFO;                             10285000 T  0075
        END   DEFINEGEN;                                                            10286000 T  0078
                                                                        82 IS   81 LONG, NEXT SEG     3



PROCEDURE DBLSTMT;                                                                  12002000 T  0468
        BEGIN                                                                       12003000 T  0468
        REAL S,T;                                                                   12004000 T  0468
                                                                        START OF SEGMENT **********    84
STACK(F+2) = S
STACK(F+3) = T
        LABEL L1,L2,L3,EXIT;                                                        12005000 T  0000
        S←0;                                                                        12006000 T  0000
        IF STEPI≠LEFTPAREN THEN ERR(281)                                            12007000 T  0000
        ELSE                                                                        12008000 T  0002
   L1:  BEGIN                                                                       12009000 T  0003
             IF STEPI=COMMA THEN                                                    12010000 T  0004
             BEGIN                                                                  12011000 T  0005
             DPTOG←TRUE;                                                            12012000 T  0005
             IF STEPI=ADOP THEN STEPIT;                                             12013000 T  0006
             EMITNUM(NLO);                                                          12014000 T  0008
             EMITNUM(IF ELBAT[I-1].ADDRESS =SUB THEN -NHI ELSE NHI);                12015000 T  0009
             DPTOG←FALSE;                                                           12016000 T  0013
             STEPIT;                                                                12017000 T  0014
             GO TO L2;                                                              12018000 T  0014
        END;                                                                        12019000 T  0015
        IF TABLE(I+1)=COMMA THEN                                                    12020000 T  0015
             BEGIN                                                                  12021000 T  0016
             IF ELCLASS=ADOP  OR ELCLASS=MULOP THEN                                 12022000 T  0017
                  BEGIN                                                             12023000 T  0019
                  EMITO(ELBAT[I].ADDRESS+1);                                        12024000 T  0019
                  IF S←S-1≤0 THEN FLAG(282); STEPIT;                                12025000 T  0021
                  GO TO L3;                                                         12026000 T  0025
                  END;                                                              12027000 T  0025
             IF ELCLASS=ASSIGNOP THEN                                               12028000 T  0025
                  BEGIN                                                             12029000 T  0026
                  IF S≠1 THEN FLAG(283); S←0;    STEPIT;                            12030000 T  0026
                  DO                                                                12031000 T  0030
                  BEGIN                                                             12032000 T  0030
              IF ELCLASS  ≠COMMA THEN BEGIN ERR(284);GO EXIT END;                   12033000 T  0030
                  STEPIT;                                                           12034000 T  0032
                  IF ELCLASS≤INTID AND ELCLASS≥REALID THEN                          12035000 T  0033
                  BEGIN  EMITN(ELBAT[I].ADDRESS); STEPIT END                        12036000 T  0034
                                        ELSE VARIABLE(FL);                          12037000 T  0037
                             EMITO(STD) END UNTIL S←S+1=2 ;                         12038000 T  0038
                  IF ELCLASS≠RTPAREN THEN ERR(285) ELSE STEPIT;                     12039000 T  0041
                  GO TO EXIT;                                                       12040000 T  0044
```

```
                    END;                                                                    12041000 T  0045
               IF ELCLASS≤INTID AND ELCLASS≥BOOID THEN                                      12042000 T  0045
                    BEGIN                                                                   12043000 T  0046
                    CHECKER(T←ELBAT[I]);                                                    12044000 T  0047
                    STEPIT;STEPIT;                                                          12045000 T  0048
                    AEXP;                                                                   12046000 T  0049
                    EMITV(T,ADDRESS);                                                       12047000 T  0050
                    GO TO L2;                                                               12048000 T  0051
                    END;                                                                    12049000 T  0052
               END    ;                                                                     12050000 T  0052
                    AEXP;                                                                   12051000 T  0052
                    IF ELCLASS≠COMMA THEN BEGIN ERR(284);GO EXIT                            12052000 T  0052
                                                                              END;          12053000 T  0055
                    STEPIT; AEXP; EMITO(XCH);                                               12054000 T  0055
L2:       S←S+1;                                                                            12055000 T  0056
L3:       IF ELCLASS≠COMMA THEN BEGIN ERR(284);GO TO EXIT END;                              12056000 T  0058
          GO TO L1;                                                                         12057000 T  0061
EXIT:END                                                                                    12058000 T  0062
     END  DBLSTMT;                                                                          12059000 T  0062
                                                                                  84 IS   65 LONG, NEXT SEG    3


          REAL PROCEDURE FIXDEFINEINFO(T); VALUE T; REAL T;                                 12101000 T  0468
               BEGIN REAL K,S,P,J,EL;                                                       12102000 T  0468
                                                                                 START OF SEGMENT **********   85
STACK(F+3) = K
STACK(F+4) = S
STACK(F+5) = P
STACK(F+6) = J
STACK(F+7) = EL
                    STREAM PROCEDURE SET(S,D,K,E); VALUE K,E;                               12103000 T  0000
PRT(662) = SET
               BEGIN SI←S;SI←SI+11;DI←D;DI←DI+3;DS←K CHR;                                   12104000 T  0000
                    SI←LOC E; SI←SI+6; DS←2 CHR;                                        12105000 T  0001
               END;                                                                         12106000 T  0002


                    MACROID←TRUE;                                                           12107000 T  0002
                    P←(FIXDEFINEINFO←T).ADDRESS;                                            12108000 T  0003
                    K←COUNT;                                                                12109000 T  0005
                    S←SCRAM;                                                                12110000 T  0006
                    STREAMTOG←TRUE & STREAMTOG[1:3:45] ;                                    12110100 T  0007
                    STOPDEFINE←TRUE;                                                        12111000 T  0008
                    EL←TABLE(NXTELBT);                                                      12112000 T  0009
                    NXTELBT←NXTELBT-1;                                                      12113000 T  0010
                    IF EL≠LEFTPAREN AND EL≠LFTBRKET THEN                                    12114000 T  0012
                         FLAG(141)                                                          12115000 T  0013
                    ELSE DO BEGIN J←J+1;                                                    12116000 T  0014
                         SET(INFO[T.LINKR,T.LINKC],ACCUM[1],K,64×J+12);                     12117000 T  0017
                         ACCUM[1].[12:6]←K+2;                                               12118000 T  0022
                         ACCUM[0]←0;                                                        12119000 T  0025
                         ACCUM[0].CLASS←DEFINEDID;                                          12120000 T  0026
                         COUNT←K+2;                                                         12121000 T  0028
```

```
                    SCRAM←ACCUM[1] MOD 125;                                                                  12122000 T  0030
                    E;                                                                                       12123000 T  0031
                    DEFINEGEN(TRUE,0);                                                                       12124000 T  0032
                END UNTIL EL←ELBAT[NXTELBT].CLASS≠COMMA;                                                     12125000 T  0033
                IF EL≠RTPAREN AND EL≠RTBRKET OR J≠P THEN FLAG(141);                                          12126000 T  0035
                MACROID←FALSE;                                                                           12127000 T  0039
                STREAMTOG←STREAMTOG,[1:45] ;                                                             12127100 T  0040
            END;                                                                                         12128000 T  0042
                                                                                                       85 IS   46 LONG, NEXT SEG    3


        PROCEDURE SCATTERELBAT;                                                                          13197000 T  0468
            BEGIN                                                                                        13198000 T  0468
                REAL T;                                                                                  13199000 T  0468
                                                                                                     START OF SEGMENT **********   86
STACK(F+2) = T
                T ← ELBAT[I];                                                                            13200000 T  0000
                KLASSF ← T.CLASS;                                                                        13201000 T  0001
                FORMALF ← BOOLEAN(T.FORMAL);                                                             13202000 T  0002
                VONF ← BOOLEAN(T.VU);                                                                    13203000 T  0003
                LEVELF ← T.LVL;                                                                          13204000 T  0004
                ADDRSF ← T.ADDRESS;                                                                      13205000 T  0006
                INCRF ← T.INCR;                                                                          13206000 T  0007
                LINKF ← T.LINK;                                                                          13207000 T  0008
            END  SCATTERELBAT;                                                                           13208000 T  0009
                                                                                                       86 IS   12 LONG, NEXT SEG    3


        PROCEDURE CHKSOB;                                                                                13209000 T  0468
PRT(663) = CHKSOB
            IF GTA1[J←J-1]≠0 THEN FLAG(23);                                                              13210000 T  0468


        DEFINE                                                                                           13211000 T  0472
            ADDC=532480#,                                                                                13212000 T  0472
            SUBC=1581056#,                                                                               13213000 T  0472
            EMITSTORE=EMITPAIR#;                                                                         13214000 T  0472
         PROCEDURE PURGE(STOPPER);                                                                       13215000 T  0472
                VALUE STOPPER;                                                                           13216000 T  0472
                REAL STOPPER;                                                                            13217000 T  0472
         BEGIN                                                                                           13218000 T  0472
           INTEGER POINTER;                                                                              13219000 T  0472
                                                                                                     START OF SEGMENT **********   87
STACK(F+2) = POINTER
            LABEL RECOV; DEFINE ELCLASS = KLASSF#;                                                       13220000 T  0000
            REAL J,N,OCR,TL,ADD;                                                                         13221000 T  0000
STACK(F+3) = J
STACK(F+4) = N
STACK(F+5) = OCR
STACK(F+6) = TL
STACK(F+7) = ADD
```

```
  POINTER←LASTINFO;                                                                  13222000 T  0000
WHILE POINTER ≥ STOPPER                                                              13223000 T  0000
DO                                                                                   13224000 T  0001
  BEGIN                                                                              13225000 T  0002
        IF ELCLASS←(GT1←TAKE(POINTER)).CLASS=NONLITNO                                13226000 T  0002
        THEN BEGIN                                                                   13227000 T  0004
        NCII←NCII+1;                                                                 13228000 T  0005
             EMITNUM(TAKE(POINTER+1));                                               13229000 T  0006
             EMITSTORE(MAXSTACK,STD);                                                13230000 T  0008
             MAXSTACK←(G←MAXSTACK)+1;                                                13231000 T  0009
             J←L; L←GT1.LINK;                                                        13232000 T  0011
             DO                                                                      13233000 T  0013
               BEGIN                                                                 13234000 T  0014
                  GT4←GET(L);                                                        13235000 T  0014
                  EMITV(G)                                                           13236000 T  0015
               END                                                                   13237000 T  0015
             UNTIL (L←GT4)=4095;                                                     13238000 T  0016
             L←J;                                                                    13239000 T  0017
             POINTER←POINTER-GT1.INCR                                                13240000 T  0018
             END                                                                     13241000 T  0018
        ELSE                                                                         13242000 T  0020
        BEGIN                                                                        13243000 T  0020
        IF NOT BOOLEAN(GT1.FORMAL)                                                   13244000 T  0022
          THEN BEGIN                                                                 13245000 T  0022
             IF ELCLASS = LABELID                                                    13246000 T  0023
               THEN BEGIN                                                            13247000 T  0023
                 ADD ← GT1.ADDRESS;                                                  13248000 T  0024
                 IF NOT BOOLEAN(OCR←TAKE(GIT(POINTER))).[1:1]                        13249000 T  0026
                    THEN IF OCR.[36:12] ≠ 0 OR ADD ≠ 0                               13250000 T  0028
                       THEN BEGIN GT1 ← 160; GO TO RECOV END;                        13251000 T  0030
             IF ADD ≠ 0 THEN                                                         13252000 T  0033
     PROGDESCBLDR(ADD,TRUE,OCR.[36:10],LDES) END                                     13252500 T  0033
     ELSE IF FALSE                                                                   13253000 T  0036
            THEN BEGIN                                                               13254000 T  0037
              IF TAKE(POINTER+1) < 0                                                 13255000 T  0037
                THEN BEGIN GT1 ← 162; GO TO RECOV END;                               13256000 T  0038
              OCR ←(J ←  TAKE(GIT(POINTER))).[24:12];                                13257000 T  0041
              N ← GET( (J←J.[36:12])+4); TL ← L;                                     13258000 T  0043
              IF ADD ← GT1.ADDRESS ≠ 0                                               13259000 T  0047
               THEN BEGIN                                                            13260000 T  0048
                                                                                     13261000 T  0049
                 IF OCR ≠ 0                                                          13262000 T  0049
     THEN BEGIN L←OCR=2; CALLSWITCH(POINTER); EMITO(BFW);END;                        13263000 T  0049
                 L←J+11; EMITL(15); EMITO(RTS);                                      13264000 T  0053
                 FOR J ← 4 STEP 4 UNTIL N                                            13265000 T  0056
                   DO BEGIN                                                          13266000 T  0056
                     EMITL(GNAT(GET(L)×4096+GET(L+1)));                              13267000 T  0057
                     EMITO(RTS) END END                                              13268000 T  0060
               ELSE BEGIN                                                            13269000 T  0063
                 L ← J+13;                                                           13270000 T  0066
                 FOR J ← 4 STEP 4 UNTIL N                                            13271000 T  0067
                   DO BEGIN                                                          13272000 T  0067
                     GT1 ← GET(L)×4096+GET(L+1);                                     13273000 T  0068
                     GOGEN(GT1,BFW) END;END;                                         13274000 T  0071
          L ← TL END                                                                 13277000 T  0074
        ELSE IF ELCLASS ≥ PROCID AND ELCLASS ≤ INTPROCID                             13278000 T  0075
         THEN IF TAKE(POINTER+1) < 0                                                 13279000 T  0078
```

```
                    THEN BEGIN GT1 ← 161;                                          13280000 T  0080
   RECOV:             MOVE(9,INFO[POINTER.LINKR,POINTER.LINKC],ACCUM);            13281000 T  0082
                      Q ← ACCUM[1]; FLAG(GT1); ERRORTOG ← TRUE END                13282000 T  0086
            END;                                                                   13283000 T  0089
       GT2←TAKE(POINTER+1);                                                        13284000 T  0089
       GT3←GT2.PURPT;                                                              13285000 T  0090
   STACKHEAD[(0&GT2[12:12:36])MOD 125]←TAKE(POINTER).LINK;                         13286000 T  0092
       POINTER←POINTER-GT3                                                         13287000 T  0095
            END                                                                    13288000 T  0096
     END ;                                                                         13289000 T  0097
  LASTINFO←POINTER;                                                              13290000 T  0097
  NEXTINFO←STOPPER                                                               13291000 T  0098
 END;                                                                            13292000 T  0098
                                                                  87 IS  103 LONG, NEXT SEG     3



PROCEDURE E;                                                                       13293000 T  0472
COMMENT                                                                            13294000 T  0472
       E IS THE PROCEDURE WHICH PLACES AN ENTRY IN INFO AND                        13295000 T  0472
       HOOKS IT INTO STACKHEAD. THE PREVIOUS STACKHEAD LINK                        13296000 T  0472
       IS SAVED IN THE LINK OF THE ELBAT WORD IN THE NEW ENTRY                     13297000 T  0472
       E PREVENTS AN ENTRY FROM OVERFLOWING A ROW,STARTING AT THE                  13298000 T  0472
       BEGINNING OF THE NEXT ROW IF NECESSARY                          ;           13299000 T  0472
 BEGIN                                                                             13300000 T  0472
    REAL WORDCOUNT,RINX;                                                           13301000 T  0472
                                                                  START OF SEGMENT **********  88
STACK(F+2) = WORDCOUNT
STACK(F+3) = RINX
            IF RINX←(NEXTINFO+WORDCOUNT←(COUNT+18)DIV 8     ).LINKR ≠             13302000 T  0000
                 NEXTINFO.LINKR                                                    13303000 T  0003
   THEN BEGIN PUT(125&(RINX×256←NEXTINFO)[27:40:8],NEXTINFO);                      13304000 T  0003
          NEXTINFO←256×RINX END;                                                   13305000 T  0007
          IF SPECTOG THEN                                                          13305100 T  0009
          IF NOT MACROID THEN                                                      13305200 T  0009
               UNHOOK;                                                             13305300 T  0010
                                                                                   13306000 T  0011
          ACCUM[0].INCR←WORDCOUNT;                                                 13307000 T  0011
          IF NOT INLINETOG OR MACROID THEN BEGIN                                   13307500 T  0013
           ACCUM[0].LINK ←STACKHEAD[SCRAM];STACKHEAD[SCRAM]←NEXTINFO;              13308000 T  0015
                            END;                                                   13308500 T  0019
           ACCUM[1].PURPT←NEXTINFO-LASTINFO;                                       13309000 T  0019
   MOVE(WORDCOUNT,ACCUM,INFO[NEXTINFO.LINKR,NEXTINFO.LINKC]);                      13310000 T  0022
          LASTINFO←NEXTINFO;                                                       13311000 T  0025
           NEXTINFO←NEXTINFO+WORDCOUNT                                             13312000 T  0026
 END;                                                                              13313000 T  0026
                                                                  88 IS   30 LONG, NEXT SEG     3



PROCEDURE ENTRY(TYPE);                                                             13314000 T  0472
          VALUE TYPE;                                                              13315000 T  0472
          REAL TYPE;                                                               13316000 T  0472
COMMENT                                                                            13317000 T  0472
       ENTRY ASSUMES THAT I IS POINTING AT AN IDENTIFIER WHICH                     13318000 T  0472
```

```
      IS BEING DECLARED AND MAKES UP THE ELBAT ENTRY FOR IT                          13319000 T  0472
      ACCORD TO TYPE        .IF THE ENTRY IS AN ARRAY AND NOT                        13320000 T  0472
      A SPECIFICATION THEN A DESCRIPTOR IS PLACED IN THE STACK                       13321000 T  0472
      FOR THE UPCOMING COMMUNICATE TO GET STORAGE FOR THE ARRAY(S)    ;              13322000 T  0472
BEGIN                                                                                13323000 T  0472
    J←0;I←I+1;                                                                       13324000 T  0472
    DO                                                                               13325000 T  0475
      BEGIN                                                                          13326000 T  0475
           STOPDEFINE ←TRUE; STEPIT; SCATTERELBAT;                                   13327000 T  0475
           IF FORMALF←SPECTOG                                                        13328000 T  0476
              THEN                                                                   13329000 T  0476
               BEGIN                                                                 13330000 T  0477
                IF ELCLASS≠SECRET                                                    13331000 T  0478
                   THEN FLAG(002);                                                   13332000 T  0478
                                BUP←BUP+1                                            13333000 T  0480
;    KLASSF←TYPE;MAKEUPACCUM; E;J←J+1;                                               13333500 T  0480
               END                                                                   13334000 T  0484
              ELSE                                                                   13335000 T  0484
               BEGIN                                                                 13336000 T  0484
                IF ELCLASS>IDMAX                                                     13337000 T  0484
              THEN IF ELCLASS= POLISHV THEN ELCLASS←TYPE ELSE FLAG(3);               13338000 T  0485
                IF LEVELF=LEVEL                                                      13339000 T  0489
                  THEN FLAG(001);                                                    13340000 T  0489
                VONF←P2;                                                             13341000 T  0491
                FORMALF←PTOG;                                                        13341100 T  0492
    KLASSF←TYPE; MAKEUPACCUM;E; J←J+1;                                               13342000 T  0492
    IF ((FORMALF←PTOG) OR(STREAMTOG AND NOT STOPGSP)) AND NOT P2                     13343000 T  0495
      THEN ADDRSF←PJ←PJ+1                                                            13344000 T  0497
       ELSE IF STOPGSP                                                               13345000 T  0499
              THEN ADDRSF←0                                                          13346000 T  0501
                           ELSE ADDRSF:=GETSPACE(P2,LASTINFO+1);                     13347000 T  0502
       PUT(TAKE(LASTINFO)& ADDRSF[16:37:11],LASTINFO);                               13348000 T  0505
     END END                                                                         13349000 T  0507
                                                                                     13350000 T  0507
      UNTIL STEPI≠COMMA OR STOPENTRY; GTA1[0]←J                                      13351000 T  0507
 END;                                                                                13352000 T  0510



  PROCEDURE UNHOOK;                                                                  13353000 T  0511
COMMENT                                                                              13354000 T  0511
   UNHOOK ASSUMES THAT THE WORD IN ELBAT[I] POINTS TO A PSUEDO ENTRY                 13355000 T  0511
   FOR APARAMETER,ITS JOB IS TO UNHOOK THAT FALSE ENTRY SO THAT                      13356000 T  0511
   E WILL WORK AS NORMAL.                                                  ;         13357000 T  0511
  BEGIN                                                                              13358000 T  0511
   REAL LINKT,A,LINKP;                                                               13359000 T  0511
                                                                    START OF SEGMENT **********    89
STACK(F+2) = LINKT
STACK(F+3) = A
STACK(F+4) = LINKP
       LABEL L;                                                                      13360000 T  0000
             LINKT←STACKHEAD[SCRAM]     ; LINKP←ELBAT[I].LINK;                     13361000 T  0000
             IF LINKT=LINKP THEN STACKHEAD[SCRAM]←TAKE(LINKT).LINK                   13362000 T  0002
                            ELSE                                                     13363000 T  0005
          L: IF A←TAKE(LINKT).LINK=LINKP                                             13364000 T  0006
             THEN PUT((TAKE(LINKT))&(TAKE(A))[35:35:13],LINKT)                       13365000 T  0008
```

```
                    ELSE BEGIN LINKT←A; GO TO L END;                                                    13366000 T  0012
      END;                                                                                              13367000 T  0014
                                                                                                  89 IS   17 LONG, NEXT SEG     3



    PROCEDURE MAKEUPACCUM;                                                                              13368000 T  0511
      BEGIN                                                                                             13369000 T  0511
        IF PTOG                                                                                         13370000 T  0511
          THEN GT1←LEVELF ELSE GT1←LEVEL;                                                               13371000 T  0512
        ACCUM[0]← ABS(ELBAT[I] & KLASSF[2:41:7] & REAL(FORMALF)[9:47:1]                                 13372000 T  0514
                    & REAL(VONF)[10:47:1] & GT1[11:43:5] &ADDRSF[16:37:11]                              13373000 T  0517
                    )                                                                                   13374000 T  0520
      END;                                                                                              13375000 T  0520



    PROCEDURE ARRAE;                                                                                    13376000 T  0521
PRT(664) = ARRAE
       BEGIN                                                                                            13377000 T  0521
        INTEGER SAVEINFO;                                                                               13378000 T  0521
                                                                                                  START OF SEGMENT **********   90
STACK(F+2) = SAVEINFO
        LABEL BETA1;                                                                                    13379000 T  0000
              TYPEV←REALARRAYID;                                                                        13380000 T  0000
               IF T1←GTA1[J←J-1]=0 THEN J←J+1                                                           13381000 T  0000
                                   ELSE                                                                 13382000 T  0004
                                     IF T1=OWNV THEN                                                    13383000 T  0005
                                   BEGIN                                                                13384000 T  0006
                                        P2←TRUE;IF SPECTOG THEN                                         13385000 T  0006
                                        FLAG(13)                                                        13386000 T  0007
                                     END                                                                13387000 T  0008
                                        ELSE                                                            13388000 T  0009
                                   TYPEV←REALARRAYID+T1-REALV;                                          13389000 T  0009
 BETA1: ENTER(TYPEV);                                                                                   13390000 T  0011
          IF ELCLASS≠LFTBRKET THEN FLAG(16);                                                            13391000 T  0012
          IF STEPI=LITNO THEN                                                                           13392000 T  0014
              BEGIN                                                                                     13393000 T  0015
                 SAVEINFO←ELBAT[I].ADDRESS;                                                             13394000 T  0016
                 IF STEPI≠RTBRKET THEN FLAG(53);                                                        13395000 T  0017
                                                FILLSTMT(SAVEINFO);                                     13396000 T  0020
SAVEINFO←1;                                                                                             13397000 T  0020
          END                                                                                           13398000 T  0021
              ELSE                                                                                      13399000 T  0021
         BEGIN IF ELCLASS≠ASTRISK THEN FLAG(56);                                                        13400000 T  0021
               SAVEINFO←1;                                                                              13401000 T  0024
              WHILE STEPI≠RTBRKET DO                                                                    13402000 T  0024
                   BEGIN IF ELCLASS≠COMMA AND                                                           13403000 T  0026
                         STEPI≠ASTRISK THEN FLAG(56);                                                   13404000 T  0027
                         SAVEINFO←SAVEINFO+1                                                            13405000 T  0029
                   END;STEPIT;                                                                          13406000 T  0030

END;       PUT(TAKE(LASTINFO)&SAVEINFO[27:40:8],LASTINFO);                                              13407000 T  0032
J ← 1 ;      GTA1[0] ← 0     ;                                                                          13408000 T  0032
IF ELCLASS=COMMA THEN BEGIN STEPIT;GO TO BETA1 END                                                      13408500 T  0034
                                                                                                        13409000 T  0036
```

```
END ARRAE;                                                                    13410000 T  0038
                                                                    90 IS   41 LONG, NEXT SEG    3



PROCEDURE PUTNBUMP(X);                                                        13589000 T  0521
            VALUE X;                                                          13590000 T  0521
            REAL X;                                                           13591000 T  0521
 BEGIN                                                                        13592000 T  0521
       INFO[NEXTINFO.LINKR,NEXTINFO.LINKC]←X;                                 13593000 T  0521
       NEXTINFO←NEXTINFO+1                                                    13594000 T  0525
 END ;                                                                        13595000 T  0525



 PROCEDURE JUMPCHKX;                                                          13596000 T  0526
COMMENT THIS PROCEDURE IS CALLED AT THE START OF ANY EXECUTABLE CODE          13597000 T  0526
        WHICH THE BLOCK MIGHT EMIT.IT DETERMINES WHETHER ANY JUMPS            13598000 T  0526
        ARROUND NONEXECUTABLE CODE MAY BE WAITING AND WHETHER IT              13599000 T  0526
        IS THE FIRST EXECUTABLE CODE;                                         13600000 T  0526
IF NOT SPECTOG THEN                                                           13601000 T  0526
BEGIN                                                                         13602000 T  0527
   IF AJUMP                                                                   13603000 T  0528
      THEN                                                                    13604000 T  0528
   BEGIN ADJUST;                                                              13605000 T  0528
         EMITB(BFW,SAVEL,L)                                                   13606000 T  0529
   END ELSE                                                                   13607000 T  0530
         IF FIRSTX=4095                                                       13608000 T  0530
           THEN                                                               13609000 T  0531
             BEGIN                                                            13610000 T  0531
               ADJUST;                                                        13611000 T  0532
               FIRSTX←L;                                                      13612000 T  0532
             END;                                                             13613000 T  0533
       AJUMP←FALSE                                                            13614000 T  0533
END;                                                                          13615000 T  0533



 PROCEDURE JUMPCHKNX;                                                         13616000 T  0536
COMMENT JUMPCHKNX DETERMINES WHETHER ANY EXECUTABLE CODE HAS BEEN             13617000 T  0536
        EMITTED AND IF SO WHETHER IT WAS JUST PREVIOUS TO THE                 13618000 T  0536
        NON EXECUTABLE ABOUT TO BE EMITTED.IF BOTH THEN L IS BUMPED           13619000 T  0536
        AND SAVED FOR A LATER BRANCH;                                         13620000 T  0536
IF NOT SPECTOG THEN                                                           13621000 T  0536
BEGIN                                                                         13622000 T  0536
   IF FIRSTX≠4095                                                             13623000 T  0537
     THEN                                                                     13624000 T  0537
      BEGIN                                                                   13625000 T  0537
       IF NOT AJUMP                                                           13626000 T  0538
         THEN                                                                 13627000 T  0538
            SAVEL←BUMPL;                                                      13628000 T  0538
      AJUMP←TRUE                                                              13629000 T  0541
      END;ADJUST                                                              13630000 T  0541
END;                                                                          13631000 T  0541
```

```
                PROCEDURE SEGMENTSTART(SAVECODE);VALUE SAVECODE;BOOLEAN SAVECODE;                13632000 T  0544
                                                                                                13632100 T  0544
                    BEGIN                                                                       13633000 T  0544
                    STREAM PROCEDURE PRINT(SAVECODE,ADR,FIEL); VALUE SAVECODE,ADR;                    START OF SEGMENT **********   91

PRT(665) = PRINT
                        BEGIN                                                                   13634000 T  0000
                        LABEL L1;                                                               13635000 T  0000
                        DI:=FIEL; DS:=8 LIT" ";                                                 13636000 T  0000
                        SI:=FIEL; DS:=9 WDS; DI:=DI-3;                                          13637000 T  0001
                        SAVECODE(DS:=38 LIT "START OF SAVE SEGMENT; BASE ADDRESS = ";           13638000 T  0002
                          JUMP OUT TO L1);                                                      13639000 T  0008
                        DS:=38 LIT " START OF REL SEGMENT; DISK ADDRESS = ";                    13640000 T  0009
                L1:                                                                             13641000 T  0014
                        SI:=LOC ADR; DS:=5 DEC;                                                 13642000 T  0014
                        END PRINT;                                                              13643000 T  0014



                    MOVE(1,SAVECODE,CODE(0));                                                   13651000 T  0014
                    IF SAVECODE AND INTOG AND NOT DECKTOG THEN FLAG(57);                        13651100 T  0019
                    IF LISTER OR SEGSTOG THEN                                                   13652000 T  0022
                        BEGIN                                                                   13652500 T  0024
                        PRINT(SAVECODE,IF SAVECODE THEN CORADR ELSE DISKADR,LIN[*]);            13653000 T  0024
                        IF NOHEADING THEN DATIME; WRITELINE;                                    13653500 T  0027
                        END;                                                                    13654000 T  0039
                    END SEGMENTSTART;                                                           13655000 T  0039
                                                                                              91 IS   40 LONG, NEXT SEG     3



                PROCEDURE SEGMENT(SIZE,FR); VALUE SIZE,FR; INTEGER SIZE,FR;                     13657000 T  0544
                    BEGIN                                                                       13660000 T  0544
                    STREAM PROCEDURE PRINT(SIZE,FIEL); VALUE SIZE;                              13661000 T  0544
                                                                                                  START OF SEGMENT **********   92

PRT(666) = PRINT
                        BEGIN                                                                   13663000 T  0000
                        DI:=FIEL; DS:=8 LIT" ";                                                 13665000 T  0000
                        SI:=FIEL; DS:=14 WDS;                                                   13667000 T  0001
                        DI:=DI-16; DS:=6 LIT"SIZE= ";                                           13668000 T  0002
                        SI:=LOC SIZE; DS:=4 DEC; DS:=6 LIT" WORDS"                              13670000 T  0003
                        END PRINT;                                                              13673000 T  0004



                    STREAM PROCEDURE DOIT(C,A,I,S,F,W); VALUE C,A,F,W;                          13673100 T  0005

PRT(667) = DOIT
                        BEGIN  LOCAL N;                                                         13673150 T  0005
                        DI:=S; DS:=8 LIT" "; SI:=S; DS:=9 WDS;                                  13673200 T  0005
                        DI:=DI-8; SI:=LOC W; DS:=4 DEC;                                         13673250 T  0007
                        SI:=I; SI:=SI+10;DI:=LOC N; DI:=DI+7; DS:=CHR;                          13673300 T  0007
```

```
        DI:=S; SI:=LOC F; SI:=SI+7; DS:=CHR; SI:=LOC C;                    13673350 T  0009
        DS:=3 DEC; DS:=4 DEC;SI:=I; SI:=SI+11;DS:=N CHR;                   13673400 T  0010
        END DOIT;                                                          13673450 T  0011



    IF LISTER OR SEGSTOG THEN                                              13674000 T  0012
        BEGIN                                                              13674500 T  0013
        PRINT(SIZE,LIN[*]);                                                13675000 T  0013
        IF NOHEADING THEN DATIME; WRITELINE;                               13676000 T  0015
        END;                                                               13677000 T  0026
    IF STUFFTOG THEN IF FR>0 THEN IF LEVEL>1 THEN                          13677100 T  0026
        BEGIN                                                              13677150 T  0029
        KLASSF:=TAKE(PROINFO).CLASS;                                       13677200 T  0030
       IF FR > 1024 THEN FR+FR-1024;                                       13677250 T  0032
        DOIT(KLASSF,FR,INFO[PROINFO.LINKR,PROINFO.LINKC],                  13677300 T  0034
             TWXA[0],SAF,SIZE);                                            13677400 T  0037
        WRITE(STUFF,10,TWXA[*]);                                           13677500 T  0039
        END;                                                               13677600 T  0043
    IF SIZE>SEGSIZEMAX THEN SEGSIZEMAX:=SIZE;                              13678000 T  0043
    END SEGMENT;                                                           13681000 T  0045
                                                                  92 IS   47 LONG, NEXT SEG     3



            STREAM PROCEDURE MOVECODE(EDOC,TEDOC);                         13683000 T  0544
PRT(670) = MOVECODE
    BEGIN LOCAL T1,T2,T3;                                                  13684000 T  0544
      SI←EDOC;T1←SI;                                                       13685000 T  0544
      SI←TEDOC;T2←SI;                                                      13686000 T  0544
      SI←LOC EDOC;                                                         13687000 T  0545
      SI←SI+3;                                                             13688000 T  0545
      DI←LOC T3;                                                           13689000 T  0545
      DI←DI+5;                                                             13690000 T  0545
      SKIP 3 DB;                                                           13691000 T  0546
    15(IF SB THEN DS← 1 SET ELSE DS←1 RESET;SKIP 1 SB);                    13692000 T  0546
      SI← LOC EDOC;                                                        13693000 T  0548
      DI← LOC T2;                                                          13694000 T  0548
      DS← 5 CHR;                                                           13695000 T  0548
      3(IF SB THEN DS←1 SET ELSE DS←1 RESET;SKIP 1 SB);                    13696000 T  0549
      DI←T3;                                                               13697000 T  0551
      SI←LOC T2;                                                           13698000 T  0551
      DS←WDS;                                                              13699000 T  0551
      DI←LOC T3;                                                           13700000 T  0551
      DI←DI+5;                                                             13701000 T  0552
      SKIP 3 DB;                                                           13702000 T  0552
      SI←LOC TEDOC;                                                        13703000 T  0552
      SI←SI+3;                                                             13704000 T  0552
    15(IF SB THEN DS←1 SET ELSE DS← 1 RESET;SKIP 1 SB);                    13705000 T  0553
      SI← LOC TEDOC;                                                       13706000 T  0555
      DI← LOC T1;                                                          13707000 T  0555
      DS← 5 CHR;                                                           13708000 T  0555
     3(IF SB THEN DS←1 SET ELSE DS←1 RESET;SKIP 1 SB);                     13709000 T  0555
      DI←T3;                                                               13710000 T  0557
      SI←LOC T1;                                                           13711000 T  0558
```

```
   DS←WDS                                                                    13712000 T  0558
 END;                                                                        13713000 T  0558



PROCEDURE ENTER(TYPE);                                                       13714000 T  0558
            VALUE TYPE;                                                      13715000 T  0558
       REAL TYPE;                                                            13716000 T  0558
BEGIN                                                                        13717000 T  0558
      G←GTA1[J←J-1];                                                         13718000 T  0558
      IF NOT SPECTOG                                                         13719000 T  0561
        THEN                                                                 13720000 T  0561
          BEGIN                                                              13721000 T  0561
            IF NOT P2                                                        13722000 T  0562
               THEN IF P2←(G=OWNV)                                           13723000 T  0562
                       THEN G←GTA1[J←J-1];                                   13724000 T  0563
            IF NOT P3                                                        13725000 T  0566
               THEN IF P3←(G=SAVEV)                                          13726000 T  0566
                       THEN G←GTA1[J←J-1]                                    13727000 T  0568
          END;                                                               13728000 T  0570
      IF G≠0 THEN FLAG(25) ELSE ENTRY(TYPE)                                  13729000 T  0571
END;                                                                         13730000 T  0574



PROCEDURE HTTEOAP(GOTSTORAGE,RELAD,STOPPER,PRTAD);                           13731000 T  0575
PRT(671) = HTTEOAP                                                           13732000 T  0575
                  VALUE GOTSTORAGE,RELAD,STOPPER,PRTAD;                      13733000 T  0575
                  BOOLEAN GOTSTORAGE;                                        13734000 T  0575
                  REAL RELAD,STOPPER,PRTAD;                                  13735000 T  0575
               BEGIN                                                         13736000 T  0575
                  IF FUNCTOG                                                 13737000 T  0575
                    THEN                                                     13738000 T  0575
                       BEGIN                                                 13739000 T  0575
                            EMITV(513);                                      13740000 T  0576
                            EMITO(RTN)                                       13741000 T  0576
                       END                                                   13742000 T  0577
                    ELSE                                                     13743000 T  0577
                             EMITO(XIT);                                     13744000 T  0578
                  CONSTANTCLEAN;                                             13745000 T  0579
                  PURGE(STOPPER);                                            13746000 T  0579
            MOVE(1,CODE(0),Z); PROGDESCBLDR(PRTAD,BOOLEAN(Z),(L+3)DIV 4,PDES);13747000 T  0586
            END HTTEOAP;



          PROCEDURE INLINE;                                                  13748000 T  0586
           BEGIN                                                             13749000 T  0586
            INTEGER SN,LN,P,LS,J; BOOLEAN MKST;                              13750000 T  0586
                                                                  START OF SEGMENT **********   93
STACK(F+2) = SN
STACK(F+3) = LN
STACK(F+4) = P
STACK(F+5) = LS
```

```
STACK(F+6) = J
STACK(F+7) = MKST
                    BOOLEAN FLIPFLOP;                                                    13750500 T  0000
STACK(F+10) = FLIPFLOP
                  INTEGER PN;                                                            13750600 T  0000
STACK(F+11) = PN
                 LABEL L1,L2,L3;                                                         13751000 T  0000
                   PN←1 ;                                                                13751100 T  0000
                        FLIPFLOP←INLINETOG←TRUE;P←0;MKST←FALSE;LS←L;EMITO(NOP);           13752000 T  0000
                  IF STEPI≠LEFTPAREN THEN FLAG(59);                                      13753000 T  0005
                  IF TABLE(I+1)=COLON THEN BEGIN STEPIT;GO TO L2 END ;                   13753100 T  0007
               L1: IF STEPI>IDMAX THEN BEGIN FLAG(465); GO TO L2 END ;                  13754000 T  0010
                    ACCUM[0]←0&P[16:37:11]&LOCLID[2:41:7]&SCRAM[35:35:13];  ENTER FORMAL PARAMS INTO INFO
                       E;IF FLIPFLOP THEN BEGIN FLIPFLOP←FALSE;LN←SN←LASTINFO END;    13755000 T  0013
                       IF STEPI=COMMA OR ELCLASS=COLON OR ELCLASS=RTPAREN              13755500 T  0018
                    THEN BEGIN I←I-2;STEPIT END                                       13756000 T  0021
                    ELSE IF ELCLASS≠ASSIGNOP THEN FLAG(60) ELSE STEPIT;               13757000 T  0023
                    AEXP;                                                             13758000 T  0026
                L2:  IF ELCLASS=COLON THEN                                            13759000 T  0030
                    BEGIN IF MKST THEN FLAG(99); MKST←TRUE; EMITO(MKS); P←P+2;        13760000 T  0030
                          IF TABLE(I+1)≠RTPAREN THEN GO TO L1; STEPIT                 13761000 T  0031
                          ;PN←2;                                                      13761100 T  0036
                    END ELSE P←P+1;   COUNT UP PARAMS WE HAVE SEEN                    13761110 T  0038
                    IF ELCLASS=COMMA THEN GO TO L1;                                   13761200 T  0040
                    IF ELCLASS≠RTPAREN THEN FLAG(61);                                 13762000 T  0041
                    IF NOT MKST THEN                                                  13763000 T  0043
                       BEGIN J←L;L←LS;EMITO(MKS);L←J END;                             13764000 T  0045
                  IF STEPI ≠ SEMICOLON THEN FLAG(62);                                 13765000 T  0045
                           EMITO(584);                                                13766000 T  0049
                                                                                      13766100 T  0051
                                                                                      13766200 T  0052
                                                                                      13766300 T  0052
                                                                                      13766400 T  0052
                                                                                      13766500 T  0052
                                                                                      13767000 T  0052
  L3:ELBAT[I]←TAKE(SN);SCATTERELBAT;ADDRSF←P-ADDRSF;                                   13768000 T  0055
      PUT(ELBAT[I]&ADDRSF[16:37:11]&STACKHEAD[LINKF][33:33:15],SN);                    13769000 T  0058
         STACKHEAD[LINKF]←SN;  SN←SN+INCRF;                                           13770000 T  0061
         IF ADDRSF≠PN THEN GO TO L3 ;                                                 13770500 T  0062
         INLINETOG←  FALSE;                                                           13770600 T  0063
         PN←NEXTINFO;                                                                 13771000 T  0063
         STREAMTOG←TRUE;STREAMWORDS;IF STEPI≠BEGINV THEN STREAMSTMT                   13772000 T  0066
         ELSE BEGIN STEPIT;COMPOUNDTAIL END;                                          13773000 T  0068
         STREAMTOG←FALSE;PURGE(PN);STREAMWORDS;PURGE(LN);EMITL(16);                   13773500 T  0072
                                                                                      13774000 T  0072
END INLINE;                                                                  93 IS   76 LONG, NEXT SEG    3


 COMMENT THIS SECTION CONTAINS THE BLOCK ROUTINE  ;                                   14000000 T  0586
PROCEDURE BLOCK(SOP);                                                                 14001000 T  0586
          VALUE SOP;                                                                  14002000 T  0586
          BOOLEAN SOP;                                                                14003000 T  0586
COMMENT SOP IS TRUE IF THE BLOCK WAS CALLED BY ITSELF THROUGH THE                     14004000 T  0586
        PROCEDURE DECLARATION-OTHERWISE IT WAS CALLED BY STATEMENT.                   14005000 T  0586
        THE BLOCK ROUTINE IS RESPONSIBLE FOR HANDLING THE BLOCK                       14006000 T  0586
        STRUCTURE OF AN ALGOL PROGRAM-SEGMENTING EACH BLOCK,HANDLING                  14007000 T  0586
```

```
                    ALL DECLARATIONS,DOING NECESSARY BOOKKEEPING REGARDING EACH          14008000 T  0586
                    BLOCK, AND SUPPLYING THE SCANNER WITH ALL NECESSARY INFORMATION      14009000 T  0586
                    ABOUT DECLARED IDENTIFIERS.                                         14010000 T  0586
                    IT ALSO WRITES EACH SEGMENT ONTO THE PCT;                           14011000 T  0586
                BEGIN                                                                   14012000 T  0586
                 LABEL OWNERR,SAVERR,BOOLEANDEC,REALDEC,ALPHADEC,INTEGERDEC,            14013000 T  0586
                                                                  START OF SEGMENT **********   94
                    LABELDEC,DUMPDEC,SUBDEC,OUTDEC,INDEC,MONITORDEC,                    14014000 T  0000
                    SWITCHDEC,PROCEDUREDEC,ARRAYDEC,NAMEDEC,FILEDEC,                    14015000 T  0000
                      GOTSCHK,                                                          14016000 T  0000
                       STREAMERR,DEFINEDEC,CALLSTATEMENT,HF,START;                      14017000 T  0000
                 SWITCH DECLSW := OWNERR,SAVERR,BOOLEANDEC,REALDEC,INTEGERDEC,ALPHADEC, 14018000 T  0000
                                  LABELDEC,DUMPDEC,SUBDEC,OUTDEC,INDEC,MONITORDEC,      14019000 T  0002
                                  SWITCHDEC,PROCEDUREDEC,ARRAYDEC,NAMEDEC,FILEDEC,      14020000 T  0002
                                 STREAMERR,DEFINEDEC;                                   14021000 T  0002
                DEFINE NLOCS=10#,LOCBEGIN=PRTI#,                                        14022000 T  0012
                       LBP=[36:12]#,                                                    14023000 T  0012
                       SPACEITDOWN = BEGIN   WRITE(LINE[DBL]); WRITE(LINE[DBL])  END#;  14023100 T  0012
                                                                                        14024000 T  0012
                BOOLEAN GOTSTORAGE;                                                     14025000 T  0012
STACK(F+2) := GOTSTORAGE
                 INTEGER PINFOO,BLKAD;                                                  14026000 T  0012
STACK(F+3) := PINFOO
STACK(F+4) := BLKAD
                      COMMENT LOCAL TO BLOCK TO SAVE WHERE A PROCEDURE IS EMTERED       14027000 T  0012
                      IN INFO;                                                          14028000 T  0012
                REAL MAXSTACKO,LASTINFOT,RELAD,LO,TSUBLEVEL,STACKCTRO;                  14029000 T  0012
STACK(F+5) := MAXSTACKO
STACK(F+6) := LASTINFOT
STACK(F+7) := RELAD
STACK(F+10) := LO
STACK(F+11) := TSUBLEVEL
STACK(F+12) := STACKCTRO
                  INTEGER SGNOO,LOLD,SAVELO,PRTIO,NINFOO;                               14030000 T  0012
STACK(F+13) := SGNOO
STACK(F+14) := LOLD
STACK(F+15) := SAVELO
STACK(F+16) := PRTIO
STACK(F+17) := NINFOO
                    INTEGER NCIIO;                                                      14031000 T  0012
STACK(F+20) := NCIIO
                     INTEGER PROAD ;                                                    14032000 T  0012
STACK(F+21) := PROAD
                      INTEGER FIRSTXO;                                                  14033000 T  0012
STACK(F+22) := FIRSTXO
                   BOOLEAN FUNCTOGO,AJUMPO;                                             14034000 T  0012
STACK(F+23) := FUNCTOGO
STACK(F+24) := AJUMPO
                        BEGINCTR:=BEGINCTR+1;                                           14035000 T  0012
                        IF SOP                                                          14036000 T  0014
                                    THEN BEGIN BLKAD:=PROADO;                           14037000 T  0014
                                  IF LASTENTRY ≠ 0                                      14038000 T  0015
                                     THEN BEGIN GT1:=BUMPL;                             14039000 T  0016
                                                CONSTANTCLEAN;                          14040000 T  0018
                                                EMITB(BFW,GT1,L)                        14041000 T  0019
                                           END                                          14042000 T  0020
                                END                                                     14043000 T  0020
```

```
                ELSE BEGIN BLKAD:=GETSPACE(TRUE,-6); % SEG. DESCR.                14044000 T  0020
                                                                                  14045000 T  0022
                                                                                  14046000 T  0022
                                                                                  14047000 T  0022
                     END;                                                         14048000 T  0022
                                                                                  14049000 T  0022
                                                                                  14050000 T  0022
       FIRSTXO←FIRSTX;                                                            14051000 T  0022
    FIRSTX←0;                                                                     14052000 T  0023
 LEVEL←LEVEL+1;                                                                   14053000 T  0024
    LOLD←L;FUNCTOGO←FUNCTOG;AJUMPO←AJUMP;PRTIO←PRTI;SGNOO←SGNO;                    14054000 T  0025
    SAVELO←SAVEL;AJUMP←FALSE; L←0;NINFOO←NEXTINFO;                                14055000 T  0029
  NCIIO←NCII;                                                                     14056000 T  0032
  NCII←0;                                                                         14057000 T  0033
  STACKCTRO←STACKCTR;                                                             14058000 T  0033
                                                                                  14059000 T  0034
                                                                                  14061000 T  0034
 ELBAT[I].CLASS←SEMICOLON;                                                        14062000 T  0034
START: IF TABLE(I)≠SEMICOLON                                                      14063000 T  0037
         THEN                                                                     14064000 T  0037
           BEGIN                                                                  14065000 T  0038
             FLAG(0);                                                             14066000 T  0038
             I←I-1                                                                14067000 T  0039
           END;                                                                   14068000 T  0039
       GTA1[0]←J←0;                                                               14069000 T  0040
       IF SPECTOG                                                                 14070000 T  0042
          THEN                                                                    14071000 T  0042
            BEGIN                                                                 14072000 T  0042
                                 IF BUP=PJ                                        14073000 T  0043
                                   THEN                                           14074000 T  0043
                                     BEGIN                                        14075000 T  0044
                                        BEGIN LABEL GETLP;                        14076000 T  0044
PRT(672) = *SEGMENT DESCRIPTOR*

                                                                          START OF SEGMENT **********    95
                                              IF STREAMTOG THEN F←0 ELSE          14077000 T  0000
                                       F←FZERO;                                   14078000 T  0001
                                     BUP←LASTINFO;                                14079000 T  0002
                                        DO                                        14080000 T  0003
                                         BEGIN                                    14081000 T  0004
                                     IF NOT STREAMTOG THEN                        14082000 T  0004
                                          BUP←LASTINFO;                           14083000 T  0004
                                          GETLP: G←TAKE(BUP);                     14084000 T  0005
                                                 IF K←G.ADDRESS≠PJ                14085000 T  0007
                                                   THEN                           14086000 T  0008
                                                     BEGIN                        14087000 T  0009
                  IF BUP ≠ BUP:=BUP^ TAKE(BUP + 1).PURPT THEN                     14088000 P  0009
                                                       GO TO GETLP                14089000 T  0012
                                                     END;                         14090000 T  0012
                   TYPEV←G.CLASS;                                                 14091000 T  0013
                                                        G.ADDRESS←F←F+1;          14115000 T  0014
                                                   PUT(G,BUP); G.INCR←GT1;        14116000 T  0017
                                                   PUT(G,MARK←PJ)                 14117000 T  0020
                                                  ;BUP←BUP^TAKE(BUP+1).PURPT      14118000 T  0020
                                         END                                      14119000 T  0023
                                       UNTIL PJ←PJ-1=0                            14120000 T  0024
                                     END;                                         14121000 T  0025
PRT(673) = *SEGMENT DESCRIPTOR*
```

```
                                SPECTOG←FALSE;
                          GO TO HF
                         END
                  END;
                    STACKCT ← 0;
          WHILE STEPI=DECLARATORS
          DO
           BEGIN
             GTA1[J←J+1]←ELBAT[I].ADDRESS;
             STORDEFINE←ERRORTOG←TRUE;
           END;
IF J =0 THEN GO TO CALLSTATEMENT;
          P2←P3←FALSE;
          GO TO DECLSW[GTA1[J]];
OWNERR:FLAG(20);J←J+1;GO TO REALDEC;
SAVERR:FLAG(21);J←J+1;GO TO REALDEC;
STREAMERR:     IF ELCLASS = LEFTPAREN THEN
               BEGIN
                    I ← I - 1;
                    GO TO CALLSTATEMENT;
               END;
                    FLAG(22);
                    J ← J + 1;
                    GO TO PROCEDUREDEC;
REALDEC:P3←TRUE;ENTER(REALID);GO TO START;
ALPHADEC:P3←TRUE;ENTER(ALFAID);GO TO START;
BOOLEANDEC:P3←TRUE;ENTER(BOOID);GO TO START;
INTEGERDEC:P3←TRUE;ENTER(INTID);GO TO START;
          MONITORDEC:IF SPECTOG
                     THEN BEGIN COMMENT ERROR 463 MEANS THAT A MONITOR
                                DECLARATION APPEARS IN THE SPECIFICATION
                                PART OF A PROCEDURE;
                          FLAG(463);
                         END;
                DO UNTIL FALSE;
          DUMPDEC:IF SPECTOG
                   THEN BEGIN COMMENT ERROR 464 MEANS A DUMP DECLARATION
                              APPEARS IN THE SPECIFICATION PART OF A
                              PROCEDURE;
                             FLAG(464);
                        END;
                DO UNTIL FALSE;
ARRAYDEC: ARRAE; GO TO START;
FILEDEC: INDEC: OUTDEC:
GOTSCHK;GOTSTORAGE← NOT SPECTOG OR GOTSTORAGE;GO TO START;
NAMEDEC: IF T1←GTA1[J←J-1]≠ARRAYV THEN J←J+1;
          TYPEV←NAMEID;
          IF T1←GTA1[J←J-1]=0 THEN J←J+1
                                  ELSE
                                IF T1=OWNV
                                  THEN
                                     BEGIN
                                       P2←TRUE; IF SPECTOG THEN
                                                  FLAG(013)
                                     END
                                  ELSE
```

```
95 IS   27 LONG, NEXT SEG   94
                                                  14122000 T  0045
                                                  14123000 T  0045
                                                  14124000 T  0046
                                                  14125000 T  0046
                                                  14125500 T  0046
                                                  14126000 T  0047
                                                  14127000 T  0047
                                                  14128000 T  0048
                                                  14129000 T  0048
                                                  14130000 T  0051
                                                  14131000 T  0052
                                                  14132000 T  0053
                                                  14133000 T  0054
                                                  14134000 T  0055
                                                  14135000 T  0058
                                                  14136000 T  0060
                                        % 6       14137000 T  0063
                                        % 6       14137100 T  0064
                                        % 6       14137200 T  0065
                                        % 6       14137300 T  0066
                                        % 6       14137400 T  0067
                                        % 6       14137500 T  0067
                                        % 6       14137600 T  0067
                                        % 6       14137700 T  0069
                                                  14138000 T  0069
                                                  14139000 T  0072
                                                  14140000 T  0074
                                                  14141000 T  0076
                                                  14142000 T  0078
                                                  14143000 T  0078
                                                  14144000 T  0078
                                                  14145000 T  0078
                                                  14146000 T  0078
                                                  14147000 T  0079
                                                  14148000 T  0079
                                                  14149000 T  0080
                                                  14150000 T  0081
                                                  14151000 T  0081
                                                  14152000 T  0081
                                                  14153000 T  0081
                                                  14154000 T  0082
                                                  14155000 T  0082
                                                  14156000 T  0083
                                                  14158000 T  0085
                                                  14160000 T  0085
                                                  14161000 T  0087
                                                  14161010 T  0091
                                                  14161020 T  0092
                                                  14161030 T  0095
                                                  14161040 T  0096
                                                  14161050 T  0097
                                                  14161060 T  0097
                                                  14161070 T  0098
                                                  14161080 T  0099
                                                  14161090 T  0099
                                                  14161100 T  0100
```

```
                                                    TYPEV←NAMEID←T1=REALV;
                          ENTER(TYPEV); GO TO START;
                SUBDEC:
                    BEGIN REAL TYPEV,T;
PRT(674) = *SEGMENT DESCRIPTOR*

STACK(F+25) = TYPEV
STACK(F+26) = T
                     IF GTA1[J←J-1]=REALV THEN TYPEV←REALSUBID ELSE TYPEV←SUBID;
                STOPGSP←TRUE;
                    JUMPCHKNX;ENTRY(TYPEV);IF ELCLASS≠SEMICOLON THEN FLAG(57);
                STOPGSP←FALSE;
                     STEPIT;
                          T←NEXTINFO;
                PUTNBUMP(L); STMT; EMITO(LFU);    IF TYPEV=REALSUBID THEN
                          IF GET(L-2)≠533 THEN FLAG(58);PUT(TAKE(T)&L[24:36:12],T);
                CONSTANTCLEAN;
                    END;
PRT(675) = *SEGMENT DESCRIPTOR*

                     GO TO START;
```

```
            LABELDEC:IF SPECTOG AND FUNCTOG THEN FLAG(24);
                     STOPENTRY←STOPGSP←TRUE;
                     I←I-1;
                     DO
                       BEGIN
                          STOPDEFINE←TRUE;
                        STEPIT;
                          ENTRY(LABELID);
                          PUTNBUMP(0)
                       END
                     UNTIL ELCLASS≠COMMA;
                     STOPENTRY←STOPGSP←FALSE;
                     GO TO START;
            SWITCHDEC:
              BEGIN
                LABEL START;
PRT(676) = *SEGMENT DESCRIPTOR*

                    INTEGER GT1,GT2,GT4,GT5;
```

```
14161110 T  0100
14161120 T  0100
14162000 T  0102
14163000 T  0103
14163500 T  0104

START OF SEGMENT **********   96

14164000 T  0000
14164500 T  0004
14165000 T  0005
14166    T  0008
14166000 T  0009
14166500 T  0009
14167000 T  0010
14168000 T  0013
14168500 T  0019
14169000 T  0019

 96 IS   21 LONG, NEXT SEG   94
14170000 T  0105
14171000 T  0105
14172000 T  0105
14173000 T  0105
14174000 T  0105
14175000 T  0105
14176000 T  0105
14177000 T  0105
14178000 T  0105
14179000 T  0105
14180000 T  0105
14181000 T  0105
14182000 T  0105
14183000 T  0105
14184000 T  0105
14185000 T  0105
14186000 T  0105
14187000 T  0105
14188000 T  0108
14189000 T  0109
14190000 T  0110
14191000 T  0111
14192000 T  0111
14193000 T  0111
14194000 T  0112
14195000 T  0113
14196000 T  0113
14197000 T  0113
14198000 T  0115
14199000 T  0116
14200000 T  0116
14201000 T  0117
14202000 T  0117

START OF SEGMENT **********   97
14203000 T  0000
```

```
STACK(F+25) = GT1
STACK(F+26) = GT2
STACK(F+27) = GT4
STACK(F+30) = GT5
                    BOOLEAN TB1;
STACK(F+31) = TB1
                    STOPENTRY←NOT SPECTOG;STOPGSP←TRUE;
                   SCATTERELBAT; GT1←0; TB1←FALSE;
                      ENTRY(SWITCHID);
                 GT2←NEXTINFO; PUTNBUMP(0);
                   DO
                      BEGIN
                         IF STEPI≠LABELID OR ELBAT[I].LVL≠LEVEL THEN FLAG(63);
                         PUTNBUMP(ELBAT[I]);GT1←GT1+1
                      END
                              COMMENT
                   UNTIL STEPI≠COMMA;

                      PUT(GT1,GT2);
                       STOPENTRY ← STOPGSP ← FALSE;
                  END SWITCHDEC;
PRT(677) = *SEGMENT DESCRIPTOR*

                GO TO START;
                  DEFINEDEC:
                    BEGIN LABEL START;
PRT(700) = *SEGMENT DESCRIPTOR*

                        REAL J,K;
STACK(F+25) = J
STACK(F+26) = K
                BOOLEAN STREAM PROCEDURE PARM(S,D,K,J); VALUE K,J;

PRT(701) = PARM
     BEGIN SI←S;SI←SI+2; DI←D;DI←DI+2;
          IF K SC≠DC THEN TALLY←1;
          DI←LOC J;DI←DI+7;
          IF SC≠DC THEN TALLY←1;
          PARM←TALLY;
     END;



STOPENTRY←STOPGSP←TRUE;I←I-1;
 DO
  BEGIN
          STOPDEFINE←TRUE;
     STEPIT; MOVE(9,ACCUM[1],GTA1);
     K←COUNT+1; J←GTA1[0]; ENTRY(DEFINEDID);
     GTA1[0]←J←"100000"; J←0;
     IF ELCLASS=LEFTPAREN OR ELCLASS=LFTBRKET THEN
     BEGIN
          DO BEGIN STOPDEFINE←TRUE;
               STEPIT;
               IF (J←J+1)>9 OR PARM(ACCUM[1],GTA1,K,J) OR
                    K>62 THEN BEGIN ERR(141); GO TO START END;
               STOPDEFINE←TRUE;
```

```
                                                  14204000 T  0000

                                                  14205000 T  0000
                                                  14206000 T  0001
                                                  14207000 T  0003
                                                  14217000 T  0004
                                                  14218000 T  0006
                                                  14219000 T  0006
                                                  14220000 T  0006
                                                  14221000 T  0010
                                                  14222000 T  0011
                                                  14222500 T  0012
                                                  14223000 T  0012
                                                  14223500 T  0013
                                                  14224000 T  0013
                                                  14251000 T  0014
                                                  14252000 T  0016

                                          97 IS   17 LONG, NEXT SEG   94
                                                  14253000 T  0118
                                                  14254000 T  0118
                                                  14254050 T  0119

                                        START OF SEGMENT **********   98
                                                  14254100 T  0000

                                                  14254200 T  0000

                                                  14254300 T  0000
                                                  14254400 T  0001
                                                  14254500 T  0002
                                                  14254600 T  0002
                                                  14254700 T  0003
                                                  14254800 T  0003



                                                  14255000 T  0004
                                                  14256000 T  0007
                                                  14257000 T  0008
                                                  14258000 T  0008
                                                  14259000 T  0008
                                                  14259010 T  0011
                                                  14259015 T  0014
                                                  14259020 T  0016
                                                  14259030 T  0018
                                                  14259060 T  0018
                                                  14259070 T  0019
                                                  14259080 T  0020
                                                  14259090 T  0024
                                                  14259100 T  0029
```

```
                        END UNTIL STEPI≠COMMA;                                                                  14259110 T  0029
                        IF ELCLASS≠RTPAREN AND ELCLASS≠RTBRKET THEN ERR(141);                                   14259120 T  0031
                        STOPDEFINE←TRUE;                                                                        14259130 T  0034
                        STEPIT;                                                                                 14259140 T  0035
                        PUT(TAKE(LASTINFO)&J[16:37:11],LASTINFO);                                               14259150 T  0035
                    END;                                                                                        14259160 T  0038
                   IF ELCLASS≠RELOP                                                                             14260000 T  0038
                     THEN                                                                                       14261000 T  0038
                        BEGIN                                                                                   14262000 T  0038
                          FLAG(30);                                                                             14263000 T  0039
                          I←I-1;                                                                                14264000 T  0040
                        END;                                                                                    14265000 T  0041
                    MACROID←TRUE;                                                                               14265900 T  0041
                    DEFINEGEN(FALSE,J);                                                                         14266000 T  0042
                    MACROID←FALSE;                                                                              14266100 T  0043
                  END                                                                                           14267000 T  0043
                 UNTIL STEPI≠COMMA;                                                                             14268000 T  0043
              START: STOPENTRY←STOPGSP←FALSE; END; GO TO START;                                                 14269000 T  0045
PRT(702) = *SEGMENT DESCRIPTOR*                                                                                 98 IS   48 LONG, NEXT SEG   94
                                                                                                                        T  0120
            PROCEDUREDEC;                                                                                       14270000 T  0120
              BEGIN                                                                                             14271000 T  0121
                LABEL START,START1;                                                                             14272000 T  0121
PRT(703) = *SEGMENT DESCRIPTOR*                                                                                 START OF SEGMENT **********   99
                LABEL START2;                                                                                   14273000 T  0000
               BOOLEAN FWDTOG; COMMENT THIS TOGGLE IS THE FORWARD DEC INDICATOR;                                14274000 T  0000
STACK(F+25) = FWDTOG
                   IF NOT SPECTOG THEN FUNCTOG←FALSE;                                                           14275000 T  0000
              FWDTOG←FALSE  ;                                                                                   14276000 T  0001
                       MAXSTACKO← MAXSTACK;                                                                     14277000 T  0002
              IF G←GTA1[J←J-1]≠STREAMV                                                                          14278000 T  0003
               THEN                                                                                             14279000 T  0005
                 BEGIN STREAMTUG←TRUE;                                                                          14280000 T  0005
                   IF G←GTA1[J←J-1]=0 THEN TYPEV←STRPROCID                                                      14281000 T  0007
                                      ELSE                                                                      14282000 T  0010
                                       BEGIN                                                                    14283000 T  0010
                                        IF TYPEV←PROCID +G>INTSTRPROCID OR                                      14284000 T  0011
                                           TYPEV <BOOSTRPROCID                                                  14285000 T  0013
                                         THEN FLAG(004);                                                        14286000 T  0013
                                    IF NOT SPECTOG THEN                                                         14287000 T  0015
                                         FUNCTOG←TRUE;                                                          14288000 T  0015
                                         CHKSOB                                                                 14289000 T  0017
                  END                    END                                                                    14290000 T  0017
                  ELSE                                                                                          14291000 T  0017
                IF G=SAVEV OR G=0 THEN TYPEV←PROCID                                                             14292000 T  0017
                           ELSE                                                                                 14293000 T  0020
                            IF TYPEV←REALSTRPROCID+G<BOOPROCID OR TYPEV>INTPROCID                               14294000 T  0021
                              THEN FLAG(005)                                                                    14295000 T  0023
                       ELSE BEGIN FUNCTOG←TRUE;G←GTA1[J←J-1];                                                   14296000 T  0025
                                  END;                                                                          14297000 T  0028
                IF NOT STREAMTUG THEN SEGMENTSTART(G=SAVEV);                                                    14298000 T  0028
                SAF ← G=SAVEV;                                                                                  14299000 T  0031
                                                                                                                14300000 T  0032
                                                                                                                14301000 T  0032
                                                                                                                14302000 T  0032
```

```
    MODE←MODE+1;                                                                14303000 T  0032
    LO←PROINFO;                                                                 14304000 T  0033
    SCATTERELBAT;                                                               14305000 T  0034
COMMENT CHECK TO SEE IF DECLARED FORWARD PREVIOUSLY                      ;      14306000 T  0034
    IF LEVELF=LEVEL                                                             14307000 T  0034
      THEN                                                                      14308000 T  0035
        BEGIN                                                                   14309000 T  0035
          IF G←TAKE(LINKF+1)≥0                                                  14310000 T  0036
            THEN FLAG(006);                                                     14311000 T  0037
          FWDTOG←TRUE;                                                          14312000 T  0039
          PROAD←ADDRSF;                                                         14313000 T  0040
          PROINFO←ELBAT[I];MARK←LINKF+INCRF;STEPIT                              14314000 T  0041
               ;PUT(=G,LINKF+1);                                                14315000 T  0043
        END                                                                     14316000 T  0045
      ELSE                                                                      14317000 T  0045
        BEGIN STOPENTRY←TRUE; P2←TRUE;                                          14318000 T  0045
          STOPGSP←LEVEL>1 AND STREAMTOG;                                        14318500 T  0047
          ENTRY(TYPEV);     MARK←NEXTINFO;PUTNBUMP(0);                          14319000 T  0049
          STOPGSP←FALSE;                                                        14319500 T  0051
          PROINFO←TAKE(LASTINFO)& LASTINFO[35:35:13];PROAD←ADDRSF;               14320000 T  0052
       P2←STOPENTRY←FALSE                                                       14321000 T  0055
        END;                                                                    14322000 T  0055
                  PJ←0; LEVEL←LEVEL+1;                                          14323000 T  0056
         IF STREAMTOG THEN STREAMWORDS;                                         14324000 T  0058
       IF ELCLASS=SEMICOLON THEN GO TO START1;                                  14325000 T  0059
      IF ELCLASS≠LEFTPAREN THEN FLAG(007);                                      14326000 T  0061
COMMENT: THE FOLLOWING 8  STATEMENTS FOOL THE SCANNER AND BLOCK,PUTTING         14327000 T  0063
          FORMAL PARAMETER ENTRIES IN THE ZERO ROW OF INFO;                     14328000 T  0063
        RR1←NEXTINFO;                                                           14329000 T  0063
       LASTINFOT←LASTINFO; LASTINFO←NEXTINFO←1;                                 14330000 T  0063
                 PUTNBUMP(0);                                                   14331000 T  0065
               PTOG←TRUE;    I←I+1;                                           14332000 T  0066
      ENTRY(SECRET);                                                            14333000 T  0068
    IF FWDTOG THEN                                                              14333100 T  0069
          BEGIN                                                                 14333200 T  0069
          IF GT1:=TAKE(MARK).[40:8] ≠ PJ THEN FLAG(48); % WRONG                 14333300 T  0070
            % NUMBER OF PARAMETERS. WE DON"T WANT TO CLOBBER INFO.              14333400 T  0073
          END                                                                   14333500 T  0073
ELSE                                                                            14333600 T  0073
  PUT(PJ,MARK);                                                                 14334000 T  0073
            P←PJ;                                                               14335000 T  0075
    IF ELCLASS≠RTPAREN                                                          14336000 T  0075
      THEN FLAG(008);                                                           14337000 T  0076
    IF STEPI≠SEMICOLON                                                          14338000 T  0077
       THEN FLAG(009);                                                          14339000 T  0078
COMMENT MARK PARAMETERS VALUE IF THERE IS A VALUE PART;                         14340000 T  0080
   IF STEPI=VALUEV                                                              14341000 T  0080
      THEN                                                                      14342000 T  0080
        BEGIN                                                                   14343000 T  0081
          DO                                                                    14344000 T  0081
            IF STEPI≠SECRET                                                     14345000 T  0082
              THEN FLAG(010)                                                    14346000 T  0082
              ELSE                                                              14347000 T  0083
                 BEGIN                                                          14348000 T  0084
                    IF G←ELBAT[I].ADDRESS=0 OR G>PJ                             14349000 T  0084
                     THEN                                                       14350000 T  0087
                        FLAG(010);                                              14351000 T  0087
```

```
                                        G←TAKE(ELBAT[I]);                                    14352000 T  0089
                                        PUT(G&1[10:47:1],ELBAT[I])                           14353000 T  0090
                        END                                                                  14354000 T  0092
                  UNTIL                                                                      14355000 T  0092
                     STEPI≠COMMA;                                                            14356000 T  0092
               IF ELCLASS≠SEMICOLON                                                          14357000 T  0094
                  THEN FLAG(011)                                                             14358000 T  0094
                  ELSE STEPIT                                                                14359000 T  0095
                    END;I←I-1;                                                               14360000 T  0096
                IF STREAMTOG                                                                 14361000 T  0098
                  THEN                                                                       14362000 T  0098
                    BEGIN                                                                    14363000 T  0098
                              BUP←PJ; SPECTOG←TRUE;GO TO START1                              14364000 T  0099
                    END                                                                      14365000 T  0101
                  ELSE                                                                       14366000 T  0101
                    BEGIN                                                                    14367000 T  0101
                      SPECTOG←TRUE;                                                          14368000 T  0101
                      BUP←0;                                                                 14369000 T  0102
                      IF ELCLASS≠DECLARATORS                                                 14370000 T  0103
                         THEN FLAG(012)                                                      14371000 T  0103
                    END;                                                                     14372000 T  0104
           START:PTOG←FALSE;LASTINFO←LASTINFOT;NEXTINFO←IF FWDTOG THEN RR1 ELSE               14373000 T  0105
                      MARK←PJ+1;                                                             14374000 T  0109
           START1:PINFOO←NEXTINFO;                                                           14375000 T  0110
           START2: END;                                                                      14376000 T  0111
PRT(704) = *SEGMENT DESCRIPTOR*                                                  99 IS  113 LONG, NEXT SEG   94
                IF SPECTOG OR STREAMTOG                                                      14377000 T  0122
                   THEN                                                                      14378000 T  0122
                        GO TO START;                                                         14379000 T  0122
           COMMENT IF SPECTOG IS ON THEN THE BLOCK WILL PROCESS THE SPECIFICATION             14380000 T  0123
                   PART SIMILARY TO DECLARATIONS WITH A FEW NECESSARY VARIATIONS;             14381000 T  0123
           HF:                                                                               14382000 T  0123
              BEGIN                                                                          14383000 T  0124
                LABEL START,STOP;                                                            14384000 T  0124
PRT(705) = *SEGMENT DESCRIPTOR*                                                  START OF SEGMENT **********  100
                    DEFINE  TESTLEV = LEVEL>2 #;                                             14384100 T  0000
             IF STREAMTOG                                                                    14385000 T  0000
               THEN BEGIN                                                                    14386000 T  0000
                IF TESTLEV THEN JUMPCHKNX ELSE SEGMENTSTART(TRUE);PJ←P;                      14387000 T  0000
                           PTOG←FALSE;                                                       14388000 T  0004
                    PUT(TAKE(GIT(PROINFO))&L[28:36:12],GIT(PROINFO));                        14388100 T  0005
               IF TESTLEV THEN BEGIN EMITO(584);          END;                               14389000 T  0008
                       IF STEPI=BEGINV                                                       14393000 T  0010
                         THEN                                                                14394000 T  0011
                            BEGIN                                                            14395000 T  0011
                              WHILE STEPI=DECLARATORS OR ELCLASS=LOCALV                       14396000 T  0012
                                 DO                                                          14397000 T  0014
                                  BEGIN                                                      14398000 T  0015
                                    IF ELBAT[I].ADDRESS=LABELV                               14399000 T  0015
                                     THEN                                                    14400000 T  0016
                                        BEGIN                                                14401000 T  0017
                                          STOPDEFINE←STOPGSP←STOPENTRY←TRUE;                 14402000 T  0017
             DO BEGIN STOPDEFINE←TRUE;STEPIT;ENTRY(STLABID);PUTNBUMP(0) END UNTIL             14403000 T  0019
                        ELCLASS≠COMMA;STOPGSP←STOPENTRY←FALSE                                14404000 T  0022
                                        END                                                  14405000 T  0024
```

```
                                        ELSE                                                                    14406000 T  0025
                                          BEGIN                                                                 14407000 T  0025
                                                  I←I+1;                                                        14408000 T  0025
                                            ENTRY(LOCLID)                                                       14409000 T  0027
                                          END                                                                   14410000 T  0027
                                    END;                                                                        14411000 T  0027
                          IF FUNCTOG  THEN                                                                      14411100 T  0028
                               PUT((Z←TAKE(PROINFO))&LOCLID[2:41:7] &                                           14411200 T  0028
                                        (PJ+2←REAL(TESTLEV))[16:37:11],PROINFO);                                14411300 T  0031
                                   COMPOUNDTAIL                                                                 14412000 T  0034
                              END                                                                               14413000 T  0034
                            ELSE                                                                                14414000 T  0035
                          BEGIN                                                                                 14415000 T  0035
                          IF   FUNCTOG  THEN                                                                    14415100 T  0035
                               PUT(( Z←TAKE(PROINFO))& LOCLID[2:41:7]&                                          14415200 T  0035
                                        (PJ+2←REAL(TESTLEV))[16:37:11],PROINFO);                                14415300 T  0038
                          STREAMSTMT;                                                                           14415400 T  0041
                          END;                                                                                  14415500 T  0042
                     COMMENT THE FOLLOWING BLOCK CONSTITUTES THE STREAM PROCEDURE PURGE;                        14416000 T  0042
                          BEGIN                                                                                 14417000 T  0042
                          REAL NLOC,NLAB;                                                                       14418000 T  0042
PRT(706) = *SEGMENT DESCRIPTOR*

                                                                                                  START OF SEGMENT **********  101
STACK(F+25) = NLOC
STACK(F+26) = NLAB
                          DEFINE SES=18#,SED=6#,TRW=5#;                                                         14419000 T  0000
                          DEFINE LOC=[36:12]#,LASTGT=[24:12]#;                                                  14420000 T  0000
                          J←  LASTINFO;                                                                         14421000 T  0000
                               NLOC←NLAB←0;                                                                     14422000 T  0000
                          DO                                                                                    14423000 T  0002
                          BEGIN                                                                                 14424000 T  0002
                          IF(GT1←TAKE(J)).CLASS=LUCLID THEN                                                     14425000 T  0002
                               BEGIN                                                                            14426000 T  0004
                               IF BOOLEAN(GT1,FORMAL) THEN                                                      14427000 T  0004
                                    BEGIN                                                                       14428000 T  0005
                                    IF GT1<0 THEN                                                               14429000 T  0006
                                    PUT(TAKE(GT2←MARK←P=GT1.ADDRESS+1)&FILEID[2:41:7]                           14430000 T  0006
                                         ,GT2);                                                                 14431000 T  0011
                                    END                                                                         14432000 T  0012
                               ELSE  NLOC←NLOC+1;                                                               14433000 T  0012
                               END                                                                              14434000 T  0014
                          ELSE                                                                                  14435000 T  0014
                               BEGIN                                                                            14436000 T  0014
                               IF  GT1.ADDRESS≠0 THEN NLAB←NLAB+1;                                              14437000 T  0014
                               IF(GT3←TAKE(GIT(J))).LASTGT≠0 AND GT3.LOC =0 THEN                                14438000 T  0017
                                    BEGIN                                                                       14439000 T  0021
                                    MOVE(9,INFO[0,J],ACCUM[0]);                                                 14440000 T  0022
                                    Q←ACCUM[1];                                                                 14441000 T  0024
                                    FLAG(267);                                                                  14442000 T  0025
                                    ERRORTOG←TRUE;                                                              14443000 T  0026
                                    END;                                                                        14444000 T  0027
                               END;                                                                             14445000 T  0027
                          G←(GT2←TAKE(J+1)).PURPT;                                                              14446000 T  0027
                          IF  GT1.[2:8] ≠ STLABID×2+1  THEN                                                     14447000 T  0030
                               STACKHEAD[(0&GT2[12:12:36])MOD 125]←TAKE(J).LINK;                                14448000 T  0032
                          END UNTIL J←J-G≤1;                                                                    14449000 T  0036
                                                                                                                14450000 T  0038
```

```
                    IF   TESTLEV   THEN BEGIN EMITC(1,0); EMITO(BFW) END                    14451000 T  0038
                                   ELSE  EMIT(0);                                           14451100 T  0041
               PUT(TAKE(MARK)&NLOC[1:42:6]&L[16:36:12]&P[40:40:8],MARK);                    14451200 T  0043
                    IF FUNCTOG THEN                                                         14452000 T  0047
                         PUT(Z, PROINFO);                                                   14457000 T  0047
                    STREAMWORDS;                                                            14460000 T  0049
                    STREAMTOG←FALSE;                                                        14461000 T  0049
                IF NOT TESTLEV THEN BEGIN PROGDESCBLDR(PROAD,TRUE,(L+3)DIV 4,CHAR);         14461100 T  0050
                                          SEGMENT((L+3)DIV 4,PROINFO.ADDRESS);              14461200 T  0054
                                               RIGHT(L); L←0;                               14461300 T  0056
                                   END;                                                     14461400 T  0058
                    IF LISTER AND FORMATOG THEN SPACEITDOWN;                                14461500 T  0058
                    END;                                                                    14462000 T  0068
PRT(707) = *SEGMENT DESCRIPTOR*
                                                                    101 IS   69 LONG, NEXT SEG  100
                        LASTINFO←LASTINFOT;NEXTINFO←MARK+P+1;                               14463000 T  0043
                    END                                                                     14464000 T  0045
                ELSE                                                                        14465000 T  0045
                    BEGIN                                                                   14466000 T  0045
                 IF STEPI=FORWARDV                                                          14467000 T  0046
                   THEN                                                                     14468000 T  0046
                    BEGIN                                                                   14469000 T  0047
                      PUT(-TAKE(G←PROINFO,LINK+1),G);                                       14470000 T  0047
                           PURGE(PINFOO);                                                   14471000 T  0050
                      STEPIT                                                                14472000 T  0051
                    END                                                                     14473000 T  0051
                   ELSE                                                                     14474000 T  0052
                    BEGIN                                                                   14475000 T  0052
                 PROADO←PROAD;                                                              14476000 T  0052
                  TSUBLEVEL←SUBLEVEL;SUBLEVEL←LEVEL ;STACKCTRO←STACKCTR;                    14477000 T  0053
    IF MODE=1 THEN FRSTLEVEL←LEVEL;STACKCTR←513+REAL(FUNCTOG);                              14478000 T  0055
          IF ELCLASS = BEGINV THEN                                                          14479000 T  0058
     BEGIN                                                                                  14481000 T  0059
            CALLINFO←(CALLX←CALLX+1)+1;                                                     14481100 T  0060
            NESTCTR←STACKCTR;                                                               14481200 T  0062
              BLOCK(TRUE)                                                                   14482000 T  0063
            ; PURGE(PINFOO);                                                                14483000 T  0063
            IF NESTOG THEN                                                                  14483100 T  0064
            BEGIN GT1←TAKE(PROINFO).ADDRESS;                                                14483200 T  0064
                 NESTPRT[GT1]←0&PROINFO[35:35:13]&CALLINFO[22:35:13];                       14483300 T  0067
                 CALL[CALLINFO-1]←(TAKE(GIT(PROINFO))+NESTCTR-511)&                         14483400 T  0070
                      CALLX[22:35:13];                                                      14483500 T  0076
            END;                                                                            14483600 T  0077
            L←0;                                                                            14483700 T  0077
              GO TO STOP END;                                                               14484000 T  0078
               BEGIN                                                                        14485000 T  0079
FLAG(052);                                                                                  14486000 T  0079
                RELAD←L  ;                                                                  14487000 T  0079
         STMT;                                                                              14488000 T  0080
              HTTEOAP(FALSE,RELAD,PINFOO,PROAD);                                            14489000 T  0081
               END;                                                                         14490000 T  0082
  STOP;                                                                                     14491000 T  0082
     SUBLEVEL←TSUBLEVEL;                                                                    14492000 T  0083
     STACKCTR←STACKCTRO;                                                                    14493000 T  0083
     IF LISTER AND FORMATOG THEN SPACEITDOWN;                                               14493500 T  0084
       END;                                                                                 14494000 T  0094
       END;                                                                                 14495000 T  0094
```

```
                    PROINFO←LO;
                    IF JUMPCTR=LEVEL
                       THEN
                              JUMPCTR←LEVEL-1;
                          LEVEL←LEVEL-1;
                         MODE←MODE-1;
                           MAXSTACK←MAXSTACKO;
                 START!END;
PRT(710) = *SEGMENT DESCRIPTOR*

                       GO TO START;
                CALLSTATEMENT! FOULED ← L;
                    JUMPCHKX;IF SOP THEN BEGIN Z←STACKCTR-513;WHILE Z←Z-120
                                                 DO EMITL(0)   END;
                          IF SPECTOG THEN BEGIN
                                            FLAG(12);GO TO HF
                                          END;
                          BEGINCTR ← BEGINCTR-1;
                          IF ERRORTOG
                            THEN COMPOUNDTAIL
                            ELSE
                              BEGIN
                               STMT;
                               IF ELCLASS←TABLE(I+1)=DECLARATORS
                                 THEN
                                   BEGIN
                                     ELBAT[I].CLASS←SEMICOLON;
                                     BEGINCTR←BEGINCTR+1;
                                     GO TO START
                                   END
                                 ELSE
                                   COMPOUNDTAIL
                              END;
                    FUNCTOG←FUNCTOGO;
                    IF SOP THEN HTTEOAP(FALSE,FIRSTX,NINFOO,BLKAD)
                        ELSE BEGIN IF NESTOG THEN SORTNEST; PURGE(NINFOO); END;
                    SEGMENT((L+3)DIV 4,PROADO);
                        IF LEVEL>1 THEN RIGHT(L);
                   IF LEVEL ← LEVEL-1 = 0 THEN CONSTANTCLEAN;

                   AJUMP←AJUMPO;

                      FIRSTX←FIRSTXO;
                   SAVEL←SAVELO;
                   STACKCTR←STACKCTRO;


END BLOCK;
```

```
14496000 T  0094
14497000 T  0095
14498000 T  0095
14499000 T  0095
14500000 T  0097
14501000 T  0098
14502000 T  0100
14503000 T  0100

100 IS  102 LONG, NEXT SEG    94
14504000 T  0125
14505000 T  0125
14506000 T  0126
14506500 T  0131
14507000 T  0133
14508000 T  0134
14509000 T  0135
14510000 T  0135
14511000 T  0136
14512000 T  0136
14513000 T  0137
14514000 T  0138
14515000 T  0138
14516000 T  0139
14517000 T  0140
14518000 T  0141
14519000 T  0141
14520000 T  0144
14521000 T  0145
14522000 T  0146
14523000 T  0146
14524000 T  0146
14525000 T  0146
14599000 T  0147
14600000 T  0147
14601000 T  0149
14602000 T  0152
14603000 T  0154
14604000 T  0156
14605000 T  0159
14606000 T  0159
14607000 T  0160
14608000 T  0160
14609000 T  0160
14610000 T  0161
14611000 T  0162
14612000 T  0162
14613000 T  0162
94 IS  170 LONG, NEXT SEG     3
```

```
COMMENT THIS SECTIUN CONTAINS THE VARIABLE ROUTINE AND ITS SIDEKICKS;
```

```
15000000 T  0586
15001000 T  0586
15002000 T  0586
15003000 T  0586
```

```
COMMENT  THE FOLLOWING BLOCK HANDLES THE FOLLOWING CASES
         OF SIMPLE VARIABLES:
            1. V ← EXP          ,WHERE V IS FORMAL-CALL BY NAME.
            2. V ← EXP          ,ALL V EXCEPT FORMAL-NAME.
            3. V.[S:L] ← EXP    ,WHERE V IS FORMAL-CALL BY NAME.
            4. V.[S:L] ← EXP    ,ALL V EXCEPT FORMAL-NAME.
            5. V.[S:L]          ,ALL V.
            6. V                ,ALL V.
         CODE EMITED FOR THE ABOVE CASES IS AS FOLLOWS:
            1. VN,EXP,M*,XCH,←.
            2. EXP,M*,VL,←.
            3. VN,DUP,COC,EXP,T,M*,XCH,←.
            4. VV,EXP,T,M*,VL,←.
            5. ZEROL,VV,T .
            6. VV .
            WHERE VN = DESC V
                  EXP= ARITH. OR BOOLEAN EXPRESSION,AS REQUIRED.
                  M* = CALL ON MONITOR ROUTINE,IF REQUIRED.
                  VL = LITC V
                  VV = OPDC V
                  ←  = STORE INSTRUCTION(ISD,ISN,SND OR STD).
                  T  = BIT TRANSFER CODE(DIA,DIB,TRB).
                  ZEROL = LITC 0
                  DUP,COC,XCH = THE INSTRUCTIONS DUP,COC,AND XCH.
```

```
15004000 T  0586
15005000 T  0586
15006000 T  0586
15007000 T  0586
15008000 T  0586
15009000 T  0586
15012000 T  0586
15013000 T  0586
15014000 T  0586
15015000 T  0586
15016000 T  0586
15017000 T  0586
15018000 T  0586
15019000 T  0586
15020000 T  0586
15021000 T  0586
15022000 T  0586
15023000 T  0586
15024000 T  0586
15025000 T  0586
15026000 T  0586
15027000 T  0586
15028000 T  0586
15029000 T  0586
15030000 T  0586
15031000 T  0586
15032000 T  0586
15033000 T  0586
15034000 T  0586
15035000 T  0586
15036000 T  0586
15037000 T  0586
15038000 T  0586
15039000 T  0586
15040000 T  0586
15041000 T  0586
15042000 T  0586
15043000 T  0586
15044000 T  0586
15045000 T  0586
15046000 T  0586
15047000 T  0586
15048000 T  0586
15049000 T  0586
15050000 T  0586
15051000 T  0586
15052000 T  0586
15053000 T  0586
15054000 T  0586
15055000 T  0586
15056000 T  0586
15057000 T  0586
15058000 T  0586
15059000 T  0586
15060000 T  0586
15061000 T  0586
15062000 T  0586
```

```
                        OF COURSE, EXP WILL CAUSE RECURSION,IN GENERAL,AND THUS          15063000 T  0586
                        THE PARAMETER P1 AND THE LOCALS CAN NOT BE HANDLED IN A          15064000 T  0586
                        GLOBAL FASHION.                                                  15065000 T  0586
                        THE PARAMETER P1 IS USED TO TELL THE VARIABLE ROUTINE            15066000 T  0586
                        WHO CALLED IT.  SOME OF THE CODE GENERATION AND SOME             15067000 T  0586
                        SYNTAX CHECKS DEPEND UPON A PARTICULAR VALUE OF P1 .             15068000 T  0586
                                                                                         15069000 T  0586
                PROCEDURE VARIABLE(P1); INTEGER P1;                                      15070000 T  0586
                        BEGIN                                                            15071000 T  0586
                        REAL TALL,  COMMENT ELBAT WORD FOR VARIABLE;                     15072000 T  0586
                                                                        START OF SEGMENT ********** 102
STACK(F+2) = TALL
                             T1  ,  COMMENT 1ST INTEGER OF PARTIAL WORD SYNTAX;          15073000 T  0000
STACK(F+3) = T1
                             T2  ,  COMMENT 2ND INTEGER OF PARTIAL WORD SYNTAX;          15074000 T  0000
STACK(F+4) = T2
                             J   ;  COMMENT SUBSCRIPT COUNTER ;                          15075000 T  0000
STACK(F+5) = J
                        LABEL EXIT,L1,LAST,NEXT,JAZZ,ITUP,LASS;                          15076000 T  0000
                        DEFINE FORMALNAME=[9:2]=2#, LONGID=NAMEID#;                      15076100 T  0000
                        BOOLEAN SPCLMON;                                                 15076200 T  0000
STACK(F+6) = SPCLMON
                        TALL←ELBAT[I] ;                                                  15077000 T  0000
                        IF   ELCLASS ≤ INTPROCID  THEN                                   15078000 T  0001
                           BEGIN                                                         15079000 T  0001
                             IF TALL.LINK ≠PROINFO.LINK THEN                             15080000 T  0002
                                  BEGIN ERR(211); GO TO EXIT END;                        15081000 T  0004
                COMMENT  211  VARIABLE←FUNCTION IDENTIFIER USED OUTSIDE OF ITS SCOPE*;   15082000 T  0005
                        TALL←TALL &(ELCLASS+4)[2:41:7] & 513[16:37:11];                  15083000 T  0005
                           END                                                           15084000 T  0009
                        ELSE  CHECKER(TALL);                                             15085000 T  0009
                        IF TALL.CLASS ≤ INTID  THEN                                      15086000 T  0010
                   BEGIN                                                                 15087000 T  0011
                                                                                         15088000 T  0012
                                                                                         15089000 T  0012
                        IF  STEPI= ASSIGNOP THEN                                         15090000 T  0012
                        BEGIN STACKCT ← 1;                                               15091000 T  0013
                L1:          IF  TALL.FORMALNAME  THEN                                   15092000 T  0014
                             BEGIN                                                       15093000 T  0016
                             EMITN(TALL.ADDRESS);                                        15094000 T  0016
                             IF T1≠0 THEN BEGIN EMITO(DUP);EMITU(COC) END;               15095000 T  0018
                             END                                                         15096000 T  0020
                             ELSE IF T1≠0 THEN EMITV(TALL.ADDRESS)                       15097000 T  0020
                        ; STACKCT ← REAL(T1≠0); STEPIT;                                  15098000 T  0022
                        AEXP;                                                            15099000 T  0025
                             EMITO(48-T2 ,T1 ,T2);                                       15100000 T  0026
                                                                                         15101000 T  0027
                        STACKCT ← 0;                                                     15101500 T  0027
                        GT1 ← IF  TALL.CLASS =INTID  THEN IF  P1= FS                     15102000 T  0028
                                  THEN ISD ELSE ISN ELSE                                 15103000 T  0030
                                  IF P1 = FS THEN STD ELSE SND ;                         15104000 T  0033
                        IF TALL.FORMALNAME  THEN                                         15105000 T  0035
                             BEGIN                                                       15106000 T  0037
                             EMITO(XCH); IF TALL.ADDRESS>1023 THEN EMITO(PRTE);          15106100 T  0037
                             EMITO(GT1);                                                 15106200 T  0040
                             END                                                         15106300 T  0041
                        ELSE EMITPAIR(TALL.ADDRESS,GT1);                                 15107000 T  0041
```

```
                END                                                                 15108000 T  0043
           ELSE                                                                     15109000 T  0043
                BEGIN                                                               15110000 T  0043
                IF P1=FL THEN BEGIN                                                 15110100 T  0044
                     IF ELCLASS < AMPERSAND THEN EMITN(TALL,ADDRESS)                15110200 T  0045
                          ELSE EMITV(TALL,ADDRESS);                                 15110300 T  0046
                     GO TO EXIT   END;                                              15110400 T  0049
                IF  ELCLASS= PERIOD THEN                                            15111000 T  0050
                   BEGIN  IF DOTSYNTAX(T1,T2) THEN GO TO EXIT ;                     15112000 T  0050
                          IF STEPI=ASSIGNOP  THEN                                   15113000 T  0053
                              IF P1≠ FS  THEN                                       15114000 T  0054
                              BEGIN ERR(201);GO  TO  EXIT END                       15115000 T  0055
                                  ELSE  GO TO L1                                    15116000 T  0057
                                                                                    15117000 T  0057
                   END ;                                                            15118000 T  0057
                IF P1≠ FP THEN BEGIN ERR(202); GO TO EXIT END;                      15119000 T  0057
COMMENT  202    VARIABLE= A  VARIABLE APPEARS WHICH IS NOT FOLLOWED      *          15120000 T  0059
                         BY A LEFT ARROW OR PERIOD                       *;         15121000 T  0059
COMMENT  201    VARIABLE= A PARTIAL WORD DESIGNATOR IS NOT THE           *          15122000 T  0059
                         LEFT=MOST OF A LEFT PART LIST                   *;         15123000 T  0059
                EMITI(TALL,T1,T2);                                                  15124000 T  0059
                                                                                    15125000 T  0060
                END     ;                                                           15126000 T  0060
                END  OF SIMPLE VARIABLES                                            15127000 T  0060
                ELSE                                                                15128000 T  0060
                IF TALL.CLASS≠LABELID THEN                                          15128100 T  0060
    COMMENT     THE FOLLOWING BLOCK HANDLES THESE CASES OF SUBSCRIPTED              15129000 T  0062
                VARIABLES:                                                          15130000 T  0062
                   1. V[*]          ,ROW DESIGNATOR FOR SINGLE=DIMENSION.          15131000 T  0062
                   2. V[R,*]        ,ROW DESIGNATOR FOR MULTI=DIMENSION.           15132000 T  0062
                   3. V[R]          ,ARRAY ELEMENT,NAME OR VALUE.                  15133000 T  0062
                   4. V[R],[S:L]    ,PARTIAL WORD DESIGNATOR, VALUE.               15134000 T  0062
                   5. V[R] ←        ,ASSIGNMENT TO ARRAY ELEMENT.                  15135000 T  0062
                   6. V[R],[S:L] ← ,ASSIGNMENT TO PARTIAL WORD,LEFT=MOST.          15136000 T  0062
R IS A K=ORDER SUBSCRIPT LIST,I.E. R= R1,R2,...,RK.                                 15137000 T  0062
IN THE CASE OF NO MONITORING ON V, THE FOLLOWING CODE                               15138000 T  0062
IS EMITTED FOR THE ABOVE CASES:                                                     15139000 T  0062
        1. CASE #1 IS A SPECIAL CASE OF #2,NAMELY,SINGLE                            15140000 T  0062
           DIMENSION. THE CODE EMITTED IS:                                          15141000 T  0062
              VL,LOD .                                                              15142000 T  0062
           EXECUTION: PLACES ARRAY DESCRIPTER IN REG A.                             15143000 T  0062
        2. THIS CODE IS BASIC TO THE SUBSCRIPTION PROCESS.                          15144000 T  0062
           EACH SUBSCRIPT GENERATES THE FOLLOWING SEQUENCE                          15145000 T  0062
           OF CODE:                                                                 15146000 T  0062
              AEXP,L*,IF FIRST SUBSCRIPT THEN VN ELSE CDC                           15147000 T  0062
                ,LOD.                                                               15148000 T  0062
              FOR A K=ORDER SUBSCRIPTION,K=1 SEQUENCE ARE                           15149000 T  0062
              PRODUCED. THE AEXP IN EACH SEQUENCE REFERS TO                         15150000 T  0062
              THE CODE PRODUCED BY THE ARITHMETIC EXPRESSION                        15151000 T  0062
              PROCEDURE FOR THE ACTUAL SUBSCRIPT EXPRESSIONS.                       15152000 T  0062
              (* REFERS TO THE CODE PRODUCED FOR SUBTRACTING                       15153000 T  0062
              NON=ZERO LOWER BOUNDS FROM THE SUBSCRIPT                              15154000 T  0062
              EXPRESSION(L* YIELDS NO CODE FOR ZERO BOUNDS).                        15155000 T  0062
              EXECUTION: PLACES ARRAY ROW DESCRIPTOR IN REG A                       15156000 T  0062
                       . THE SPECIFIC ROW DEPENDS UPON THE                          15157000 T  0062
                       VALUES OF THE K=1 SUBSCRIPTS.                                15158000 T  0062
FOR THE REMAINING CASES,                                                            15159000 T  0062
```

```
                    SEQUENCES OF CODE ARE EMITED AS IN CASE #2.                  15160000 T  0062
                    HOWEVER,THE ACTUAL SEQUENCES ARE:                            15161000 T  0062
                     ONE SEQUENCE ,(AEXP,L*),FOR THE 1ST SUBSCRIPT.              15162000 T  0062
                     K-1 SEQUENCES,(IF FIRST SUBSCRIPT THEN VN                   15163000 T  0062
                     ELSE CDC,LOD,AEXP,L*), FOR THE REMAINING                    15164000 T  0062
                     SUBSCRIPTS,IF K>1.                                          15165000 T  0062
                    AT THIS POINT, CASES #3-6 ARE DIFFERENTIATED                 15166000 T  0062
                    AND ADDITION CODE,PARTICULAR TO EACH CASE,IS                 15167000 T  0062
                    EMITTED.                                                     15168000 T  0062
                 3. ADD THE SEQUENCE:                                            15169000 T  0062
                      IF FIRST SUBSCRIPT THEN VV ELSE CDC.                       15170000 T  0062
                    EXECUTION: THE ARRAY ELEMENT IS PUT IN REG A.                15171000 T  0062
                 4. ADD THE SEQUENCE:                                            15172000 T  0062
                      IF FIRST SUBSCRIPT THEN VV ELSE CDC,ZEROL,                 15173000 T  0062
                        XCH,T.                                                   15174000 T  0062
                 5. ADD THE SEQUENCE:                                            15175000 T  0062
                      IF FIRST SUBSCRIPT THEN VN ELSE CDC,EXP,                   15176000 T  0062
                        XCH,*.                                                   15177000 T  0062
                 6. ADD THE SEQUENCE:                                            15178000 T  0062
                      IF FIRST SUBSCRIPT THEN VN ELSE CDC,DUP,LOD,               15179000 T  0062
                        EXP,T, XCH,*.                                            15180000 T  0062
          EXP,T,*,ZEROL,ETC. HAVE SAME MEANINGS AS DEFINED IN                    15181000 T  0062
          SIMPLE VARIABLE BLOCK.                                                 15182000 T  0062
BEGIN                                                                            15183000 T  0062
                                                                                 15184000 T  0063
                                                                                 15184100 T  0063
                                                                                 15184200 T  0063
                                                                                 15184300 T  0063
                                                                                 15184400 T  0063
          IF STEPI ≠ LFTBRKET THEN                                               15233000 T  0063
          BEGIN                                                                  15233002 T  0064
          IF ELCLASS = PERIOD THEN                                               15233003 T  0064
               BEGIN                                                             15233004 T  0065
               IF DOTSYNTAX(T1,T2) THEN GO TO EXIT;                              15233005 T  0065
               IF STEPI = ASSIGNOP THEN                                          15233006 T  0067
                    BEGIN                                                        15233007 T  0068
                    IF  P1≠FS  THEN BEGIN ERR(209); GO EXIT END;                 15233008 T  0069
                    IF TALL.CLASS ≤ INTARRAYID THEN                              15233009 T  0071
                    BEGIN EMITPAIR(TALL.ADDRESS,LOD)  END                        15233010 T  0072
               ELSE EMITN(TALL.ADDRESS); STACKCT ← STACKCT+1;                    15233011 T  0074
JAZZ:               STEPIT;   AEXP;                                              15233012 T  0077
                    EMITD(48-T2,T1,T2);                                          15233013 T  0079
                    EMITPAIR(TALL.ADDRESS,                                       15233014 T  0080
                         IF P1=FS THEN STD ELSE SND);                            15233015 T  0081
               STACKCT ← 0; END                                                  15233016 T  0084
               ELSE BEGIN                                                        15233017 T  0085
ITUP:          EMITI(TALL,T1,T2);                                                15233018 T  0085
                                                                                 15233019 T  0087
                                                                                 15233020 T  0087
                                                                                 15233021 T  0087
                                                                                 15233022 T  0087
                    END;                                                         15233023 T  0087
               GO TO EXIT ;                                                      15233024 T  0087
               END;                                                              15233025 T  0087
          IF ELCLASS = ASSIGNOP THEN  GO TO JAZZ ELSE GO TO ITUP ;               15233026 T  0087
          END;                                                                   15233027 T  0089
               J ← 0;                                                            15234000 T  0089
```

```
                STACKCT ← 0;                                                                    15234500 T  0090
COMMENT  207    VARIABLE=MISSING LEFTBRACKET ON SUBSCRIPTED VARIABLE  *;                         15235000 T  0091
    NEXT;       IF STEPI = FACTOP THEN                                                          15253000 T  0091
                     BEGIN                                                                      15254000 T  0092
                     IF   J+1≠ TALL.INCR THEN                                                   15255000 T  0092
                               BEGIN ERR(203);GO EXIT END;                                      15256000 T  0094
COMMENT  203    VARIABLE= THE NUMBER OF SUBSCRIPTS USED IN A ROW        *                       15257000 T  0096
                          ROW DESIGNATER  DOES NOT MATCH THE ARRAY      *                       15258000 T  0096
                          DECLARATION.                                  *;                      15259000 T  0096
                          IF STEPI ≠ RTBRKET THEN                                               15260000 T  0096
                               BEGIN ERR(204);GO EXIT END;                                      15261000 T  0097
COMMENT  204    VARIABLE= COMPILER EXPECTS A  ]  IN A ROW DESIGNATER    *;                      15262000 T  0098
                                                                                                15263000 T  0098
COMMENT  205    VARIABLE= A ROW DESIGNATER APPEARS OUTSIDE OF A FILL    *                       15264000 T  0098
                          STATEMENT OR ACTUAL PARAMETER LIST.           *;                      15265000 T  0098
                          IF  J=0  THEN                                                         15266000 T  0098
                               EMITPAIR(TALL.ADDRESS,LOD);                                      15267000 T  0099
                     STLB←0;                                                                    15273000 T  0101
                          STEPIT;                                                               15274000 T  0102
                          GO TO EXIT;                                                           15275000 T  0102
                     END OF ROW DESIGNATOR PORTION  ;                                           15276000 T  0103
                IF ELCLASS=LITNO AND ELBAT[I].ADDRESS=0 AND TABLE(I+1)=RTBRKET                  15276010 T  0103
                     AND TALL.CLASS≥NAMEID THEN                                                 15276020 T  0107
                     BEGIN                                                                      15276030 T  0109
                     I←I+1;                                                                     15276040 T  0109
                     IF STEPI=ASSIGNOP THEN BEGIN                                               15276050 T  0111
 LASS;                    IF T1≠0 THEN EMITV(TALL.ADDRESS);                                     15276060 T  0112
                          STEPIT;  AEXP;  EMITO(48=T2,T1,T2);                                   15276070 T  0115
                          EMITN(TALL,ADDRESS);                                                  15276080 T  0118
                          EMITO(IF TALL.CLASS≠NAMEID THEN                                       15276090 T  0119
                               IF P1=FS THEN ISD ELSE ISN ELSE                                  15276100 T  0121
                               IF P1=FS THEN STD ELSE SND);                                     15276110 T  0124
                     STACKCT ← 0;                                                               15276115 T  0126
                          GO TO EXIT     END                                                    15276120 T  0127
                     ELSE                                                                       15276130 T  0128
                     IF ELCLASS = PERIOD THEN BEGIN                                             15276140 T  0128
                          IF DOTSYNTAX(T1,T2) THEN GO TO EXIT;                                  15276150 T  0129
                          IF STEPI = ASSIGNOP THEN IF P1=FS THEN GO TO LASS                     15276160 T  0131
                          ELSE BEGIN ERR(209); GO EXIT END;                                     15276170 T  0133
                               END;                                                             15276180 T  0135
                     IF P1=FS THEN BEGIN ERR(210); GO EXIT END;                                 15276190 T  0135
                                                                                                15276200 T  0138
                     EMITI(IF P1=FL THEN TALL ELSE TALL&REALIO[2:41:7],T1,T2);                  15276210 T  0138
                                                                                                15276220 T  0142
                     GO TO EXIT;                                                                15276230 T  0142
                     END;                                                                       15276240 T  0142
                     AEXP;                                                                      15277000 T  0142
                     STACKCT ← 1;                                                               15278000 T  0143
                     J ← J + 1;                                                                 15280000 T  0144
                     IF  ELCLASS = COMMA THEN                                                   15287000 T  0145
                          BEGIN                                                                 15288000 T  0146
COMMENT  *****            MONITOR FUNCTION M4 GOES HERE ;                                       15289000 T  0146
    IF J = 1 THEN EMITV(TALL.ADDRESS) ELSE EMITO(COC);                                          15290000 T  0146
                                                                                                15291000 T  0150
                               GO TO NEXT;                                                      15292000 T  0150
                          END OF SUBSCRIPT COMMA HANDLER ;                                      15293000 T  0150
                     IF ELCLASS ≠ RTBRKET THEN BEGIN ERR(206);GO EXIT END;                      15294000 T  0150
```

```
COMMENT  206  VARIABLE= MISSING RIGHT BRACKET ON SUBSCRIPTED VARIABLE*;          15295000 T  0153
         GT1←IF TALL.CLASS≥NAMEID THEN 1 ELSE TALL.INCR;                        15295100 T  0153
         IF J≠GT1  THEN                                                         15296000 T  0157
                        BEGIN ERR(208);GO TO EXIT END;                          15297000 T  0157
COMMENT  208  VARIABLE= NUMBER OF SUBSCRIPTS DOES NOT MATCH WITH       *        15298000 T  0159
                        ARRAY DECLARATION.                             *;       15299000 T  0159
              IF STEPI = ASSIGNOP  THEN                                         15300000 T  0159
                 BEGIN                                                          15301000 T  0160
  LAST:  IF J=1 THEN EMITN(TALL.ADDRESS) ELSE EMITO(CDC);                       15302000 T  0161
         IF  TALL.CLASS ≥ LONGID THEN EMITO(INX);                               15303000 T  0164
              IF  T1= 0 THEN                                                    15304000 T  0167
                        BEGIN IF P1= FR THEN GO TO EXIT END                     15305000 T  0168
                   ELSE BEGIN EMITO(DUP);EMITO(LOD)END;   STEPIT;               15306000 T  0169
              AEXP;                                                             15307000 T  0172
                   EMITO(48=T2,T1,T2) ;                                         15308000 T  0172
                   EMITO(XCH);                                                  15309000 T  0174
                   IF TALL.ADDRESS>1023 THEN EMITO(PRTE);                       15310000 T  0175
         EMITO(IF TALL.CLASS MOD 2 = INTARRAYID MOD 2 THEN                      15333000 T  0177
                        IF  P1 = FS THEN ISD ELSE ISN ELSE                      15334000 T  0180
                        IF P1=FS THEN STD ELSE SND);                            15335000 T  0183
              STACKCT ← 0;                                                      15335500 T  0186
                   P1←0  ;                                                      15336000 T  0186
                   GO TO EXIT ;                                                 15337000 T  0187
                 END OF ASSIGNMENT STATEMENT SUBSCRIPTED VARIABLES;             15338000 T  0188
              IF  ELCLASS=PERIOD  THEN                                          15339000 T  0188
                 BEGIN                                                          15340000 T  0189
                   IF DOTSYNTAX(T1,T2) THEN GO TO EXIT;                         15341000 T  0189
                   IF STEPI = ASSIGNOP THEN IF P1=FS THEN GO TO LAST            15342000 T  0191
                        ELSE BEGIN ERR(209); GO EXIT END;                       15343000 T  0193
              IF J≠1 THEN EMITO(COC) ELSE IF TALL.CLASS ≥ LONGID THEN           15344000 T  0195
                   BEGIN EMITN(TALL.ADDRESS);EMITO(INX);EMITO(LOD) END          15344100 T  0199
                   ELSE  EMITV(TALL.ADDRESS);                                   15344200 T  0202
                      END                                                       15345000 T  0204
              ELSE                                                              15346000 T  0204
COMMENT  *****        MONITOR FUNCTION M10 GOES HERE   ;                        15347000 T  0204
                      BEGIN COMMENT MONITOR FUNCTION M10;                       15348000 T  0204
                           SPCLMON←P1 = FP OR ELCLASS ≥ AMPERSAND;              15349000 T  0204
                           IF J = 1                                             15350000 T  0206
              THEN IF TALL.CLASS ≥ LONGID THEN                                  15351000 T  0207
                   BEGIN                                                        15351100 T  0209
                   EMITN(TALL.ADDRESS); EMITO(INX);                             15351200 T  0209
                   IF SPCLMON   THEN  EMITO(LOD) ;                              15351300 T  0211
                   END   ELSE  IF  SPCLMON                                      15351400 T  0213
                                  THEN EMITV(TALL.ADDRESS)                      15352000 T  0213
                                  ELSE EMITN(TALL.ADDRESS)                      15353000 T  0214
                           ELSE EMITO(IF SPCLMON                                15354000 T  0216
                                         THEN COC                               15355000 T  0218
                                         ELSE CDC);                             15356000 T  0219
                           IF P1 =FS THEN ERR(210);                             15364000 T  0220
                           GO TO     EXIT;                                      15365000 T  0222
                      END;                                                      15366000 T  0222
              IF P1=FS THEN BEGIN ERR(210); GO TO EXIT END ;                    15367000 T  0222
COMMENT  210  VARIABLE=MISSING LEFT ARROW OR PERIOD.                   *;       15368000 T  0225
              STACKCT ← 0;                                                      15369000 T  0225
              IF T1 ≠ 0 THEN BEGIN EMITI(0,T1,T2); P1 ← 0 END;                  15370000 T  0226
              END OF SUBSCRIPTED VARIABLES                                      15376000 T  0229
              ELSE                                                              15376100 T  0229
```

```
                        BEGIN COMMENT LABELID;                                                  15376200 T  0229
                        T1:=TAKE(T2:=GIT(TALL));                                                15376300 T  0230
                        PUT(L,T2);                                                              15376400 T  0232
                        IF T1=0 THEN T1:=L;                                                     15376500 T  0233
                        IF (T1←L-T1) DIV 4 > 127 THEN BEGIN T1←0;FLAG(50);END;                  15376600 T  0235
                        EMIT(T1×4+3);                                                           15376700 T  0239
                        STEPIT;                                                                 15376800 T  0241
                        END OF LABELID;                                                         15376900 T  0241
                EXIT :  END  OF THE VARIABLE ROUTINE;                                           15377000 T  0241
                                                                                      102 IS  246 LONG, NEXT SEG     3



        COMMENT THIS SECTION GENERATES CODE FOR STREAM PROCEDURES;                              16000000 T  0586
        COMMENT  DO  LABEL DECS UPON APPEARANCE OF LABEL  ;                                     16000050 T  0586
        PROCEDURE DECLARELABEL ;                                                                16000100 T  0586
PRT(711) = DECLARELABEL
                        BEGIN                                                                   16000200 T  0586
                        KLASSF ← STLABID;                                                       16000300 T  0586
                        VONF ← FORMALF ← FALSE;                                                 16000400 T  0587
                        ADDRSF ← 0;                                                             16000500 T  0589
                        MAKEUPACCUM;  E; PUTNBUMP(0);                                           16000600 T  0589
                        ELBAT[I] ← ACCUM[0]& LASTINFO[35:35:13];                                16000700 T  0591
                        END;                                                                    16000800 T  0594

        PROCEDURE STREAMSTMT ;                                                                  16001000 T  0594
                BEGIN                                                                           16002000 T  0594
                DEFINE LFTPAREN=LEFTPAREN#,LOC=[36:12]#,LASTGT=[24:12]#,                        16003000 T  0594
                                                                                      START OF SEGMENT **********  103
                     LOCFLD=36:36:12#,LGTFLD=24:24:12#;                                         16004000 T  0000
                DEFINE LEVEL=LVL#,ADDOP=ADOP#;                                                  16005000 T  0000
                                                                                                16006000 T  0000
        DEFINE                                                                                  16007000 T  0000
                JFW  =  39#,   COMMENT  7.5.5.1  JUMP FORWARD UNCONDITIONAL  ;                   16008000 T  0000
                 RCA =  40#, COMMENT    7.5.7.6  RECALL CONTROL ADDRESS      ;                  16009000 T  0000
                JRV  =  47#,  COMMENT   7.5.5.2  JUMP REVERSE UNCONDITIONAL  ;                  16010000 T  0000
                CRF  =  35#,  COMMENT    7.5.10.6 CALL REPEAT FIELD          ;                  16011000 T  0000
                 BNS =  42#,  COMMENT   7.5.5.5  BEGIN LOOP                  ;                  16012000 T  0000
                NOP  =   1#,  COMMENT                                       ;                   16013000 T  0000
                ENS  =  41#,  COMMENT   7.5.5.6  END LOOP                   ;                   16014000 T  0000
                TAN  =  30#,  COMMENT   7.5.3.7  TEST FOR ALPHAMERIC        ;                   16015000 T  0000
                BIT  =  31#,  COMMENT   7.5.3.8  TEST BIT                   ;                   16016000 T  0000
                 JFC =  37#,  COMMENT   7.5.5.3  JUMP FORWARD CONDITIONAL   ;                   16017000 T  0000
                SED  =  06#,  COMMENT   7.5.7.8  SET DESTINATION ADDRESS    ;                   16018000 T  0000
                RSA  =  43#,  COMMENT   7.5.7.4  RECALL SOURCE ADDRESS      ;                   16019000 T  0000
                TRP  =  60#,  COMMENT   7.5.2.2  TRANSFER PROGRAM CHARACTERS ;                  16020000 T  0000
                BSS  =   3#,  COMMENT   7.5.6.6  SKIP SOURCE BIT            ;                   16021000 T  0000
                BSD  =   2#,  COMMENT   7.5.6.5  SKIP DESTINATION BITS      ;                   16022000 T  0000
                SEC  =  34#,     COMMENT   7.5.10.1  SET COUNT              ;                   16023000 T  0000
                JNS  =  38#;   COMMENT  7.5.5.7  JUMP OUT LOOP              ;                   16023100 T  0000
        PROCEDURE ADJUST;;
PRT(712) = ADJUST
```

```
                    COMMENT FIXC EMITS  BASICLY FORWARD JUMPS. HOWEVER IN THE CASE          16024000 T  0000
                            OF INSTRUCTIONS INTERPTED AS JUMPS BECAUSE OF A CRF ON         16025000 T  0000
                            A VALUE = 0 AND THE JUMP ≥ 64 SYLLABLES A JFW 1 AND            16026000 T  0000
                            A RCA L (L IS STACK ADDRESS OF A PSEUDO LABEL WHICH            16027000 T  0000
                            MUST ALSO BE MANUFACTURED) IS EMITTED.  ;                      16028000 T  0000
                PROCEDURE FIXC(S); VALUE S; REAL S;                                        16029000 T  0000
PRT(713) = FIXC
                        BEGIN                                                              16030000 T  0000
                        REAL SAVL,D,F;                                                     16031000 T  0000
                                                                                  START OF SEGMENT **********  104
STACK(F+2) = SAVL
STACK(F+3) = D
STACK(F+4) = F
                        IF D← (SAVL←L) - (L←S)-1 ≤ 63 THEN EMITC(D,GET(S))                 16032000 T  0000
                            ELSE  FLAG(700);                                               16033000 T  0004
                        L←SAVL ;                                                           16034000 T  0006
                        END  FIXC  ;                                                       16057000 T  0007
                                                                                   104 IS   10 LONG, NEXT SEG  103


                    COMMENT EMITJUMP IS CALLED BY GOTOS AND JUMPCHAIN.                     16058000 T  0000
                            THIS ROUTINE WILL EMIT A JUMP IF THE DISTANCE IS ≤ 63           16059000 T  0000
                            SYLLABLES ,OTHERWISE, IT GETS A PRT CELL AND STUFFS THE        16060000 T  0000
                            STACK ADDRESS INTO THE LABEL ENTRY IN INFO AND EMITS AN        16061000 T  0000
                            RCA ON THIS STACK CELL. AT EXECUTION TIME ACTUAL PARAPART      16062000 T  0000
                            INSURES US THAT THIS CELL WILL CONATIN A LABEL DESCRIPTOR       16063000 T  0000
                            POINTING TO OUR LABEL IN QUESTION.  ;                          16064000 T  0000
                PROCEDURE EMITJUMP( E); VALUE E; REAL E;                                   16065000 T  0000
PRT(714) = EMITJUMP
                        BEGIN                                                              16066000 T  0000
                        REAL T,D;                                                          16067000 T  0000
                                                                                  START OF SEGMENT **********  105
STACK(F+2) = T
STACK(F+3) = D
                        REAL ADDR;                                                         16068000 T  0000
STACK(F+4) = ADDR
                        IF ABS(                                                            16069000 T  0000
                        D←(T←TAKE(GIT(E)),LOC)-L-1)≥64 THEN                                16070000 T  0000
                            FLAG(700)                                                      16071000 T  0004
                        ELSE EMITC(D,IF D <0 THEN JRV ELSE JFW);                           16079000 T  0005
                        END  EMIT JUMP;                                                    16080000 T  0009
                                                                                   105 IS   12 LONG, NEXT SEG  103


                    COMMENT   WHEN JUMPCHAIN IS CALLED THERE IS A LINKEDLIST IN THE CODE   16081000 T  0000
                            ARRAY WHERE JFWS MUST BE PLACED. THE 1ST LINK IS POINTED        16082000 T  0000
                            TO BY THE LOC FIELD OF EACH LABEL ENTRY IN INFO. THE LAST      16083000 T  0000
                            LINK IS = 4096.  ;                                             16084000 T  0000
                PROCEDURE JUMPCHAIN( E); VALUE E;REAL E;                                   16085000 T  0000
PRT(715) = JUMPCHAIN
                        BEGIN                                                              16086000 T  0000
                        REAL SAVL ,LINK;                                                   16087000 T  0000
                                                                                  START OF SEGMENT **********  106
```

```
STACK(F+2) = SAVL
STACK(F+3) = LINK
                    SAVL ← L;                                                    16088000 T  0000
                    L ←  TAKE(GIT(E)),LASTGT ;                                   16089000 T  0000
                    WHILE L≠ 4095 DO                                             16090000 T  0003
                         BEGIN                                                   16091000 T  0004
                         LINK ← GET(L);                                          16092000 T  0004
                         EMITJUMP( E);                                           16093000 T  0005
                         L ← LINK                                                16094000 T  0006
                         END;                                                    16095000 T  0006
                    L←SAVL;                                                      16096000 T  0009
                    END  JUMPCHAIN  ;                                            16097000 T  0009
                                                                  106 IS   12 LONG, NEXT SEG  103


               COMMENT NESTS COMPILES THE NEST STATEMENT,                        16098000 T  0000
                         A VARIABLE NEST INDEX CAUSES THE CODE,                  16099000 T  0000
                              CRF V, BNS 0 ,NOP,NOP, TO BE GENERATED INITIALLY.  16100000 T  0000
                         AT THE RIGHT PAREN THE BNS IS FIXED WITH THE LENGTH OF  16101000 T  0000
                         THE NEST (NUMBER OF SYLLABLES) IF THE LENGTH ≤63,OTHERWISE 16102000 T  0000
                         IT IS FIXED WITH A 1 AND THE NOPS REPLACED WITH JFW 1,  16103000 T  0000
                         RCA P.  THIS IS DONE BECAUSE THE VALUE OF V AT EXECUTION 16104000 T  0000
                         MAY = 0 AND THIS CODE CAUSES A JUMP AROUND THE NEST.    16105000 T  0000
                         JUMPOUT INFO IS REMEMBERED IN A RECURSIVE CELL AND      16106000 T  0000
                         NEST LEVEL INCREASED BY ONE.                            16107000 T  0000
                         WHEN THE RIGHT PAREN IS REACHED,(IF THE STATEMENTS IN   16108000 T  0000
                         THE NEST COMPILED), JOINFO IS CHECKED FOR THE EXISTANCE 16109000 T  0000
                         OF JUMPOUT STATEMENTS IN THE NEST,IF SO,THE THE JUMPS   16110000 T  0000
                         ARE FIXED BY FAKING TOTOS INTO COMPILING THE REQUIRED   16111000 T  0000
                         JUMPS.                                                  16112000 T  0000
                         FINALLY THE BNS IS FIXED,IF REQUIRED,AND NEST LEVEL     16113000 T  0000
                         AND JOINFO RESTORED TO THEIR ORIGINAL VALUES.  ;        16114000 T  0000
                    PROCEDURE NESTS;                                             16115000 T  0000
PRT(716) = NESTS
                    BEGIN                                                        16116000 T  0000
                    LABEL EXIT;                                                  16117000 T  0000
                                                          START OF SEGMENT **********  107
                    REAL JOINT,BNSFIX;                                           16118000 T  0000
STACK(F+2) = JOINT
STACK(F+3) = BNSFIX
                    IF ELCLASS≠LITNO THEN                                        16119000 T  0000
                         BEGIN                                                   16120000 T  0000
                         EMITC(ELBAT[I].ADDRESS,CRF); BNSFIX← L;                 16121000 T  0001
                         EMIT(BNS);                                              16122000 T  0003
                         END                                                     16123000 T  0004
                    ELSE EMITC(ELBAT[I].ADDRESS,BNS);                            16124000 T  0004
                    IF STEPI ≠ LFTPAREN THEN BEGIN ERR(262); GO TO EXIT END;     16125000 T  0006
                    NESTLEVEL←NESTLEVEL + 1;                                     16126000 T  0009
                    JOINT ← JOINFO;                                              16127000 T  0010
                    JOINFO ← 0;                                                  16128000 T  0011
                    DO   BEGIN                                                   16129000 T  0012
                         STEPIT; ERRORTOG ← TRUE; STREAMSTMT                     16130000 T  0013
                         END UNTIL ELCLASS ≠ SEMICOLON ;                         16131000 T  0014
                    IF ELCLASS ≠ RTPAREN THEN BEGIN ERR(262);GO TO EXIT END;     16132000 T  0016
                    EMIT (  ENS);                                                16133000 T  0018
```

```
          IF JOINFO ≠ 0 THEN                                                    16134000 T  0019
               BEGIN                                                            16135000 T  0020
     COMMENT   PREPARE TO CALL JUMPCHAIN FORJUMPOUTS;                           16136000 T  0020
               ADJUST;                                                          16137000 T  0020
               PUT(TAKE(GIT(JOINFO))&L[LOCFLD],GIT(JOINFO));                    16138000 T  0021
               JUMPCHAIN(TAKE(JOINFO)&JOINFO[35:35:13]);                        16139000 T  0024
               END;                                                             16140000 T  0026
          IF BNSFIX ≠ 0 THEN FIXC(BNSFIX);                                      16141000 T  0026
          NESTLEVEL ← NESTLEVEL-1;                                              16142000 T  0028
          JOINFO ← JOINT ;                                                      16143000 T  0030
     EXIT: END  NESTS  ;                                                        16144000 T  0030
                                                                  107 IS   34 LONG, NEXT SEG  103


     COMMENT   LABELS HANDLES STREAM LABELS.                                    16145000 T  0000
               ALL LABELS ARE ADJUSTED TO THE BEGINING OF THE NEXT              16146000 T  0000
               WORD (IN THE PROGRAMSTREAM).                                     16147000 T  0000
               IF A GO TO HAS NOT BEEN ENCOUNTERED BEFORE THE LABEL             16148000 T  0000
               THEN THE NEST LEVEL FIELD IS ENTERED AND THE DEFINED BIT,        16149000 T  0000
               [1:1], SET TO ONE. FOR DEFINED LABELS,IF WHERE A GO TO           16150000 T  0000
               HAS APPEARED, A CHECK IS MADE THAT THE CURRENT NEST LEVEL        16151000 T  0000
               MATCHES THE LEVEL OF THE LABEL.                                  16152000 T  0000
               MULTIPLE OCCURANCES ARE ALSO CHECKED FOR AND FLAGGED.            16153000 T  0000
               FINALLY,JUMPCHAIN IS CALLED TO FIX UP ANY FORWARD GO TOS         16154000 T  0000
               AND GET A PRT LOCATION FOR ANY JUMPS ≥64 SYLLABLES.  ;           16155000 T  0000
PROCEDURE LABELS;                                                               16156000 T  0000
PRT(717) = LABELS
                    BEGIN                                                       16157000 T  0000
                    REAL GT1;                                                   16157100 T  0000
                                                                  START OF SEGMENT **********  108
STACK(F+2) = GT1
                    ADJUST;                                                     16158000 T  0000
                    GT1 ← ELBAT[I];                                             16159000 T  0000
                    IF STEPI ≠ COLON THEN ERR(258)                              16160000 T  0001
                    ELSE                                                        16161000 T  0003
                         BEGIN                                                  16162000 T  0003
                         IF TAKE(GT2+GIT(GT1)).LOC ≠ 0 THEN FLAG(259);          16163000 T  0004
                         IF GT1>0 THEN                                          16164000 T  0008
                              BEGIN                                             16165000 T  0009
                              PUT(-(TAKE(GT1)&NESTLEVEL[11:43:5]),GT1);         16166000 T  0009
                              PUT(-L,GT2)                                       16167000 T  0012
                              END                                               16168000 T  0013
                         ELSE                                                   16169000 T  0013
                              BEGIN                                             16170000 T  0013
                              IF GT1.LEVEL≠NESTLEVEL THEN FLAG(257);            16171000 T  0014
                              PUT((-L)&TAKE(GT2)[LGTFLD],GT2);                  16172000 T  0016
                              JUMPCHAIN(GT1);                                   16173000 T  0019
                              END;                                              16174000 T  0020
                         END                                                    16175000 T  0020
                    ;  STEPIT;                                                 16176000 T  0020
                    END  LABELS  ;                                              16177000 T  0020
                                                                  108 IS   83 LONG, NEXT SEG  103
```

```
                    COMMENT   IFS COMPILES IF STATEMENTS.                                          16178000 T  0000
                              FIRST THE TEST IS COMPILED. NOTE THAT IN THE                        16179000 T  0000
                              CONSTRUCTS  "SC RELOP DC" AND  "SC RELOP STRING" THAT               16180000 T  0000
                              THE SYLLABLE EMITTED IS FETCHED FROM ONE OF TWO FIELDS              16181000 T  0000
                              IN THE ELBAT WORD FOR THE RELATIONAL OPERATOR. OTHERWISE             16182000 T  0000
                              THE CODE IS EMITTED STRAIGHTAWAY.                                   16183000 T  0000
                              A TEST IS MADE TO SEE WHETHER THE STATEMENT AFTER THE               16184000 T  0000
                              "THEN" COULD POSSIBLY BE LONGER THAN 63 SYLLABLES,AND IF            16185000 T  0000
                              SO, Z NOPS ARE EMITTED FOR FIXC IN CASE A RCA WILL HAVE             16186000 T  0000
                              TO BE GENERATED.                                                    16187000 T  0000
                              THIS PROCEDURE DOES NO OPTIMAZATION IN THE CASES                    16188000 T  0000
                              IF THEN GO TO L,IF THEN STATEMENT ELSE GO TO L, OR                  16189000 T  0000
                              IF THEN GO TO L1 ELSE GO TO L2   ;                                  16190000 T  0000
                    PROCEDURE  IFS;   BEGIN                                                       16191000 T  0000
PRT(720) = IFS
                         DEFINE COMPARECODE=[42:6]#,TESTCODE=[36:6]#;                             16192000 T  0000
                                                                                  START OF SEGMENT **********  109
                         LABEL  IFSB,IFTOG,IFSC,EXIT;                                             16193000 T  0000
                         SWITCH IFSW ← IFSB,IFTOG,IFSC;                                           16194000 T  0000
                         REAL ADDR,FIX1,FIX2  ;                                                   16195000 T  0004
STACK(F+2) = ADDR
STACK(F+3) = FIX1
STACK(F+4) = FIX2
                         ADDR←1  ;                                                                16196000 T  0004
                         GO TO IFSW[STEPI -SBV+1] ;                                               16197000 T  0005
                         IF ELCLASS=LOCLID THEN                                                   16198000 T  0009
                              BEGIN                                                               16199000 T  0009
                              EMITC(ELBAT[I],ADDRESS,CRF);                                        16200000 T  0010
                              ADDR←0;                                                             16201000 T  0012
                              END                                                                 16202000 T  0012
                         ELSE                                                                     16203000 T  0012
                              IF ELCLASS=LITNO THEN ADDR ← ELBAT[I].ADDRESS                       16204000 T  0012
                         ELSE BEGIN ERR(250); GO TO EXIT END;                                     16205000 T  0014
                         IF STEPI ≠ SCV THEN BEGIN ERR(263);GO TO EXIT END;                       16206000 T  0017
                  IFSC;  IF STEPI ≠ RELOP THEN BEGIN ERR(264);GO TO EXIT END;                     16207000 T  0020
                         IF STEPI = DCV THEN EMITC( ADDR,ELBAT[I-1],COMPARECODE)                  16208000 T  0023
                         ELSE                                                                     16209000 T  0026
                              IF ELCLASS = STRNGCON THEN                                          16210000 T  0027
                                   EMITC(ACCUM[1],[18:6],ELBAT[I-1].TESTCODE)                     16211000 T  0028
                         ELSE                                                                     16212000 T  0031
                         IF ELCLASS=LITNO THEN EMITC(C,ELBAT[I-1].TESTCODE) ELSE                  16212500 T  0032
                         IF ELCLASS$IDMAX AND Q="5ALPHA" THEN EMITC(17,TAN)                       16213000 T  0036
                         ELSE BEGIN ERR(265);GO TO EXIT END;                                      16214000 T  0039
                         GO TO IFTOG  ;                                                           16215000 T  0043
                   IFSB: EMITC(1,BIT);                                                            16216000 T  0043
                  IFTOG: IF STEPI ≠ THENV THEN BEGIN ERR(266); GO TO EXIT  END;                   16217000 T  0045
                         FIX1 ← L;                                                                16218000 T  0047
                         EMIT(JFC);                                                               16219000 T  0048
                         IF STEPI≠ELSEV THEN%                                                     16220000 T  0049
                         STREAMSTMT;                                                              16229000 T  0050
                         IF ELCLASS= ELSEV THEN                                                   16230000 T  0051
                              BEGIN                                                               16231000 T  0052
                              FIX2 ← L;  EMIT(JFW);                                               16232000 T  0052
                              FIXC(FIX1);                                                         16233000 T  0054
                              STEPIT;                                                             16234000 T  0054
                              STREAMSTMT;                                                         16235000 T  0055
                              FIXC(FIX2);                                                         16236000 T  0055
```

```
                    END
               ELSE FIXC(FIX1);
          EXIT:END   IFS  ;
```

```
16237000 T  0056
16238000 T  0056
16239000 T  0057
109 IS   61 LONG, NEXT SEG  103
```

```
          COMMENT GOTOS HANDLES GO TO AND THE LAST PART OF JUMP OUT TO
                  STATEMENTS.
                  IF THE LABEL HAS BEEN ENCOUNTERED THEN EMITJUMP IS CALLED
                  AN PRODUCES A JRV OR RCA IN THE CASE OF JUMPS264 SYLLABL
                  ES. OTHERWISE, A LINK IS EMITTED POINTING ANY PREVIOUS
                  GO TOS IN THE CASE OF FORWARD JUMPS.
                  FINALLY, IF THE NEST LEVEL IS DEFINED THEN IT IS CHECKED
                  AGAINST THE CURRENT LEVEL MINUS THE NUMBER OF LEVELS TO
                  BE JUMPED OUT. OTHERWISE,NEST LEVEL IS DEFINED.  ;
        PROCEDURE GOTOS;
PRT(721) = GOTOS
          BEGIN
          LABEL EXIT;

          IF STEPI #TOV THEN I←I-1 ;
               IF STEPI # STLABID THEN IF ELCLASS ≤ IDMAX THEN
                    DECLARELABEL ELSE BEGIN ERR(260); GO TO EXIT END;
          IF(GT2←TAKE(GIT(GT1←ELBAT[I]))).MON=1
          OR GT2.LOC#0 THEN EMITJUMP(GT1)
          ELSE
            BEGIN   PUT(O&L[24:36:12],GIT(GT1));
               IF GT1>0 THEN
                    BEGIN
                    PUT(=(TAKE(GT1)&(NESTLEVEL-JUMPLEVEL)[11:43:5]),GT1);
                    EMITN(1023);
                    END
               ELSE
                    BEGIN
                    IF GT1.LEVEL # NESTLEVEL-JUMPLEVEL THEN FLAG(257);
                    EMIT(GT2,LASTGT);
               END;
            END;
          JUMPLEVEL←0 ;
     EXIT: END  GOTOS   ;
```

```
16240000 T  0000
16241000 T  0000
16242000 T  0000
16243000 T  0000
16244000 T  0000
16245000 T  0000
16246000 T  0000
16247000 T  0000
16248000 T  0000
16249000 T  0000
16250000 T  0000
16251000 T  0000
START OF SEGMENT **********  110
16252000 T  0000
16253000 T  0002
16253100 T  0005
16254000 T  0007
16255000 T  0010
16256000 T  0013
16257000 T  0014
16258000 T  0017
16259000 T  0017
16260000 T  0018
16261000 T  0021
16262000 T  0022
16263000 T  0022
16264000 T  0022
16265000 T  0022
16266000 T  0025
16267000 T  0027
16268000 T  0027
16269000 T  0027
16270000 T  0027
110 IS   29 LONG, NEXT SEG  103
```

```
COMMENT  RELEASES COMPILES THE STREAM RELEASE STATEMENT.
         THE CODE GENERATED IS :
              SED  FILE
              RSA  0.
         AT EXECUTION TIME THIS CAUSES AN INVALID ADDRESS WHICH IS
         INTERPETED BY THE MCP TO MEAN RELEASE THE FILE POINTED TO
         BY THE DESTINATION ADDRESS.
         THE MONITOR BIT IS SET IN INFO FOR THE LOCAL VARIABLE SO
         THAT ACUTAL PARAPART MAY BE INFORMED LATER THAT A FILE
         MUST BE PASSED FOR THIS FORMAL PARAMETER;
```

```
16271000 T  0000
16272000 T  0000
16273000 T  0000
16274000 T  0000
16275000 T  0000
16276000 T  0000
16277000 T  0000
16278000 T  0000
16279000 T  0000
16280000 T  0000
```

```
                                                                                16281000 T  0000
                                                                                16282000 T  0000
                                                                                16283000 T  0000
                                                                                16284000 T  0000
                                                                                16285000 T  0000
                                                                                16286000 T  0000
                                                                                16287000 T  0000
                                                                                16288000 T  0000
                                                                                16289000 T  0000
COMMENT    INDEXS  COMPILE STATEMENTS BEGINING WITH SI,DI,CI,TALLY               16290000 T  0000
           OR LOCALIDS .                                                        16291000 T  0000
           THREE CASES PRESENT THEMSELVES,                                      16292000 T  0000
                LETING X BE EITHER OF  SI,DI,CI OR TALLY, THEY ARE:             16293000 T  0000
                CASE I    LOCLID ← X                                            16294000 T  0000
                CASE II   X ← X ...,                                            16295000 T  0000
                CASE III  X ← EITHER LOC,LOCLID,SC OR DC.                       16296000 T  0000
           THE VARIABLE "INDEX" IS COMPUTED,DEPENDING UPON WHICH                16297000 T  0000
           CASE EXISTS,SUCH THAT ARRAY ELEMENT "MACRO[INDEX]"CONTAINS           16298000 T  0000
           THE CODE TO BE EMITTED.                                              16299000 T  0000
           EACH ELEMENT OF MACRO HAS  1-3 SYLLABLES ORDERED FROM                16300000 T  0000
           RIGHT TO LEFT, UNUSED SYLLABLES MUST = 0.    EACH MACRO              16301000 T  0000
           MAY REQUIRE AT MOST ONE REPEAT PART.                                 16302000 T  0000
           IN THIS PROCEDURE,INDEXS,THE VARIABLE "ADDR" CONTAINS THE            16303000 T  0000
           PROPER REPEAT PART BY THE TIME THE LABEL "GENERATE" IS               16304000 T  0000
           ENCOUNTERED, THE SYLLABLES ARE FETCHED FROM MACRO[TYPE]              16305000 T  0000
           ONE AT A TIME AND IF THE REPEAT PART ≠ 0 THEN"ADDR" IS               16306000 T  0000
           USED AS THE REPEAT PART,THUS BUILDING A SYLLABLE WITH                16307000 T  0000
           THE PROPER  ADDRESS AND OPERATOR .                                   16308000 T  0000
           NOTE: IF MACRO[TYPE] = 0 THEN THIS SIGNIFIES A SYNTAX                16309000 T  0000
                 ERROR.                                ;                      16310000 T  0000
      PROCEDURE INDEXS;                                                         16311000 T  0000
PRT(722) = INDEXS
                BEGIN                                                           16312000 T  0000
                LABEL EXIT,GENERATE,L,L1;                                       16313000 T  0000
                                                                    START OF SEGMENT **********  111
                INTEGER TCLASS,INDEX,ADDR,J;                                    16314000 T  0000
STACK(F+2) = TCLASS
STACK(F+3) = INDEX
STACK(F+4) = ADDR
STACK(F+5) = J
                TCLASS ← ELCLASS ;                                              16315000 T  0000
                IF STEPI ≠ ASSIGNOP THEN BEGIN ERR(251); GO TO EXIT END;        16316000 T  0000
                IF TCLASS = LOCLID THEN                                         16317000 T  0003
                     BEGIN                                                      16318000 T  0004
                     IF SIV>STEPI OR ELCLASS>TALLYV THEN GO TO L;               16319000 T  0004
                     INDEX ← 32 + ELCLASS*SIV;                                  16320000 T  0007
                     ADDR ← ELBAT[I-2],ADDRESS;                                 16321000 T  0009
                     GO TO GENERATE;                                            16322000 T  0011
                     END;                                                       16323000 T  0011
                IF TCLASS = STEPI THEN                                          16324000 T  0011
                     BEGIN                                                      16325000 T  0012
                     IF STEPI ≠ ADDOP OR STEPI≠ LITNU AND ELCLASS ≠ LOCLID THEN 16326000 T  0013
                          GO TO L;                                              16327000 T  0015
                     INDEX ← TCLASS*SIV                                         16328000 T  0017
                          +REAL(ELBAT[I-1],ADDRESS*SUB) × 4                     16329000 T  0017
                          + REAL(ELCLASS =LOCLID) × 8;                          16330000 T  0020
                     END                                                        16331000 T  0022
```

```
     ELSE                                                                                    16332000 T  0022
          BEGIN                                                                              16333000 T  0022
          INDEX ← TCLASS -SIV                                                                16334000 T  0023
               + ( IF ELCLASS = LOCLID THEN 16 ELSE                                          16335000 T  0023
                IF ELCLASS = LOCV THEN 20 ELSE                                               16336000 T  0026
                IF ELCLASS = SCV THEN 24 ELSE                                                16337000 T  0028
                IF ELCLASS= DCV  THEN 28 ELSE 25);                                           16338000 T  0030
          IF ELCLASS = LOCV THEN                                                             16339000 T  0033
               IF STEPI ≠ LOCLID THEN GO TO L;                                               16340000 T  0033
          IF ELCLASS = LITNO AND TCLASS = TALLYV THEN                                        16341000 T  0035
               BEGIN EMITC(ELBAT[I],ADDRESS,SEC);GO TO EXIT END;                             16342000 T  0037
          END ;                                                                              16343000 T  0040
     ADDR ← ELBAT[I],ADDRESS;                                                                16344000 T  0040
GENERATE;                                                                                    16345000 T  0041
     IF MACRO[INDEX]= 0 THEN                                                                 16346000 T  0042
  L:      BEGIN ERR(250);GO TO EXIT END;                                                     16347000 T  0043
     J ← 8; TCLASS ←0 ;                                                                      16348000 T  0045
 L1: MOVECHARACTERS(2,MACRO[INDEX],J+J-2,TCLASS,6 );                                         16349000 T  0046
     IF TCLASS≠0 THEN                                                                        16350000 T  0050
          BEGIN                                                                              16351000 T  0050
          EMITC(IF TCLASS≥64 THEN ADDR ELSE 0,TCLASS);                                       16352000 T  0051
          GO TO L1                                                                           16353000 T  0054
          END;                                                                               16354000 T  0054
EXIT;END  INDEXS ;                                                                           16355000 T  0054
                                                                                  111 IS   58 LONG, NEXT SEG  103

                    COMMENT  DSS COMPILES DESTINATION STREAM STATEMENTS,                     16356000 T  0000
                             DS←  LIT"STRING" IS HANDLED AS A SPECIAL CASE BECAUE THE        16357000 T  0000
                             STRING MUST BE SCANED FROM RIGHT TO LEFT,REPEATEDLY IF          16358000 T  0000
                             NECESSARY, AND EMITTED TO THE PROGRAM STREAM.  IN               16359000 T  0000
                             ALL OTHER CASES,THE ELBAT WORD CONTAINS THE OPERATOR IN         16360000 T  0000
                             THE OPCODE FIELD  ;                                             16361000 T  0000
               PROCEDURE DSS;                                                                16362000 T  0000
PRT(723) = DSS
                         BEGIN                                                               16363000 T  0000
                         INTEGER ADDR,J,K,L,T;                                               16364000 T  0000
                                                                                  START OF SEGMENT **********  112
STACK(F+2) = ADDR
STACK(F+3) = J
STACK(F+4) = K
STACK(F+5) = L
STACK(F+6) = T
                         LABEL EXIT,L1;                                                      16365000 T  0000
                         DEFINE OPCODE=[27:6]#;                                              16366000 T  0000
                         IF STEPI ≠ ASSIGNOP THEN BEGIN ERR(251); GO TO EXIT END;            16367000 T  0000
                         IF STEPI = LOCLID THEN                                              16368000 T  0002
                              BEGIN                                                          16369000 T  0003
                              EMITC(ELBAT[I],ADDRESS,CRF);                                   16370000 T  0004
                              ADDR← 0;                                                       16371000 T  0006
                              IF STEPI = LITV THEN GO TO L1                                  16372000 T  0006
                              END                                                            16373000 T  0007
                         ELSE IF ELCLASS= LITNO THEN                                         16374000 T  0008
                              BEGIN                                                          16375000 T  0009
                              ADDR ← ELBAT[I],ADDRESS; STEPIT ;                              16376000 T  0010
```

```
                    END                                                                          16377000 T  0012
               ELSE ADDR ← 1 ;                                                                   16378000 T  0012
               IF Q = "4FILLO" THEN EMITC(ADDR,10) ELSE                          %E              16378500 T  0013
               IF ELCLASS = TRNSFER THEN EMITC(ADDR,ELBAT[I].OPCODE)                             16379000 T  0015
               ELSE                                                                              16380000 T  0019
               IF ELCLASS = LITV THEN                                                            16381000 T  0020
                    BEGIN                                                                        16382000 T  0021
                    EMITC(ADDR,TRP);                                                             16383000 T  0021
                    IF STEPI≠STRNGCON THEN                                                       16384000 T  0022
                         BEGIN ERR(255);GO TO EXIT END;                                          16384500 T  0023
                    IF ADDR MOD 2 ≠ 0 THEN                                                       16385000 T  0025
                         BEGIN                                                                   16386000 T  0026
                         EMIT(ACCUM[1].[18:6]);  J ← 1;                                          16387000 T  0027
                         END ;                                                                   16388000 T  0029
                    FOR K ←J+2 STEP 2 UNTIL ADDR DO                                              16389000 T  0029
                         BEGIN                                                                   16390000 T  0033
                         FOR L ←6,7 DO                                                           16391000 T  0033
                         MOVECHARACTERS(1,ACCUM[1],2+(IF J+J+1>COUNT THEN J+1                    16392000 T  0038
STACK(F+7) = *TEMPORARY STORAGE*
                                   ELSE J),T,L  );                                               16393000 T  0041
                         EMIT(T);                                                                16394000 T  0044
                    END  END                                                                     16395000 T  0045
               ELSE                                                                              16396000 T  0046
          L1:  ERR(250);                                                                         16397000 T  0046
     EXIT:END  DSS  ;                                                                            16398000 T  0047
                                                                                  112 IS   52 LONG, NEXT SEG  103

          COMMENT SKIPS  COMPILES THE SKIP BIT STATEMENT.                                        16399000 T  0000
                    IF THE REPEAT INDEX IS A LOCALID THEN A CRF IS EMITTED.                      16400000 T  0000
                    A  BSS OR BSD IS THEN EMITTED FOR SKIP SOURCE BITS (SB)                      16401000 T  0000
                    OR   SKIP DESTINATION BITS (DB) RESPECTIVELY ;                               16402000 T  0000
     PROCEDURE SKIPS ;                                                                           16403000 T  0000
PRT(724) = SKIPS
               BEGIN                                                                             16404000 T  0000
               REAL ADDR;                                                                        16405000 T  0000
                                                                                  START OF SEGMENT **********  113
STACK(F+2) = ADDR
               IF STEPI = LOCLID THEN                                                            16406000 T  0000
                    BEGIN                                                                        16407000 T  0001
                    EMITC(ELBAT[I],ADDRESS,CRF); ADDR←0; STEPIT;                                 16408000 T  0001
                    END                                                                          16409000 T  0004
               ELSE IF ELCLASS = LITNO THEN                                                      16410000 T  0004
                    BEGIN                                                                        16411000 T  0005
                    ADDR← ELBAT[I],ADDRESS; STEPIT                                               16412000 T  0006
                    END                                                                          16413000 T  0007
               ELSE ADDR ← 1 ;                                                                   16414000 T  0008
               IF ELCLASS =SBV THEN EMITC(ADDR,BSS)                                              16415000 T  0009
               ELSE                                                                              16416000 T  0011
               IF ELCLASS =DBV THEN  EMITC(ADDR,BSD)                                             16417000 T  0011
               ELSE ERR(250);                                                                    16418000 T  0014
               END  SKIPS  ;                                                                     16419000 T  0015
                                                                                  113 IS   18 LONG, NEXT SEG  103
```

```
                    COMMENT JUMPS COMPILES JUMP OUT AND JUMP OUT TO STATEMENTS.                16420000 T  0000
                            JUMP OUT TO STATEMENTS CAUSE JUMP LEVEL TO BE SET TO               16421000 T  0000
                            THE NUMBER OF LEVELS SPECIFIED. THEN THIS NUMBER OF                16422000 T  0000
                            JNS ARE EMITTED AND GOTOS IS CALLED TO COMPILE THE                 16423000 T  0000
                            JUMP INSTRUCTION.                                                  16424000 T  0000
                            SIMPLE JUMP OUTS ARE HANDLED BY EMITTING ONE JNS,ENTERING          16425000 T  0000
                            A PSEUDO STLABID IN INFO AND SETTING ELBAT[I] SUCH THAT            16426000 T  0000
                            THE GOTOS PROCEDURE WILL PERFORM THE ACTION OF SETTING             16427000 T  0000
                            UP THE LINKS FOR LATER FIX UPS. THE NEST STATEMENT CAUSES          16428000 T  0000
                            THESE FIX UPS(IF EMITTING OF JUMP INSTRUCTIONS) BY CALLING         16429000 T  0000
                            GO TOS WHEN THE RIGHT PAREN IS ENCOUNTERED.     ;                  16430000 T  0000
                PROCEDURE JUMPS;                                                               16431000 T  0000
PRT(725) = JUMPS
                        BEGIN                                                                  16432000 T  0000
                        JUMPLEVEL←1;                                                           16433000 T  0000
                            IF STEPI≠DECLARATORS THEN IF ACCUM[1]≠"3OUTOO" THEN                16434000 T  0001
                                                        FLAG(261);                             16434100 T  0004
                        IF STEPI = LITNU THEN JUMPLEVEL← ELBAT[I].ADDRESS                      16435000 T  0005
                        ELSE BEGIN                                                             16436000 T  0007
                            IF ELCLASS≠ TOV AND ELCLASS≠ STLABID THEN                          16437000 T  0010
                            BEGIN                                                              16438000 T  0011
                    COMMENT    SIMPLE JUMP OUT STATEMENT;                                      16439000 T  0012
                            IF JOINFO = 0 THEN                                                 16440000 T  0012
                                BEGIN                                                          16441000 T  0013
                                JOINFO ← NEXTINFO ;                                            16442000 T  0013
                                PUTNBUMP(STACKHEAD[0].LINK&(STLABID×2+1)                       16443000 T  0014
                                    [2:40:8]&2[27:40:8 ]);                                     16444000 T  0016
                                PUTNBUMP(0&(JOINFO-LASTINFO    )[ 4:40:8]);                    16445000 T  0018
                                PUTNBUMP (0);                                                  16446000 T  0021
                                LASTINFO ← JOINFO;                                             16447000 T  0021
                                END;                                                           16448000 T  0022
                            ELBAT[I←  I-1]←  TAKE(JOINFO)&JOINFO[35:35:13];                    16449000 T  0022
                            END; I←I-1 ;                                                       16450000 T  0026
                    END;                                                                       16451000 T  0027
                    FOR GT1← 1 STEP 1 UNTIL JUMPLEVEL DO                                       16452000 T  0027
                            EMIT(  JNS);                                                       16453000 T  0029
                    GOTOS;                                                                     16454000 T  0032
                    END JUMPS;                                                                 16455000 T  0032


COMMENT STREAMSTMT  ENVOKES THE APPROPRIATE PROCEDURE TO HANDLE                                16456000 T  0032
            THE VARIOUS AND SUNDRY STREAM PROCEDURE STATEMENTS.                                16457000 T  0032
            THE STATEMENTS ARE BROKEN DOWN AS FOLLOWS:                                         16458000 T  0032
                IDENTIFIED BY          PROCEDURE ENVOKED                                       16459000 T  0032
                  END                    GO TO FINI                                           16460000 T  0032
                  SEMICOLON              GO TO FINI                                           16461000 T  0032
                  )                      GO TO FINI                                           16462000 T  0032
                  IF                     IFS                                                  16463000 T  0032
                  GO                     GOTOS                                                16464000 T  0032
                  RELEASE                RELEASES                                             16465000 T  0032
                BEGIN                    COMPOUNDTAIL                                         16466000 T  0032
                SI,DI,CI,TALLY,LOCALID INDEXS                                                  16467000 T  0032
                DS                       DSS                                                   16468000 T  0032
                SKIP                     SKIPS                                                 16469000 T  0032
                  JUMP                   JUMPS                                                 16470000 T  0032
```

```
                    LABELID                    LABELS                                                    16471000 T  0032
                    LITERAL NO.,LOCALID(   NESTS                                                         16472000 T  0032
              UPON EXITING,STREAMSTMT ASSURES THAT "I" POINTS TO                                         16473000 T  0032
              THE SEMICOLON ,END OR ) IN SYNTACICALLY CORRECT PROGRAMS;                                  16474000 T  0032
         LABEL  L,L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,EXIT,FINI,START;                                        16475000 T  0032
         SWITCH TYPE ← FINI,L,FINI,L3,L4,L5,L6,L7,L7,L7,L7,L8,L9,L10;                                    16476000 T  0032
    START:    GO TO TYPE[ ELCLASS-ENDV+1];                                                             16477000 T  0043
              IF ELCLASS= RTPAREN THEN GO TO FINI ;                                                    16478000 T  0047
              IF ELCLASS= STLABID THEN GO TO L2 ;                                                      16481000 T  0048
                                                                                                       16482000 T  0049
         IF ELCLASS <IDMAX AND ELCLASS≠LOCLID THEN BEGIN                                               16482100 T  0049
              DECLARELABEL; GO TO L2;  END;                                                            16482200 T  0051
         IF ELCLASS = LITNO OR ELCLASS = LOCLID AND TABLE(I+1)                                         16482300 T  0052
               = LFTPAREN THEN GO TO L1;                                                               16482400 T  0055
         IF ELCLASS = LOCLID  THEN  GO TO L7;                                                          16482500 T  0057
       L:     ERR( 250 ); GO TO FINI ;                                                                16483000 T  0058
       L1:    NESTS;     GO TO EXIT;                                                                  16484000 T  0060
       L2:    LABELS;    GO TO START;                                                                 16485000 T  0062
       L3:    IFS;       GO TO FINI;                                                                  16486000 T  0063
       L4:    GOTOS;     GO TO EXIT;                                                                  16487000 T  0064
L5:
       L6:    I←I+1 ; COMPOUNDTAIL; GO TO FINI;                                                       16488000 T  0065
       L7:    INDEXS;    GO TO EXIT;                                                                  16489000 T  0065
       L8:    DSS;       GO TO EXIT;                                                                  16490000 T  0067
       L9:    SKIPS;     GO TO EXIT;                                                                  16491000 T  0069
       L10:   JUMPS;     GO TO EXIT;                                                                  16492000 T  0070
     EXIT:    STEPIT;                                                                                 16493000 T  0071
     FINI:    END   STREAMSTMT;                                                                       16494000 T  0072
                                                                                                       16495000 T  0072
                                                                                  103 IS    74 LONG, NEXT SEG    3


                    MOVE(1,I,CODE(0));                                                                 16495100 T  0594
                    TIME1 ← TIME(1); PROGRAM;                                                          16495200 T  0599
                    ENDOFITALL:END MAIN BLOCK                                                          16495210 T  0600
                    END.                                                                               16495300 T  0601
PRT(726) = *SEGMENT DESCRIPTOR*

                                                                                    3 IS   604 LONG, NEXT SEG    2
                                                                                    2 IS    41 LONG, NEXT SEG    1
PRT(527) = EXP INTRINSIC, SEGMENT NUMBER =  114.
PRT(526) = LN INTRINSIC, SEGMENT NUMBER =  115.
PRT(343) = OUTPUT(#) INTRINSIC, SEGMENT NUMBER =  116.
PRT(5) = BLOCK CONTROL INTRINSIC, SEGMENT NUMBER =  117.
PRT(530) = X TO THE I INTRINSIC, SEGMENT NUMBER =  118.
PRT(373) = GO TO SOLVER  INTRINSIC, SEGMENT NUMBER =  119.
PRT(14) = ALGOL WRITE   INTRINSIC, SEGMENT NUMBER =  120.
PRT(15) = ALGOL READ    INTRINSIC, SEGMENT NUMBER =  121.
PRT(16) = ALGOL SELECT  INTRINSIC, SEGMENT NUMBER =  122.
PRT(411) = DYNAMIC DIALS INTRINSIC, SEGMENT NUMBER =  123.
PRT(201) = FILE ATTRBUTS INTRINSIC, SEGMENT NUMBER =  124.

                                                                                    1 IS    2 LONG, NEXT SEG    0
                                                                                  125 IS   69 LONG, NEXT SEG    0
NUMBER OF ERRORS DETECTED =   0,  COMPILATION TIME =  816 SECONDS.

PRT SIZE = 471; TOTAL SEGMENT SIZE =  7276 WORDS; DISK SIZE = 360 SEGS; NO. PGM. SEGS = 125

ESTIMATED CORE STORAGE REQUIRED = 13928 WORDS.
```

ESTIMATED AUXILIARY MEMORY REQUIRED =   0 WORDS.

NUMBER OF CARD-IMAGES PROCESSED =   6918.

A
0003 *02062500*
02063500

A
0003 *02054000*
02055000

A
0003 *02022000*
02023000

A
0003 *03050000*

A
0003 *04311000*
04311050 04311060 04315000 *04315000*

A
0003 *03049000*

A
0003 *03047100*

A
0003 *04500000*
04529000 04534000 04542000 04548000 04550000 04551000 04556000

A
0012 *02264100*
02265100 02265150 02265250 02265300 02265600 02265650 02265700 02265750 02265850 02265900 02265950
02266000

```
A
  0036 *05130000*
       *05131000* 05132000

A
  0035 *05065000*
        05065000  05066000

A
  0041 *05334100*
        05334100  05411100

A
  0043 *05415000*
        05417000  05419000  05421000  05422000

A
  0043 *05413000*
        05426000  05427000  05431000  05432000
```

```
A
  0065 *08015000*
        08181000  08188000

A
  0065 *08027000*
       *08035000**08037000*

A
  0065 *08020000*
        08021000

A
  0089 *13359000*
       *13364000* 13365000  13366000

A
  0092 *13673100*
        13673100
```

ABS
0035 *05065000*

05066000 05092000

ACC
0063 *07653500*

07654500 08027000 08083000 08084000

ACCUM
0003 *01304000*

02128500 02251000 02255000 02259000 02278000 02280000 02327000 02366100 02371000 02499000 02502000
02581000 02590000 02592000 02602600 02647000 02650000 02652000 02686000 02686500 02696000 02698000
02700000 02705000 02754000 02822000 02846000 02865000 02881000 02958500 05043000 05044000 05325460
07086200 07086300 07668500 07670000 07671000 10264700 10264900 10269000 10274000 12117000 12118000
12119000 12120000 12122000 13281000 13282000 13307000 13308000 13309000 13310000 13372000 13755000
14259000 14259080 14440000 14441000 16000700 16211000 16387000 16392000 16434000

ACCUM
0003 *02041000*

02043000

ACCUM
0007 *02089500*

02122500

ACCUM
0034 *05016000*

05028000

ACCUM
0054 *07038100*

07038500 07536000 07537000

ACCUM
0083 *10236000*

10239000

ACLASS
0054 *07038000*

07041000 07047000 07057000 07069000 *07081000* 07089000

ACTUALPARAPART
0003 *07035000*

07410000 07438000

ADD
0003 *01633000*

05226000 07525000 08126000 08214000 13221000

ADD
0087 *13221000*

13248000 13250000 13252000 13252500 *13259000*

ADDC
0003 *13212000*

ADDITIONAL
0062 *07595000*

07601000 07603000 07610000

ADDOP
0103 *16005000*

16326000

ADDR
0105 *16068000*

ADDR
0109 *16195000*

*16196000**16201000**16204000* 16208000

ADDR
0113 *16405000*

*16408000**16412000**16414000* 16415000 16417000

ADDR

0112 *16364000*

*16371000**16376000**16378000* 16378500 16379000 16383000 16385000 16389000

ADDR
0111 *16314000*

*16321000**16344000* 16352000

ADDRES
0065 *08010000*

08055000 08067000 08068000 08075000 08179000 08212000

ADDRESS
0003 *01152000*

02850000 02896000 02900000 02923000 04508000 05131000 05217000 05233000 05280000 05281000 05344570
05344615 05344620 05344660 05344700 05344710 06016000 06072000 06108000 06185000 06324000 06325000
06326000 07071000 07090000 07393000 07439000 07745000 07768100 07994000 07995000 08021000 08029000
08044000 08055000 10277000 12015000 12024000 12036000 12047000 12108000 13205000 13248000 13259000
13394000 14085000 *14115000* 14129000 14349000 14399000 14430000 14437000 14461200 14483200 15094000
15097000 15106100 15107000 15110200 15110300 15233010 15233011 15233014 15267000 15276010 15276060
15276080 15290000 15302000 15310000 15344100 15344200 15351200 15352000 15353000 16121000 16124000
16200000 16204000 16321000 16329000 16342000 16344000 16370000 16376000 16408000 16412000 16435000

ADDRESS
0003 *04022000*

04023000 04024000

ADDRESS
0003 *04017000*

04018000 04019000

ADDRESS
0003 *04030000*

04032000 04033000

ADDRESS
0003 *05199000*

05201000

ADDRESS
0055 *07393000*

07397000 07397400 07397500 07413000

ADDRSF
0003 *01604000*

07397000 07677000 *13205000**13344000**13346000**13347000* 13348000 13373000 *13767000* 13767000 13768000
13770000 14313000 14320000 *16000500*

ADDVALUE
0003 *01000860*

02193000 *02592000**09028920*

ADES
0003 *01299010*

ADJUST
0003 *04083000*

04087000 04144000 04172000 06208500 07080000 07459000 07579000 07585100 07586000 07600200 07607100
07727040 08190000 13605000 13611000 13630000 16023100

ADJUST
0103 *16023100*

16137000 16158000

ADOP
0003 *01278000*

05344550 05344670 06013000 07666000 08029000 08031000 12013000 12022000 16005000 16326000

ADR
0091 *13633000*

13633000 13642000 13657000

ADRES
0037 *05215000*

05217000 05219000 *05219000* 05226000 05227000

AEXP
0003 *03001000*

03006000 *06002000* 06007000 06057000 06105000 06174000 06202000 06305000 06410000 06416000 06422000

07045000 07062000 07086000 07487000 07493000 08101000 08123000 08131000 08135000 08142000 12046000
12051000 12054000 13759000 15099000 15233012 15276070 15277000 15307000

AGAIN
0016 *02360000*

02372000 02377000 02390000 02393000 02395000 02397000 02404000 02418000 02421100 02431000 02460000
02470000 02472000 02474000 02480000 02493000 02496000 02501000 02506000 02508000 02513000 02516000
02520000 02522000 02564000 02567000 02601000

AGAIN
0044 *05439000*

05445000 05454000 05457000

AGAIN
0064 *07726000*

07727000 07727100 07741000

AGIN
0003 *02001860*

02001880 02001886

AJUMP
0003 *01596000*

13603000 *13614000* 13626000 *13629000* 14054000 *14055000**14606000*

AJUMPO
0094 *14034000*

14054000 14606000

AKKUM
0003 *01693000*

ALFAARRAYID
0003 *01204000*

ALFAID
0003 *01200000*

14139000

ALFAPROCID
  0003 *01196000*

ALFASTRPROCID
  0003 *01192000*

ALFAV
  0003 *01284000*

ALL
  0065 *08052000*
        08054000 08055000 08056000 08057000

ALLTHU
  0028 *04168000*
        04176000 *04186000*

ALONG
  0017 *02325000*
        02336000 02340000 02354000 *02358000*

ALPHADEC
  0094 *14013000*
        14018000 *14139000*

ALPHASIZE
  0003 *01707000*

ALPHAWORDS
  0012 *02264700*
        02264900 02265600 02265850

AMPERSAND

```
  0003 *01266500*
        06034000  06180000  07084000  15110200  15349000

ANDOP
  0003 *01276000*
        02964500  02971000  02983500

ANOTHER
  0053 *07007000*
       *07009000* 07012000  07015000  07024000

ANOTHER
  0054 *07037000*
       *07041000* 07095000

ARGH
  0021 *02638000*
       *02649000* 02685000  02718000  02753000  02899000

ARITHCOMP
  0003 *03004000*
        06036000 *06041000* 06052000  06055000

ARITHSEC
  0003 *03002000*
        06006000 *06011000* 06023000

ARRAE
  0003 *13376000*
        13410000  14156000

ARRAYDEC
  0094 *14015000*
        14020000 *14156000*

ARRAYMONFILE
  0003 *01526000*
```

0003 *02054000*

02055000

B

0003 *01717000*

B

0003 *02062500*

02063500

B

0003 *04247000*

04248000

B

0003 *02980000*

$0_2$983500 *$0_2$983500* $0_2$984000 $0_2$984500

B

0003 *03047100*

B

0003 *04311000*

04311070 04311080 04317000 *04317000*

B

0003 *04500000*

$04_26$100 $04_28$000 $04_24$2000 04544000 04548000 $04_25$1000 04556000

B

0003 *02955500*

B

0003 *03041000*

B
0003 *03050000*

B
0017 *02323000*
02343000 02352000 02353000 *02356000* 02357000

B
0024 *02973500*
*02976000**02978000**02978500* 02978500 02979000

B
0023 *02970000*
02970500 02971000 02971500

B
0033 *04505000*
04507000

B
0065 *08027000*
*08035000**08037000*

B
0065 *08010000*
08044000 08047000 08089000 08112000 08116000 *08122000**08169000* 08183000 08199000 *08209000* 08213000

B
0077 *09393240*
09393300 09393310 09393330 09393350 09393370 *09393370* 09393400 *09393410* 09393410 09393430 *09393440*
09393440

BACK
0058 *07491000*
07493000 07495000

BACK
  0066 *08084000*

       08089000

BACK
  0082 *10259000*

      10263300 10276500 10278000 10279000

BACKFIX
  0066 *08091000*

      *08097000**08099000**08106000* 08127000 *08146000**08147000* 08157000 08168000 08171000

BAE
  0003 *03046000*

      *07458000**07459000* 07459000

BANA
  0003 *03048000*

       05225000 *06419000* 06423000 07523000

BASE
  0076 *09370000*

       09371000 09374000

BASENUM
  0003 *01000860*

       02192500 *02581000**09028920*

BATMAN
  0003 *01001600*

      02961500 02966500 *02967000* 02968000 02978000

BBC
  0003 *01634000*

       07487000 08221000

BBW
  0003 *01635000*

       06207000 07444000 07495000 08047000 08078000 08166000 08168000

BEGINCTR
0003 *01499000*

07008000 *07008000**07021000* 07021000 07023000 *07024000**14035000* 14035000 *14510000* 14510000 *14520000*
14520000

BEGINV
0003 *01234000*

05108000 09252000 13771000 14393000 14479000

BETA1
0090 *13379000*

13390000 13409000

BEXP
0003 *03006000*

06410000 07487000 07493000 08135000 08142000

BFC
0003 *01636000*

06301000 07496000 07572000 07573000 07575000 07579000 07583000 07585100 08151000

BFW
0003 *01637000*

04218000 06307000 07088000 07512000 07516400 07518000 07526000 07529000 07572000 07586000 08125000
08127000 08146000 08148000 08215000 08228000 13263000 13274000 13606000 14041000 14451000

BIT
0003 *02307000*

02311000

BIT
0103 *16015000*

16216000

BITEDUST
0035 *05067000*

05087000 05088000 05089000 05090000

```
BITOP
  0003 *01269000*
       06087000

BLANKET
  0003 *01737350*
       01737500  02265100  02265250  02265600  02265700  02265850  02265950  05325450  09034500

BLKAD
  0094 *14026000*
       *14037000**14044000* 14600000

BLOCK
  0003 *03067000*
       07750000  07768200 *14001000* 14482000  14613000  16495210

BLOCKCTR
  0003 *01430000*

BNS
  0103 *16011000*
       16122000  16124000

BNSFIX
  0107 *16118000*
       *16121000* 16141000

BOOARRAYID
  0003 *01202000*


BOOID
  0003 *01198000*
       02967000  02977500  07047000  08057000  08058000  12042000  14140000

BOOLCOMP
  0003 *02955500*
       02971000 *02980000* 02983000  02985000
```

BOOLEANDEC
  0094 *14013000*

       14018000 *14140000*

BOOLEXP
  0003 *02065600*

       02343000 *02969000**02971500* 02972000 02976000

BOOLPRIM
  0003 *02955000*

       02970500 *02972500**02979000* 02979500 02982500

BOOPROCID
  0003 *01194000*

       14294000

BOOSTRPROCID
  0003 *01190000*

       14285000

BOOV
  0003 *01282000*

BOTTOM
  0044 *05439000*

       05446000 05451000 *05455000*

BRANCH
  0003 *04116000*

      04120400 04120400 04122000

BRANCH
  0003 *03037000*

BRANCHTYPE

0003 *03025000*

BRANCHTYPE
0003 *07537000*
07540000 07544000

BRET
0065 *08011000*
08016000 *08033000* 08033000 08034000 08035000 08046000 08094000 *08124000**08180000**08223000**08232000*

BRNCH
0066 *08092000*
08148000 *08152000*

BSD
0103 *16021000*
16417000

BSS
0103 *16020000*
16415000

BTYPE
0003 *01452000*

BUG
0004 *01802000*
04279000 04282500

BUILDLINE
0002 *00504700*
02526000 02526000

BUMPL
0003 *01477000*
04216000 06207000 06297000 06300000 07444000 07459000 07487000 07493000 07495000 07540000 07577000

07585000 08047000 08078000 08106000 08146000 08168000 08190000 08221000 13628000 14039000

BUP
0003 *01695000*

13333000 13333000 14073000 *14079000**14083000* 14084000 14088000 *14088000* 14088000 14116000 14118000
14118000 14118000 *14364000**14369000*

B2D
0003 *01717000*

04149000 *04247000**04248000* 04279000 04281500 04282000 04282500 04285000 04285500 05376100 09434000
09447080

C
0003 *01562000*

02682000 02686000 *02701000* 02705000 *02758000* 02789000 *02789000**02801000**02803000* 02803000 02811000
02811000 02827000 *02842000* 02842000 02849000 02850000 *02923000* 06087000 06090000 06093000 06099100
06099200 06168000 07600300 07604120 07667000 *07667000* 07669000 07727050 08035000 16212500

C
0003 *03047100*

C
0003 *04203000*

04205000 04209000 04211000 *04216000* 04218000

C
0046 *06064000*

06064000 06066000 06194000 07035000 07036000

C
0041 *05334100*

05334100 05334500 05400000 05411100

C
0065 *08020000*

08021000

C
0092 *13673100*

13673100 13673350

CALL
0003 *01721000*
05402100 05405000 05407000 05409000 05428000 05430000 07397400 07397500 08072000 14483400

CALL
0065 *08072000*
08079000 08098000 08108000 08130000 08217000

CALLA
0003 *01720000*
01721000 05402100 05405000 05407000 05409000 05428000 05430000 07397400 07397500 14483400

CALLINFO
0003 *01722000*
07397300 *14481100* 14483300 14483400

CALLSTATEMENT
0094 *14017000*
14132000 14137300 *14505000*

CALLSWITCH
0003 *05315000*
05316000 13263000

CALLX
0003 *01722000*
07397300 *07397500* 07397500 14481100 *14481100* 14483500

CALL1?
0003 *01721000*
05402100 05405000 05407000 05409000 05428000 05430000 07397400 07397500 14483400

CARD
0003 *01557000*
02213750 02215500 02216250 09281500

CARD

0003 *02188000*

02188000 02188500 02202750

CARDCOUNT
0002 *00504150*

02234500 *02234500*

CARDLAST
0009 *02210000*

02210750 *02216000*

CARDNUMBER
0002 *00504100*

02228000

CARDONLY
0009 *02210000*

02210750 *02215250*

CARDOPTION
0016 *02363000*

02429000 02436000

CBUFF
0003 *01561056*

02213750 02214000 02215500 02215750 02216250 02216500 07025010 07029000

CD
0063 *07653500*

07654500 07655000 08083000 08084000

CDC
0003 *01638000*

15302000 15356000

CELL
0040 *05325040*

05325050 05325130 05325150 05325180 05325220 05325330 05325350 05325360 05325380 05325390 05325400

05325410 05325420

CHANGESEQ
0003 *01741200*

01741300 02193250

CHAR
0003 *01299040*

14461100

CHAR
0082 *10234000*

10242000 10252000 10255000

CHARCOUNT
0082 *10230000*

10242000 10250000 10251000 10256000 *10256000**10261000* 10285000

CHECKBIT
0003 *01000920*

01001210 02496000

CHECKER
0003 *05113000*

05120000 05216000 07070500 07398000 08054000 12044000 15085000

CHECKTOG
0003 *01001210*

CHKSOB
0003 *13209000*

14289000

CHS
0003 *01639000*

06021000 08189000 08219000

CIV

0003 *01238000*

CLASS
0003 *01148000*

*02705100**02850000**02851000* 02886000 02920000 02924000 *02966500* 02968000 04509000 05350120 07044000
08057000 12120000 12125000 13201000 13226000 13677200 14062000 14091000 14425000 14519000 15086000
15102000 15128100 15233009 15276020 15276090 15295100 15303000 15333000 15344000 15351000

CLCR
0003 *01330000*

*02216500**02217750* 02218000 02218250 02224500

CLCR
0009 *02202750*

02203500 02209000

CNT
0003 *02001838*

02001838 02001844

CNT
0003 *02001858*

02001858 02001866 02001870 02001892

COC
0003 *01640000*

15095000 15290000 15344000 15355000

COCT
0063 *07653500*

07656500 07657000 07671000

CODE
0003 *01556900*

04147000 04296000 05312000 07662000 07668500 07669000 07671500 08999200 08999575 08999700 09393350
09393400 09393430 09400000 09401000 13651000 13746000 16495100

CODE

0003 *04130000*

04130000 04134000 04231000

CODISK
0003 *01561300*

08999150 08999175 08999275 08999300 08999350 08999375 08999550 08999575 08999600 08999700 08999725
09398000

COL
0041 *05344000*

COLON
0003 *01260000*

02637000 05276000 06318000 06320000 07597500 07600100 07600600 07604020 07727010 07727030 07727070
13753100 13756000 13760000 16160000

COLON
0021 *02637000*

02641000 *02663000*

COM
0003 *01641000*

COMCOUNT
0007 *02089500*

02090500 02111500 02118000

COMMA
0003 *01261000*

06114000 07067000 07095000 07604140 07673000 08145000 08150000 08153000 10263700 10282000 12010000
12020000 12033000 12052000 12056000 12125000 13351000 13403000 13409000 13756000 13762000 14197000
14223000 14259110 14268000 14356000 14404000 15287000 15293000

COMMANTS
0007 *02092000*

02097500 02105000

COMMENTS

```
  0007 *02092000*
         02106500  02107500  02108000

COMMENTV
  0003 *01253000*
         02886000  02920000  08034000

COMMON
  0054 *07037000*
         07045000  07054000  07058000  07064000  07074000  07076000  07084000 *07094000*

COMPAR
  0009 *02210500*
         02216750  02218500 *02224250*

COMPARE
  0003 *02188000*
         02190000  02190250  02203500  02226000  02228150  02232000  02233500  02411000  02486000

COMPARECODE
  0109 *16192000*
         16208000

COMPLETE
  0021 *02637000*
         02658000  02667000  02680000  02705200  02715000  02757000  02852000  02877000  02877020  02893000  02894000
        *02909000*

COMPOST
  0021 *02639000*
         02880000 *02882000*

COMPOUNDTAIL
  0003 *07006000*
         07033000  07768140  07768200  13772000  14412000  14512000  14524000  16489000

CONSTANA
  0065 *08013000*
         08015000  08074000  08099000  08181000  08188000 *08211000**08224000**08227000*
```

CONSTANB
0065 *08013000*

08043000 08112000 08116000 *08121000* 08183000 08194000 *08205000* 08213000

CONSTANC
0065 *08013000*

08185000 08195000 *08204000* 08218000

CONSTANTCLEAN
0003 *03034000*

04163000 04217000 07459000 07496000 07519000 07585000 08170000 08190000 13744000 14040000 14168500
14604000

CONV
0003 *02041000*

02043000 02228000 02251000 02255000 02259000 02502000 02581000 02592000

CONVERT
0003 *02248000*

02260000 02758000 02796000 02803000 02823000

COP
0003 *01371000*

02550000 04280000 04283500

CORADR
0003 *01344000*

05248500 05250000 *05250000* 09282600 09388000 13653000

CORE
0076 *09370000*

09371000 09375000 09393240

CORNER
0003 *02011000*

02011000 02012000

COUNT

```
  0003 *01313000*

       02128500  02129000  02251000  02252000  02278000 *02327000* 02366200 *02371000* 02373000 *02499000**02590000*
      *02602700**02647000* 02685000  02686000 *02691000* 02695000 *02698000* 02698000  02699000 *02699000* 02703000
       02705000  02790000  02796000  02798000  02803000  02810000  02817000  02880000  02881000 *02889000**02958500*
       05325070  06062000  07668000  07670000  07671000  10241000  12109000 *12121000* 13302000  14259010  16392000


COUNT
  0003 *02062500*

       02062500  02064000


COUNT
  0007 *02090000*

       02121500


COUNT
  0034 *05016000*

       05017000  05028000


COUNT
  0035 *05067000*

       05067000  05071000  05077000  05080000


COUNT
  0046 *06062000*

       06074000  06075000 *06075000* 06103000  06112000 *06112000* 06119000 *06119000* 06120000  06121000  06122000


COUNT
  0040 *05325070*

       05325150  05325160  05325210  05325320  05325330


COUNT
  0083 *10241000*

      *10242000* 10245000 *10247000* 10247000  10249000  10252000  10255000  10256000


COUNT
  0083 *10236000*

       10237000  10240000
```

```
COUNTV
  0007 *02089500*
         02090500  02121000  02123500

CPLUS1
  0003 *01713000*



CPLUS2
  0003 *01714000*



CREL
  0028 *04167000*
        *04173000**04183000* 04187000

CRF
  0103 *16010000*
         16121000  16200000  16370000  16408000

CROSSHATCH
  0003 *01262000*
         02640000  10279000

CROSSHATCH
  0021 *02640000*
         02642000 *02714000*

D
  0003 *02001858*
         02001890  02001895  02001896

D
  0003 *02001838*
         02001842  02001848  02001850

D
  0003 *03047100*
```

```
D
  0012 *02264400*
         02264500

D
  0012 *02264700*
         02264800

D
  0028 *04166000*
       *04175000* 04184000

D
  0063 *07658500*
         07659500  07660000  08020000  08027000  08052000  08063000  08072000  08080000  08083000  08084000

D
  0082 *10257100*
         10257200  12101000

D
  0085 *12103000*
         12104000

D
  0098 *14254200*
         14254300

D
  0105 *16067000*
       *16070000* 16079000

D
  0104 *16031000*
       *16032000* 16032000

DA
```

```
  0003 *01559020*


DATE
  0005 *01823000*

         01823000  01825000


DATER
  0005 *01823000*

         01825000  01827000  01833000


DATIME
  0003 *01820000*

         01837000  02183000  02195000  02195750  02214200  02228750  02264950  02421000  05039600  05325490  07025020
         13653500  13676000


DBLSTMT
  0003 *03060000*

         07753000 *12002000* 12059000


DBV
  0003 *01243000*

         16417000


DCINTYPE
  0003 *01561420*

         09430060


DCV
  0003 *01248000*

         16208000  16338000


DDES
  0003 *01299000*

         07677000


DEBLANK
  0007 *02091500*

         02094500  02100000
```

DEBUG
  0003 *04277500*
        04286000 04297000

DEBUGBIT
  0003 *01000930*
        01001220 02531000 02533000 02602500 04148000 04297000

DEBUGTOG
  0003 *01001220*
        02533000 02602500 04148000 04297000

DEBUGWORD
  0003 *04130000*
        04149000

DECK
  0003 *01561500*
        09435000 09447000 09447090

DECKBIT
  0003 *01000940*
        01001230 02474000 09405000 09414010 09421000 09447010 13651100

DECKTOG
  0003 *01001230*
        09405000 09414010 09421000 09447010 13651100

DECLARATORS
  0003 *01214000*
        07767000 10277000 14126000 14370000 14396000 14516000 16434000

DECLARELABEL
  0003 *16000100*
        16253100 16482200

DECLSW

```
  0094 *14018000*

       14134000

DEFINEARRAY
  0003 *01491000*

       02202030  02202040  02202050  02202060  02212500  02217250  02217500  02217750  02218000  02366100  02602600
       02719000  02721000  02900000  02902000  02903000

DEFINECTR
  0003 *01481000*

       02715000 *10262000* 10278000 *10278000* 10280000  10283000 *10283000**10284000*

DEFINEDEC
  0094 *14017000*

       14021000 *14254000*

DEFINEDID
  0003 *01180000*

       02894000  12120000  14259010

DEFINEGEN
  0003 *10228000*

       10286000  12124000  14266000

DEFINEINDEX
  0003 *01481000*

       02212500  02718000  02719000 *02721000* 02721000  02898000  02900000  02902000  02903000 *02905000* 02905000


DEFINEV
  0003 *01298000*

       10278000

DEL
  0003 *01642000*



DEPTH
  0003 *05400000*
```

*05402100**05403000* 05407000 *05409000* 05426000 05431000

DESC
0065 *08016000*
08098000 08108000

DEST
0003 *02045000*
02048000

DIA
0003 *01680000*
04315000 04550000

DIALA
0003 *01502000*
*04315000**08152000*

DIALB
0003 *01502000*
*04317000**08152000*

DIB
0003 *01681000*
04317000

DIQ
0003 *01237000*

DISK
0003 *01561400*
01561400 01561600 01828500 09425900 09429000 09430075 09430100

DISK
0076 *09370000*
09371000 09376000

DISKADR

```
  0003 *01344000*
       05252000 *05254000* 05254000 *09253100* 09389000  13653000

DOIT
  0041 *05334100*
       05350160  05350200

DOIT
  0092 *13673100*
       13673450  13677300

DOLLAR
  0021 *02639000*
       02641000 *02729000*

DOLLARCARD
  0003 *02065500*
       02229000  02231000 *02319000* 02604000  02730000

DOLLARTOG
  0003 *01001540*
       *02366000**02603000*

DOLLAR2TOG
  0003 *01691500*
       *02211800**02230100* 02234600  02421000

DOSTMT
  0003 *07482000*
       07488000  07759000

DOT
  0003 *03015000*
       06178000 *06338000* 06346000

DOT
  0021 *02637000*
       02641000 *02668000*
```

```
DOTSYNTAX
  0003 *05270000*
       *05286000* 05286000  06341000  15112000  15233005  15276150  15341000

DOV
  0003 *01225000*
        07494000  08159000  08186000

DPTOG
  0003 *01690000*
        02253000  02784000  02794000  02829000 *12012000**12016000*

DSKIP
  0003 *02045000*
        02046000  02049000

DSS
  0103 *16362000*
        16398000  16491000

DSV
  0003 *01240000*


DTYPE
  0003 *01452000*


DUMMY
  0003 *01001470*


DUMPDEC
  0094 *14014000*
         14019000 *14149000*

DUMPEE
```

```
  0003 *01547000*


DUMPINFO
  0003 *02264000*
         02266100  02468000

DUMPOR
  0003 *01552000*


DUMPV
  0003 *01287000*


DUP
  0003 *01643000*
         07524000  15095000  15306000

DWA
  0012 *02264100*
         02265500  02265600  02265700

E
  0003 *03054000*
         04099000  12123000 *13293000* 13333500  13342000  13755500  16000600

E
  0003 *04500000*
         04526200  04530000  04536000  04543000  04549000  04555000

E
  0026 *04099000*
         04100000  04102000 *04112000*

E
  0033 *04504000*
         04506000  04508000 *04509000* 04509000  04522000
```

E
0085 *12103000*
12103000 12105000

E
0103 *16065000*
16070000

E
0103 *16085000*
16089000 16093000

EDITLINE
0003 *02183500*
02187000 02195000 02195750 02214200 02228750 02421000 05038000 07025020

EDOC
0003 *13683000*
13685000 13687000 13693000 13715000 13716000 13732000 13733000 14002000 14003000

EDUCINDEX
0003 *01710000*

EL
0046 *06061900*
06104300 06107200

EL
0085 *12102000*
12112000 12114000 *12125000* 12126000

ELBAT
0003 *01319000*
02265100 02265250 02910000 02916000 02920000 02923000 02924000 05089000 05189000 05216000 05217000
05218000 05222000 05280000 05281000 05344570 05344615 05344620 05344660 05344700 05344710 05350120
05350160 05350180 06016000 06044000 06072000 06104300 06108000 06166000 06185000 06324000 06325000
06326000 07031000 07070000 07070500 07071000 07396000 07430000 07516000 07516300 07518000 07521000

07554000 07566000 07572000 07583000 07599000 07667000 07745000 07768100 07994000 07995000 08029000
08034000 08037000 08176000 09424000 09425900 09427500 09429000 09432000 09433000 09434000 09435000
09437000 09447000 09447010 09447070 09447080 09447090 10277000 12015000 12024000 12036000 12044000
12125000 13200000 13361000 13372000 13394000 13767000 13768000 14062000 14129000 14220000 14221000
14314000 14349000 14352000 14353000 14399000 14519000 15077000 15276010 16000700 16121000 16124000
16159000 16200000 16204000 16208000 16211000 16212500 16254000 16321000 16329000 16342000 16344000
16370000 16376000 16379000 16408000 16412000 16435000 16449000

ELBATWORD
0003 *05113000*

05116000 05118000

ELBATWORD
0003 *05207000*

05208000 05209000

ELBATWORD
0062 *07595000*

07599000 07601000

ELBW
0059 *07503000*

07521000 07527000

ELCLASS
0003 *01328000*

05002000 *05003000* 05109000 05304000 05344550 06004000 06013000 06034000 06036000 06048000 06052000
06055000 06076000 06084000 06104100 06105000 06106300 06107100 06114000 06115000 06138000 06139000
06141000 06146000 06151000 06175000 06180000 06198000 06302000 06316000 06340000 06411000 06416000
06422000 07012000 07013000 07018000 07021000 07026000 *07032000* 07052000 07053000 07075000 07084000
07095000 07096000 07399000 07404000 07409000 07431000 07485000 07494000 07506000 07507000 07511000
07550000 07553000 *07556000* 07578000 07600300 07600400 07604110 07665000 07665500 07666000 07666500
07668000 07727000 07727010 07727030 07727050 07728000 *07749000* *07751000* *07768130* 08031000 08032000
08038000 08103000 08113000 08128000 08133000 08139000 08145000 08150000 08153000 08159000 08182000
08184000 08186000 08229000 10258100 12022000 12028000 12033000 12035000 12039000 12042000 12052000
12056000 13220000 13331000 13337000 *13338000* 13338000 13391000 13400000 13403000 13409000 13756000
13758000 13760000 13762000 13763000 14137000 14165000 14197000 14259020 14259120 14260000 14325000
14326000 14336000 14357000 14370000 14396000 14404000 14479000 *14516000* 15078000 15083000 15110200
15111000 15233003 15233026 15276010 15276140 15287000 15294000 15339000 15349000 16119000 16131000
16132000 16198000 16204000 16210000 16212500 16213000 16230000 16253000 16315000 16319000 16320000
16326000 16330000 16335000 16336000 16337000 16338000 16339000 16341000 16374000 16379000 16381000
16410000 16415000 16417000 16437000 16477000 16478000 16481000 16482100 16482300 16482500

ELCLASS
0082 *10258100*

10263400 10263700 10264100 10264300 10264410 10277000 10279000 *10282000*

ELCLASS
  0087 *13220000*

13226000 13246000 13278000

ELSEBRANCH
  0050 *06295000*

06300000 06307000

ELSEV
  0003 *01227000*

06302000 07567000 07578000 16220000 16230000

EMIT
  0003 *03028000*

04004000 04007000 04014000 04019000 04024000 04086000 04221000 *04291000* 04302000 04315000 04317000
04318000 04531000 04535000 04550000 04551000 04554000 06054000 06054100 06082000 06087000 06089000
06099100 06104600 06148000 06166000 06191000 06308000 07093100 07431000 07436000 07544000 07604070
07995000 08049000 08050000 08122000 08148000 08151000 08227000 14451100 15376700 16080000 16122000
16133000 16219000 16232000 16266000 16387000 16394000 16453000

EMITB
  0003 *03036000*

04114000 04125000 04218000 06207000 06301000 06307000 07088000 07444000 07487000 07495000 07496000
07540000 07579000 07585100 07586000 07608000 08047000 08078000 08086000 08127000 08146000 08168000
08215000 08221000 13606000 14041000

EMITC
  0003 *04010000*

04014000 14451000 16032000 16079000 16121000 16124000 16200000 16208000 16211000 16212500 16213000
16216000 16342000 16352000 16370000 16378500 16379000 16383000 16408000 16415000 16417000

EMITD
  0003 *03050000*

04311000 04322000 04556000 05203000 06332000 07093000 07093200 15100000 15233013 15276070 15308000

EMITI
  0003 *04500000*

06344000 15124000 15233018 15276210 15370000

EMITJUMP
0103 *16065000*

16093000 16255000

EMITL
0003 *04003000*

04032000 04205000 04544000 05222000 06110000 06206000 08046000 08067000 08077000 13264000 13267000
13773000 14506500

EMITLNG
0003 *04098000*

04112000 06148000 07573000 07575000 08150000 08220000

EMITN
0003 *04022000*

04024000 04212000 04524000 05202000 07092000 08075000 08179000 08212000 12036000 15094000 15110200
15233011 15276080 15302000 15344100 15351200 15353000 16261000

EMITNUM
0003 *03049000*

04121000 *04203000* 04225000 06168000 07089000 07442000 08021000 08089000 12014000 12015000 13229000

EMITO
0003 *04006000*

04018000 04023000 04033000 04034000 04109000 04112000 04122000 04212100 04223100 04311090 04507000
04531000 04537000 04545000 05226000 05316000 06021000 06077000 06110300 06172500 06187000 06203000
07078000 07087000 07091000 07408000 07434000 07512000 07516400 07524000 07525000 07526000 08044000
08049000 08050000 08065000 08068000 08070000 08079000 08094000 08123000 08125000 08126000 08166000
08189000 08214000 08219000 08228000 12024000 12038000 12054000 13263000 13264000 13268000 13740000
13743000 13752000 13761000 13765000 13766100 14167000 14389000 14451000 15095000 15106100 15106200
15276090 15290000 15302000 15303000 15306000 15309000 15310000 15333000 15344000 15344100 15351200
15351300 15354000

EMITPAIR
0003 *04028000*

04034000 04523000 05201000 05227000 07071000 07090000 08090000 13214000 13230000 15107000 15233010
15233014 15267000

EMITSTORE
0003 *13214000*

13230000

```
EMITV
  0003 *04017000*
       04019000  04184000  04184500  04521000  05219000  05226000  05316000  07413000  07439000  07516300  08021000
       08044000  08075000  08166000  08212000  12047000  13236000  13739000  15097000  15110300  15276060  15290000
       15344200  15352000

EMITWORD
  0003 *04142000*
       04152000  04187000  07604120

EMIT21
  0033 *04503000*
       04526200  04530000  04536000  04543000  04549000  04555000

ENDOFITALL
  0003 *05101100*
       *16495210*

ENDREADTAPE
  0010 *02201510*
       02202010 *02202080*

ENDTOG
  0003 *01415000*
      *02653000* 02653000  02680000  02884000 *06021000* 06022000 *06022000**07016000* 07019000 *07020000*

ENDV
  0003 *01228000*
       05109000  07013000  07018000  07748000  07749000  16477000

ENS
  0103 *16013000*
       16133000

ENTER
  0003 *03069000*
       13390000 *13714000* 14138000  14139000  14140000  14141000  14162000
```

ENTRY
  0003 *03055000*
       13314000 13729000 14165000 14194000 14207000 14259010 14319000 14333000 14403000 14409000

EOF
  0009 *02210250*
       02216250 *02217000*

EOF
  0067 *08999050*
       08999150 08999375 *08999400*

EOF
  0077 *09394000*
       09399000 *09413000*

EOFT
  0010 *02201510*
       02201750 *02202020*

EPART
  0021 *02639000*
       02810000

EQL
  0003 *01644000*

EQUAL
  0003 *02062500*
       02064000 02064500 02881000

EQVOP
  0003 *01274000*
       02965500 02971000 06036000

ERR
  0003 *01718000*
       02211500 04145000 04300000 *05105000* 05109000 05110000 05285000 06005000 06086000 06095000 06107000

06109000 06116000 06142000 06151000 06154000 06176000 06302000 06334000 06411000 06415000 06417000
06421000 06423000 07015000 07053000 07097000 07099000 07405000 07412000 07437000 07485000 07494000
07513000 07517000 07598000 07600700 07604130 07667500 07672500 07674000 07727080 07732000 07994000
08059000 08134000 08160000 08178000 08229000 12007000 12033000 12039000 12052000 12056000 14259090
14259120 15081000 15115000 15119000 15233008 15256000 15261000 15276170 15276190 15294000 15297000
15343000 15364000 15367000 16125000 16132000 16160000 16205000 16206000 16207000 16214000 16217000
16253100 16316000 16347000 16367000 16384500 16397000 16418000 16483000

ERRMAX
0003 *01001550*

02211500 *02502000**09028910*

ERRNUM
0003 *01716000*

ERRNUM
0003 *01718000*

ERRNUM
0003 *05105000*

05106000 05107100 05107200

ERRNUM
0003 *05012000*

05044000 05091000

ERRNUM
0034 *05016000*

05017000 05026000

ERROR
0007 *02091500*

02114000 02119000

ERROR
0003 *05110000*

07667500 07672500 07674000

ERROR

0022 *02958000*

02959000

ERRORCOUNT
0002 *00501000*

02211500 *05035000* 05035000 05087000 09028000 *09028900* 09386000 09391000

ERRORTOG
0003 *01412000*

05032000 *05047000**05119000**07009000**07063000**07645000**07750000**13282000**14130000**14443000* 14511000
16130000

EXAMIN
0002 *00511000*

00512000 00529000 00531000 00533000 00536000 02214200 02225750 02228250 02230000 02230100 02665000
02680000 02697000 02745000 02750000 02753000 02786000 02791000 02807000 02813000 02888000

EXIT
0003 *01758000*

01763000 01764500 01782000

EXIT
0009 *02210000*

02215750 *02225000*

EXIT
0007 *02091500*

02104500 02112000 02114500 02115500 02118500 02119500

EXIT
0016 *02360000*

02375000 02411000 02413000 02486000 02488000 *02602000*

EXIT
0022 *02957000*

02958000 02959000 02962500 *02967500*

EXIT
0035 *05049000*

```
        05094000 *05100000*

EXIT
  0038 *05273000*
        05284000 *05286000*

EXIT
  0032 *04311010*
        04311100 *04322100*

EXIT
  0033 *04501000*
        04526200  04532000  04538000  04546000  04552000 *04558000*

EXIT
  0044 *05439000*
        05454000  05457000 *05458000*

EXIT
  0041 *05334000*
        05344760  05372000 *05375000*

EXIT
  0049 *06195000*
        06200000  06204000 *06209000*

EXIT
  0046 *06061000*
        06069000  06086000  06096000  06107000  06109000  06116000 *06125000*

EXIT
  0047 *06137000*
        06142000  06149000  06154000  06176000 *06182000*

EXIT
  0040 *05325030*



EXIT
```

0054 *07037000*

07097000 07099000 *07101000*

EXIT
0059 *07504000*

07514000 07516600 07517000 07519000 *07531000*

EXIT
0051 *06315000*

06333000 *06335000*

EXIT
0052 *06339000*

06341000 *06346000*

EXIT
0055 *07394000*

07402000 07405000 *07425000*

EXIT
0060 *07549000*

07555000 *07557000*

EXIT
0062 *07594000*

07598000 07600700 *07646000*

EXIT
0061 *07562000*

07573000 07574000 07576000 07579100 07584000 *07587000*

EXIT
0064 *07726000*

07727080 07732000 07733000 07735000 07737000 07739000 07743000 07746000 07749000 07751000 07753000
07755000 07757000 07759000 07761000 07763000 07765000 07768160 07771000

EXIT
0065 *08017000*

08092000 08178000 08222000 08229000 *08232000*

EXIT
  0066 *08092000*
       08134000 08160000 *08172000*

EXIT
  0084 *12005000*
       12033000 12040000 12052000 12056000 *12058000*

EXIT
  0082 *10259000*
       10263900 10275000 *10284000*

EXIT
  0103 *16475000*
       16484000 16487000 16490000 16491000 16492000 16493000 *16494000*

EXIT
  0109 *16193000*
       16205000 16206000 16207000 16214000 16217000 *16239000*

EXIT
  0102 *15076000*
       15081000 15110400 15112000 15115000 15119000 15233005 15233008 15233024 15256000 15261000 15275000
       15276120 15276150 15276170 15276190 15276230 15294000 15297000 15305000 15337000 15341000 15343000
       15365000 15367000 *15377000*

EXIT
  0107 *16117000*
       16125000 16132000 *16144000*

EXIT
  0112 *16365000*
       16367000 16384500 *16398000*

EXIT
  0110 *16251000*
       16253100 *16270000*

EXIT
  0111 *16313000*

       16316000 16342000 16347000 *16355000*

EXPECT
  0003 *03009000*

EXPECT
  0003 *06060000*

       06119000

EXPRSS
  0003 *03007000*

       06057000 06299000

F
  0003 *01688000*

       05068000 *14077000**14078000**14115000* 14115000 16031000

F
  0003 *02262000*

       02262000 02263000

F
  0018 *02441000*

       02443000

F
  0018 *02446000*

       02448000 02635000 02955500 02980000 03009000 03011000 03019000 03020000 03024000 03025000 03028000
       03029000 03036000 03037000 03038000 03039500 03041000 03042000 03047000 03047100 03049000 03050000
       03051000 03051001 03053000 03055000 03057000 03067000 03068000 03070000 03071000 04003000 04006000
       04010000 04011000 04017000 04022000 04029000 04030000 04115000 04116000

F
  0035 *05068000*

       05069000 05076000

F

0067 *08999075*

08999100

F

0077 *09393700*

09393710 09393720 09393730 10228000 10234000

F

0092 *13673100*

13673100 13673350

F

0104 *16031000*

FACTOP

0003 *01299100*

15253000

FCR

0003 *01330000*

02182750 02194000 02195000 02195750 *02214000* 02214200 *02225250* 02225750 02228250 02228750 02230000
02230100 02421000 02731000 05038000 *07025010* 07025020 *07025030*

FCR

0003 *02016000*

02016000 02017000

FEIL

0003 *04130000*

04132000

FEJ

0018 *02441000*

02444000 02445000 02452000

FETCH

0003 *02262000*

02263000 08999175 09400000 09401000

```
FIEL
  0003 *02056000*
         02058000  02059000

FIEL
  0076 *09364000*
         09365000  09366000  09367000

FIEL
  0091 *13633000*
         13636000  13637000

FIEL
  0092 *13661000*
         13665000  13667000

FIL
  0076 *09370000*
         09372000

FILEDEC
  0094 *14015000*
         14020000 *14158000*

FILEID
  0003 *01299300*
         14430000

FILEV
  0003 *01296000*



FILLIT
  0040 *05325040*
         05325440  05325460  05325470  05325480

FILLSTMT
```

0003 *07647000*

07677500 13396000

FINAL
0082 *10258300*

10263900 10264400 10264420

FINALAX
0076 *09378000*

09384100

FINDOPTION
0003 *02307000*

02317000 02318000 02327000 02967000

FINI
0103 *16475000*

16476000 16478000 16483000 16486000 16489000 *16495000*

FINIS
0007 *02091500*

02095500 02096500 02099500 02110000 02125500

FINISHED
0029 *04204000*

04220000 *04225000*

FINISHNUMBER
0021 *02638000*

02847000

FIRST
0003 *05271000*

05280000 05282000

FIRST
0051 *06314000*

06324000 06327000 06332000

```
FIRST
  0052 *06339000*
       06341000  06344000

FIRSTTIME
  0009 *02210000*
       02210750 *02213500*

FIRSTX
  0003 *01621000*
       13608000 *13612000* 13623000  14051000 *14052000* 14600000 *14608000*

FIRSTXO
  0094 *14033000*
      *14051000* 14608000

FIX
  0018 *02446000*
       02451000  02454000

FIX
  0056 *07427000*
      *07430000**07433000* 07439000  07440000

FIX
  0066 *08082000*
       08090000  08157000  08171000

FIXC
  0103 *16029000*
       16057000  16141000  16233000  16236000  16238000

FIXDEFINEINFO
  0003 *01717900*
       02896000 *12101000**12108000*

FIXHDR
```

0077 *09393700*

09393750 09430075

FIX1

0109 *16195000*

16218000 16233000 16238000

FIX2

0109 *16195000*

16232000 16236000

FL

0003 *01420000*

06140000 07050000 07058000 07073000 07083000 12037000 15110100 15276210

FLAG

0003 *01716000*

02226500 02316000 02421000 02501000 02593000 02599000 02650000 02686500 02696000 02824000 02828000
02899000 02958000 02959000 02962000 02976500 02977500 04013000 04176000 *05012000* 05101000 05106000
05119000 05344610 05344661 05347000 05348000 05370000 06104520 06110200 06200000 07024000 07025000
07061000 07600000 07600420 07600900 07604000 07604068 07671500 07728000 07747000 09282600 10243000
10264420 12025000 12030000 12115000 12126000 13210000 13282000 13332000 13338000 13340000 13386000
13391000 13395000 13400000 13404000 13651100 13729000 13753000 13754000 13758000 13761000 13763000
13766000 14066000 14135000 14136000 14137500 14146000 14153000 14161080 14165000 14168000 14187000
14220000 14263000 14286000 14295000 14311000 14326000 14333300 14337000 14339000 14346000 14351000
14358000 14371000 14442000 14486000 14508000 15376600 16033000 16071000 16163000 16171000 16265000
16434100

FLIPFLOP

0093 *13750500*

*13752000**13755500* 13755500

FOOT

0066 *08091000*

08162000 08166000

FORLIST

0065 *08080000*

08156000 08172000 08231000

FORMAL

0003 *01149000*

05209000 05218000 05236000 07070000 09133800 13202000 13244000 14427000

FORMALF
0003 *01591000*

07400000 07411000 *13202000**13328000**13341100**13343000* 13372000 *16000400*

FORMALNAME
0102 *15076100*

15092000 15105000

FORMALV
0065 *08014000*

08056000 08065000 08098000 08108000 08130000 08179000 08212000 08217000 08227000

FORMATBIT
0003 *01000950*

01001240 02567000 14461500 14493500

FORMATOG
0003 *01001240*

14461500 14493500

FORSTMT
0003 *03052000*

07755000 *08008000* 08232000

FORV
0003 *01223000*

FORWARDBRANCH
0066 *08091000*

08152000 08157000 08169000 08171000

FORWARDV
0003 *01254000*

14467000

FORWART
0066 *08084000*

08086000

FOULED
0003 *01583100*

04119500 04119500 *04151000**04207500**07483500**07491500**07604010**14505000*

FOUND
0015 *02309000*

02313000 *02315000*

FOUND
0029 *04204000*

04209000 *04221000*

FP
0003 *01420000*

06164000 15119000 15349000

FPART
0021 *02639000*

02682000 *02790000*

FR
0003 *01420000*

08228000 15305000

FR
0003 *03042000*

FR
0003 *13657000*

13677100 13677250 *13677250* 13677250 13677300

FRMTID

```
  0003 *01182000*


FROM
  0003 *04116000*
         04120000  04120100  04120500

FROM
  0003 *03037000*


FROM
  0003 *03038000*


FROM
  0003 *03029000*


FROM
  0003 *03011000*


FROM
  0003 *07391000*
         07402000  07404000

FROM
  0003 *06194000*
         06198000

FRONT
  0058 *07491000*
        *07493000* 07496000

FRSTLEVEL
  0003 *01402000*
         05116000 *14478000*
```

FS
  0003 *01420000*

        07739000 15102000 15104000 15114000 15233008 15233015 15276100 15276110 15276160 15276190 15334000
        15335000 15342000 15364000 15367000

FUNCMONFILE
  0003 *01543000*

FUNCTOG
  0003 *01586000*

        13736000 14054000 14187000 *14275000**14288000**14296000* 14411100 14415100 14452000 14478000 *14599000*

FUNCTOGO
  0094 *14034000*

       14054000 14599000

FWDTOG
  0099 *14274000*

       *14276000**14312000* 14333100 14373000

FZERO
  0003 *01692000*

       0534459₂ 05344662 05375100 14078000

G
  0003 *01610000*

       13231000 13236000 *13718000* 13723000 *13724000* 13726000 *13727000* 13729000 *14084000* 14085000 14091000
        14115000 14116000 14117000 *14278000**14281000* 14284000 14292000 14294000 *14296000* 14298000 14299000
       14310000 14315000 *14349000* 14349000 *14352000* 14353000 *14446000* 14449000 *14470000* 14470000

GENERATE
  0111 *16313000*

        16322000 *16345000*

GEQ
  0003 *01647000*

        08050000

GET
  0003 *03019000*

       04102000 04105000 04182000 *05307000* 05312000 *05312000* 07604050 07607000 07608000 08086000 13235000
       13258000 13267000 13273000 14168000 16032000 16092000

GETF
  0002 *00523000*

       00524000 00526000 00527000 00529000

GETLP
  0095 *14076000*

       14084000 14089000

GETSPACE
  0003 *03051000*

       05132000 *05331000**05375000* 05378000 08162000 13347000 14044000

GETSYL
  0039 *05309000*

       05310000 05312000

GETVOID
  0003 *01756000*

       01784000 02410000 02485000

GIT
  0003 *01719000*

       02901000 *05123000**05124000* 06104300 07410000 07412000 07438000 07441000 07527000 07539000 07601000
       13249000 13257000 14388100 14438000 14483400 15376300 16070000 16089000 16138000 16163000 16254000
       16257000

GNAT
  0003 *03020000*

       *05128000**05131000**05132000* 05133000 05316000 07516300 13267000

GNC
  0007 *02091500*

       02097000 02104000

GOGEN
  0003 *03023000*

        07518000 07529000 *07535000* 07545000 07572000 07573000 07575000 07583000 13274000

GOMCP
  0059 *07504000*

GOSTMT
  0003 *07501000*

        07531000 07763000

GOT
  0077 *09393060*

        09393130 09393170 09393190

GOTOS
  0103 *16249000*

        16270000 16454000 16487000

GOTSCHK
  0094 *14016000*

        14160000

GOTSTORAGE
  0003 *13733000*

GOTSTORAGE
  0094 *14025000*

        14160000 *14160000*

GOV
  0003 *01232000*

        07550000 07556000

GRINCH

0077 *09393060*

09393130 09393170 09393190

GRINCH

0077 *09393250*

09393350 09393400 09393430

GS

0003 *05325010*

05325460 05325470 05325480

GS

0041 *05344000*

*0534459$_2$**05344600**0534461$_5$**05344620* 05344620 *05344660* 05344661 05344662 *05344662* 05344662 *05344700*
05344720 *05344740* 05344740 05344750 *05350000* 05350160 05350200 *05371000* 05373000 *05373000* 05375000
05375100 *05376000* 05376000 05376100

GTA1

0003 *01697000*

13210000 13351000 13381000 13408500 13718000 13724000 13727000 14069000 14129000 14134000 14161000
14161020 14164000 14259000 14259010 14259015 14259080 14278000 14281000 14296000

GTI1

0003 *01384500*

*02192500**02193000* 02193250

GTR

0003 *01650000*

GT1

0003 *01384000*

02203500 02205000 *02452000* 02456000 *02652000* 02653000 02655000 *02658000* 02659000 02716000 *02719000*
02720000 02723000 *02823000* 02830000 *02832000* 02832000 02833000 02834000 02838000 02839000 *02875000*
02876000 02877010 02878000 02879000 02881000 02882000 *02886000* 02894000 *04105000* 04107000 04109000
04110000 *04120100**04120500* 04121000 04122000 *04508000* 04521000 04523000 04524000 *05116000* 05117000
05231000 05233000 05235000 *05235000**05447000* 05449000 *07042000* 07043000 07044000 *07066000* 07067000
07436000 *07539000* 07540000 07544000 07545000 *07600950**07727020* 07727070 *07994000* 07995000 *08058000*
08060000 *09034100* 09034200 09133800 *09133900* 09134000 09134100 09134200 *09253000* 09253500 09254000
09255000 *09414040* 09414080 09414090 *09415000* 09415500 09416000 *09416000* 09416000 09417000 *09417000*
09417000 09418000 *10250000* 10251000 *13226000* 13232000 13240000 13244000 13248000 *13251000**13256000*
13259000 *13273000* 13274000 *13280000* 13282000 *13371000* 13373000 *14039000* 14041000 14116000 14203000

*14333300**14425000* 14427000 14429000 14430000 14437000 14447000 *14483200* 14483300 *15102000* 15106200
15107000 *15295100* 15296000 16157100 *16254000* 16255000 16257000 16258000 16260000 16265000 16452000

GT1
0097 *14203000*

14206000 14221000 *14221000* 14224000

GT1
0108 *16157100*

16159000 16163000 16164000 16166000 16171000 16173000

GT2
0003 *01384000*

02721000 02723000 02834000 02839000 *02876000* 02877010 02878000 02879000 02881000 02882000 *05233000*
05234000 *05448000* 05450000 *07539000* 07545000 07595000 *07601000* 07604040 07604050 07604060 07604070
07604090 *07604090* 07605000 07607000 07607100 07608000 *07609000* 09133800 09133900 09134000 *09134100*
13284000 13285000 13286000 14203000 *14430000* 14431000 *14446000* 14448000 *16163000* 16167000 16172000
16254000 16255000 16266000

GT2
0097 *14203000*

14217000 14224000

GT3
0003 *01384000*

01721000 05402100 *05402100**05405000* 05405000 *05407000* 05407000 *05409000* 05409000 05428000 *05428000*
05430000 05430000 07397400 *07397400**07397500* 07397500 *07527000* 07528000 07595000 *07601000* 07610000
09134100 09134200 *13285000* 13287000 *14438000* 14438000 *14483400* 14483400

GT4
0003 *01384000*

02904000 *06104300* 06104500 *06104500**06104510* 06104510 *06104520* 06104600 *07527000* 07527000 07528000
07529000 07596000 *07601000* 07603000 07610000 *13235000* 13238000 14203000

GT4
0097 *14203000*

GT5
0003 *01384000*

07028000 07029000 07596000 *07604050* 07604080 07604090 *07607000* 07609000 14203000

GT5
  0097 *14203000*

H
  0005 *01822000*
       01828000 01833000 01835000

H
  0003 *05315000*
        05316000

H
  0035 *05068000*
        05071100 05071200 05085000

HEXIZE
  0003 *02001858*
        02001897 02001898

HEXOP
  0003 *01268000*
        01299200 06071000 06089000

HF
  0094 *14017000*
        14123000 *14382000* 14508000

HOLE
  0055 *07393000*
       07396000 07398000 07410000 07412000

HTTEOAP
  0003 *13731000*
        13747000 14489000 14600000

I

```
0003 *01319000*

     *02917000* 02917000  04237000 *05002000* 05002000  05003000 *05003000**05107000* 05107000 *05107100* 05107100
     *05107200* 05107200  05189000  05216000  05217000  05218000  05221000  05222000  05224000 *05225000* 05225000
      05280000  05281000  05344570 *05344600* 05344600  05344615  05344620  05344660 *05344670* 05344670  05344700
      05344710  05344770 *05344770* 05350120  05350160  05350180  05414000  05438000  06016000  06044000  06072000
      06104300  06108000  06166000  06185000  06324000  06325000  06326000 *07008000* 07008000  07031000 *07031000*
      07066000  07070000  07070500  07071000  07396000  07397210  07401000  07430000  07516000  07516300  07518000
      07521000  07554000  07556000 *07556000* 07566000  07572000  07583000  07599000 *07600200* 07600200 *07600500*
      07600500  07600950  07604020 *07604130* 07604130 *07604150* 07604150  07667000  07727040 *07727040**07727060*
      07727060  07745000  07748000  07749000 *07749000**07750000* 07750000  07751000 *07751000* 07768100 *07768130*
      07768130  07768170 *07768170**07768180* 07768180  07994000  07995000  08029000  08034000  08037000 *08110000*
      08110000  08111000 *08120000* 08120000 *08174000* 08176000 *08223000* 08999050  08999500 *09391000* 09393270
     *09414030* 09414040  09414100 *09414100**09415000* 09415000  09418000  09419000 *09419000* 09419000 *09420020*
     *09424000* 09424000  09426000 *09427000* 09427100  09427500 *09427500**09432000* 09433000  09434000 *09436000*
      09436000  09447020  09447030  09447040  10258200  12015000  12020000  12024000  12036000  12044000  13200000
      13324000 *13324000* 13361000  13372000  13394000  13753100  13757000 *13757000* 13761100  13767000  13768000
      14062000  14063000 *14067000* 14067000  14129000  14137200 *14137200* 14189000 *14189000* 14220000  14221000
     *14255000* 14255000  14264000 *14264000* 14314000  14332000 *14332000* 14349000  14352000  14353000 *14360000*
      14360000  14399000  14408000 *14408000* 14516000  14519000  15077000  15276010 *15276040* 15276040  16000700
      16121000  16124000  16159000  16200000  16204000  16208000  16211000  16212500 *16252000* 16252000  16254000
      16321000  16329000  16342000  16344000  16370000  16376000  16379000  16408000  16412000  16435000  16449000
     *16449000**16450000* 16450000  16482300 *16489000* 16489000  16495100

I
  0003 *05400000*

       05402000  05402200  05402300  05404000  05409200  05409500

I
  0030 *04237000*
         04239000  04241000 *04242000* 04243000

I
  0043 *05415000*
         05415000  05419000  05437000  06060000

I
  0044 *05438000*
        *05444000* 05445000  05447000  05455000 *05456000* 05456000  05458000  05459000

I
  0043 *05414000*
         05424000  05425000  05425100 *05425400* 05425400  05426000 *05430000* 05430500  05431000

I
  0041 *05334100*
```

```
        05334400  05334600

I
  0067 *08999050*
       *08999175* 08999225  08999350

I
  0068 *08999500*
       *08999525* 08999550  08999650

I
  0079 *09393270*
       *09393300* 09393310

I
  0082 *10258200*
        10277000  10282000

I
  0092 *13673100*
        13673300  13673400

ID
  0015 *02310000*
       *02312000* 02313000

ID
  0035 *05067000*
        05067000  05070000  05246000  05247000  05325010

ID
  0040 *05325040*
        05325320

IDBIT
  0054 *07039000*
       *07047000* 07050000  07068000  07082000

IDBLDR
```

IFSW
  0109 *16194000*
       16197000

IFTOG
  0109 *16193000*
       16194000 16215000 *16217000*

IFV
  0003 *01231000*
       06004000

IMPFUN
  0003 *03039000*
       06158000 *06183000*

INCR
  0003 *01153000*
       02658000 05124000 05189000 05222000 06080000 06189000 13206000 13240000 13307000 *14116000* 15255000
       15295100

INCRF
  0003 *01607000*
       13206000 13769000 14314000

INDEC
  0094 *14014000*
       14019000 14158000

INDEX
  0003 *05005000*
       05006000

INDEX
  0003 *03053000*

INDEX

0003 *05008000*

05009000

INDEX

0003 *07036000*

07042000 07098000

INDEX

0062 *07595000*

07601000 07610000

INDEX

0111 *16314000*

*16320000**16328000**16334000* 16346000 16349000

INDEXS

0103 *16311000*

16355000 16490000

INFO

0003 *01007000*

02212250 02214250 02226000 02228000 02234750 02265600 02265700 02265850 02265950 02368000 02410000
02485000 02602800 02877010 02878000 02879000 02881000 02923000 04172000 04187000 04209000 04210000
04211000 04221000 04223000 05006000 05009000 05039500 05045000 05088000 05233000 05325470 05350160
05350200 05426000 05431000 05449000 05450000 07086400 08035000 09034200 09034500 09085000 09121650
09253500 09254000 09255000 09384100 09386000 09392500 10251000 10254000 10264800 12117000 13281000
13310000 13593000 13677300 14440000

INFO

0003 *02005000*

02006000

INFO

0016 *02320200*

02320600

INFO

0077 *09392500*

09406000 09427500 09447060

INFO
0083 *10236000*

10238000

INLINE
0003 *03010000*

07746000 *13748000* 13774000

INLINETOG
0003 *01695500*

13307500 *13752000**13770500*

INT
0065 *08012000*

08060000 08070000

INTARRAYID
0003 *01205000*

04522000 08057000 15233009 15333000

INTBIT
0003 *01000960*

01001250 *02577000* 02577000 *09253050* 09253050 09253100 09394100 09414010 09430050 13651100

INTEGERDEC
0094 *14013000*

14018000 *14141000*

INTID
0003 *01201000*

04509000 08038000 08061000 12035000 12042000 14141000 15086000 15102000

INTNAMEID
0003 *01205400*

INTOG
0003 *01001250*

02577000 09253050 09253100 09394100 09414010 09430050 13651100

INTPROCID
  0003 *01197000*

        13278000 14294000 15078000

INTRINSICADR
  0003 *01299500*

        09253100

INTRNSICPROCID
  0003 *01188000*

INTSTRPROCID
  0003 *01193000*

        14284000

INTV
  0003 *01285000*

INV
  0003 *01290000*

INX
  0003 *01652000*

        04531000 07091000 15303000 15344100 15351200

IOCLASS
  0003 *01233000*

IOSTMT
  0003 *03012000*

        07765000 *07993000*

IPART

0021 *02639000*

02643000 *02758000*

IS

0033 *04501000*

04527000 04540000 *04548000*

ISD

0003 *01653000*

15103000 15276100 15334000

ISKIP

0083 *10236000*

10237000 10238000

ISN

0003 *01654000*

15103000 15276100 15334000

ISOLATE

0003 *01270000*

06092000

ITUP

0102 *15076000*

15233018 15233026

J

0003 *01618000*

04166000 05401000 05414000 05438000 07393100 08999050 08999500 09393010 12102000 *13210000* 13210000
13221000 *13324000* 13333500 *13333500**13342000* 13342000 13351000 13381000 *13381000**13408500**13718000*
13718000 13724000 *13724000* 13727000 *13727000* 13750000 *14069000* 14129000 *14129000* 14132000 14134000
14135000 *14135000* 14136000 *14136000* 14137600 *14137600**14161000* 14161000 *14161000* 14161000 14161020
14161020 14164000 *14164000* 14254100 *14278000* 14278000 *14281000* 14281000 *14296000* 14296000 *14421000*
14425000 14438000 14440000 14446000 14448000 *14449000* 14449000 15075000 16314000 16364000

J

0003 *10228000*

10263110 10264600 10264900

J

0028 *04166000*

04170000 04172000 04187000

J

0042 *05401000*

05405000 *05405000* 05406000 05407000

J

0044 *05438000*

05444000 05446000 05448000 05452000 *05453000* 05453000

J

0043 *05414000*

05428000 *05428000* 05428000 05429000 05430000

J

0055 *07393100*

07397300 07397400

J

0067 *08999050*

08999125 08999225 08999275 *08999275* 08999350

J

0068 *08999500*

08999625 08999650 08999700 *08999750* 08999750

J

0079 *09393270*

09393290 09393350 *09393360* 09393360 09393400 09393420 *09393420* 09393430 *09393450* 09393450

J

0077 *09393010*

09393270 09414022 09414024 09414026 *09414026**09414030* 09414060 *09414060* 09414070 09414110 *09420020*
09425000 *09425000**09428000* 09428000 09447050 09447060 09447080

J

0082 *10257100*

10257100 10257500 10257700

J
0087 *13221000*
13232000 13239000 *13257000**13258000* 13258000 13264000 13265000 13270000 13271000

J
0085 *12102000*
12116000 *12116000* 12117000 12126000

J
0093 *13750000*
13765000 13765000

J
0098 *14254100*
14259010 14259015 *14259015**14259080* 14259080 14259150 14266000

J
0098 *14254200*
14254200 14254500

J
0102 *15075000*
15234000 15255000 15266000 *15280000* 15280000 15290000 15296000 15302000 15344000 15350000

J
0111 *16314000*
16348000 16349000 *16349000*

J
0112 *16364000*
16387000 16389000 *16392000* 16392000 *16392000* 16393000

JAZZ
0102 *15076000*
15233012 15233026

JEDEN

0012 *02264100*

02265000 02265100 02265350 02265450 02265600 02265700 02265850 02265950

JFC
0103 *16016000*

16219000

JFW
0103 *16007000*

16079000 16232000

JNS
0103 *16023000*

16453000

JOINFO
0003 *01482000*

16127000 *16128000* 16134000 16138000 16139000 *16143000* 16440000 *16442000* 16445000 16447000 16449000

JOINT
0107 *16118000*

16127000 16143000

JRV
0103 *16009000*

16079000

JUMPCHAIN
0103 *16085000*

16097000 16139000 16173000

JUMPCHKNX
0003 *03058000*

13616000 14165000 14387000

JUMPCHKX
0003 *03059000*

13596000 14506000

JUMPCTR
  0003 *01459000*
       14497000 *14499000*

JUMPLEVEL
  0003 *01485000*
       16260000 16265000 *16269000**16433000**16435000* 16452000

JUMPS
  0103 *16431000*
       16455000 16493000

JUMPV
  0003 *01242000*

JUNK
  0003 *01430000*

K
  0003 *01693000*
       02091500 05401000 05414000 05438000 08016000 08999050 09393010 10258100 12102000 *14085000* 14254100
       16364000

K
  0007 *02091500*
       02099000

K
  0042 *05401000*
       05405000 05406000

K
  0043 *05414000*
       05428000 05429000

K

0044 *05438000*

05444000 05452000 05454000 *05454000* 05455000 *05457000* 05457000

K

0067 *08999050*

08999200 08999275 *08999275* 08999275 08999350

K

0065 *08016000*

08033000 08034000 08035000 08094000 08180000 08223000 08232000

K

0077 *09393010*

09447030 09447060

K

0085 *12103000*

12103000 12104000

K

0082 *10258100*

10263200 10263800 10264200 *10264200**10264400* 10264400

K

0085 *12102000*

12109000 12117000 12118000 12121000

K

0098 *14254200*

14254200 14254400

K

0098 *14254100*

14259010 14259080 14259090

K

0112 *16364000*

16389000

KEY
0003 *04231000*

04239000 04241000

KLASSF
0003 *01603000*

05350100 05350200 *13201000* 13220000 *13226000* 13246000 13278000 *13333500**13342000* 13372000 *13677200*
13677300 *16000300*

KNOT
0024 *02973500*

02974500 02978500

L
0003 *01428000*

01477000 01710000 01758000 02091500 02127500 02182750 02195000 02195750 02214200 02228750 02421000
04086000 04102000 04105000 *04105000* 04105000 *04109000* 04109000 04119000 *04120000**04124000* 04145000
04145000 04147000 04149000 *04151000* 04151000 *04172000* 04172000 04175000 04182000 *04185000**04186000*
04207500 04210000 *04216000* 04216000 04218000 04223000 04232000 04279000 04282500 04293000 04296000
04298000 *04298000**04300000* 05038000 05049000 06054100 *06054100* 06104400 06104500 06104510 06148000
06148000 06206000 06207000 *06207000**06297000* 06297000 *06300000* 06300000 06301000 06307000 *06308000*
06308000 07025020 07079000 07081000 07088000 07411000 *07411000**07433000* 07433000 *07440000* 07440000
07443000 07444000 *07444000* 07459000 *07459000* 07483500 07485000 07487000 *07487000* 07491500 07493000
07493000 07495000 *07495000* 07496000 07540000 *07540000* 07545000 07577000 *07577000* 07579000 07583000
07583000 07585000 *07585000* 07585100 07586000 07597000 *07600300**07600440* 07600800 07604010 07604040
07604060 07604067 07604085 *07604085* 07604090 *07604100* 07607100 07608000 07610000 07727020 *07727050*
07727070 08046000 08047000 *08047000* 08050000 *08050000* 08077000 *08078000* 08078000 *08089000* 08093000
08097000 08106000 *08106000* 08122000 08124000 08127000 08146000 *08146000* 08147000 *08148000* 08148000
08150000 08150000 08152000 08160000 *08168000* 08168000 08169000 08170000 08171000 08179000 *08190000*
08190000 08191000 08215000 08221000 *08221000* 08227000 08228000 *08231000* 09282600 09388000 *09395000*
09395000 09396000 09397000 09416000 *13232000* 13232000 13235000 *13238000**13239000* 13258000 *13263000*
13264000 13267000 *13270000* 13273000 *13277000* 13360000 13606000 13612000 13628000 *13628000* 13746000
13752000 *13765000* 13765000 *13765000**14039000* 14039000 14041000 14054000 *14055000* 14167000 14168000
14388100 14451200 14461100 14461200 14461300 *14461300**14483700* 14487000 14505000 14602000 14603000
15376400 15376500 15376600 *16032000* 16032000 *16034000* 16070000 16088000 *16089000* 16090000 16092000
*16094000**16096000* 16121000 16138000 16167000 16172000 16218000 16232000 16257000 16313000 16364000
16475000

L
0003 *01758000*

01762000 01764000

L
0003 *02183500*

02183750 02186000 03047000 03047100

L
0003 *01742100*
01742110

L
0007 *02127500*
02128000 02135500

L
0003 *01719000*

L
0007 *02091500*
02101000

L
0003 *05333000*
05376100

L
0003 *03051001*

L
0003 *03020000*

L
0003 *03039500*

L
0003 *05307000*
05312000

L

0003 *05128000*

05131000 05132000

L

0003 *05411100*

L

0003 *05123000*

05124000

L

0003 *05437000*

05440000 05441000 05442000 05444000 05458000

L

0003 *08999450*

08999525

L

0030 *04232000*

04241000 *04243000*

L

0035 *05050000*

05051000 05052000

L

0035 *05059000*

05060000

L

0035 *05049000*

05092000 05094000 05098000

L

0046 *06064000*

06065000

L
0043 *05415000*
05418000

L
0089 *13360000*
13364000 13366000

L
0103 *16475000*
16476000 *16483000*

L
0112 *16364000*
16391000 16393000

L
0111 *16313000*
16319000 16327000 16340000 *16347000*

LABELBAT
0003 *03025000*

LABELBAT
0003 *07537000*
07539000

LABELDEC
0094 *14014000*
14019000 *14187000*

LABELID
0003 *01206000*
01479000 05344520 05350120 05350160 06104100 06106300 06107100 07047000 07075000 07506000 07553000
07645000 07664000 13246000 13337000 13754000 14194000 14220000 15128100 15376900 16213000 16253000
16482100

```
LABELR
  0003 *07593000*
         07646000  07741000

LABELS
  0103 *16156000*
         16177000  16485000

LABELV
  0003 *01286000*
         14399000

LABLMONFILE
  0003 *01535000*


LAMPER
  0047 *06137000*
         06164000  06166000  06168000 *06179000*

LASS
  0102 *15076000*
        *15276060* 15276160

LAST
  0102 *15076000*
        *15302000* 15342000

LASTCRDPATCH
  0003 *01001500*


LASTENTRY
  0003 *01376000*
         04170000 *04189000* 04208000  04210000  04211000 *04214000* 04214000  14038000

LASTGT
  0103 *16003000*
         16089000  16266000
```

LASTGT
0101 *14420000*

14438000

LASTINFO
0003 *01619000*

02724000 02725000 02726000 *02726000* 05350200 *09033000* 10261000 10263100 10263110 10264800 13222000
13290000 13309000 *13311000* 13347000 13348000 13408000 13755500 14079000 14083000 14259150 14320000
14330000 14330000 *14373000* 14421000 *14463000* 16000700 16445000 *16447000*

LASTINFOT
0094 *14029000*

14330000 14373000 14463000

LASTRESULT
0082 *10258000*

10262000 10272000 *10276000*

LASTSEQROW
0003 *01570000*

02214250 02226000 02228000 02234750 02368000 02410000 02485000 02602800 05039500 05045000 07086400
09033000 09034200 09034500 09253500 09254000 09255000 09384100 09386000

LASTSEQUENCE
0003 *01569000*

02214250 02226000 02228000 02234750 02368000 02410000 02485000 02602800 05039500 05045000 07086400
09033000 09034200 09034500 09253500 09254000 09255000 09384100 09386000

LASTUSED
0003 *01391000*

02194500 *02204000**02209250* 02212000 02212250 *02212750* 02212750 02228050 02234000 *02332000**02430000*
*02438000**02458000**02722000* 02900000 *02901000**07030000**09029000**09037000* 09282000

LBC
0003 *01645000*

LBP
0094 *14023000*

LBU
0003 *01646000*

LCARD
0008 *02191000*
02191250 *02194750*

LCR
0003 *01330000*
02129500 02193250 *02202000**02204000**02209000* 02213000 *02214000* 02214250 02214500 *02215750* 02225250
02226000 02228150 02232000 02233500 02234750 02235750 02368000 02369000 02410000 02485000 02602800
02716000 02717000 *02719000* 02902000 *02903000* 02904000 02905000 *07025010**07025030*

LCR
0003 *02005000*
02005000 02006000

LCR
0003 *01756000*
01756000 01767000 01774000

LCR
0016 *02320200*
02320200 02320600 02321000

LDES
0003 *01299030*
05248000 05250000 09415500 13252500

LDOT
0047 *06137000*
06144000 06156000 06158000 06160000 06162000 06170000 06172500 *06178000*

LEFTPAREN
0003 *01212000*
02975000 06069000 06415000 07409000 07437000 07768110 10264100 12007000 12114000 13753000 14137000

```
        14259020  14326000  16003000  16125000  16482400

LENGTH1
  0016 *02360000*
        02364000 *02374000*

LENGTH2
  0016 *02360000*
       *02380000*

LENGTH3
  0016 *02360000*
        02364000  02381000

LENGTH4
  0016 *02360000*
        02364000 *02399000*

LENGTH5
  0016 *02361000*
        02364000 *02476000*

LENGTH6
  0016 *02361000*
        02365000 *02518000*

LENGTH7
  0016 *02361000*
        02365000 *02569000*

LENGTH8
  0016 *02361000*
        02574000

LENGTH9
  0016 *02361000*
        02365000  02575000

LEQ
```

```
  0003 *01655000*

       08050000


LEVEL
  0003 *01402000*

       05208000  05344400  05350100  05350120  13339000  13371000  13677100  14053000 *14053000* 14220000  14307000
       14318500 *14323000* 14323000  14384100  14387000  14389000  14411300  14415300  14451000  14461100  14477000
       14478000  14497000  14499000 *14500000* 14500000  14603000 *14604000* 14604000  16005000


LEVEL
  0103 *16005000*

       16171000  16265000


LEVELF
  0003 *01605000*

       *13204000* 13339000  13371000  14307000


LFC
  0003 *01648000*


LFTBRKET
  0003 *01263000*

       05274000  06139000  06316000  06421000  10264100  12114000  13391000  14259020  15233000


LFTPAREN
  0103 *16003000*

       16125000  16482400


LFU
  0003 *01649000*

       14167000


LGTFLD
  0103 *16004000*

       16172000


LIB
  0009 *02202500*
```

LIN
0003 *01559010*

02181000 02181250 02182750 02194000 02195000 02195750 02214200 02226750 02228750 02421000 04149000
04150000 05038000 05039500 05039600 05044000 05046000 05096000 05097500 05099000 05325450 05325460
05325470 05325480 05325490 06122000 06123000 07025020 07086300 07086500 09984100 09384500 09386000
09387000 09388000 09389500 13653000 13653500 13675000 13676000

LIN
0040 *05325040*

05325090 05331000 05333000

LINE
0003 *01558000*

01829000 02181000 02181250 02195000 02195750 02214200 02228750 02264950 02265150 02265300 02265450
02265650 02265750 02265900 02266000 02421000 02464000 04150000 04279000 04282500 05039600 05046000
05087000 05088000 05089000 05090000 05097500 05099000 05325490 05427000 05432000 05434000 06123000
07025020 07086500 09384500 09387000 09389500 09390000 13653500 13676000 14023100 14461500 14493500

LINE
0003 *02183500*

02184250

LINE
0034 *05016000*

05019000 05021000 05024000

LINE
0035 *05067000*

05069000 05071200 05076000 05085000 05246000 05247000

LINE
0054 *07038100*

07038200 07038300 07391000 07536000 07537000 07647000

LINE
0076 *09378000*

09379000 09380000 09393060 09393250

```
LINK
  0003 *01154000*

       02725000  04166000  05124000  05132000  05189000  05233000  05234000  05447000  05448000  06079000  06189000
       06190000  07595000  09133800  09134000  13207000  13232000  13286000  13308000  13361000  13362000  13364000
       14448000  14470000  15080000  16087000  16443000

LINK
  0028 *04166000*

       *04182000* 04185000

LINK
  0062 *07595000*

       07601000  07604040  07604050  07604060  07604070  07604090  07605000  07607000  07607100  07608000  07609000


LINK
  0106 *16087000*

       *16092000* 16094000

LINKC
  0003 *01156000*

       01721000  02212250  02876000  05006000  05009000  05325470  05350180  05350200  05402100  05405000  05407000
       05409000  05426000  05428000  05430000  05431000  05449000  05450000  07397400  07397500  09393110  09393330
       09414070  09418000  09419000  09433000  09447060  10251000  10264800  12117000  13281000  13310000  13593000
       13677300  14483400

LINKF
  0003 *01606000*

       *13207000* 13768000  13769000  14310000  14314000  14315000

LINKP
  0089 *13359000*

       *13361000* 13362000  13364000

LINKR
  0003 *01155000*

       01721000  02212250  02875000  05006000  05009000  05325470  05350160  05350200  05402100  05405000  05407000
       05409000  05426000  05428000  05430000  05431000  05449000  05450000  07397400  07397500  09393090  09393280
       09414070  09418000  09419000  09433000  09447060  10251000  10254000  10264800  12117000  13281000  13302000
       13303000  13310000  13593000  13677300  14483400

LINKR
```

0079 *09393280*

09393350 09393400 09393430

LINKR
0078 *09393090*

09393130 09393170 09393190

LINKT
0089 *13359000*

13361000 13362000 13364000 13365000 *13366000*

LINKTOG
0003 *01399000*

04100000 *04213000**04222000**04295000* 06021000 *06022000*

LISTABIT
0003 *01000970*

01001260 02194250 02511000 02602500

LISTATOG
0003 *01001260*

02194250 02602500

LISTBIT
0003 *01000980*

01001270 02402000 02602500 *09028000*

LISTER
0003 *01001570*

02195000 02195750 02214200 02464000 *02602500* 05036000 07025020 07086100 *09028000* 13652000 13674000
14461500 14493500

LISTID
0003 *01181000*

LISTOG
0003 *01001270*

02602500 09028000

LISTPBIT
  0003 *01000990*

01001280 02195000 02228750 02480000

LISTPTOG
  0003 *01001280*

02195000 02228750

LITERAL
  0003 *04003000*

04004000

LITNO
  0003 *01210000*

02850000 05275000 05277000 0534461O 06085000 06094000 06317000 06319000 06321000 07600300 07665000
07666500 07727050 07994000 13392000 15276010 16119000 16204000 16212500 16326000 16341000 16374000
16410000 16435000 16482300

LITV
  0003 *01250000*

16372000 16381000

LN
  0093 *13750000*

13755500 13773000

LND
  0003 *01656000*

04545000

LNG
  0003 *01657000*

04112000

LO
  0094 *14029000*

14304000 14496000

```
LOC
  0101 *14420000*
         14438000

LOC
  0103 *16003000*
         16070000  16163000  16255000

LOCAL
  0003 *05206000*
         05210000 *05210000* 07516000  07554000  07599000

LOCALV
  0003 *01249000*
         14396000

LOCBEGIN
  0094 *14022000*

LOCFLD
  0103 *16004000*
         16138000

LOCLID
  0003 *01179000*
         13755000  14409000  14411200  14415200  14425000  16198000  16317000  16326000  16330000  16335000  16340000
         16368000  16406000  16482100  16482300  16482500

LOCV
  0003 *01247000*
         16336000  16339000

LOD
  0003 *01658000*
         04523000  05201000  05227000  05316000  06172500  07071000  07090000  08079000  15233010  15267000  15306000
         15344100  15351300

LOLD
```

```
  0094 *14030000*
       *14054000*

LONGID
  0102 *15076100*
        15303000  15344000  15351000

LOR
  0003 *01659000*


LOWER
  0003 *05299000*
        05304000

LPRT
  0003 *01483000*


LQV
  0003 *01660000*


LS
  0093 *13750000*
       *13752000* 13765000

LSQ
  0076 *09364000*
        09366000

LSS
  0003 *01661000*


LSTRTN
  0003 *01430000*
```

```
LSTSEQ
  0034 *05016000*
       05020000  05023000

LTAPE
  0008 *02191000*
       02191250 *02195500*

LVL
  0003 *01151000*
       05116000  05208000  07432000  13204000  14220000  16005000  16171000  16265000

L1
  0035 *05049000*
       *05096000* 05099100

L1
  0041 *05334000*

L1
  0047 *06128000*
       06133000 *06152000*

L1
  0063 *07658000*
       07667500  07672500  07673000 *07675000*

L1
  0064 *07713000*
       07720000 *07729000*

L1
  0069 *09025000*

L1
```

```
  0084 *12005000*
       *12009000* 12057000

L1
  0091 *13635000*
        13639000  13641000

L1
  0093 *13751000*
       *13754000* 13761100  13762000

L1
  0102 *15076000*
       *15092000* 15116000

L1
  0103 *16475000*
        16482400 *16484000*

L1
  0112 *16365000*
        16372000 *16397000*

L1
  0111 *16313000*
       *16349000* 16353000

L10
  0047 *06128000*
        06133000  06152000

L10
  0064 *07713000*
        07720000 *07734000*

L10
  0103 *16475000*
        16476000 *16493000*
```

```
L11
  0047 *06129000*
         06134000 *06157000*

L11
  0064 *07714000*
         07721000  07729000

L12
  0047 *06129000*
         06134000  06152000

L12
  0064 *07714000*
         07721000 *07736000*

L13
  0047 *06129000*
         06134000  06152000

L13
  0064 *07714000*
         07721000  07729000

L14
  0047 *06129000*
         06134000 *06159000*

L14
  0064 *07714000*
         07721000  07729000

L15
  0047 *06129000*
         06134000  06159000

L15
```

```
  0064 *07714000*
         07721000  07729000

L16
  0047 *06129000*
         06134000  06152000

L16
  0064 *07714000*
         07721000  07729000

L17
  0047 *06129000*
         06134000  06152000

L17
  0064 *07714000*
         07721000  07729000

L18
  0047 *06129000*
         06134000 *06161000*

L18
  0064 *07714000*
         07721000  07734000

L19
  0047 *06129000*
         06134000  06161000

L19
  0064 *07714000*
         07721000  07734000

L2
  0041 *05334000*
        *05373000*
```

```
L2
  0047 *06128000*
         06133000  06152000

L2
  0064 *07713000*
         07720000  07729000

L2
  0084 *12005000*
         12018000  12048000 *12055000*

L2
  0093 *13751000*
         13753100  13754000 *13760000*

L2
  0103 *16475000*
         16481000  16482200 *16485000*

L20
  0047 *06129000*
         06134000  06152000

L20
  0064 *07714000*
         07721000  07729000

L21
  0047 *06130000*
         06135000  06152000

L21
  0064 *07715000*
         07722000  07729000

L22
```

```
  0047 *06130000*
         06135000 *06163000*

L22
  0064 *07715000*
         07722000 *07738000*

L23
  0047 *06130000*
         06135000  06163000

L23
  0064 *07715000*
         07722000  07738000

L24
  0047 *06130000*
         06135000  06152000

L24
  0064 *07715000*
         07722000  07729000

L25
  0047 *06130000*
         06135000  06152000

L25
  0064 *07715000*
         07722000  07729000

L26
  0047 *06130000*
         06135000  06163000

L26
  0064 *07715000*
         07722000  07738000
```

L27
  0047 *06130000*
         06135000 06163000

L27
  0064 *07715000*
         07722000 07738000

L28
  0047 *06130000*
         06135000 06152000

L28
  0064 *07715000*
         07722000 07729000

L29
  0047 *06130000*
         06135000 06152000

L29
  0064 *07715000*
         07722000 07729000

L3
  0047 *06128000*
         06133000 06152000

L3
  0064 *07713000*
         07720000 07729000

L3
  0084 *12005000*
         12026000 *12056000*

L3

0093 *13751000*

13767000 13770000

L3

0103 *16475000*

16476000 *16486000*

L30

0047 *06130000*

06135000 06163000

L30

0064 *07715000*

07722000 07738000

L31

0047 *06131000*

06136000 06163000

L31

0064 *07716000*

07723000 07738000

L32

0047 *06131000*

06136000 06153000

L32

0064 *07716000*

07723000 *07740000*

L33

0047 *06131000*

06136000 *06165000*

L33

0064 *07716000*

07723000 07730000

L34
0047 *06131000*
06136000 *06167000*

L34
0064 *07716000*
07723000 07730000

L35
0047 *06131000*
06136000 06165000

L35
0064 *07716000*
07723000 07730000

L36
0047 *06131000*
06136000 06167000

L36
0064 *07716000*
07723000 07730000

L37
0047 *06131000*
06136000 *06173000*

L37
0064 *07716000*
07723000 07730000

L38
0047 *06131000*
06136000 *06169000*

L38

```
  0064 *07716000*
         07723000 *07742000*

L39
  0047 *06131000*
         06136000 *06171000*

L39
  0064 *07716000*
         07723000  07730000

L4
  0047 *06128000*
         06133000  06152000

L4
  0064 *07713000*
         07720000  07729000

L4
  0103 *16475000*
         16476000 *16487000*

L40
  0064 *07716000*
         07723000 *07744000*

L41
  0064 *07717000*
         07724000  07770000

L42
  0064 *07717000*
         07724000 *07752000*

L43
  0064 *07717000*
         07724000 *07754000*
```

L44
0064 *07717000*
07724000 *07756000*

L45
0064 *07717000*
07724000 *07758000*

L46
0064 *07717000*
07724000 *07769000*

L47
0064 *07717000*
07724000 07769000

L48
0064 *07717000*
07724000 07769000

L49
0064 *07717000*
07724000 07770000

L5
0047 *06128000*
06133000 06152000

L5
0064 *07713000*
07720000 07729000

L5
0103 *16475000*
16476000 *16488000*

L50

0064 *07717000*

07724000 07769000

L51

0064 *07718000*

07725000 *07760000*

L52

0064 *07718000*

07725000 *07762000*

L53

0064 *07718000*

07725000 *07764000*

L54

0064 *07718000*

07725000 *07766000*

L6

0047 *06128000*

06133000 06152000

L6

0064 *07713000*

07720000 07729000

L6

0103 *16475000*

16476000 16489000

L7

0047 *06128000*

06133000 *06155000*

L7

0064 *07713000*

07720000 *07732000*

L7
  0103 *16475000*
        16476000 16482500 *16490000*

L8
  0047 *06128000*
        06133000 06152000

L8
  0064 *07713000*
        07720000 07732000

L8
  0103 *16475000*
        16476000 *16491000*

L9
  0047 *06128000*
        06133000 06152000

L9
  0064 *07713000*
        07720000 07729000

L9
  0103 *16475000*
        16476000 *16492000*

M
  0044 *05438000*
       05441000 05442000 05443000 05444000 05445000

M
  0042 *05401000*
       *05404000**05407000* 05407000 *05409000* 05409000 05409300 05409400

M

MAXSTACK
0003 *01683000*
13230000 *13231000* 13231000 14277000 *14502000*

MAXSTACKO
0094 *14029000*
14277000 14502000

MAXTLCR
0003 *01331000*
02439000 *02456000*

MCPBIT
0003 *01001000*
01001290 02397000

MCPTOG
0003 *01001290*

MCPTYPE
0003 *01561410*
09430060

MDESC
0069 *09005000*
09034200 09258500 09254000 09255000 09418000 09419100

MEDIUM
0003 *01001580*
02182750 02195000 02195750 *02203250**02204250**02205500**02214100* 02214200 02228750 02421000 05038000
07025020

MERGEBIT
0003 *01001010*
01001300 02425000 *02425500* 02425500 02426000 02427000 02428000 02435000 02436000

MERGEOPTION

0016 *02363000*

02428000 *02437000*

MERGETOG
0003 *01001300*

02425500 02427000 02428000 02436000

MIN
0005 *01822000*

01828000 01834000

MKABS
0003 *02022000*

02023000 02202000 02202050 02214000 02215750 02216500 02217750 02226000 02410500 02411000 02456000
02485500 02486000 02903000 05091000 07025010 07025030 08999175 08999200 09400000 09401000

MKS
0003 *01662000*

07408000 07434000 13761000 13765000

MKST
0093 *13750000*

*13752000**13761000* 13761000 13764000

MODE
0003 *01402000*

04175500 04184000 04212100 04223100 05115000 05345000 07397200 07600410 07748000 08093000 14303000
14303000 14478000 *14501000* 14501000

MON
0003 *01147000*

16254000

MONITORDEC
0094 *14014000*

14019000 *14142000*

MONITORV
0003 *01291000*

MOVE
  0003 *02054000*

02212250 02366100 02412000 02487000 02602600 02916000 04147000 05039500 07029000 07669000 08999075
08999550 08999575 08999700 09393130 09393170 09393190 09393350 09393400 09393430 09433000 09447060
13281000 13310000 13651000 13746000 14259000 14440000 16495100

MOVE
  0067 *08999075*

08999100 08999275 08999350

MOVECHARACTERS
  0003 *02045000*

02705000 07668500 16349000 16392000

MOVECODE
  0003 *13683000*

MOVEIT
  0021 *02638000*

02704000

MRCLEAN
  0003 *01379000*

04164000 *09040000*

MUL
  0003 *01663000*

08049000

MULOP
  0003 *01278500*

12022000

MYCLASS
  0003 *01001590*

02968000 02971000 02974000 *02974500* 02975000 02976500 02977500 *02978500* 02982000 02983000

```
M0
  0040 *05325080*
         05325240  05325320

M1
  0040 *05325080*
         05325250  05325350

M2
  0040 *05325080*
         05325260  05325360

M3
  0040 *05325080*
         05325270  05325380

M4
  0040 *05325080*
         05325280  05325390

M5
  0040 *05325080*
         05325290  05325400

M6
  0040 *05325080*
         05325300  05325410

M7
  0040 *05325080*
         05325310  05325420

N
  0003 *01693000*
         01759000  02249000  04204000  04237000  05334200  09393000  10258100  13221000  13673150

N
```

0003 *01737350*

01737350 01737450

N

0003 *01759000*

01781000

N

0003 *02041000*

02041000 02043000

N

0003 *02045000*

02046000 02050000

N

0003 *05325010*

05325460 05325470 05325480

N

0012 *02264700*

02264700 02264850 03036000 03037000 03047000 03047100 03050000 04115000 04116000

N

0012 *02264400*

02264400 02264550

N

0011 *02249000*

02251000 02252000 02255000 02259000

N

0029 *04204000*

04208000 04209000 04221000 04223000

N

0035 *05067000*

05067000 05075000 05246000 05247000 05271000 05298000 05299000

N
0030 *04237000*
04238000 04239000 *04240000* 04240000

N
0040 *05325040*
05325050 05325230

N
0046 *06064000*
06064000 06067000 07035000 07036000

N
0043 *05415000*
05415000 05421000 05437000

N
0041 *05334200*
05334400 05334600

N
0054 *07038100*
07038100 07038500

N
0067 *08999075*
08999075 08999100 08999450

N
0063 *07653500*
07653500 07655500

N
0077 *09393000*
09400000 09402100 09403000 09405500 09406000 09407000 *09413000* 09417000 09420000 *09430050* 09430075

N

```
  0076 *09378000*
        09378000  09381000  09393060  09393240

N
  0077 *09393700*
        09393700  09393730  10228000

N
  0077 *09393240*
        09393300

N
  0087 *13221000*
       *13258000* 13265000  13271000

N
  0082 *10257100*
        10257100  10257300

N
  0082 *10258100*
       *10263110* 10264700  10264900

N
  0092 *13673150*
        13673300  13673400

NAMEDEC
  0094 *14015000*
        14020000 *14161000*

NAMEID
  0003 *01205200*
        07046000  07048000  14161010  14161120  15076100  15276020  15276090  15295100  15303000  15344000  15351000

NAMEV
  0003 *01295000*
```

```
NCII
  0003 *01624000*
       13228000 *13228000* 14056000 *14057000*

NCIIO
  0094 *14031000*
       *14056000*

NCR
  0002 *00511000*
       00511000  00512000

NCR
  0003 *01330000*
       02128500  02129000  02129500 *02213000**02214000**02225250**02229000* 02229000  02230750 *02230750* 02410000
       02485000  02665000  02680000  02697000 *02720000* 02731000  02745000  02750000  02753000  02786000  02791000
       02807000  02813000  02888000  02902000 *02903000*

NCR
  0003 *01756000*
       01756000  01761000  01768000  01773000  01780000

NCR
  0007 *02089500*
       02126000

NCR
  0003 *02183500*
       02183750  02184500

NCRV
  0007 *02090000*
       02090500  02092500  02102000  02124500

NEQ
  0003 *01664000*
```

NER
  0076 *09364000*

        09364000 09368000

NESTBIT
  0003 *01001020*

        01001310 02472000

NESTCTR
  0003 *01722000*

       05375100 05375100 *14481200* 14483400

NESTCUR
  0003 *01722000*

       05402200 05402200 05409100 05409300 *05409300*

NESTFORM
  0043 *05415000*

        05426000 05431000

NESTLEVEL
  0003 *01484000*

       16126000 16126000 16142000 *16142000* 16166000 16171000 16260000 16265000

NESTOG
  0003 *01001310*

        01723000

NESTOG
  0003 *01723000*

        07397100 14483100 14601000

NESTPRT
  0003 *01724000*

        05402000 05402200 05402300 05404000 05409200 05409500 05425000 05425400 05430500 05447000 05448000
        14483300

NESTS
  0103 *16115000*

16144000 16484000

NESTSORT
  0003 *05411100*
       05425200 *05437000* 05442000 05443000

NEW
  0003 *02016000*
       02018000

NEWBASE
  0003 *01001490*
       02192000 *02192500**02583000**09028920*

NEWBIT
  0003 *01001030*
       01001320 02193750 02388000 09282500

NEWINCL
  0003 *01001330*

NEWINCLBIT
  0003 *01001040*
       01001330

NEWINX
  0003 *01000860*

NEWTAPBUF
  0003 *01490510*

NEWTAPE
  0003 *01560000*
       02194000 09282500

NEWTOG

```
  0003 *01001320*
        02193750  09282500

NEXT
  0003 *02956000*
       *02968000* 02968500  02974000  02974500  02978500  15076000

NEXT
  0102 *15076000*
       *15253000* 15292000

NEXTINFO
  0003 *01620000*
        02265350 *02726000**09033000* 10250000  10254000  10261000  10263000 *10263100* 10285000 *10285000**13291000*
        13302000  13303000  13304000 *13305000* 13308000  13309000  13310000  13311000  13312000 *13312000* 13593000
       *13594000* 13594000  13770600  14055000  14166500  14217000  14319000  14329000 *14330000**14373000* 14375000
       *14463000* 16442000

NEXTLINK
  0062 *07596000*
        07604050  07604080  07604090  07607000  07609000

NHI
  0003 *01689000*
       *02681000**02785000* 02800000  02801000  02833000  02835000  02838000  02840000  12015000

NINFOO
  0094 *14030000*
       *14055000* 14600000  14601000

NLAB
  0101 *14418000*
       *14422000**14437000* 14437000

NLO
  0003 *01689000*
       *02681000**02785000* 02800000  02833000  02835000  02838000  02840000  12014000

NLOC
```

0101 *14418000*

14422000 14433000 *14433000* 14451200

NLOCS
0094 *14022000*

NODIMPART
0003 *01532000*

07098000 07412000

NOHEADING
0003 *01001480*

01836000 02183000 02195000 02195750 02214200 02228750 02264950 02421000 05039600 05046000 05325490
07025020 *09028050* 09362000 13653500 13676000

NONLITNO
0003 *01209000*

02851000 02924000 07604120 07665000 07666500 08032000 13226000

NONSAVNDX
0077 *09393000*

09405500 09406000 *09407000* 09407000 09427100

NOP
0003 *01665000*

04212100 04223100 07078000 08094000 13752000 16012000

NOP
0103 *16012000*

NORMAL
0032 *04311010*

04311060 04311080 *04311120*

NOTOP
0003 *01272000*

02964000 02974500 06146000

```
NUMBEREND
  0021 *02640000*
       02687000 *02845000*

NUMBERS
  0007 *02091500*
       02096000  02102500  02117000

NUMLE
  0065 *08080000*
       08095000  08162000  08165000

NXTELBT
  0003 *01319000*
       02644000  02910000 *02912000* 02912000 *02917000**08110000**08174000**09038000* 10258200  10263400  10263500
      *10263500* 10277000  10282000  12112000 *12113000* 12113000  12125000

N1
  0005 *01822000*
      *01828500* 01835500

N2
  0005 *01822000*
      *01828500* 01835500

OCR
  0087 *13221000*
      *13249000* 13250000  13252500 *13257000* 13262000  13263000

OCTALWORDS
  0012 *02264400*
       02264650  02265100  02265250  02265700  02265950

OCTIZE
  0003 *02001838*
       02001852  02001854  02686000

OK
```

OPDC
  0065 *08016000*

        08108000 08130000 08217000

OPERATOR
  0003 *01271000*

        06042000 06076000

OPERATOR
  0003 *04011000*

        04014000

OPERATOR
  0003 *04006000*

        04007000

OPERATOR
  0003 *04030000*

        04034000

OPERATOR
  0045 *06042000*

       06044000 06054000

OPERATORS
  0003 *01299200*

        06071000

OPINX
  0003 *01000800*

       02311000 02312000 *02312000* 02314000 02316000 02317000 02334000 02338000 02342000 02356000

OPTIONLENGTH
  0016 *02364000*

        02373000

OPTIONS

0003 *01000904*

02312000 02314000 02317000 02331000 02334000 02338000 02352000 02356000 02426000 09251208 09414024
09414026

OPTIONWORD
0003 *01000910*

01001210 01001220 01001230 01001240 01001250 01001260 01001270 01001280 01001290 01001300 01001310
01001320 01001330 01001340 01001350 01001360 01001370 01001380 01001390 01001400 01001420 01001430
01001440 01001450 01001460 02191500 02193750 02194250 02195000 02195750 02214200 02228100 02228750
02231500 02232500 02235250 *02330000* 02335000 *02335000** *02339000* 02339000 02353000 *02353000* 02357000
02357000 02421000 02425500 02427000 02428000 02436000 02533000 02577000 02602500 04148000 04150000
04278500 04297000 05039600 05046000 05048000 05097500 05099000 05325490 05350100 05376100 06123000
07025020 07086500 09028000 09253050 09253100 09282500 09384500 09387000 09389500 09394100 09405000
09414010 09421000 09430050 09447010 13651100 13652000 13653500 13674000 13676000 13677100 14461500
14493500

OROP
0003 *01275000*

02965000 02984000

OUTDEC
0094 *14014000*

14019000 14158000

OUTPUTSOURCE
0003 *02190500*

02196500 02230500 02235000

OUTV
0003 *01289000*

OWNERR
0094 *14013000*

14018000 *14135000*

OWNV
0003 *01280000*

13383000 13723000 14161040

P

```
  0003 *01489000*
       05414100  12102000  13750000 *14335000* 14387000  14430000  14451200  14463000

P
  0003 *02635000*
       02644000  02917000 *02917000* 02920000  02923000  02924000

P
  0035 *05050000*
       05050000  05052000  05055000

P
  0035 *05059000*
       05059000  05061000

P
  0043 *05414100*
       05425300  05425400

P
  0085 *12102000*
       *12108000* 12126000

P
  0093 *13750000*
       *13752000* 13755000  13761000 *13761000* 13761200 *13761200* 13767000

PACK
  0003 *04270000*
        04276000  04296000

PACKIN
  0082 *10258300*
        10264000  10264200  10264400  10264420 *10264910*

PACKINFO
  0083 *10236000*
        10240000  10251000  10254000
```

```
PAN
  0076 *09364000*
         09386000

PANA
  0003 *03021000*
         06186000 *06413000* 06417000

PARM
  0098 *14254200*
         14254700  14259080

PARSE
  0003 *03014000*
         06035000  06181000 *06313000* 06335000

PASSFORMAT
  0003 *05214000*
         05227000

PCTR
  0054 *07038000*
        *07040000* 07042000 *07094000* 07094000  07098000

PDES
  0003 *01299020*
         13746000

PEN
  0076 *09370000*
         09377000  09388000

PERCENT
  0021 *02637000*
         02641000 *02731000* 02906000

PERIOD
```

0003 *01264000*

06105000 06106100 06340000 07021000 07026000 15111000 15233003 15276140 15339000

PERMANENT
0003 *05333000*

05345000

PINFOO
0094 *14026000*

14375000 14471000 14483000 14489000

PJ
0003 *01617000*

13344000 *13344000* 14073000 14085000 14117000 *14120000* 14120000 *14323000* 14333300 14334000 14335000
14349000 14364000 14374000 *14387000* 14411300 14415300

PLUG
0065 *08020000*

08116000 08188000 08213000 08218000

PN
0093 *13750600*

*13751100**13761110* 13770000 *13770600* 13773000

POINTER
0087 *13219000*

13222000 13223000 13226000 13229000 13240000 *13240000* 13249000 13255000 13257000 13263000 13279000
13281000 13284000 13286000 *13287000* 13287000 13290000

POL
0054 *07037000*

07052000 07059000

POLISHER
0003 *03009000*

06060000 06170000 07052000 07512000 07743000

POLISHV
0003 *01212100*

07052000 07511000 13338000

PORS
0003 *05325010*
05325460 05325470 05325480

PORS
0040 *05325040*
05325050 05325090 05331000 05333000

POSITION
0003 *04270000*
04271000 04273000 04311000 04500000 04503000 04505000 05008000

PRIMARY
0003 *03005000*
06016000 06018000 06023000 06035000 06049000 *06126000* 06147000 06172000 06181000 06182000 07076000

PRINT
0092 *13661000*
13673000 13675000

PRINT
0091 *13633000*
13643000 13653000

PRINTCARD
0003 *02182500*
02195000 02195750 02214200 02228750 02421000 07025020

PRINTDOLLARBIT
0003 *01001060*
01001350 02228750 02377000

PRINTDOLLARTOG
0003 *01001350*
02228750

PRL
  0003 *01666000*

PROAD
  0094 *14032000*
       *14313000**14320000* 14461100 14476000 14489000

PROADO
  0003 *01613000*
        14037000 *14476000* 14602000

PROCEDUREDEC
  0094 *14015000*
        14020000 14137700 *14270000*

PROCID
  0003 *01187000*
        07399000 07404000 13278000 14284000 14292000

PROCSTMT
  0003 *03029000*
        06162000 *07391000* 07425000 07735000

PROCV
  0003 *01293000*

PROGDESCBLDR
  0003 *03047000*
       05245000 05255000 07677000 13252500 13746000 14461100

PROGRAM
  0003 *09003000*
        09448000 16495200

PROINFO
  0003 *01608000*
        07090000 13677200 13677300 14304000 *14314000**14320000* 14388100 14411200 14411300 14415200 14415300

```
        14457000  14461200  14470000  14483200  14483300  14483400 *14496000* 15080000

PRT
  0003 *01343000*
        05248000  05252000  09414040  09415000

PRTAD
  0003 *13734000*
        13746000

PRTADR
  0003 *05247000*
       *05247500* 05247500  05248000  05252000

PRTBASE
  0003 *01342000*
        01343000  01724000  01725000  05424000  09039000  09414030  09415000  09419100

PRTBIT
  0003 *01001070*
        01001360  02395000  05376100

PRTE
  0003 *01667000*
        04018000  04023000  04033000  05316000  06110300  08068000  15106100  15310000

PRTI
  0003 *01703000*
       *05349000* 07600440 *09039000* 14022000  14054000

PRTIMAX
  0003 *01703000*
        05347000 *05350000* 05350000 *09039000* 09388000  09414100  09419000

PRTIO
  0094 *14030000*
       *14054000*

PRTOG
```

```
  0003 *01001360*
       05376100

PRTOP
  0003 *01342000*
       01343000  01724000  01725000  05424000

PRTSIZ
  0076 *09370000*
       09371000  09372000

PTOG
  0003 *01592000*
       13341100  13343000  13370000 *14332000**14373000**14388000*

PUNCHBIT
  0003 *01001080*
       01001370  02506000  05048000

PUNCHTOG
  0003 *01001370*
       05048000

PURGE
  0003 *03068000*
      *13215000* 13745000  13773000  14471000  14483000  14601000

PURGEBIT
  0003 *01001090*
       01001380  02508000

PURGETOG
  0003 *01001380*


PURPT
  0003 *01628000*
       02726000  13285000  13309000  14088000  14118000  14446000
```

PUSHEE
0077 *09393240*

09393480 09396000 09402100 09406000

PUSHER
0077 *09393050*

*09393140**09393200* 09393230 09424000 09427500

PUT
0003 *05008000*

05132000 06104400 07545000 07610000 09133900 09134100 13304000 13348000 13365000 13408000 13768000
14116000 14117000 14168000 14224000 14259150 14315000 14334000 14353000 14388100 14411200 14415200
14430000 144512oo 14457000 14470000 15376400 16138000 16166000 16167000 16172000 16257000 16260000

PUTNBUMP
0003 *03057000*

09034000 *13589000* 14167000 14195000 14217000 14221000 14319000 14331000 14403000 16000600 16443000
16445000 16446000

PUTOGETHER
0082 *10234000*

10257000 10263900 10268000 10269000 10270000 10273000 10274000

PUTSEQNO
0003 *02005000*

02006000 02202060 02214250 02218000 02234750 02368000 02716000 02904000

P1
0003 *03057000*

P1
0003 *15070000*

15102000 15104000 15110100 15114000 15119000 15233008 15233015 15276100 15276110 15276160 15276190
15276210 15305000 15334000 15335000 *15336000* 15342000 15349000 15364000 15367000 *15370000*

P1
0035 *05050000*

05096000

P2
0003 *01588000*

05059000 13341000 13343000 13347000 *13385000* 13722000 *13723000**14133000**14161070**14318000**14321000*

P2
0035 *05059000*

05096000

P3
0003 *01589000*

13725000 *13726000**14133000**14138000**14139000**14140000**14141000*

Q
0003 *01316000*

01822000 02280000 02313000 02314000 *02327000* 02343000 *02371000* 02375000 02376000 02378000 02382000
02384000 02386000 02389000 02392000 02394000 02396000 02400000 02406000 02420000 02422000 *02424000*
02433000 02462000 02467000 02469000 02471000 02473000 02477000 02479000 02481000 02495000 02497000
02499000 02505000 02507000 02509000 02515000 02519000 02521000 02523000 02529000 02566000 02576000
02588000 *02590000**02650000**02686500**02696000**02698000* 02700000 *02754000**02822000**02846000**02865000*
02879000 *02958500* 02960500 02961000 02964000 02964500 02965000 02965500 04311020 05043000 05044000
05343000 05414100 07086200 07086300 08015000 09393010 *13282000**14441000* 16213000 16378500

Q
0002 *00523000*

00523000 00524000 00526000 01716000 01717000 01717700 01717900 01718000 01719000

Q
0005 *01822000*

01833000 *01833000**01834000*

Q
0003 *04231000*

04239000 04241000 04243000

Q
0032 *04311020*

*04311050**04311070* 04311070 04311090

Q
0043 *05414100*
05425100 05425100 *05425200* 05425200 05425300

Q
0041 *05343000*
*05344750**05371000**05374000* 05376100

Q
0065 *08015000*
08093000 08094000 08185000 08200000 08210000 08218000

Q
0077 *09393010*
09395700 09405500 *09405500* 09447020

Q
0082 *10257100*
10257100 10257700

QUOTE
0021 *02637000*
02641000 *02690000*

R
0003 *02183500*
02183750 02185750 02202500

RANGE
0003 *05297000*
05304000

RCA
0103 *16008000*

RDV
0003 *01668000*

```
READACARD
  0003 *02065000*
         02130500 *02196750* 02238000  02731000  09035000

READTAPE
  0009 *02198500*
         02202250  02208500  02219500

REALARRAYID
  0003 *01203000*
         07056000  07057000  13380000  13389000

REALDEC
  0094 *14013000*
         14018000  14135000  14136000 *14138000*

REALID
  0003 *01199000*
         07060000  07069000  08038000  12035000  14138000  15276210

REALPROCID
  0003 *01195000*


REALSTRPROCID
  0003 *01191000*
         14294000

REALSUBID
  0003 *01184000*
         14164000  14167000

REALV
  0003 *01283000*
         13389000  14161120  14164000

RECOV
```

0087 *13220000*

13251000 13256000 *13281000*

RED
0003 *02011000*

02011000

REED
0003 *08999000*

08999175 08999425 09399000

REGO
0065 *08010000*

08015000 *08093000* 08094000 *08111000* 08120000 *08160000**08170000* 08185000 08200000 *08210000* 08218000
08231000

RELAD
0003 *13734000*

RELAD
0094 *14029000*

14487000 14489000

RELOP
0003 *01277000*

05344500 14260000 16207000

REMCOUNT
0082 *10230000*

10245000 10247000 *10248000**10249000* 10249000 *10263000*

REPEAT
0003 *04011000*

04013000 04014000

RESIZE
0003 *02056000*

RESTORESEQNUM
  0016 *02320200*

        02602800

RESULT
  0003 *01386000*

        02128500 02129000 02278000 *02327000* 02366200 *02371000**02499000* 02500000 02579000 02586000 *02590000*
        02591000 02598000 *02602700**02647000* 02648000 *02663000**02666000**02667000**02684000**02694000**02698000*
       *02700000**02744000**02747000**02749000**02751000**02752000**02754000**02757000**02788000**02793000**02806000*
       *02809000**02812000**02816000**02820000**02846000**02889000**02890000**02958500* 02959000 02959500 10265000
        10272000 10276000

RESULT
  0007 *02089500*

        02092500

RESULTSWITCH
  0021 *02643000*

        02648000

RESULTV
  0007 *02089500*

        02090500 02094000

RETURNSTORE
  0065 *08010000*

        08087000 08090000 *08166000**08168000**08190000* 08193000 *08207000* 08215000

RIGHT
  0003 *03039500*

        07675500 *08999450* 08999800 14461300 14603000

RINX
  0088 *13301000*

       13302000 13304000 13305000

ROSE
  0021 *02639000*

       02875000 02879000 02881000

ROUND
  0062 *07594000*
       07600000 07600900 07604000 *07645000*

ROW
  0041 *05344000*

RP
  0047 *06137000*

RRB1
  0003 *01522000*

RRB2
  0003 *01525000*

RR1
  0002 *00507000*
       *00529000**00530000* 01557000 *14329000* 14373000

RR10
  0002 *00507000*

RR11
  0002 *00507000*
       00529000 00531000 00533000 00536000

RR2
  0002 *00507000*
       00530000 01557000

RR3

0002 *00507000*

*00531000**00532000* 01558000

RR4

0002 *00507000*

00532000 01558000

RR5

0002 *00507000*

*00534000**00535000* 01560000

RR6

0002 *00507000*

*00534000**00535000* 01560000

RR7

0002 *00507000*

*00534000**00535000* 01560000

RR8

0002 *00507000*

*00537000**00538000* 01561000

RR9

0002 *00507000*

*00537000**00538000* 01561000 07595000 *07599000* 07601000

RSA

0103 *16018000*

RTBRKET

0003 *01265000*

05278000 06141000 06322000 06422000 10264300 12126000 13395000 13402000 14259120 15260000 15276010
15294000

RTN

0003 *01669000*

13740000

RTPAREN
  0003 *01266000*
       02639000 02976500 06115000 06175000 06416000 07067000 07096000 10264300 12039000 12126000 13756000
       13761100 13763000 14259120 14336000 16132000 16478000

RTPAREN
  0021 *02639000*
       02642000 *02744000*

RTS
  0003 *01670000*
       07087000 13264000 13268000

S
  0003 *02001858*
       02001868

S
  0003 *02001838*
       02001842

S
  0003 *03051001*

S
  0003 *03067000*

S
  0003 *04291000*
       04296000 04297000

S
  0003 *04277500*
       04280000 04280500 04281500 04282000 04283500 04284000 04285000 04285500

S

0012 *02264400*

02264500

S

0012 *02264700*

02264800 02307000

S

0033 *04502000*

04548000 04551000

S

0039 *05309000*

05309000 05311000 05325010

S

0042 *05401000*

05407000 *05407000*

S

0047 *06132000*

06138000

S

0043 *05416000*

05418000 05420000

S

0041 *05334100*

05334300 05334500

S

0046 *06063000*

06074000 *06074000**06103000* 06103000

S

0065 *08027000*

08029000

```
S
  0065 *08063000*
       *08065000* 08069000

S
  0064 *07719000*
        07727000

S
  0065 *08072000*
        08075000  08079000

S
  0084 *12004000*
       *12006000* 12025000 *12025000* 12030000 *12030000**12038000* 12038000  12055000 *12055000*

S
  0082 *10257100*
        10257200  10257800

S
  0085 *12103000*
        12104000  13216000  13217000  13315000  13316000  13590000  13591000  13632000

S
  0085 *12102000*
       *12110000*

S
  0098 *14254200*
        14254300  15070000  16029000  16065000  16085000

S
  0092 *13673100*
        13673200  13673350

S
```

0103 *16029000*

16032000

SAF

0003 *01300200*

13677400 *14299000*

SAN

0003 *01677100*

SAV

0003 *05247000*

05248000

SAVECODE

0003 *13632000*

13651000 13651100 13653000

SAVECODE

0091 *13633000*

13633000 13638000 13657000

SAVEINFO

0090 *13378000*

13394000 13396000 *13397000**13401000* 13405000 *13405000* 13408000

SAVEINX

0017 *02324000*

02331000 *02342000* 02352000

SAVEL

0003 *01623000*

13606000 *13628000* 14055000 *14609000*

SAVELO

0094 *14030000*

14055000 14609000

SAVERR
  0094 *14013000*
       14018000 *14136000*

SAVETIME
  0002 *00503000*

SAVEV
  0003 *01281000*
       13726000 14292000 14298000 14299000

SAVINFO
  0077 *09392300*
       09395100 09396000 09402100 09414070 09414110 09418000 09419000 09419100 09424000 09433000

SAVL
  0104 *16031000*
       16032000 16034000

SAVL
  0106 *16087000*
       16088000 16096000

SAVNDX
  0077 *09393000*
       09397000 09402100 *09403000* 09403000 09413000 *09414120**09420000* 09426000 09436000

SBIT
  0003 *07036000*
       07046000

SBV
  0003 *01244000*
       16197000 16415000

SCAN

0007 *02089500*

02127000 02128500

SCAN
0082 *10257100*

10264800

SCANAGAIN
0021 *02639000*

02646000 02650000 02686500 02696000 02728000 02737000 02890000

SCANNER
0003 *02066000*

02137000 02279000 02327000 02371000 02499000 02590000 02647000 02663000 02666000 02684000 02694000
02698000 02744000 02747000 02749000 02751000 02752000 02754000 02788000 02793000 02806000 02809000
02812000 02816000 02820000 02889000 02890000 02958500

SCATTERELBAT
0003 *03035000*

07395000 *13197000* 13208000 13327000 13767000 14206000 14305000

SCI
0003 *01677050*

SCLASS
0054 *07038000*

*07044000**07046000* 07048000 07056000 07060000 *07079000* 07088000

SCOUNT
0016 *02365100*

02366200 02602700

SCRAM
0003 *01301000*

*02865000**09385000* 09386000 12110000 *12122000* 13308000 13361000 13362000 13755000

SCS
0003 *01677150*

SCV
  0003 *01246000*

        16206000 16337000

SEARCH
  0003 *04231000*

       04243000 04244000 04280000 04280500 04283500 04284000

SEC
  0103 *16022000*

        16342000

SECOND
  0003 *05271000*

       05281000 05282000

SECOND
  0051 *06314000*

       06325000 06328000 06332000

SECOND
  0052 *06339000*

        06341000 06344000

SECRET
  0003 *01628000*

        13331000 14333000 14345000

SED
  0103 *16017000*

SED
  0101 *14419000*

SEGMENT

```
  0003 *03042000*
        07676500 *13657000* 13681000  14461200  14602000

SEGMENTSTART
  0003 *03041000*
        07661500 *13632000* 13655000  14298000  14387000

SEGSBIT
  0003 *01001100*
        01001390  02470000  13652000  13674000

SEGSIZEMAX
  0003 *01687000*
       *13678000* 13678000

SEGSTOG
  0003 *01001390*
        13652000  13674000

SEMICOLON
  0003 *01230000*
        05109000  07012000  07032000  10264410  13766000  14062000  14063000  14165000  14325000  14338000  14357000
        14519000  16131000

SEQ
  0003 *01756000*



SEQ
  0003 *04130000*
        04130000  04132000  04142000  04203000  04231000  04247000

SEQ
  0054 *07038100*
        07038300  07038600

SEQ
  0076 *09378000*
        09382000  09393050  09393240
```

SEQBIT
  0003 *01001110*

        01001400 02191500 02393000

SEQCOMPARE
  0009 *02202500*

        02209750 02224500

SEQERRBIT
  0003 *01001120*

        01001420 02520000

SEQERRTOG
  0003 *01001420*

SEQTOG
  0003 *01001400*

        02191500

SEQUENCEERROR
  0003 *01742100*

        01742110 02226750

SEQXEQTOG
  0003 *01001560*

       02525000 02525000

SES
  0101 *14419000*

SET
  0085 *12103000*

        12117000

SETTING

```
  0003 *01000802*
       *02327000* 02338000  02352000  02356000  02427000

SGAVL
  0003 *01369000*


SGNO
  0003 *01370000*
        14054000

SGNOO
  0094 *14030000*
       *14054000*

SIGNA
  0065 *08012000*
        08014000 *08056000* 08065000  08098000  08108000  08130000  08179000  08181000  08189000  08212000  08217000
       *08227000*

SIGNB
  0065 *08012000*
        08050000  08112000 *08121000* 08126000  08183000  08196000 *08203000* 08214000

SIGNC
  0065 *08012000*
        08014000  08044000 *08112000* 08185000  08197000 *08202000* 08219000 *08220000*

SIMPARITH
  0003 *03003000*
        06006000 *06032000* 07085000

SIMPGO
  0003 *03026000*
       *07548000**07555000* 07557000  07564000  07570000  07581000

SIMPI
  0065 *08052000*
       *08061000* 08061000  08176000  08211000  08224000
```

SIMPLE
0065 *08027000*

*08033000**08038000* 08039000 08112000 08181000 08183000 08185000

SIMPLEB
0065 *08014000*

08044000 08112000 08220000

SIMPLEV
0065 *08015000*

08074000 08099000 08211000 08224000 08227000

SINGLBIT
0003 *01001130*

01001430 02195000 02195750 02214200 02228750 02421000 02522000 04150000 04278500 05039600 05046000
05097500 05099000 05325490 06123000 07025020 07086500 09384500 09387000 09389500 13653500 13676000

SINGLTOG
0003 *01001430*

02181000 02195000 02195750 02214200 02228750 02421000 04150000 04278500 05039600 05046000 05097500
05099000 05325490 06123000 07025020 07086500 09384500 09387000 09389500 13653500 13676000

SIV
0003 *01236000*

16319000 16320000 16328000 16334000

SIZE
0003 *03042000*

SIZE
0003 *05247000*

05248500 05249000 05250000 05252000 05254000

SIZE
0003 *07647000*

07663000 07675500 07676500 07677000

```
SIZE
  0003 *13657000*
        13675000  13677400  13678000

SIZE
  0092 *13661000*
        13661000  13670000

SKAN
  0003 *02277000*
        02327000  02371000  02499000  02590000  02958500

SKANAGAIN
  0016 *02360000*
       *02370000* 02378000  02383000  02385000  02389000  02465000  02468000  02478000  02503000  02527000  02577250
        02584000  02595000  02599000

SKBIT
  0063 *07653500*
        07653500  07655000  08020000  08027000  08083000  08084000

SKIPCOUNT
  0083 *10241000*
       *10247000* 10253000  10254000  10256000

SKIPIT
  0024 *02974000*
        02974500  02978500

SKIPS
  0103 *16403000*
        16419000  16492000

SKIPV
  0003 *01241000*



SKP
```

```
  0003 *02001838*
         02001838  02001842

SKP
  0003 *02041000*
         02041000  02043000

SKP
  0003 *02001858*
         02001858  02001890

SKSC
  0082 *10259000*
        *10263500* 10283000

SN
  0093 *13750000*
        *13755500* 13767000  13768000 *13769000* 13769000

SND
  0003 *01671000*
         15104000  15233015  15276110  15335000

SOP
  0003 *14003000*
         14036000  14506000  14600000

SORCE
  0003 *02045000*
         02048000

SORTNEST
  0003 *05412000*
         14601000

SORTPRT
  0003 *01725000*
         05425100  05425400  05447000  05448000  05452000  05455000  05459000
```

SPACEITDOWN
0094 *14023100*

14461500 14493500

SPCLMON
0102 *15076200*

15349000 15351300 15351400 15354000

SPECIAL
0003 *01003000*

02655000 02666000 02755000 02961500 02963000 07031000 09198000

SPECIALCHAR
0021 *02637000*

02643000 *02651000*

SPECIALSWITCH
0021 *02641000*

02659000

SPECTOG
0003 *01593000*

05034000 13305100 13328000 13385000 13601000 13621000 13719000 14070000 *14122000* 14142000 14149000
14160000 14161070 14187000 14205000 14275000 14287000 *14364000**14368000* 14377000 14507000

SPRT
0003 *01698000*

SRESULT
0016 *02365100*

02366200 02602700

SSKIP
0003 *02045000*

02046000 02049000

SSP

0003 *01672000*

05226000

SSS
0046 *06063500*

06068500 06125000

STACKCT
0003 *01566010*

04527000 *06054150* 06158000 *06158000**06172000* 06172000 *06174500* 06174500 *06179500* 06179500 *06296500*
*06299500**06305000**06332500**07041200**07100500**07726990**14125500**15091000**15098000**15101500**15233011*
15233011 *15233016**15234500**15276115**15278000**15335500**15369000*

STACKCTR
0003 *01683000*

05370000 *05371000* 05371000 06068500 *06125000* 14058000 14477000 *14478000* 14481200 *14493000* 14506000
14610000

STACKCTRO
0094 *14029000*

*14058000**14477000* 14493000 14610000

STACKHEAD
0003 *01310000*

02724000 02865000 05090000 05233000 05234000 09134100 09134200 09405500 09447030 09447040 13286000
13308000 13361000 13362000 13768000 13769000 14448000 16443000

START
0066 *08084000*

08086000

START
0094 *14017000*

14063000 14138000 14139000 14140000 14141000 14156000 14160000 14162000 14170000 14199000 14202000
14253000 14254050 14269000 14272000 14379000 14384000 14504000 14521000

START
0097 *14202000*

```
START
  0099 *14272000*
       *14373000*

START
  0098 *14254050*
        14259090 *14269000*

START
  0100 *14384000*
       *14503000*

START
  0103 *16475000*
       *16477000* 16485000

STARTINTRSC
  0069 *09024000*
```

```
START1
  0099 *14272000*
        14325000  14364000 *14375000*

START2
  0099 *14273000*
       *14376000*

STD
  0003 *01673000*
        08090000  12038000  13230000  15104000  15233015  15276110  15335000

STEPI
  0003 *05003000*
       *05003000* 05108000  05274000  05275000  05276000  05277000  05278000  05344500  05344520  05344610  05344670
        06069000  06071000  06085000  06094000  06106100  06199000  06317000  06318000  06319000  06320000  06321000
        06322000  06415000  06421000  07018000  07041000  07437000  07505000  07552000  07567000  07597500  07600100
        07600600  07604110  07604140  07645000  07662000  07664000  07666500  07673000  07727070  07767000  07768100
        07994000  08029000  08178000  09252000  12007000  12010000  12013000  13351000  13392000  13395000  13402000
        13404000  13753000  13754000  13756000  13766000  13771000  14126000  14220000  14223000  14259110  14268000
```

14338000 14341000 14345000 14356000 14393000 14396000 14467000 15090000 15113000 15233000 15233006
15253000 15260000 15276050 15276160 15300000 15342000 16125000 16160000 16197000 16206000 16207000
16208000 16217000 16220000 16252000 16253000 16316000 16319000 16324000 16326000 16340000 16367000
16368000 16372000 16384000 16406000 16434000 16435000

STEPIT
0003 *05002000*

05344690 06015000 06035000 06049000 06102000 06104700 06106200 06111000 06117000 06140000 06143000
06147000 06166000 06168000 06172000 06174000 06177000 06181000 06201000 06208000 06303000 06331000
06345000 06410000 06411000 06416000 06417000 06422000 06423000 07010000 07072000 07100000 07407000
07485000 07487000 07493000 07495000 07505000 07516500 07518000 07552000 07569000 07573000 07580000
07584000 07604100 07727020 07727090 07996000 08031000 08039000 08120000 08131000 08135000 08142000
08155000 08161000 08175000 08192000 08231000 12013000 12017000 12025000 12030000 12034000 12036000
12039000 12045000 12054000 13327000 13406000 13409000 13753100 13757000 13758000 13761100 13772000
14166000 14193000 14259000 14259070 14259140 14314000 14359000 14403000 14472000 15098000 15233012
15274000 15276070 15306000 15376800 16130000 16176000 16234000 16376000 16408000 16412000 16494000

STEPV
0003 *01255000*

08103000 08182000

STLABID
0003 *01178000*

14403000 14447000 16000300 16253000 16437000 16443000 16481000

STLB
0003 *01471000*

15273000

STMT
0003 *03027000*

05108000 07011000 07485000 07495000 07574000 07577000 07585100 *07711000* 07771000 08163000 08201000
09275000 14167000 14488000 14515000

STMTSTART
0065 *08010000*

08157000 *08169000**08191000* 08193000 *08206000* 08221000

STUP
0100 *14384000*

14484000 *14491000*

```
STUPDEFINE
  0003 *01597000*
        02893000 *02911000**02926000**07017000**10263300**10281000**12111000**13327000**14130000**14192000**14258000*
       *14259060**14259100**14259130**14402000**14403000*

STUPENTRY
  0003 *01594000*
        13351000 *14188000**14198000**14205000**14251000**14255000**14269000**14318000**14321000**14402000**14404000*


STUPGSP
  0003 *01694000*
        13343000  13345000 *14164500**14166000**14188000**14198000**14205000**14251000**14255000**14269000**14318500*
       *14319500**14402000**14404000*

STUPPER
  0003 *03068000*



STUPPER
  0003 *13217000*
        13223000  13291000

STUPPER
  0003 *13734000*
        13745000

STURE
  0065 *08063000*
        08070000  08130000  08135000  08138000  08216000

STURE
  0066 *08084000*
        08089000

STUREFIX
  0066 *08091000*
       *08093000* 08157000  08171000

STREAMERR
```

05350300 09281000 13677500

STUFFBIT
0003 *01001140*
01001440 02516000 05350100 13677100

STUFFF
0003 *05199000*
05203000

STUFFTOG
0003 *01001440*
05350100 13677100

ST1
0007 *02090000*
02090500 02120000 02124500 02126000

ST2
0007 *02090000*
02090500 02121000 02121500

SUB
0003 *01674000*
06016000 08030000 08044000 08214000 12015000 16329000

SUBC
0003 *13213000*

SUBDEC
0094 *14014000*
14019000 *14163000*

SUBHAND
0003 *03011000*
06156000 *06194000* 06209000 07733000

SUBID

```
  0003 *01185000*

       06198000  14164000

SUBLEVEL
  0003 *01402000*

       05117000 *14477000* 14477000 *14492000*

SUBOP
  0003 *01556500*

       05344570  05344710

SUBV
  0003 *01288000*



SUPERFRMTID
  0003 *01183000*

       05221000  05224000

SVARMONFILE
  0003 *01529000*



SW
  0008 *02191250*

       02194500

SW
  0017 *02326000*

       02328000

SWITCHDEC
  0094 *14015000*

       14020000 *14200000* 14252000

SWITCHID
  0003 *01186000*

       07507000  14207000
```

SWITCHIT
  0016 *02321000*

02359000 02377000 02388000 02393000 02395000 02397000 02402000 02415000 02421100 02425000 02435000
02470000 02472000 02474000 02480000 02490000 02496000 02506000 02508000 02511000 02516000 02520000
02522000 02531000 02567000 02600000

SWITCHV
  0003 *01292000*

SWITMONFILE
  0003 *01539000*

SYLLABLE
  0003 *03028000*

SYLLABLE
  0003 *03019000*

SYLLABLE
  0003 *04270000*

04271000 04274000 04311000 04500000

SYMBOL
  0003 *02183500*

02183750 02185250

T
  0003 *01717900*

T
  0003 *02057000*

02058000 02059000

T
0003 *01564000*
02057000 02249000 02323000 *02655000* 02658000 *02666000**02692000* 02695000 *02701000* 02705100 *02721000*
02722000 *02724000* 02726000 *02755000* 02798000 *02813000* 02823000 *02823000* 02824000 02826000 *02826000*
02827000 02842000 *02848000* 02850000 02851000 *02865000* 02875000 02876000 *02877000* 02877010 *02877020*
*02878000**02882000* 02882000 02886000 02896000 *02896000* 02896000 02900000 02901000 02910000 02957500
02981500 05068000 05096000 *05233000* 05234000 05236000 05401000 05414000 *08057000* 08057000 08058000
08059000 08061000 *08069000* 08070000 *09282600**09388000* 09388000 09389000 09393020 12004000 13199000
14163500 16067000 16364000

T
0003 *03050000*

T
0003 *04311000*
04311030 04312000 04318000

T
0003 *12101000*
12108000 12117000

T
0018 *02446000*
02446000 02448000 02449000 03024000 03025000 03036000 03037000 03042000 03047000 03047100 03050000
03051000 03051001 04010000 04011000 04029000 04030000 04115000 04116000

T
0018 *02441000*
02441000 02443000 02444000

T
0017 *02323000*

T
0011 *02249000*
*02251000**02257000**02259000* 02259000 02260000

T
0025 *02981500*

02982500 02983000 02983500 02984000 02984500

T
0022 *02957500*
*02960500**02961000* 02961000 02961500 *02964000* 02966500

T
0035 *05068000*
05074000 05081000 05086000

T
0043 *05414000*
05425400 05426000 05428000 *05430500* 05431000

T
0042 *05401000*
05402000 05402100 05403000 05404000 05405000 05409000 *05409300* 05409300 *05409400* 05409400 05409500

T
0067 *08999075*
08999075 08999100

T
0076 *09364000*
09364000 09369000

T
0077 *09393020*

T
0084 *12004000*
12044000 12047000

T
0086 *13199000*
13200000 13201000 13202000 13203000 13204000 13205000 13206000 13207000

T
0096 *14163500*
14166500 14168000

T
0105 *16067000*
16070000

T
0112 *16364000*
16393000 16394000

TABLE
0003 *02635000*
02920000 **02924000* 02927000 05002000 05003000 05221000 05224000 07066000 07397210 07401000 07600950
07604020 07748000 07751000 07768130 10263400 10282000 12020000 12112000 13753100 13761100 14063000
14516000 15276010 16482300

TAKE
0003 *01717700*
02724000 02725000 *03053000** *05005000** *05006000* 05124000 05131000 05189000 06082000 06104300 06191000
07042000 07090000 07098000 07412000 07441000 07527000 07529000 07539000 07601000 09133900 09134100
10263110 13226000 13229000 13249000 13255000 13257000 13279000 13284000 13286000 13348000 13362000
13364000 13365000 13408000 13677200 13767000 14084000 14088000 14118000 14168000 14259150 14310000
14320000 14333300 14352000 14388100 14411200 14415200 14425000 14430000 14438000 14446000 14448000
14451200 14470000 14483200 14483400 15376300 16070000 16089000 16138000 16139000 16163000 16166000
16172000 16254000 16260000 16449000

TAKEFRST
0003 *05188000*
05189000 05222000 06197000 07435000

TALL
0102 *15072000*
15077000 15080000 *15083000* 15083000 15085000 15086000 15092000 15094000 15097000 15102000 15105000
15106100 15107000 15110200 15110300 15124000 15128100 15233009 15233010 15233011 15233014 15233018
15255000 15267000 15276020 15276060 15276080 15276090 15276210 15290000 15295100 15302000 15303000
15310000 15333000 15344000 15344100 15344200 15351000 15351200 15352000 15353000 15376300

TALLYV
0003 *01239000*
16319000 16341000

```
TAN
  0103 *16014000*
        16213000

TAPE
  0003 *01561000*
        02201750  02452000  02454000  02457000  09282000

TAPE
  0003 *02188000*
        02188000  02188500  02202750

TAPELAST
  0009 *02210000*
        02210750 *02219250*

TBUFF
  0003 *01561056*
        02201750  02202000  02456000  02457000  07025030

TB1
  0003 *01457000*
       *07506000**07507000* 07516200  07518000 *10243000**10260000* 10275000  14204000

TB1
  0097 *14204000*
       *14206000*

TCLASS
  0111 *16314000*
       *16315000* 16317000  16324000  16328000  16334000  16341000 *16348000* 16349000  16350000  16352000

TCOUNT
  0003 *01566000*
        02251000  02252000 *02758000**02790000* 02796000  02798000  02803000 *02810000**02817000*

TEDOC
```

0003 *13683000*

13686000 13703000 13706000 13732000 13734000

TEMPL
0028 *04166000*

04172000 04175000 04184500 04186000

TEMP1
0003 *02001860*

02001866 02001868 02001890

TEMP2
0003 *02001860*

TEN
0003 *01340000*

$0_2$796000 $0_2$799000 $0_2$803000 $0_2$826000 $0_2$833000 $0_2$835000 $0_2$838000 $0_2$840000 09041000

TEST
0065 *08041000*

08051000 08131000 08220000

TESTCODE
0109 *16192000*

16211000 16212500

TESTLEV
0100 *14384100*

14387000 14389000 14411300 14415300 14451000 14461100

TESTVOID
0009 *02210500*

02232500 *02234000*

THENBRANCH
0050 *06295000*

06297000 06301000

THENV
0003 *01256000*
06411000 16217000

THERE
0003 *01737350*
01737450

THI
0003 *01689000*
02250000 02255000 02256000 02257000 02785000 02797000 02799000 02800000

THIRD
0051 *06314000*
06326000 06327000 06328000 06332000

TIME1
0003 *01300000*
01828000 09385000 *16495200*

TL
0027 *04118000*
04119000 04124000

TL
0057 *07483000*
07485000 07487000

TL
0087 *13221000*
13258000 13277000

TLCR
0003 *01330000*
02202050 02202060 02202070 02224500 *02456000*

TLCR

0009 *02202750*

02203500 02204000

TLU
0003 *01689000*

02250000 02255000 02256000 02785000 02796000 02797000 02799000 02800000

TOGGLEV
0003 *01245000*

TOLOC
0069 *09005000*

09006000

TOP
0044 *05439000*

05445000 *05452000*

TOTALNO
0003 *01000860*

02192000 *02192500* 02193000 *02193000**02582000*

TOV
0003 *01257000*

07505000 07552000 16252000 16437000

TOWARDS
0003 *04116000*

04119500 04120100 04120200 04120500

TOWARDS
0003 *03037000*

TRB
0003 *01682000*

04318000

```
TRNSFER
  0003 *01251000*
         16379000

TRP
  0103 *16019000*
         16383000

TRUTHV
  0003 *01208000*


TRW
  0101 *14419000*


TSSINTYPE
  0003 *01561430*
         09430060

TSSTOG
  0077 *09393020*
        *09414026* 09430050

TSUBLEVEL
  0094 *14029000*
        *14477000* 14492000

TURNONSTOPLIGHT
  0003 *02011000*
         02202070  02214500  02218250  02235750  02369000  02717000  02905000

TWXA
  0003 *01561700*
         05350180  05350300  13677400  13677500

TYPE
```

0003 *03055000*

TYPE
0003 *05247000*
05248000 05249000 05250000 05253000

TYPE
0003 *13316000*
13333500 13338000 13342000

TYPE
0003 *13716000*
13729000

TYPE
0050 *06295000*
06299000

TYPE
0103 *16476000*
16477000

TYPEV
0003 *01611000*
*13380000**13389000* 13390000 *14091000**14161010**14161120* 14162000 14163500 *14281000**14284000* 14285000
14292000 14294000 *14294000* 14319000

TYPEV
0003 *03071000*

TYPEV
0096 *14163500*
14164000 14165000 14167000

T1
0003 *01693000*
02001860 04278000 04502000 05049000 06062000 06184000 06196000 07427000 07562000 08173000 *13381000*

13383000 13389000 13684000 *14161000**14161020* 14161040 14161120 15073000

T1
0003 *02001860*
02001874 02001876

T1
0003 *13684000*
13685000 13707000 13711000 13732000 13734000

T1
0033 *04502000*
04550000 04551000

T1
0031 *04278000*
04280500 04281000 04281500 *04284000* 04284500 04285000

T1
0035 *05049000*
05088000 *05091000* 05091000 05096000 05097000 *05097000**05099000* 05099000

T1
0046 *06062000*
06072000 06077000 *06079000* 06080000 06081000 06082000 06087000 06089000 *06104200* 06104600 *06107200*
06108000 06110000 06110100

T1
0048 *06184000*
06185000 06187000 *06189000* 06191000

T1
0049 *06196000*
06197000 06206000 06207000

T1
0056 *07427000*
07435000 07436000 *07440000* 07442000 07443000

```
T1
  0065 *08173000*
       *08179000**08193000* 08202000  08203000  08204000  08205000  08206000  08207000  08225000

T1
  0061 *07562000*
       *07566000* 07572000  07573000  07575000 *07577000* 07579000  07583000  07585100

T1
  0102 *15073000*
        15095000  15097000  15098000  15100000  15112000  15124000  15233005  15233013  15233018  15276060  15276070
        15276150  15276210  15304000  15308000  15341000  15370000 *15376300**15376500* 15376500  15376600 *15376600*
        15376700

T2
  0003 *01693000*
        02001860  04502000  05049000  06062000  06184000  07562000  07657500  08173000  13684000  15074000

T2
  0003 *02001860*
        02001874  02001876

T2
  0003 *13684000*
        13686000  13694000  13698000

T2
  0035 *05049000*
       *05092000* 05093000  05095000  05098000

T2
  0033 *04502000*
       *04550000*

T2
  0048 *06184000*
       *06185000* 06189000 *06190000* 06190000  06191000

T2
```

0046 *06062000*

06072000 06074000 06075000 06079000 *06080000* 06080000 06081000 *06084000* 06087000 06089000 06090000
06090000 06092000 *06093000* 06099100 06099200 *06104300* 06104400 *06106000**06106200* 06110200

T2
0065 *08173000*

08198000 08208000

T2
0063 *07657500*

07663000 07668500 07669000 07671500

T2
0061 *07562000*

07583000 07583000 *07585000* 07586000

T2
0102 *15074000*

15100000 15112000 15124000 15233005 15233013 15233018 15276070 15276150 15276210 15308000 15341000
15370000 *15376300* 15376400

T3
0003 *13684000*

13689000 13697000 13700000 13710000 13732000 13734000

T3
0035 *05049000*

05091000 05092000 *05093000* 05094000 05096000 *05098000*

T3
0065 *08173000*

08199000 08209000

T4
0035 *05049000*

05095000 05096000

T4
0065 *08173000*

08200000 08210000

U
0003 *05411100*

U
0003 *05437000*
05440000 05441000 05443000 05446000 05454000 05457000 05458000

UNHOOK
0003 *01626000*
13305300 *13353000*

UNKNOWNID
0003 *01177000*
06151000

UNTILV
0003 *01226000*
07018000 07485000 08113000 08128000 08184000

UPPER
0003 *05299000*
05304000

USEROPINX
0003 *01001170*
02335000 02339000 02353000 02357000 02600000 09414022

USESWITCH
0009 *02210750*
02212000

USETHESWITCH
0009 *02210250*
02211750 02228100 02228150 02229500 02231500 02232250 02234000 02234600 02235250

V

0065 *08010000*

08015000 08078000 08099000 08181000 08188000 *08225000**08227000*

VAL

0003 *01741200*

01741300

VALUEV

0003 *01258000*

14341000

VARIABLE

0003 *03038000*

06140000 06164000 07050000 07058000 07073000 07083000 07402000 07739000 08228000 12037000 *15070000*
15377000

VBIT

0054 *07039000*

07043000 07045000

VD

0003 *01150000*

07043000 13203000

VOIDBIT

0003 *01001150*

01001450 02231500 02415000

VOIDCR

0003 *01785000*

02231750 02232000 *02232000**02410500* 02410500 02411000

VOIDING

0003 *01001450*

02231500

VOIDPLACE

0003 *01785000*

02232000 02410500 02412000

```
VOIDRANGE
  0016 *02365200*
       02410000  02411000  02412000  02485000  02486000  02487000

VOIDTAPE
  0003 *01001460*
       02228100  02232500

VOIDTBIT
  0003 *01001160*
       01001460  02228100  02232500  02490000

VOIDTCR
  0003 *01785000*
       02228125  02228150  02233000  02233500 *02233500* 02485500 *02485500* 02486000

VOIDTPLACE
  0003 *01785000*
       *02233500* 02485500  02487000

VONF
  0003 *01590000*
       *13203000**13341000* 13373000 *16000400*

VP
  0003 *01756000*
        01761000  01780000

VRET
  0065 *08010000*
       *08054000* 08077000 *08176000* 08198000 *08208000* 08211000  08224000 *08228000*

W
  0003 *02054000*
        02054000  02055000

W
```

0003 *01717700*

W
0039 *05309000*
05311000 05315000 05325010

W
0092 *13673100*
13673100 13673250

WD
0069 *09005000*
09005000 09006000

WHAT
0078 *09393080*
09393170 09393180 09393190

WHATISIT
0016 *02362000*
02364000 02365000 02379000 02398000 02475000 02517000 02568000 *02578000*

WHILESTMT
0003 *07490000*
07496000 07757000

WHILEV
0003 *01224000*
08113000 08133000 08139000

WHO
0078 *09393080*
09393110 09393130 09393170 *09393190* 09393200

WHOLE
0056 *07427000*
07430000 07432000 07438000 07439000 *07441000* 07441000 07442000 07444000

WITHV
  0003 *01259000*

WOP
  0003 *01371000*
        02534000 02536000 04280500 04281000 04281500 04284000 04284500 04285000

WORD
  0003 *04142000*
        04147000 04149000

WORD
  0003 *04270000*
        04273000 04277500 04291000 04311000 04500000 04503000 04504000 05005000 05008000 05012000

WORD
  0003 *05008000*
        05009000

WORDCOUNT
  0088 *13301000*
       13302000 13307000 13310000 13312000

WRITEAX
  0054 *07038100*
        07086300

WRITELINE
  0003 *02181000*
        02183000 02195000 02195750 02214200 02228750 02421000 04150000 05039600 05046000 05097500 05099000
        05325490 06123000 07025020 07086500 09384500 09387000 09389500 13653500 13676000

WRITEOUT
  0046 *06064000*
        06120000

WRITEPRT

0003 *05325010*

05325500 05376100

WRITERROR

0034 *05016000*

05031000 05044000

WRITNEW

0003 *02016000*

02020000 02194000

X

0003 *13591000*

13593000

X

0035 *05067000*

05072000 05074000 05105000 05113000 05123000 05128000 05199000 05207000 05246000 05247000 05271000
05298000 05299000 05307000

X

0079 *09393270*

09393400 09393410 09393420 09393430

XBIT

0016 *02321000*

02327000 02335000 02339000 02353000 02357000

XCH

0003 *01675000*

04507000 06203000 08065000 08123000 08228000 12054000 15106100 15309000

XIT

0003 *01676000*

02210500 05325080 13743000

XIT

0009 *02210500*

02213250 02214750 *02237750*

```
XIT
  0040 *05325080*
        05325340  05325350  05325370  05325380  05325390  05325400  05325410  05325430

XITR
  0003 *01430000*



XMAS
  0077 *09393050*
        09393110  09393130  09393140  09393170  09393180 *09393180* 09393190  09393200

XMODE
  0003 *01001530*
        02328000 *02332000**02367000**02383000**02385000* 02408000  02425500 *02478000* 02483000  02525000  02577000


XMODEO
  0017 *02325000*
        02326000 *02329000*

XMODE1
  0017 *02325000*
        02326000 *02333000*

XMODE2
  0017 *02325000*
        02326000 *02337000*

XMODE3
  0017 *02325000*
        02326000 *02341000*

XMODE4
  0017 *02325000*
        02326000 *02355000*

Y
```

```
  0077 *09393240*
       09393290 *09393330* 09393350  09393400 *09393430* 09393440  09393450

Z
  0003 *01490000*
       06063500  13746000 *14411200**14415200* 14457000  14506000 *14506000*

Z
  0046 *06063500*
       *06099100*

ZEERO
  0063 *07658500*
       07660500  07662000

ZP1
  0003 *01677000*
```

```
LABEL  OLPA    OLPA    00176366CC USER=0000000; REMOVE ESPOL/DISK;COMPILE ESPOL/DISK TSPOL LIBRARY;TSPO TSPOL  /ESPOL
```