

?USER = SITE

PACKET 168
INPUT 6 CARDS FROM CRA
TIME 1220
DATE 77116 TUESDAY, 04/26/77

DAN ROSS

*** MESSAGE OF THE DAY ***

SYSTEM WAS COLD STARTED 4/15/77 USING 4/11/77 BACKUP TAPES
THE FORTRAN COMPILER WAS RECOMPILED 7 APR 77. REPORT BUGZ TO DAN ROSS.
PLEASE USE USER CARDS IN FRONT OF ALL BATCH JOBS OR PACKETS

WHEN YOU GET TIRED OF THIS MESSAGE -
SUGGEST A BETTER ONE.

B5700
LISP
from Ft. Monmouth

*** BURROUGHS B5700 TSMCP MARK XVI.0.69 AND INTRINSICS MARK XVI.0.00 ***

?USER= SITE

?COMPILE LISP/LISP XALGOL LIBRARY

?XALGOL FILE TAPE= LISP/APTLIB DISK SERIAL

?DATA CARD

4: XALGOL/LISP/SITE= 4 BOJ 1220 10/08/76
CDC IN CARD DC: XALGOL/LISP= 4
DKA IN SER LISP APTLIB: XALGOL/LISP= 4
PBD0168 OUT 011 LINE: XALGOL/LISP= 4
DKA OUT RDM LISP LISP: XALGOL/LISP= 4

4TI

TIME FOR XALGOL/LISP= 4 IS 1:14 IN 1:56
DKA LOK LISP LISP: XALGOL/LISP= 4
DKA REL LISP APTLIB: XALGOL/LISP= 4
PBD0168 REL 011 LINE 1874: XALGOL/LISP= 4
CDC REL CARD DC: XALGOL/LISP= 4

?END

XALGOL/LISP/SITE= 4, PST= 2:07 EOJ
FOR XALGOL/LISP= 4: PST= 126, IOT= 49, CORE=15360
PKT#0168 REMOVED

LISP /LISP
=====

BEGIN %8 SEPTEMBER 1969

%
% CONSTANT DEFINES
%

FILE DATA REMOTE(2,10);
DEFINE MEMORYROWS = 127#,NOOFPROCESSOPS=44#,STKSZ=2#;
REAL LASTUSEDSPACE,XADDR,LINKTONIL,CARADDR,NEXTTOKEN;
REAL HASH,PRIORITY,TKNO,LABTAB,T,XX1,XX2;
POINTER CP;ARRAY JUNKARRAY[0:0],STACK[0:STKSZ,0:511];
DEFINE CHR=JUNKARRAY[0]#,LEXTREE=HASH#;
REAL LASTRPID,STACKPOSITION;
BOOLEAN TRACE;
REAL ATOMUNDEF,NULLIST,AT0,AT1,AT2,AT3,AT4,AT5,AT6,AT7,RC1,RC9;
ARRAY SYSMARRAY[0:2], TYPEARRAY[0:15];
ARRAY PROCESSARRAY[0:NOOFPROCESSOPS];
ARRAY MEMORY[0:MEMORYROWS, 0:511];
ARRAY C1[0:0], C2[0:0],C3[0:0],C4[0:0],C5[0:0],C6[0:0];
REAL CURROW;
DEFINE TYPE = [22:4]#,
ADDR=[18:19]#;
DEFINE TYPEGARBAGE =0#,
TYPECONS=1#,
TYPELIST=2#,
TYPEAREA=3#,
TYPEFIELD=4#,
TYPESYMB=5#,
TYPELOGIC=6#,
TYPELINK =7#,
TYPEPROCESS=8#,
TYPEMONITOR=9#,
TYPEGENERAL=10#,
TYPETOKEN=11#,
TYPECODE=11#

%
% TOKEN ASSIGNMENTS
%
% 1 TO 30= ALGOL RETURN LABELS
% 100 = ASGN
% 101 = REF
% 102 = CAR
% 103 = CDR
% 104 = CONS
% 105 = NEWSYMB
% 106 = AMTSPACE
% 107 = MAKE
% 108 = PRINT

START OF SEGMENT	T	*****	2
00001000	T	0000	
00002000	T	0000	
00003000	T	0000	
00004000	T	0000	
00005000	T	0000	
00006000	T	0003	
00007000	T	0003	
00008000	T	0003	
00009000	T	0003	
00010000	T	0007	
00011000	T	0007	
00012000	T	0007	
00013000	T	0007	
00014000	T	0007	
00015000	T	0010	
00016000	T	0012	
00016050	T	0014	
00016060	T	0025	
00017000	T	0025	
00018000	T	0025	
00019000	T	0025	
00020000	T	0025	
00021000	T	0025	
00022000	T	0025	
00023000	T	0025	
00024000	T	0025	
00025000	T	0025	
00026000	T	0025	
00027000	T	0025	
00028000	T	0025	
00029000	T	0025	
00030000	T	0025	
00031000	T	0025	
00032000	T	0025	
00033000	T	0025	
00034000	T	0025	
00035000	T	0025	
00036000	T	0025	
00037000	T	0025	
00038000	T	0025	
00039000	T	0025	
00040000	T	0025	
00041000	T	0025	
00042000	T	0025	
00043000	T	0025	
00044000	T	0025	

```

X 109 = INLEX
X 110 = GCL
X 111 = CREATE
X 112 = DELETE
X 113 = SUSPEND
X 114 = RESUME
X 115 = RETURN
X 116 = IDENTITY
X 117 = HALT
X 118 =
X 119 =
X 120 = TRACE
X 121 = EQLIST
X 122 = QUOTE
X 123 = ASGNENV
X 124 = REFENV
X 125 = EXE
X 126 = DEBUG
X 127 = LDMODE
X 128 = CONSTRUCT
X 129 = AREA
X 130 = TOKEN
X 131 = SINGLESTEP
X 132 = UP
X 133 = DP
X 134 = READ
X 135 =
X 136 = LIST
X 137 = COND
X 138 = EQ
X 139 = ATOM
X 140 = BLOCKOP
X 141 = NEQ
X 142 = IF
X 143 = NONTERM
X144=USERL
X145=LTAB
X146=COPY
X147=ADD
X148=SUB
X149=MUL
X150=DIV
X151=EX
X152=NEG
X
X 512 TO 576 = SPECIAL CHARACTERS
X
X 118 = ALTERNATIVE OP IN BNF INTERPRETER
X 1000 = RESULT VARIABLE FOR BASE LANGUAGE
X 1001 = LIST FORMAL PARAMETER
X 1002 = QUOTED FORMAL PARAMETER
X 1003 = QUOTED LIST FORMAL PARAMETER
X 1004 = BASE LANGUAGE INTERPRETER
X 1005 = BNF INTERPRETER
X 1006 = STRING VARIABLE FOR BNF
X 1007 = QUOTED SYMBOL IN SCANNER
X 1008 = SINGLE STEP VARIABLE

```

```

00045000 T 0025
00046000 T 0025
00047000 T 0025
00048000 T 0025
00049000 T 0025
00050000 T 0025
00051000 T 0025
00052000 T 0025
00053000 T 0025
00054000 T 0025
00055000 T 0025
00056000 T 0025
00057000 T 0025
00058000 T 0025
00059000 T 0025
00060000 T 0025
00061000 T 0025
00062000 T 0025
00063000 T 0025
00064000 T 0025
00065000 T 0025
00066000 T 0025
00067000 T 0025
00068000 T 0025
00069000 T 0025
00070000 T 0025
00071000 T 0025
00072000 T 0025
00073000 T 0025
00074000 T 0025
00075000 T 0025
00076000 T 0025
00077000 T 0025
00078000 T 0025
00079000 T 0025
00079010 T 0025
00079020 T 0025
00079030 T 0025
00079035 T 0025
00079040 T 0025
00079045 T 0025
00079050 T 0025
00079055 T 0025
00079060 T 0025
00080000 T 0025
00081000 T 0025
00082000 T 0025
00083000 T 0025
00084000 T 0025
00085000 T 0025
00086000 T 0025
00087000 T 0025
00088000 T 0025
00089000 T 0025
00090000 T 0025
00091000 T 0025
00092000 T 0025

```

```

% 1009 = TRACE VARIABLE
% 1010 = REMOTE TERMINAL DEVICE
% 1011 = DISK DEVICE
% 1012 = # IN BNF ALTERNATIVE
% 1013 = LEXUNIT
% 1014 = ID
% 1015 = INT
% 1016 = SPCHAR
% 1017 = CODE GENERATOR INTERPRETER
% 1018 = TREE VARIABLE IN BNF INTERPRETER
% 1019 = N VARIABLE IN BNF INTERPRETER
%1020=NUMBER
%
% 2000 TO 10000 = IDENTIFIERS
%
% 262144 TO 524287 = USER TOKENS
%
% DEFINED SUBROUTINES
%
DEFINE FNS(FNS1)=PSCAN(1,FNS1,F)#,STATS=PSCAN(0,0,V1)#;
DEFINE M[M1]=(IF BOOLEAN(M1)THEN
MEMORY[(M1),[18:9],[M1],[9:9]],[22:23] ELSE
MEMORY[(M1),[18:9],[M1],[9:9]],[45:23])#;
STEPSPACE=(LASTUSEDSPACE:=LASTUSEDSPACE+1)#;
AT(AT1)=(ATO+AT1)#;
TOKEN(TOKEN1)=(TKNO+TOKEN1)#;
ATOM(ATOM1)=((T:=ATOM1,TYPE)=TYPELOGIC OR T=TYPESYMB OR T=TYPETOKEN)#;
DELETE(DELETE1)=BEGIN ASGN(AT2,DELETE1,ATOMDELETED)END#;
SUSPEND(SUSPEND1)=BEGIN IF REF(AT2,SUSPEND1) NEQ ATOMRESUMED THEN
PRIMERRR("SPND",SUSPEND1)ELSE ASGN(AT2,SUSPEND1,ATOMSUSPENDED)END#;
AMTSPACE=((O&(MEMORYROWS+1)[18:8:9])=LASTUSEDSPACE-1)#;
ATOMUND=ATOMUNDEF#;
ATOMCAR=AT1#;
ATOMCDR=ATO#;
ATOMASSO=ATO#;
ATOMPRIORITY=AT1#;
ATOMSUSPENDED=AT1#;
ATOMRESUMED=AT2#;
ATOMDELETED=AT3#;
LOGIC(LOGIC1)=MAKE(TYPELOGIC,LOGIC1)#;
DEC(DEC1)=(DEC1-1)#;
MAKE(MAKE1,MAKE2)=((MAKE2)&(MAKE1)[22:3:4])#;
QQ(QQ1)=QU(1,QQ1)#;
UQ=QU(2,0)#;
INQ(INQ1)=QU(3,INQ1)#;
GCL=QU(4,0)#;
DEFINE SMEM(SMEM1)=QU(6," "&SMEM1[41:35:36])#;
DEFINE LMEM(LMEM1)=QU(7," "&LMEM1[41:35:36])#;
DEFINE INLEX(INLEX1)=IL(1,0,0,INLEX1)#,INTLX=IL(0,0,0,0)#;
LOADMODE(LOADMODE1,LOADMODE2)=IL(2,LOADMODE1,LOADMODE2,0)#;
DEFINE ID(ID1)=(ID1,TYPE=TYPECODE AND ID1.ADDR LSS 10000
AND ID1.ADDR GEQ 2000)#;
INT(INT1)=(INT1,TYPE=TYPELOGIC)#;
NBR(NBR1)=(NBR1,TYPE=TYPELOGIC OR (IF NBR1,TYPE=TYPEAREA
THEN (IF REF(ATO,NBR1),ADDR EQL 3 THEN TRUE ELSE FALSE)ELSE FALSE))#;
SPCHAR(SPCHAR1)=(SPCHAR1,TYPE=TYPECODE AND SPCHAR1.ADDR
LEQ 576 AND SPCHAR1.ADDR GEQ 512)#;

```

```

00093000 T 0025
00094000 T 0025
00095000 T 0025
00096000 T 0025
00097000 T 0025
00098000 T 0025
00099000 T 0025
00100000 T 0025
00101000 T 0025
00102000 T 0025
00103000 T 0025
00103010 T 0025
00104000 T 0025
00105000 T 0025
00106000 T 0025
00107000 T 0025
00108000 T 0025
00109000 T 0025
00110000 T 0025
00111000 T 0025
00112000 T 0025
00113000 T 0025
00114000 T 0025
00115000 T 0025
00116000 T 0025
00117000 T 0025
00118000 T 0025
00119000 T 0025
00120000 T 0025
00121000 T 0025
00122000 T 0025
00123000 T 0025
00124000 T 0025
00125000 T 0025
00126000 T 0025
00127000 T 0025
00128000 T 0025
00129000 T 0025
00130000 T 0025
00131000 T 0025
00132000 T 0025
00133000 T 0025
00134000 T 0025
00135000 T 0025
00136000 T 0025
00137000 T 0025
00138000 T 0025
00139000 T 0025
00140000 T 0025
00141000 T 0025
00142000 T 0025
00143000 T 0025
00144000 T 0025
00144010 T 0025
00144020 T 0025
00145000 T 0025
00146000 T 0025

```

%	00147000	T	0025
% FORWARD PROCEDURE DECLARATIONS	00148000	T	0025
%	00149000	T	0025
REAL PROCEDURE TAIL (STRING); VALUE STRING; REAL STRING; FORWARD;	00150000	T	0025
REAL PROCEDURE USERS (F); REAL F; FORWARD;	00151000	T	0025
REAL PROCEDURE USERL (X); REAL X; FORWARD;	00152000	T	0025
REAL PROCEDURE PSCAN (IJ, X, F); REAL X, F; INTEGER IJ; FORWARD;	00153000	T	0025
REAL PROCEDURE BNFTREE (F); REAL F; FORWARD;	00154000	T	0025
PROCEDURE MEMSAVE (X, N, Y);	00155000	T	0025
VALUE N, Y; ARRAY X [0]; REAL N, Y; FORWARD;	00156000	T	0025
PROCEDURE UNSAVE (X, N, FID);	00157000	T	0025
VALUE FID; ARRAY X [0]; REAL N, FID; FORWARD;	00158000	T	0025
PROCEDURE GARBAGECOLLECT (QMEM, N);	00159000	T	0025
VALUE N; ARRAY QMEM [0]; REAL N;	00160000	T	0025
FORWARD;	00161000	T	0025
REAL PROCEDURE COPY (X); VALUE X; REAL X; FORWARD;	00162000	T	0025
PROCEDURE DOTRACE (PROCESS); VALUE PROCESS; REAL PROCESS; FORWARD;	00163000	T	0025
PROCEDURE EQLIST (X, Y); VALUE X, Y; REAL X, Y; FORWARD;	00164000	T	0025
PROCEDURE PRINT (X); VALUE X; REAL X; FORWARD;	00165000	T	0025
PROCEDURE UREAD (FIL, BUF); VALUE FIL; REAL FIL; ARRAY BUF [0]; FORWARD;	00166000	T	0025
BOOLEAN PROCEDURE EQ (X, Y); VALUE X, Y; REAL X, Y; FORWARD;	00167000	T	0025
REAL PROCEDURE LEXFIND; FORWARD;	00168000	T	0025
REAL PROCEDURE QU (P, Y);	00169000	T	0025
VALUE P, Y; INTEGER P; REAL Y;	00170000	T	0025
FORWARD;	00171000	T	0025
PROCEDURE PRIMERROR (A, V);	00172000	T	0025
VALUE A, V; REAL A, V; FORWARD;	00173000	T	0025
REAL PROCEDURE CONS (X, Y); VALUE X, Y; REAL X, Y; FORWARD;	00174000	T	0025
REAL PROCEDURE CAR (X); VALUE X; REAL X; FORWARD;	00175000	T	0025
REAL PROCEDURE CDR (X); VALUE X; REAL X; FORWARD;	00176000	T	0025
PROCEDURE MASGN (A, V); VALUE A, V; REAL A, V; FORWARD;	00177000	T	0025
PROCEDURE RESUME (PROCESS); VALUE PROCESS; REAL PROCESS; FORWARD;	00178000	T	0025
PROCEDURE RETURN (PROCESS, VAL);	00179000	T	0025
VALUE PROCESS, VAL; REAL PROCESS, VAL; FORWARD;	00180000	T	0025
PROCEDURE DEBUG; FORWARD;	00181000	T	0025
PROCEDURE INTERP; FORWARD;	00182000	T	0025
PROCEDURE SYSM; FORWARD;	00183000	T	0025
REAL PROCEDURE CREATE (START, ENV, INTERPRETER, PROCESS);	00184000	T	0025
VALUE START, ENV, INTERPRETER, PROCESS;	00185000	T	0025
REAL START, ENV, INTERPRETER, PROCESS; FORWARD;	00186000	T	0025
PROCEDURE ASGN (X, Y, Z); VALUE X, Y, Z; REAL X, Y, Z; FORWARD;	00187000	T	0025
REAL PROCEDURE NEWSYMB (ASSO); VALUE ASSO; REAL ASSO; FORWARD;	00188000	T	0025
REAL PROCEDURE REF (X, Y); VALUE X, Y; REAL X, Y;	00189000	T	0025
FORWARD;	00190000	T	0025
REAL PROCEDURE HANGON (X, Y);	00191000	T	0025
VALUE X, Y; REAL X, Y; FORWARD;	00192000	T	0025
REAL PROCEDURE IL (M, C, CM, F);	00193000	T	0025
INTEGER M; REAL C, CM, F; FORWARD;	00194000	T	0025
REAL PROCEDURE CHAR (P, N);	00195000	T	0025
POINTER P; INTEGER N; FORWARD;	00196000	T	0025
REAL PROCEDURE PMAKE (T, N); VALUE T, N; REAL T, N; FORWARD;	00197000	T	0025
REAL PROCEDURE DA (X); VALUE X; REAL X; FORWARD;	00197010	T	0025
REAL PROCEDURE AD (X); VALUE X; REAL X; FORWARD;	00197020	T	0025
REAL PROCEDURE AR (X, Y, Z); VALUE X, Y, Z; REAL X, Y, Z; FORWARD;	00197030	T	0025
PROCEDURE APLPNT (X); VALUE X; REAL X; FORWARD;	00197040	T	0025
REAL PROCEDURE TAIL (STRING); VALUE STRING; REAL STRING;	00198000	T	0025
BEGIN REAL X;	00198010	T	0025

```

IF CAR(TAIL:=X:=CDR(STRING)).TYPE=TYPEAREA AND NOT NBR(CAR(X)) THEN
ASGN(ATO,STRING,(TAIL:=CONS(USERS(CAR(X)),X))) END;

```

```

START OF SEGMENT ***** 3
00199000 T 0000
00200000 T 0012
3 IS 22 LONG, NEXT SEG 2

```

```

REAL PROCEDURE QU(P,Y);
VALUE P,Y;INTEGER P;REAL Y;
BEGIN OWN INTEGER N;REAL V;

DEFINE PR(PR1)=(PR1,[41:19])#;
FORMAT F1(I4,"":",A4,I6,"",",A4,I6);

```

```

00201000 T 0025
00202000 T 0025
00203000 T 0025
START OF SEGMENT ***** 4
00204000 T 0000
00205000 T 0000

```

```

INTEGER L,LP; ALPHA A;
LABEL B1,B2,BB,BB1;
SWITCH PSW:=B1,B2,BB,B2,B2,B2,B2,B2;
GO TO PSW[P];
BB:N:=0;
V:=MAKE(TYPELOGIC,0);
Y:=CONS(LABTAB,CONS(Y,NULLIST));
P:=8;
GO TO BB1;
B1:QU:=ATOMUND;
V:=REF(ATOMPRIORITY,Y);
BB1:N:=N+1;
A:=Y&V[45:22:23];
B2: BEGIN OWN ALPHA ARRAY X[0:N];

LABEL B3,B4,B5,B6,B7,B8,B9;
LABEL B10,QDUMP,MEMDP,MEMLD,QIN,QSET;
INTEGER I,K,IS;
ALPHA XS;
SWITCH SW:=B3,B4,B5,B10,QDUMP,MEMDP,MEMLD,QIN,QSET;
GO TO SW[P];
QSET:K:=LASTUSEDSPACE,[18:18]+2;
FOR I:=0 STEP 1 UNTIL N DO
X[I]:=MEMORY[(K+I),[17:9],[K+I],[8:9]];
LABTAB:=CAR(X[0],[45:23]);HASH:=CAR(CDR(X[0],[45:23]));
GO TO B9;
QIN: X[0]:=Y&Y[45:22:23];
GO TO B3;
QDUMP:FOR P:=0 STEP 1 UNTIL N DO
WRITE(DATA,F1,P,TYPEARRAY[X[P],[45:4]],X[P],[41:19],
TYPEARRAY[X[P],TYPE],X[P],ADDR);
GO TO B9;
MEMDP:MEMSAVE(X,N,Y); GO TO B9;
MEMLD:UNSAVE(X,N,Y);P:=9;GO TO B2;
B10:GARBAGECOLLECT(X,N);
LABTAB:=CAR(X[0],[45:23]);
HASH:=CAR(CDR(X[0],[45:23]));GO TO B9;
B3:X[N]:=A
;
I:=N;
B6:IS:=I DIV 2;

```

```

START OF SEGMENT ***** 5
5 IS 10 LONG, NEXT SEG 4
00206000 T 0000
00207000 T 0000
00208000 T 0000
00209000 T 0007
00210000 T 0010
00211000 T 0010
00212000 T 0012
00213000 T 0014
00214000 T 0015
00215000 T 0016
00216000 T 0016
00217000 T 0018
00218000 T 0020
00219000 T 0022

```

```

START OF SEGMENT ***** 6
00220000 T 0004
00221000 T 0004
00222000 T 0004
00223000 T 0004
00224000 T 0004
00225000 T 0012
00225010 T 0015
00225020 T 0016
00225030 T 0018
00225031 T 0024
00225040 T 0029
00226000 T 0029
00227000 T 0032
00228000 T 0032
00229000 T 0034
00230000 T 0043
00231000 T 0053
00232000 T 0054
00233000 T 0056
00235000 T 0061
00236000 T 0063
00237000 T 0065
00238000 T 0068
00239000 T 0069
00240000 T 0070
00241000 T 0071

```

```

IF I LSS 2 THEN GO TO B9
ELSE IF PR(X[I]) GTR PR(X[IS]) THEN
BEGIN XS:=X[IS];
X[IS]:=X[I];
X[I]:=XS;
I:=IS; GO TO B6 END
ELSE GO TO B9;
B4:QU:=X[1],[22:23];
X[0]:=X[0]&X[1][22:23];

I:=1
]
IS:= 2*I;
WHILE IS LSS N DO BEGIN
IF PR(X[IS]) GTR PR(X[IS+1])
THEN BEGIN X[I]:=X[IS];
I:=IS END
ELSE BEGIN X[I]:=X[IS+1];
I:=IS+1 END;
IS:=2*I END
]
IF IS GTR N THEN BEGIN
N:=N-1
]
FOR K:=I STEP 1 UNTIL N DO
X[K]:=X[K+1];
IF N EQL 0 THEN BEGIN
N:=N+1; GO TO B9 END END
ELSE BEGIN N:=N-1;X[I]:=X[IS]END;
P:=3; GO TO B2;
B5:P:=2;
B9: PRIORITY:=X[1],[45:23] END
END QU;

```

```

00242000 T 0072
00243000 T 0073
00244000 T 0075
00245000 T 0077
00246000 T 0078
00247000 T 0080
00248000 T 0081
00249000 T 0081
00250000 T 0083
00251000 T 0086
00252000 T 0086
00253000 T 0086
00254000 T 0087
00255000 T 0088
00256000 T 0090
00257000 T 0091
00258000 T 0095
00259000 T 0095
00260000 T 0098
00261000 T 0099
00262000 T 0100
00263000 T 0101
00264000 T 0102
00265000 T 0102
00266000 T 0103
00267000 T 0105
00268000 T 0109
00269000 T 0110
00270000 T 0112
00271000 T 0115
00272000 T 0118
00273000 T 0119
00274000 T 0121

```

```

6 IS 125 LONG, NEXT SEG 4
4 IS 30 LONG, NEXT SEG 2

```

```

PROCEDURE PRIMERROR (A,V);
VALUE A,V;REAL A,V;
BEGIN FORMAT F1("SERR:",A4,X1,A4,I6);

WRITE(DATA,F1,A,TYPEARRAY[V,TYPE],V,ADDR);
A:=0/0;
END PRIMERROR;

```

```

00275000 T 0025
00276000 T 0025
00277000 T 0025
START OF SEGMENT ***** 7
START OF SEGMENT ***** 8
8 IS 8 LONG, NEXT SEG 7
00278000 T 0000
00279000 T 0012
00280000 T 0013
7 IS 14 LONG, NEXT SEG 2

```

```

%
% BASE LANGUAGE INTERPRETER
%
%ROUTINE INTERPRET(PROCESS)=

```

```

00281000 T 0025
00282000 T 0025
00283000 T 0025
00284000 T 0025

```

```

% [ INT1: WHEN STATUS/PROCESS NEQ "SUSPENDED";
% IF STATUS/PROCESS NEQ "RESUMED" THEN DELETE(SELF);
% INSTVAL(PROCESS);
% GO TO INT1 ]];
%
%ROUTINE INSTVAL(PROCESS)=
% [ INST:= INST/NEXT/PROCESS;
% NEXT/PROCESS:= SUCCESSOR/NEXT/PROCESS;
% ARGVAL(INST, PROCESS);
% RETURN(CALLER/SELF); DELETE(SELF) ]];
%
%ROUTINE ARGVAL(ARG,PROCESS)=
% [ IF ATOM(ARG) THEN ARG:= I(ARG)/ENV/PROCESS ELSE
% ARG:= APPLY(OPVAL(OP/ARG,PROCESS),ARGLIST/ARG,PROCESS);
% RETURN(CALLER/SELF,ARG); DELETE(SELF)];
%
%ROUTINE OPVAL(OP,PROCESS)=
% [ IF NOT ATOM(OP) THEN OP:= ARGVAL(OP,PROCESS);
% IF ATOM OP THEN OP:= I(OP)/MACHINE;
% RETURN(CALLER/SELF,OP); DELETE(SELF) ]];
%
%ROUTINE APPLY(ROUTINE,ARGLIST,PROCESS)=
% IF ATOM(ROUTINE) THEN
% [RETURN(CALLER/SELF, PRIMVAL(ROUTINE,
% LISTVAL(ARGLIST,PROCESS),PROCESS)); DELETE(SELF)];
% ENV:=PARAPASS(FPARA/ROUTINE,MAKE("CONSTRUCT"),
% ARGLIST,PROCESS);
% RETURN(CALLER/SELF, CALL(ROUTINE,ENV,PROCESS));
% DELETE(SELF) ]];
%
%ROUTINE CALL(ROUTINE,ENV,PROCESS)=
% [ RESUME(CREATE(START/ROUTINE,ENV,INTERPRETER/ROUTINE,
% PROCESS));
% DELETE(SELF) ]];
%
%ROUTINE PARAPASS(FPARA,ENV,ARGLIST,PROCESS)=
% [ IF FPARA NEQ "NIL" AND ARGLIST NEQ "NIL" THEN
% IF CAR(FPARA) = "UNDEF" THEN
% CAR(CDR(FPARA))/ENV:= LISTVAL(ARGLIST,PROCESS) ELSE
% [ CAR(FPARA)/ENV:= ARGVAL(CAR(ARGLIST),PROCESS);
% ENV:= PARAPASS(CDR(FPARA),ENV,CDR(ARGLIST),PROCESS) ]];
% RETURN(CALLER/SELF,ENV) ]];
%
%ROUTINE LISTVAL(ARGLIST,PROCESS)=
% [ IF ARGLIST="NIL" THEN
% RETURN(CALLER/SELF,"NIL") ELSE
% RETURN(CALLER/SELF,CONS(ARGVAL(CAR(ARGLIST),PROCESS),
% LISTVAL(CDR(ARGLIST),PROCESS)) ]];
%
PROCEDURE INTERP;
BEGIN REAL ARG,ARGLIST,STRING,OP,ENV,FPARA,SAV,T,PROCESS;

DEFINE ROUTINE=OP#;
DEFINE GET1=CAR(ARG)#,GET=CAR((ARG:=CDR(ARG)))#;
DEFINE STACK(STACK1)=SAV:=CONS(STACK1,SAV)#,
EXIT=BEGIN T:=CAR(SAV);SAV:=CDR(SAV);
IF T.TYPE NEQ TYPECODE THEN PRIMERROR("EXIT",T);

```

```

00285000 T 0025
00286000 T 0025
00287000 T 0025
00288000 T 0025
00289000 T 0025
00290000 T 0025
00291000 T 0025
00292000 T 0025
00293000 T 0025
00294000 T 0025
00295000 T 0025
00296000 T 0025
00297000 T 0025
00298000 T 0025
00299000 T 0025
00300000 T 0025
00301000 T 0025
00302000 T 0025
00303000 T 0025
00304000 T 0025
00305000 T 0025
00306000 T 0025
00307000 T 0025
00308000 T 0025
00309000 T 0025
00310000 T 0025
00311000 T 0025
00312000 T 0025
00313000 T 0025
00314000 T 0025
00315000 T 0025
00316000 T 0025
00317000 T 0025
00318000 T 0025
00319000 T 0025
00320000 T 0025
00321000 T 0025
00322000 T 0025
00323000 T 0025
00324000 T 0025
00325000 T 0025
00326000 T 0025
00327000 T 0025
00328000 T 0025
00329000 T 0025
00330000 T 0025
00331000 T 0025
00332000 T 0025
00333000 T 0025
00334000 T 0025
00335000 T 0025
START OF SEGMENT ***** 9
00336000 T 0000
00337000 T 0000
00338000 T 0000
00339000 T 0000
00340000 T 0000

```



```

GO TO RETSWIT,ADDRJEND#;
LABEL INTERPRET,INSTVAL,ARGVAL,APPLY,CALL,OPVAL,PARAPASS,
LISTVAL,PRIMVAL,RETURNVAL,INT1,OP1,L1,
R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,
R16,R17,R18,R19,R20,R21,R22,R23,R24,R25,R26,R27,R28,R29,R30,R31,
R32,R33,R34,R35,R36,R37,R38,R39,R40,R41,R42,R43,R44,R45,
R46,LEND,R47,R48,R49,R50,R51,R52,R53,R54,R55,R56,R57;
SWITCH RETSW:=R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,
R16,R17,R18,R19,R20,R21,R22,R23,R24,R25,R26,R27,R28,R29,R30,R31,
R32,R33,R34,R35,R36,R37,R38,R39,R40,R41,R42,R43,R44,R45,R46,R47,R48,
R49,R50,R51,R52,R53,R54,R55,R56,R57;
REAL RC2,RC3,RC4,RC5,RC6,RC7,RC8,RC10;
REAL PROCEDURE UNSTCK(SAV);REAL SAV;
BEGIN UNSTCK:=CAR(SAV);SAV:=CDR(SAV)END;

```

```

00341000 T 0000
00342000 T 0000
00343000 T 0000
00344000 T 0000
00345000 T 0000
00346000 T 0000
00347000 T 0000
00348000 T 0000
00349000 T 0002
00350000 T 0002
00350001 T 0002
00351000 T 0031
00352000 T 0031
00353000 T 0031

```

```

DEFINE LV(LV1)=BEGIN STACK(MAKE(TYPECODE,LV1));
GO TO LISTVAL;END#;
DEFINE UNSTACK=UNSTCK(SAV);
LABEL BNFINSTVAL,BNFEXITVAL,BNFTERMVAL,BNFRETURNVAL,
BNFNONTERMVAL,WAITRETURN,BNFL1,BNFL2,BNFL3;
REAL INST,X,VAL;
DEFINE STR=CODE(1006)#,SLEXUNIT=CODE(1013)#,
SID=CODE(1014)#,SINT=CODE(1015)#,
SSPCHAR=CODE(1016)#,SQQUOTE=CODE(122)#,
SCGINT=CODE(1017)#,TREE=CODE(1018)#,
N=CODE(1019)#,CODE(CODE1)=TOKEN(CODE1)#;
DEFINE SNBR=CODE(1020)#;
BOOLEAN PROCEDURE LOOK1(X,Y);VALUE X,Y;REAL X,Y;
IF(X=SLEXUNIT)OR(X=SID AND ID(Y)) OR(X=SINT AND INT(Y))
OR(X=SNBR AND NBR(Y))OR(X=SSPCHAR AND SPCHAR(Y))
OR(X=Y)OR(X,TYPE EQL TYPelist)
OR(X=CODE(1012)) THEN LOOK1:=TRUE ELSE LOOK1:=FALSE;

```

```

00354000 T 0037
00355000 T 0037
00356000 T 0037
00357000 T 0037
00358000 T 0037
00359000 T 0037
00360000 T 0037
00361000 T 0037
00362000 T 0037
00363000 T 0037
00364000 T 0037
00364010 T 0037
00364020 T 0037
00364030 T 0037
00364040 T 0047
00364050 T 0063
00364060 T 0065

```

```

PROCESS:=UQ;
SAV:=REF(AT7,PROCESS);
RC2:=MAKE(TYPECODE,2);
RC3:=MAKE(TYPECODE,3);
RC4:=MAKE(TYPECODE,4);
RC5:=MAKE(TYPECODE,5);
RC6:=MAKE(TYPECODE,6);
RC7:=MAKE(TYPECODE,7);
RC8:=MAKE(TYPECODE,8);
RC10:=MAKE(TYPECODE,10);
EXIT;
INTERPRET:R1: X(PROCESS)
IF REF(AT2,PROCESS) NEQ ATOMRESUMED THEN GO TO INT1;
IF REF(AT1,PROCESS) LSS PRIORITY THEN
BEGIN STACK(RC1);T:=QQ(PROCESS);
INT1:ASGN(AT7,PROCESS,SAV);
PROCESS:=UQ;SAV:=REF(AT7,PROCESS); EXIT END;
IF TRACE THEN BEGIN ASGN(AT7,PROCESS,SAV);

```

```

00365000 T 0075
00366000 T 0077
00367000 T 0079
00368000 T 0080
00369000 T 0082
00370000 T 0084
00371000 T 0086
00372000 T 0087
00373000 T 0089
00374000 T 0091
00375000 T 0093
00376000 T 0100
00377000 T 0101
00378000 T 0103
00379000 T 0104
00380000 T 0108
00381000 T 0109
00382000 T 0119

```

```

DOTRACE(PROCESS) END;
STACK(RC1);
IF (OP:=REF(AT5,PROCESS))=TOKEN(1004) THEN GO TO INSTVAL;
IF OP=token(1005) THEN GO TO BNFINSTVAL;
IF OP=token(1017) THEN GO TO INSTVAL;
ASGN(AT1,(ENV:=PMAKE(TYPECONS,0)),PROCESS);
GO TO CALL;
INSTVAL: %(PROCESS)=ARG
ARG:=REF(AT3,PROCESS);
ASGN(AT3,PROCESS,CDR(ARG));
IF ARG=NULLLIST THEN GO TO RETURNVAL;
ARG:=CAR(ARG);
STACK(TOKEN(27));GO TO ARGVAL;R2;
ASGN(TOKEN(1000),REF(AT4,PROCESS),ARG);
EXIT;
ARGVAL: %(ARG,PROCESS)=ARG
IF ATOM(ARG) THEN ARG:=REF(ARG,REF(AT4,PROCESS))ELSE
BEGIN STACK(CDR(ARG));
OP:=CAR(ARG);
STACK(RC2);GO TO OPVAL;R2;
ARGLIST:=UNSTACK;
GO TO APPLY END; EXIT;
APPLY: %(OP,ARGLIST,PROCESS)=ARG
IF ROUTINE,TYPE=TYPECODE THEN GO TO PRIMVAL;
FPARA:=REF(AT1,ROUTINE);
ENV:=PMAKE(TYPECONS,0);
STACK(ROUTINE);
STACK(RC3);GO TO PARAPASS; R3;
ROUTINE:=UNSTACK;
%GO TO CALL;
CALL: %(OP,ENV,PROCESS)=ARG
ASGN(AT7,PROCESS,SAV);
PROCESS:=CREATE(REF(AT2,ROUTINE),ENV,REF(AT3,ROUTINE),PROCESS);
R39:ASGN(AT2,PROCESS,ATOMRESUMED);
GO TO INTERPRET;
R9:IF REF(AT5,PROCESS)=CODE(1005) THEN %BNFINT
BEGIN VAL:=UNSTACK;GO TO BNFRETURNVAL END;
ARG:=UNSTACK;
EXIT;
OPVAL: %(OP,PROCESS)=ARG
IF ATOM(OP) THEN GO TO OP1;
BEGIN
ARG:=OP;
STACK(RC4);GO TO ARGVAL;R4;
OP:=ARG END;
OP1:EXIT;
PARAPASS: %(FPARA,ENV,ARGLIST,PROCESS)=ENV
IF FPARA NEQ NULLLIST THEN IF ARGLIST NEQ NULLLIST THEN
IF CAR(FPARA)=ATOMUNDEF THEN
BEGIN STACK(ENV);STACK(CAR(CDR(FPARA)));
STACK(RC5);GO TO LISTVAL;R5;
ASGN(UNSTACK,UNSTACK,ARG)END ELSE
BEGIN STACK(FPARA);STACK(ENV);STACK(CDR(ARGLIST));
ARG:=CAR(ARGLIST);
STACK(RC6);GO TO ARGVAL;R6;
ARGLIST:=UNSTACK;ENV:=UNSTACK;FPARA:=UNSTACK;
ASGN(CAR(FPARA),ENV,ARG);

```

```

00383000 T 0121
00384000 T 0122
00385000 T 0124
00386000 T 0127
00387000 T 0128
00388000 T 0130
00389000 T 0133
00390000 T 0135
00391000 T 0135
00392000 T 0136
00393000 T 0138
00394000 T 0139
00395000 T 0140
00396000 T 0144
00397000 T 0146
00398000 T 0154
00399000 T 0154
00400000 T 0160
00401000 T 0164
00402000 T 0165
00403000 T 0168
00404000 T 0169
00405000 T 0177
00406000 T 0178
00407000 T 0179
00408000 T 0181
00409000 T 0182
00410000 T 0184
00411000 T 0188
00412000 T 0189
00413000 T 0189
00413010 T 0190
00414000 T 0191
00415000 T 0194
00416000 T 0196
00417000 T 0196
00418000 T 0199
00419000 T 0201
00420000 T 0203
00421000 T 0211
00422000 T 0211
00423000 T 0215
00424000 T 0215
00425000 T 0216
00426000 T 0220
00427000 T 0220
00428000 T 0228
00429000 T 0229
00430000 T 0231
00431000 T 0232
00432000 T 0237
00433000 T 0241
00434000 T 0244
00435000 T 0249
00436000 T 0251
00437000 T 0253
00438000 T 0258

```

```

FPARA:=CDR(FPARA);GO TO PARAPASS;END;
EXIT;
LISTVAL: %(ARGLIST,PROCESS)=ARG
IF ARGLIST NEQ NULLLIST THEN
BEGIN STACK(CDR(ARGLIST));ARG:=CAR(ARGLIST);
STACK(RC7);GO TO ARGVAL;R7;
ARGLIST:=UNSTACK;STACK(ARG);
STACK(RC8);GO TO LISTVAL;R8;
ARG:=CONS(UNSTACK,ARG);
END ELSE ARG:=ARGLIST; EXIT;
RETURNVAL: %(PROCESS)
IF(OP:=REF(AT6,PROCESS))NEQ ATOMUNDEF THEN
BEGIN ARG:=REF(TOKEN(1000),REF(AT4,PROCESS));
RETURN(OP,ARG)END;
DELETE(PROCESS);GO TO INT1 ;
BNFINSTVAL: %(PROCESS)
ASGN(AT3,PROCESS,CDR((INST:=REF(AT3,PROCESS)))));
IF INST=NULLLIST THEN GO TO BNFXITVAL ELSE
IF ATOM((ARG:=CAR(INST))) THEN GO TO BNFTERMVAL ELSE
GO TO BNFNONTERMVAL;

BNFXITVAL: %(PROCESS)
SUSPEND(PROCESS); RETURN(COPY(REF(AT6,PROCESS)),PROCESS);
ASGN(AT5,PROCESS,SCGINT); EXIT;

BNFTERMVAL: %(ARG,PROCESS)
X:=CAR((STRING:=REF(STR,(ENV:=REF(AT4,PROCESS))))));
IF ARG=SLEXUNIT THEN GO TO BNFL1;
IF ARG=SID THEN GO TO (IF ID(X) THEN BNFL1 ELSE BNFL2);
IF ARG=SINT THEN GO TO (IF INT(X) THEN BNFL1 ELSE BNFL2);
IF ARG=SSPCHAR THEN GO TO (IF SPCHAR(X) THEN BNFL1 ELSE BNFL2);
IF ARG=SNBR THEN GO TO(IF NBR(X) THEN BNFL1 ELSE BNFL2);
IF ARG=(OP:=CODE(117)) THEN GO TO PRIMVAL;
IF ARG=CODE(1012) THEN GO TO BNFXITVAL;
IF X=ARG THEN
ASGN(STR,ENV,TAIL(STRING)) ELSE GO TO BNFL2;
EXIT;
BNFL1: INST:=CONS(SQUOTE,CONS(X,NULLLIST));
X:=CREATE(CONS(INST,NULLLIST),PMAKE(TYPECONS,0),SCGINT,PROCESS);
ASGN(STR,ENV,TAIL(STRING));
ASGN(TREE,ENV,CONS(X,REF(TREE,ENV)));
ASGN(N,ENV,REF(N,ENV)+1);EXIT;
BNFL2:DELETE(PROCESS); EXIT;

BNFRETURNVAL: %(PROCESS,VAL)
ASGN(AT4,PROCESS,ENV:=COPY(REF(AT4,PROCESS)));
ASGN(TREE,ENV,CONS(VAL,REF(TREE,ENV)));
ASGN(STR,ENV,REF(STR,REF(AT4,VAL)));
ASGN(N,ENV,REF(N,ENV)+1); EXIT;

BNFNONTERMVAL: %(ARG,PROCESS)
IF CAR(ARG) NEQ CODE(118) THEN PRIMERROR("BNF ",ARG);
X:=NULLLIST;ENV:=REF(AT4,PROCESS);
ARGLIST:=CDR(ARG);
WHILE ARGLIST NEQ NULLLIST DO
IF LOOK1(CAR(CAR(ARGLIST)),CAR(REF(STR,ENV))) THEN
BEGIN ENV:=COPY(ENV);

```

```

00439000 T 0260
00440000 T 0261
00441000 T 0269
00442000 T 0270
00443000 T 0270
00444000 T 0274
00445000 T 0278
00446000 T 0281
00447000 T 0284
00448000 T 0286
00449000 T 0295
00450000 T 0296
00451000 T 0298
00452000 T 0301
00453000 T 0302
00454000 T 0305
00455000 T 0305
00456000 T 0308
00457000 T 0308
00458000 T 0314
00459000 T 0315
00460000 T 0315
00461000 T 0315
00462000 T 0323
00463000 T 0332
00464000 T 0332
00465000 T 0333
00466000 T 0337
00467000 T 0339
00468000 T 0349
00469000 T 0356
00469010 T 0366
00470000 T 0380
00471000 T 0382
00472000 T 0384
00473000 T 0385
00474000 T 0387
00475000 T 0395
00476000 T 0398
00477000 T 0402
00478000 T 0405
00479000 T 0408
00480000 T 0419
00481000 T 0428
00482000 T 0428
00484000 T 0429
00485000 T 0432
00486000 T 0439
00487000 T 0443
00488000 T 0454
00489000 T 0454
00490000 T 0455
00491000 T 0458
00492000 T 0460
00493000 T 0461
00493010 T 0463
00494000 T 0467

```

```

ASGN(TREE,ENV, NULLLIST);
ASGN(N,ENV,ATO);
X:=CONS(CREATE(CAR(ARGLIST),ENV, CODE(1005),PROCESS),X);
ARGLIST:=CDR(ARGLIST) END ELSE ARGLIST:=CDR(ARGLIST);
GO TO WAITRETURN;
WAITRETURN: X(X)
WHILE X NEQ NULLLIST DO
  BEGIN RESUME(CAR(X)); X:=CDR(X) END;
GO TO INT1;
PRIMVAL: X(OP,ARGLIST,PROCESS)=ARG
CASE ROUTINE,ADDR=100 OF BEGIN
  IF ATOM(CAR(ARGLIST)) THEN X ASGN=CODE 100
  BEGIN STACK(CAR(ARGLIST)); ARGLIST:=CDR(ARGLIST);
  LV(41);R41:ASGN(UNSTACK,GET1,ARG:=GET) END ELSE
  BEGIN LV(11);R11:ASGN(GET1,GET,(ARG:=GET))END; XASGN=CODE100
  IF ATOM(CAR(ARGLIST)) THEN X REF=CODE 101
  BEGIN STACK(CAR(ARGLIST));ARGLIST:=CDR(ARGLIST);
  LV(42);R42:ARG:=REF(UNSTACK,GET1) END ELSE
  BEGIN LV(12);R12:ARG:=REF(GET1,GET)END; XREF=CODE101
  BEGIN LV(13);R13:ARG:=CAR(GET1)END; XCAR=CODE 102
  BEGIN LV(14);R14:ARG:=CDR(GET1)END; XCDR=CODE103
  BEGIN LV(15);R15:ARG:=CONS(GET1,GET)END; XCONS=CODE 104
  BEGIN LV(16);R16:ARG:=NEWSYMB(GET1)END;
  ARG:=AMTSPACE) XAMTSPACE
  BEGIN LV(17);R17:ARG:=PMAKE(GET1,TYPE,GET)END;
  BEGIN LV(18);R18:PRINT((ARG:=GET1))END;
  BEGIN LV(19);R19:ARG:=INLEX(GET1)END;
  BEGIN SUSPEND(PROCESS);RESUME(PROCESS);GCL;
  PROCESS:=UQ;ARG:=AMTSPACE END; X GCL
  BEGIN LV(20);R20:ARG:=CREATE(GET1,GET,GET,GET)END;
  BEGIN LV(21);R21: DELETE(GET1);ARG:=ATOMUNDEF END;%DELETE
  BEGIN LV(22);R22: ARG:=GET1;SUSPEND(ARG)END;
  BEGIN LV(23);R23:RESUME((ARG:=GET1))END;
  BEGIN LV(24);R24:RETURN(GET1,(ARG:=GET))END; XRETURN=CODE115
  BEGIN LV(25);R25:ARG:=GET1 END; XIDENTITY=CODE116
  BEGIN SUSPEND(PROCESS);SMEM("RERUN1"); GO TO LEND END;%HALT = CODE 11
  PRIMERRR("C118",PROCESS);XRESERVED FOR CODE118
  PRIMERRR("C119",PROCESS);XRESERVED FOR CODE119
  BEGIN LV(33);R33:ASGN(TOKEN(1009),REF(AT4,PROCESS),
  (ARG:=GET))END; X TRACE=CODE120
  BEGIN LV(26);R26:EQLIST(GET1,(ARG:=GET))END; XEQLIST
  ARG:=CAR(ARGLIST); XQUOTE=CODE122
  IF ATOM(CAR(ARGLIST)) THEN X ASGNENV=CODE 123
  BEGIN STACK(CAR(ARGLIST)); ARGLIST:=CDR(ARGLIST);
  LV(43);R43:ASGN(UNSTACK,REF(AT4,PROCESS),(ARG:=GET1)) END ELSE
  BEGIN LV(28);R28:ASGN(GET1,REF(AT4,PROCESS),(ARG:=GET))END;%ASGNENV=CODE
  IF ATOM(CAR(ARGLIST)) THEN XREFENV=CODE 124
  BEGIN STACK(CAR(ARGLIST)); ARGLIST:=CDR(ARGLIST);
  LV(44);R44:ARG:=REF(UNSTACK,REF(AT4,PROCESS)) END ELSE
  BEGIN LV(29);R29:ARG:=REF(GET1,REF(AT4,PROCESS))END; XREFENV=CODE124
  BEGIN LV(30);R30:ARG:=GET1;GO TO ARGVAL END; XEXE=CODE125
  DEBUG; XDEBUG=CODE126
  BEGIN LV(49);R49:ARG:=LOADMODE(GET1,GET) END; XLDMODE=127
  ARG:=PMAKE(TYPECONS,0); XCONSTRUCT=CODE128
  BEGIN LV(31);R31:ARG:=PMAKE(TYPEAREA,GET1)END;
  XAREA = CODE129
  BEGIN LV(48);R48:ARG:=TOKEN(GET1,ADDR) END;%TOKEN=CODE 130

```

```

00495000 T 0469
00496000 T 0471
00497000 T 0472
00498000 T 0476
00499000 T 0482
00500000 T 0483
00501000 T 0484
00502000 T 0485
00503000 T 0488
00504000 T 0488
00505000 T 0489
00506000 T 0490
00507000 T 0495
00508000 T 0498
00509000 T 0506
00510000 T 0516
00511000 T 0521
00512000 T 0524
00513000 T 0531
00514000 T 0539
00515000 T 0544
00516000 T 0550
00517000 T 0558
00518000 T 0563
00519000 T 0567
00520000 T 0574
00521000 T 0580
00522000 T 0592
00522010 T 0600
00523000 T 0605
00524000 T 0616
00525000 T 0623
00526000 T 0633
00527000 T 0639
00528000 T 0647
00529000 T 0651
00530000 T 0662
00531000 T 0664
00532000 T 0665
00533000 T 0673
00534000 T 0676
00535000 T 0683
00536000 T 0684
00537000 T 0689
00538000 T 0692
00539000 T 0700
00540000 T 0709
00541000 T 0713
00542000 T 0717
00543000 T 0723
00544000 T 0730
00545000 T 0736
00546000 T 0737
00547000 T 0753
00548000 T 0755
00549000 T 0760
00550000 T 0760

```

FACILE BUSINESS FORMS, INC. NY

```

BEGIN LV(32);R32:ASGN(TOKEN(1008),REF(AT4,PROCESS),
(ARG:=GET1))END; % SINGLESTEP=CODE131
BEGIN ASGN(AT1,PROCESS,REF(AT1,PROCESS)+1);
ARG:=ATOMUNDEF END; %UP=CODE132
BEGIN ASGN(AT1,PROCESS,REF(AT1,PROCESS)-1);
ARG:=ATOMUNDEF END; %DP=CODE133
;%BEGIN LV(34);R34:UREAD((ARG:=GET1))END; %READ=CODE134
;% RESERVED FOR CODE 135
GO TO LISTVAL; % LIST=CODE 136
BEGIN STACK(CDR(ARGLIST)); ARG:=CAR(ARGLIST); %COND=CODE 137
STACK(TOKEN(36));GO TO ARGVAL;R36:ARGLIST:=UNSTACK;
IF ARG.TYPE=TYPELOGIC THEN
BEGIN ARG:=(IF BOOLEAN(ARG) THEN CAR(ARGLIST) ELSE REF(AT2,ARGLIST));
GO TO ARGVAL END ELSE ARG:=ATOMUNDEF END;
BEGIN LV(37);R37:ARG:=(IF EQ(GET1,GET) THEN AT(524287)
ELSE ATO) END; % EQ=CODE 138
BEGIN LV(38);R38:ARG:=GET1; % ATOM=CODE 139
ARG:=(IF ATOM(ARG)THEN AT(524287) ELSE ATO) END;
BEGIN PROCESS:=CREATE(ARGLIST,REF(AT4,PROCESS),REF(AT5,
PROCESS),PROCESS);
GO TO R39 END ; % BLOCKOP=CODE 140
BEGIN LV(40);R40:ARG:=(IF EQ(GET1,GET) THEN ATO
ELSE AT(524287)) END; % NEG=CODE 141
BEGIN STACK(CAR(CDR(ARGLIST))); ARG:=CAR(ARGLIST);%IF=CODE 142
STACK(TOKEN(45));GO TO ARGVAL;R45:ARGLIST:=UNSTACK;
IF BOOLEAN(ARG) THEN ASGN(AT3,PROCESS,ARGLIST) END;
BEGIN ENV:=REF(AT4,PROCESS);%NONTERM=CODE143
ASGN(STR,ENV,ATO);
BNFL3:IF (ARG:=REF(N,ENV)).ADDR GTR 0 THEN
BEGIN ASGN(N,ENV,ARG-1);ASGN(STR,ENV,REF(STR,ENV)+1);
ASGN(AT6,ARG:=REF(ARG,REF(TREE,ENV)),PROCESS);
STACK(CODE(46));RESUME(ARG);
GO TO INT1;R46:
ENV:=REF(AT4,PROCESS);
ASGN(REF(STR,ENV),ENV,ARG);GO TO BNFL3 END;
ARG:=REF(AT1,ENV);
ASGN(TREE,ENV,ATOMUNDEF);
ASGN(STR,ENV,ATOMUNDEF);
ASGN(N,ENV,ATOMUNDEF) END NONTERM;
BEGIN LV(47);R47:ARG:=USERL(GET1) END;%USERL=CODE 144
ARG:=LABTAB; %LTAB=CODE 145
BEGIN LV(50);R50:ARG:=COPY(GET1) END; %COPY=CODE 146
BEGIN LV(51);R51:ARG:=AR(0,GET1,GET) END; %ADD=CODE 147
BEGIN LV(52);R52:ARG:=AR(1,GET1,GET) END; %SUB=CODE 148
BEGIN LV(53);R53:ARG:=AR(2,GET1,GET) END; %MUL=CODE 149
BEGIN LV(54);R54:ARG:=AR(3,GET1,GET) END; %DIV=CODE 150
BEGIN LV(55);R55:ARG:=AR(4,GET1,GET) END; %EX =CODE 151
BEGIN LV(56);R56:ARG:=AR(2,GET1,AD(-1)) END; %NEG=CODE 152
BEGIN LV(57);R57:ARG:=IF LOOK1(GET1,GET) THEN AT(524287)
ELSE ATO END % LOOK1=CODE 153
END CASE STATEMENT;

```

```

00551000 T 0766
00552000 T 0772
00553000 T 0774
00554000 T 0776
00555000 T 0777
00556000 T 0780
00557000 T 0781
00558000 T 0781
00559000 T 0781
00560000 T 0782
00561000 T 0785
00562000 T 0789
00563000 T 0791
00564000 T 0795
00565000 T 0797
00566000 T 0804
00567000 T 0807
00568000 T 0812
00569000 T 0819
00570000 T 0821
00571000 T 0822
00572000 T 0824
00573000 T 0831
00574000 T 0834
00575000 T 0837
00576000 T 0843
00577000 T 0846
00577050 T 0847
00578000 T 0849
00579000 T 0853
00579050 T 0859
00580000 T 0863
00581000 T 0865
00582000 T 0867
00583000 T 0868
00583050 T 0871
00584000 T 0873
00585000 T 0874
00586000 T 0876
00586010 T 0878
00586020 T 0887
00586030 T 0888
00586035 T 0894
00586040 T 0902
00586045 T 0910
00586050 T 0918
00586055 T 0926
00586060 T 0934
00586070 T 0941
00586080 T 0948
00587000 T 0951

```

```

START OF SEGMENT ***** 10
10 IS 54 LONG, NEXT SEG 9

```

```

EXIT;
L1:ASGN(AT7,PROCESS,SAV);DOTRACE(PROCESS);
R10:R35:R34; % THESE ARE TO BE LATER MOVED
LEND:XX1:=PROCESS; END INTERP;

```

```

00588000 T 0951
00589000 T 0958
00590000 T 0961
00591000 T 0961

```

```

PROCEDURE GARBAGECOLLECT(QMEM,N);
VALUE N;ARRAY QMEM[0];REAL N;
BEGIN REAL RP,WP,X,XADDR,BASE,TYP,I;

LABEL L1,L2,L3,L4,L5,L6;
ARRAY TMEM[0:MEMORYROWS,0:511];
DEFINE CR(CP1,CP2)=%CONSTRUCT PROCESSING
BEGIN IF (X:=M[XADDR]).TYPE NEQ TYPEGARBAGE THEN
BEGIN TMSGN((WP:=WP+1),X);
MSGN(XADDR,MAKE(TYPEGARBAGE,WP));
CP1(RP,MAKE(XADDR,TYPE,WP),CP2)END ELSE
CP1(RP,MAKE(XADDR,TYPE,X),CP2)END#;
DEFINE LP(LP1,LP2)=
BEGIN IF (X:=CAR(XADDR)).TYPE = TYPEGARBAGE THEN
LP1(RP,MAKE(TYPELIST,X),LP2)ELSE
IF CARADDR=NULLLIST THEN LP1(RP,NULLLIST,LP2)ELSE
BEGIN LP1(RP,MAKE(TYPELIST,(WP:=WP+1)),LP2)#;
LLP(LLP1)=TMSGN(WP,X);
MSGN(CARADDR,MAKE(TYPEGARBAGE,WP));
XADDR:=CDR(XADDR);
IF (X:=CAR(XADDR)).TYPE NEQ TYPEGARBAGE THEN
IF CARADDR NEQ NULLLIST THEN
BEGIN WP:=WP+1;GO TO LLP1 END ELSE
TMSGN((WP:=WP+1),LINKTONIL)ELSE
TMSGN((WP:=WP+1),MAKE(TYPELINK,X))END END #;
DEFINE TM(TM1)=(IF BOOLEAN(TM1) THEN
TMEM[(TM1),[18:9],[TM1],[9:9]],[22:23]ELSE
TMEM[(TM1),[18:9],[TM1],[9:9]],[45:23])#;
Q[Q1]=(IF BOOLEAN(Q1) THEN QMEM[Q1],[10:10],[22:23] ELSE
QMEM[Q1],[10:10],[45:23])#;
TMSGN(TMSGN1,TMSGN2)=TA(TMSGN1,TMSGN2,TMEM)#;
QASGN(QASGN1,QASGN2)=QA(QASGN1,QASGN2,QMEM)#;
PROCEDURE TA(A,V,M);VALUE A,V;REAL A,V;ARRAY M[0,0];
IF BOOLEAN(A) THEN M[A,[18:9],A,[9:9]],[22:23]:=V ELSE
M[A,[18:9],A,[9:9]],[45:23]:=V;

```

```

PROCEDURE QA(A,V,M);VALUE A,V;REAL A,V;ARRAY M[0];
IF BOOLEAN(A) THEN M[A,[10:10]],[22:23]:=V ELSE
M[A,[10:10]],[45:23]:=V;

```

```

FORMAT F1("GCL:",I6," WORDS RECOVERED.",I6," WORDS IN USE.");

```

```

WP:=0;
% COPY FROM M TO TM UPSIDE DOWN USING Q
FOR RP:=N+N+1 STEP -1 UNTIL 0 DO

```

00592000	T	0025	
00593000	T	0025	
00594000	T	0025	
START OF SEGMENT	*****	11	
00595000	T	0000	
00596000	T	0000	
00597000	T	0002	
00598000	T	0002	
00599000	T	0002	
00600000	T	0002	
00601000	T	0002	
00602000	T	0002	
00603000	T	0002	
00604000	T	0002	
00605000	T	0002	
00606000	T	0002	
00607000	T	0002	
00608000	T	0002	
00609000	T	0002	
00610000	T	0002	
00611000	T	0002	
00612000	T	0002	
00613000	T	0002	
00614000	T	0002	
00615000	T	0002	
00616000	T	0002	
00617000	T	0002	
00618000	T	0002	
00619000	T	0002	
00620000	T	0002	
00621000	T	0002	
00622000	T	0002	
00623000	T	0002	
00624000	T	0002	
00625000	T	0008	
00626000	T	0013	
00627000	T	0013	
00628000	T	0016	
00629000	T	0020	
START OF SEGMENT	*****	12	
12 IS	12 LONG,	NEXT SEG	11
00630000	T	0020	
00631000	T	0021	
00632000	T	0021	

```

CASE (XADDR:=Q[RP]),TYPE OF
BEGIN PRIMERROR("GCL1",XADDR);%GARBAGE
L1:CP(QA,QMEM);%CONSTRUCT
LP(QA,QMEM);L2:LLP(L2);%LIST
L5:IF (X:=M[XADDR]),TYPE=TYPEGARBAGE THEN %AREA
QASGN(RP,MAKE(XADDR,TYPE,X))ELSE
BEGIN WP:=WP+1;
FOR I:=X.ADDR STEP -1 UNTIL 0 DO
TMASGN(WP+I,M[XADDR=I]);
QASGN(RP,MAKE(XADDR,TYPE,WP));
MASGN(XADDR,MAKE(TYPEGARBAGE,WP));
WP:=WP+X.ADDR END;
PRIMERROR("GCL2",XADDR);%FIELD
IF XADDR NEQ ATOMUNDEF THEN GO TO L1;%SYMBOL
;%LOGIC
PRIMERROR("GCL3",XADDR);%LINK
GO TO L5;%PROCESS
GO TO L5;%MONITOR
GO TO L5;%GENERAL
;%TOKEN
PRIMERROR("GCL4",XADDR);%TTP12
PRIMERROR("GCL5",XADDR);%TTP13
PRIMERROR("GCL6",XADDR);%TTP14
PRIMERROR("GCL7",XADDR);%TTP15
END CASE STATEMENT;

```

```

%COPY FROM M TO TM UPSIDE DOWN USING TM
FOR RP:=1 STEP 1 UNTIL WP DO
CASE (XADDR:=TM[RP]),TYPE OF BEGIN
PRIMERROR("GCL8",XADDR);%GARBAGE
L3:CP(TA,TMEM);%CONSTRUCT
LP(TA,TMEM);L4:LLP(L4);%LIST
L6:IF (X:=M[XADDR]),TYPE=TYPEGARBAGE THEN %AREA
TMASGN(RP,MAKE(XADDR,TYPE,X))ELSE
BEGIN WP:=WP+1;
FOR I:=X.ADDR STEP -1 UNTIL 0 DO
TMASGN(WP+I,M[XADDR=I]);
TMASGN(RP,MAKE(XADDR,TYPE,WP));
MASGN(XADDR,MAKE(TYPEGARBAGE,WP));
WP:=WP+X.ADDR END;
PRIMERROR("GCL9",XADDR);%FIELD
IF XADDR NEQ ATOMUNDEF THEN GO TO L3;%SYMBOL
;%LOGIC
;%LINK
GO TO L6;%PROCESS
GO TO L6;%MONITOR
GO TO L6;%GENERAL
;%TOKEN
END CASE STATEMENT;

```

```

% COPY FROM TM TO M AND INVERT
BASE:=WP+1;
LASTUSEDSPACE:=0;
FOR RP:=WP STEP -1 UNTIL 1 DO
BEGIN X:=TM[RP];

```

```

00633000 T 0026
00634000 T 0031
00635000 T 0033
00636000 T 0058
00637000 T 0094
00638000 T 0102
00639000 T 0106
00640000 T 0108
00641000 T 0112
00642000 T 0125
00643000 T 0128
00644000 T 0130
00645000 T 0133
00646000 T 0134
00647000 T 0136
00648000 T 0136
00649000 T 0137
00650000 T 0141
00651000 T 0141
00652000 T 0142
00653000 T 0142
00654000 T 0143
00655000 T 0145
00656000 T 0146
00657000 T 0148

```

```

START OF SEGMENT ***** 13
13 IS 16 LONG, NEXT SEG 11

```

```

00658000 T 0152
00659000 T 0152
00660000 T 0153
00661000 T 0161
00662000 T 0162
00663000 T 0187
00664000 T 0224
00665000 T 0232
00666000 T 0236
00667000 T 0238
00668000 T 0242
00669000 T 0255
00670000 T 0259
00671000 T 0261
00672000 T 0263
00673000 T 0265
00674000 T 0266
00675000 T 0266
00676000 T 0266
00677000 T 0269
00678000 T 0269
00679000 T 0270
00680000 T 0270

```

```

START OF SEGMENT ***** 14
14 IS 12 LONG, NEXT SEG 11

```

```

00681000 T 0272
00682000 T 0272
00683000 T 0273
00684000 T 0274
00685000 T 0275

```

```

IF (TYP:=X.TYPE) NEQ TYPELOGIC THEN
IF TYP NEQ TYPECODE THEN
IF X.ADDR NEQ 524287 THEN
X:=MAKE(TYP,BASE=X.ADDR);
MASGN(STEPSPACE,X) END;
% FIX ADDRESSES IN Q
FOR RP:=N+N+1 STEP -1 UNTIL 0 DO
IF (TYP:=Q[RP].TYPE) NEQ TYPELOGIC THEN
IF TYP NEQ TYPECODE THEN
IF Q[RP].ADDR NEQ 524287 THEN
QASGN(RP,MAKE(TYP,BASE=Q[RP].ADDR));
WRITE(DATA,F1,AMTSPACE,LASTUSEDSPACE);
END GARBAGE COLLECT;

```

```

00686000 T 0282
00687000 T 0284
00688000 T 0285
00689000 T 0287
00690000 T 0290
00691000 T 0294
00692000 T 0294
00693000 T 0299
00694000 T 0305
00695000 T 0306
00696000 T 0312
00697000 T 0323
00698000 T 0335

```

11 IS 343 LONG, NEXT SEG 2

```

REAL PROCEDURE COPY(X);VALUE X;REAL X;
BEGIN LABEL L1,L2;REAL I;

CASE X,TYPE OF BEGIN
L1;PRIMERROR("COPY",X); %GARBAGE
MASGN((COPY:=MAKE(X,TYPE,0)),COPY(M[X]));%CONSTRUCT
COPY:=(IF X=NULLIST THEN NULLIST ELSE CONS(CAR(X),COPY(CDR
(X)))); %LIST
L2;BEGIN FOR I:=M[X],ADDR STEP -1 UNTIL 0 DO %AREA
MASGN(STEPSPACE, M[X=I]);
COPY:=MAKE(X,TYPE,LASTUSEDSPACE)END;
GO TO L1; %FIELD
COPY:=X; %SYMBOL
COPY:=X; %LOGIC
GO TO L1; %LINK
GO TO L2; %PROCESS
GO TO L2; %MONITOR
GO TO L2; %GENERAL
COPY:=X; %TOKEN
END CASE STATEMENT;

```

```

00699000 T 0025
00700000 T 0025
START OF SEGMENT ***** 15
00701000 T 0000
00702000 T 0001
00703000 T 0003
00704000 T 0013
00705000 T 0017
00706000 T 0019
00707000 T 0030
00708000 T 0044
00709000 T 0046
00710000 T 0047
00711000 T 0048
00712000 T 0049
00713000 T 0050
00714000 T 0050
00715000 T 0051
00716000 T 0051
00717000 T 0053

```

START OF SEGMENT ***** 16
16 IS 12 LONG, NEXT SEG 15
00718000 T 0052
15 IS 55 LONG, NEXT SEG 2

END COPY;

```

BOOLEAN PROCEDURE EQ(X,Y);VALUE X,Y; REAL X,Y;
IF X=Y THEN EQ:=TRUE ELSE
IF X,TYPE NEQ Y,TYPE THEN EQ:= FALSE ELSE
IF X,TYPE NEQ TYPELIST THEN EQ := FALSE ELSE
BEGIN X:=CAR(X); X:=CARADDR;
Y:=CAR(Y);EQ:=X=CARADDR
END OF EQ;

```

```

00719000 T 0025
00720000 T 0025
00721000 T 0028
00722000 T 0031
00723000 T 0034
00724000 T 0037
00725000 T 0038

```

REAL PROCEDURE CAR(Y);VALUE Y;REAL Y;

00726000 T 0041

BEGIN LABEL L1,L2;

IF Y.TYPE NEQ TYPELIST THEN
IF Y.TYPE NEQ TYPELINK THEN PRIMERROR("CAR ",Y);
IF Y=NULLLIST THEN GO TO L2;
L1:CARADDR:=Y;
IF (Y:=M[Y]).TYPE=TYPELINK THEN
IF Y NEQ LINKTONIL THEN GO TO L1 ELSE
L2:BEGIN CAR:=ATOMUNDEF;
CARADDR:=NULLLIST END ELSE
CAR:=Y END CAR;

00727000 T 0041
START OF SEGMENT ***** 17
00728000 T 0000
00729000 T 0001
00730000 T 0004
00731000 T 0005
00732000 T 0006
00733000 T 0015
00734000 T 0016
00735000 T 0017
00736000 T 0018
17 IS 24 LONG, NEXT SEG 2

REAL PROCEDURE CDR(Y);VALUE Y;REAL Y;
BEGIN Y:=CAR(Y);
IF CARADDR NEQ NULLLIST THEN Y:=CAR(MAKE(TYPELIST,DEC(CARADDR)));
CDR:=MAKE(TYPELIST,CARADDR);
END CDR;

00737000 T 0041
00738000 T 0041
00739000 T 0043
00740000 T 0047
00741000 T 0049

PROCEDURE EQLIST(X,Y);VALUE X,Y;REAL X,Y;
BEGIN REAL T1,T2;LABEL LEND;

T1:=CAR(Y);T1:=CARADDR;T2:=CAR(X);
IF T1=CARADDR THEN GO TO LEND;
IF CARADDR=NULLLIST THEN PRIMERROR("EQL2",X);
MASGN(X,MAKE(TYPELINK,T1));
LEND:END EQLIST;

00742000 T 0051
00743000 T 0051
START OF SEGMENT ***** 18
00744000 T 0000
00745000 T 0003
00746000 T 0004
00747000 T 0006
00748000 T 0008
18 IS 13 LONG, NEXT SEG 2

REAL PROCEDURE CONS(X,Y);VALUE X,Y;REAL X,Y;
BEGIN IF Y.TYPE NEQ TYPELIST THEN PRIMERROR("CONS",Y);
IF Y.ADDR NEQ LASTUSEDSPACE THEN MASGN(STEPSPACE,MAKE(TYPELINK,Y));
MASGN(STEPSPACE,X);
CONS:=MAKE(TYPELIST,LASTUSEDSPACE)END CONS;

00749000 T 0051
00750000 T 0051
00751000 T 0054
00752000 T 0059
00753000 T 0061

REAL PROCEDURE REF(X,Y);VALUE X,Y;REAL X,Y;
BEGIN LABEL L1,L2,L3,L4;REAL T1,T2;

CASE Y.TYPE OF
BEGIN PRIMERROR("REF1",Y);%GARBAGE
L1:BEGIN T1:=M[Y];%CONSTRUCT
L2:IF CAR(T1) NEQ ATOMUNDEF THEN
IF CAR(T1) NEQ X THEN
BEGIN T1:=CDR(T1)-1;GO TO L2 END ELSE
REF:=CAR(CDR(T1))ELSE

00754000 T 0066
00755000 T 0066
START OF SEGMENT ***** 19
00756000 T 0000
00757000 T 0000
00758000 T 0002
00759000 T 0010
00760000 T 0012
00761000 T 0014
00762000 T 0018

```

REF:=ATOMUNDEF END;
IF X=ATOMCAR THEN REF:=CAR(Y) ELSE XLIST
IF X=ATOMCDR THEN REF:=CDR(Y) ELSE
IF X.TYPE NEQ TYPELOGIC THEN REF:=ATOMUNDEF ELSE
BEGIN FOR X:=X.ADDR STEP -1 UNTIL 2 DO Y:=CDR(Y);
REF:=CAR(Y) END;
L3:BEGIN IF X.TYPE=TYPELOGIC THEN %AREA
IF X.ADDR LEQ M(Y).ADDR THEN
BEGIN REF:=M(Y-X.ADDR);GO TO L4 END;
REF:=ATOMUNDEF;
L4:END;
PRIMERROR("REF4",Y);%FIELD
IF X=ATOMASSO THEN %SYMBOL
REF:=(IF Y=ATOMUNDEF THEN Y ELSE M(Y))ELSE
PRIMERROR("REF5",Y);
PRIMERROR("REF6",Y);%LOGIC
PRIMERROR("REF7",Y);%LINK
GO TO L3; %PROCESS
GO TO L3;%MONITOR
PRIMERROR("REF8",Y);%GENERAL
PRIMERROR("REF9",Y);%CODE
END CASE STATEMENT;

END REF;

```

```

00763000 T 0020
00764000 T 0022
00765000 T 0024
00766000 T 0027
00767000 T 0030
00768000 T 0037
00769000 T 0038
00770000 T 0040
00771000 T 0049
00772000 T 0062
00773000 T 0063
00774000 T 0063
00775000 T 0065
00776000 T 0065
00777000 T 0075
00778000 T 0078
00779000 T 0080
00780000 T 0081
00781000 T 0085
00782000 T 0085
00783000 T 0087
00784000 T 0088

```

```

START OF SEGMENT ***** 20
20 IS 12 LONG, NEXT SEG 19
00785000 T 0088
19 IS 93 LONG, NEXT SEG 2

```

```

PROCEDURE ASGN(X,Y,Z);VALUE X,Y,Z;REAL X,Y,Z;
BEGIN LABEL L1,L2,L3;REAL T1,T2;

CASE Y.TYPE OF
BEGIN PRIMERROR("ASG1",Y);%GARBAGE
L1:BEGIN T1:=M(Y);%CONSTRUCT
IF X=ATOMUNDEF THEN PRIMERROR("ASGD",Y);
L2:IF CAR(T1) NEQ ATOMUNDEF THEN
IF CAR(T1) NEQ X THEN
BEGIN T1:=DEC(CDR(T1));GO TO L2 END ELSE
IF Z NEQ ATOMUNDEF THEN
MASGN(CDR(T1),Z) ELSE
IF T1=M(Y) THEN MASGN(Y,CDR(CDR(T1)))ELSE
MASGN(T1,MAKE(TYPELINK,CDR(CDR(T1))))ELSE
IF Z NEQ ATOMUNDEF THEN MASGN(Y,CONS(X,CONS(Z,M(Y))))END;
BEGIN T1:=CAR(Y);%XLIST
IF CARADDR=NULLLIST THEN PRIMERROR("ASGO",Y);
IF X=ATOMCAR THEN MASGN(CARADDR,Z)ELSE
IF X=ATOMCDR THEN
IF Z.TYPE NEQ TYPELIST THEN PRIMERROR("ASGC",Z)ELSE
MASGN(CARADDR,MAKE(TYPELINK,CONS(M(CARADDR),Z)))ELSE
IF X.TYPE NEQ TYPELOGIC THEN PRIMERROR("ASG2",X) ELSE
BEGIN T1:=CARADDR;
FOR X:=X.ADDR STEP -1 UNTIL 2 DO
BEGIN IF (Y:=CDR(Y))=NULLLIST THEN MASGN(T1-1,(Y:=MAKE(TYPELINK,
CONS(ATOMUNDEF,NULLLIST)))));
T1:=Y END;

```

```

00786000 T 0066
00787000 T 0066
START OF SEGMENT ***** 21
00788000 T 0000
00789000 T 0000
00790000 T 0002
00791000 T 0010
00792000 T 0012
00793000 T 0014
00794000 T 0016
00795000 T 0021
00796000 T 0022
00797000 T 0024
00798000 T 0034
00799000 T 0038
00800000 T 0049
00801000 T 0050
00802000 T 0052
00803000 T 0055
00804000 T 0057
00805000 T 0061
00806000 T 0072
00807000 T 0075
00808000 T 0077
00809000 T 0082
00810000 T 0085
00811000 T 0088

```

```

MASGN(Y,Z) END END;
L3:BEGIN IF X.TYPE NEQ TYPELOGIC THEN PRIMERROR("ASG3",X);%AREA
IF (X#X.ADDR) GTR M[Y].ADDR THEN PRIMERROR("ASGA",Y);
IF X=0 THEN
IF X.TYPE NEQ TYPELOGIC OR X GTR M[Y] THEN PRIMERROR("ASGB",Z);
MASGN(Y-X,Z)END;
PRIMERROR("ASG4",Y);%FIELD
IF X=ATOMASSO AND Y NEQ ATOMUNDEF THEN MASGN(Y,Z)ELSE%SYMBOL
PRIMERROR("ASG5",Y);
PRIMERROR("ASG6",Y);%LOGIC
PRIMERROR("ASG7",Y);%LINK
GO TO L3; %PROCESS
GO TO L3;%MONITOR
PRIMERROR("ASG8",Y);%GENERAL
PRIMERROR("ASG9",Y);%CODE
END CASE STATEMENT;

```

END ASGN;

```

00812000 T 0089
00813000 T 0090
00814000 T 0093
00815000 T 0104
00816000 T 0104
00817000 T 0115
00818000 T 0117
00819000 T 0119
00820000 T 0122
00821000 T 0128
00822000 T 0130
00823000 T 0131
00824000 T 0135
00825000 T 0135
00826000 T 0137
00827000 T 0138
START OF SEGMENT ***** 22
22 IS 12 LONG, NEXT SEG 21
00828000 T 0138
21 IS 143 LONG, NEXT SEG 2

```

```

REAL PROCEDURE PMAKE(T,N);VALUE T,N;REAL T,N;
BEGIN LABEL L1,L2,L3; REAL I;

CASE T OF BEGIN
L1:PRIMERROR("MK1 ",T); %GARBAGE
L2:BEGIN MASGN(STEPSPACE,NULLLIST); %CONSTRUCT
PMAKE:=MAKE(T, LASTUSEDSPACE)END;
PMAKE:= NULLLIST; %LIST
L3:IF N.TYPE NEQ TYPELOGIC THEN PRIMERROR("MK2 ",N)ELSE%AREA
BEGIN FOR I:=N.ADDR STEP =1 UNTIL 1 DO
MASGN(STEPSPACE,ATOMUNDEF);
MASGN(STEPSPACE,N);
PMAKE:=MAKE(T, LASTUSEDSPACE)END;
GO TO L1; %FIELD
PMAKE:=NEWSYMB(N); %SYMBOL
PMAKE:=ATO; %LOGIC
GO TO L1; %LINK
GO TO L3; %PROCESS
GO TO L3; %MONITOR
GO TO L3;%GENERAL
PMAKE:=NEXTTOKEN:=NEXTTOKEN+1; %TOKEN
END CASE STATEMENT;

```

END PMAKE;

```

00829000 T 0066
00830000 T 0066
START OF SEGMENT ***** 23
00831000 T 0000
00832000 T 0000
00833000 T 0002
00834000 T 0005
00835000 T 0007
00836000 T 0008
00837000 T 0011
00838000 T 0019
00839000 T 0021
00840000 T 0023
00841000 T 0026
00842000 T 0026
00843000 T 0028
00844000 T 0029
00845000 T 0030
00846000 T 0030
00847000 T 0031
00848000 T 0031
00849000 T 0033
START OF SEGMENT ***** 24
24 IS 12 LONG, NEXT SEG 23
00850000 T 0033
23 IS 36 LONG, NEXT SEG 2

```

```

REAL PROCEDURE NEWSYMB(ASSO);VALUE ASSO;REAL ASSO;
BEGIN MASGN(STEPSPACE,ASSO);
NEWSYMB:=MAKE(TYPESYMB, LASTUSEDSPACE)END NEWSYMB;

```

```

00851000 T 0066
00852000 T 0066
00853000 T 0069

```

```

REAL PROCEDURE CHAR(P,N);
POINTER P; INTEGER N;
BEGIN REAL ARRAY X[0:0];

X[0]:=0;
IF N LSS 8 AND N GTR 0 THEN
REPLACE POINTER(X)+(8-N) BY P FOR N;
CHAR:=X[0] END; % END OF CHAR

```

```

00854000 T 0073
00855000 T 0073
00856000 T 0073
START OF SEGMENT ***** 25
00857000 T 0001
00858000 T 0003
00859000 T 0004
00860000 T 0011
25 IS 17 LONG, NEXT SEG 2

```

```

REAL PROCEDURE HANGON (X,Y);
VALUE X,Y; REAL X,Y;
BEGIN LABEL L1,L2;

POINTER P;REAL ARRAY XD[0:0],X1[0:0];REAL Z
);
P:=POINTER(STACK[STACKPOSITION.[18:7],*])+STACKPOSITION.[11:12];
X1[0]:=1;Z:=NULLIST;
WHILE Y NEQ NULLIST DO
BEGIN Z:=CONS(CAR(Y),Z);Y:=CDR(Y)END;
WHILE Z NEQ NULLIST DO
BEGIN
XD[0]:=CAR(Z);
REPLACE P:=P+1 BY POINTER(XD)+7 FOR 1;
X1[0]:=X1[0]+1;
Z:=CDR(Z) END;
REPLACE POINTER(STACK[STACKPOSITION.[18:7],*])+
STACKPOSITION.[11:12] BY POINTER(X1)+7 FOR 1;
IF X1[0] EQL 2 THEN BEGIN
X1[0]:=LOADMODE(XD[0],X1[0]),ADDR;
IF X1[0] EQL 2 OR X1[0] EQL 66 THEN BEGIN
HANGON:=TOKEN(XD[0],ADDR+512);X1[0]:=LOADMODE(XD[0],X1[0]);
GO TO L2 END;
X1[0]:=LOADMODE(XD[0],X1[0]) END;
HANGON:=LEXFIND;
L2:END OF HANGON;

```

```

00861000 T 0073
00862000 T 0073
00863000 T 0073
START OF SEGMENT ***** 26
00864000 T 0000
00865000 T 0003
00866000 T 0003
00867000 T 0008
00868000 T 0010
00869000 T 0012
00870000 T 0016
00871000 T 0017
00872000 T 0017
00873000 T 0019
00874000 T 0027
00875000 T 0029
00876000 T 0030
00877000 T 0034
00878000 T 0040
00879000 T 0042
00880000 T 0047
00881000 T 0050
00882000 T 0057
00883000 T 0057
00884000 T 0062
00885000 T 0063
26 IS 68 LONG, NEXT SEG 2

```

```

REAL PROCEDURE IL(M,C,CM,F);
INTEGER M; REAL C,CM,F;
BEGIN % PROCEDURE BLOCK;
DEFINE N=72#, NCPW=8#;

DEFINE FILIN=DATA#,
LAST="S"#;
DEFINE TOTCHAR = 64#;
DEFINE X=LEXTREE#;

```

```

00886000 T 0073
00887000 T 0073
00888000 T 0073
00889000 T 0073
START OF SEGMENT ***** 27
00890000 T 0000
00891000 T 0000
00892000 T 0000
00893000 T 0000

```

```

BEGIN % OWN BLOCK;
OWN ALPHA ARRAY D[0:N DIV NCPW];

OWN REAL ARRAY MODE [0:TOTCHAR];
OWN INTEGER COUNT;
OWN POINTER P;
OWN REAL ARRAY SS[0:63];
CASE M OF BEGIN
BEGIN INTEGER I; % M IS 0;

I:=0;
WHILE I LEQ 9 DO
BEGIN MODE[I]:=1; I:=I+1 END;
WHILE I LEQ 63 DO
BEGIN MODE[I]:=0; I:=I+1 END;
MODE[" "] := 3;
MODE[I]:=68;
FILL SS[+] WITH ".,\"["(", "&", "§", "¶", ")",";","
"/", ":", "x", "=", "]", "!", "#", "@", "£", "+", "?", "x" ;

FOR I:=0 STEP 1 UNTIL 20 DO MODE[SS[I]]:=2 ;
COUNT:=0 END ; % EO MO

BEGIN % M IS 1;
LABEL L1,L2,L3,L4,L5,L6;

INTEGER I,J,IE; REAL IL1;
BOOLEAN EOL,IE1;
REAL E;
REAL Y;
I:=0;
L4: IF COUNT EQL 0 THEN
BEGIN
UREAD(F,D[+]);
P:=POINTER(D);
GO TO L6;
END
ELSE IF COUNT EQL N
THEN BEGIN
E:=MAKE(TYPELOGIC, LAST);
J:=MODE[TOTCHAR].[5:6];
EOL:=TRUE END
ELSE L6: BEGIN
E:=MAKE(TYPELOGIC, CHAR(P,1));
J:=MODE[CHAR(P,1)]; P:=P+1;
EOL:=BOOLEAN(J.[6:11]);
J:=J.[5:6] END;
L5: CASE J OF BEGIN
BEGIN % -J IS 0;
CASE I OF BEGIN
BEGIN % I IS 0;
Y:=CONS(E, NULLIST);
I:=1;
GO TO L1 END; % EO IO
BEGIN % I IS 1;
Y:=CONS(E, Y);

```

```

00894000 T 0000
00895000 T 0000
START OF SEGMENT ***** 28
00896000 T 0004
00897000 T 0006
00898000 T 0006
00898050 T 0006
00899000 T 0009
00900000 T 0009
START OF SEGMENT ***** 29
00901000 T 0000
00902000 T 0000
00903000 T 0002
00904000 T 0005
00905000 T 0007
00906000 T 0010
00907000 T 0011
00907050 T 0012
START OF SEGMENT ***** 30
00907060 T 0014
30 IS 21 LONG, NEXT SEG 29
00907080 T 0014
00908000 T 0019
29 IS 21 LONG, NEXT SEG 28
00909000 T 0011
00910000 T 0011
START OF SEGMENT ***** 31
00911000 T 0000
00912000 T 0000
00913000 T 0000
00914000 T 0000
00915000 T 0000
00916000 T 0000
00917000 T 0001
00918000 T 0002
00919000 T 0003
00920000 T 0005
00921000 T 0005
00922000 T 0005
00923000 T 0006
00924000 T 0007
00925000 T 0009
00926000 T 0010
00927000 T 0011
00928000 T 0012
00929000 T 0013
00930000 T 0020
00931000 T 0021
00932000 T 0023
00933000 T 0023
00934000 T 0023
00935000 T 0024
00936000 T 0024
00937000 T 0026
00938000 T 0026
00939000 T 0027
00940000 T 0027

```

```

GO TO L1 END; % EO I1;
BEGIN % I IS 2
J1=2; GO TO L5 END % EO I2
END % CASES OF I;
END; % EO =J0

BEGIN % -J IS 1;
CASE I OF BEGIN
BEGIN % I IS 0;
IL1:=E.ADDR; I1:=2; IE1:=FALSE;
GO TO L1 END; % EO I0;
BEGIN % I IS 1;
Y:=CONS(E,Y);
GO TO L1 END; % EO I1;
BEGIN % I IS 2;
IF IE1 THEN
IL1:=IL1+E.ADDR/10+(IE1:=IE+1) ELSE
IL1:=IL1*10+E.ADDR;
GO TO L1 END % EO I2;
END % CASES OF I;
END; % EO =J1;

```

```

BEGIN % -J IS 2 SPECIAL;
CASE I OF BEGIN
BEGIN % I IS 0;
IL:=HANGON(X,CONS(E,NULLIST));
GO TO L3 END; % EO I0;
BEGIN % I IS 1;
IL:=HANGON(X,Y);
P:=P-1;
GO TO L2 END; % EO I1;
BEGIN % I IS 2;
IF E.ADDR EQL "." THEN
BEGIN IE1:=0; IE1:=TRUE; GO TO L1 END;
IL:=AD(IL1); IE1:=FALSE;
P:=P-1; GO TO L2 END % EO I2;
END % CASES OF I;
END; % EO =J2;

```

```

BEGIN % -J IS 3 SPACE;
CASE I OF BEGIN
BEGIN % I IS 0;
GO TO L1 END; % EO I0;
BEGIN % I IS 1;
IL:=HANGON(X,Y);
GO TO L3 END; % EO I1;
BEGIN % I IS 2;
IL:=AD(IL1); IE1:=FALSE;
GO TO L3 END % EO I2;
END % CASES OF I;
END; % EO =J3;

```

```

BEGIN % -J IS 4 IGNORE;

```

```

00941000 T 0029
00942000 T 0030
00943000 T 0030
00944000 T 0031
00945000 T 0031
START OF SEGMENT ***** 32
32 IS 3 LONG, NEXT SEG 31
00946000 T 0032
00947000 T 0032
00948000 T 0032
00949000 T 0032
00950000 T 0035
00951000 T 0036
00952000 T 0036
00953000 T 0038
00954000 T 0039
00954010 T 0039
00954020 T 0039
00955000 T 0045
00956000 T 0047
00957000 T 0048
00958000 T 0048
START OF SEGMENT ***** 33
33 IS 3 LONG, NEXT SEG 31
00959000 T 0048
00960000 T 0048
00961000 T 0049
00962000 T 0049
00963000 T 0051
00964000 T 0052
00965000 T 0052
00966000 T 0054
00967000 T 0057
00968000 T 0058
00968010 T 0058
00968020 T 0059
00969000 T 0061
00970000 T 0063
00971000 T 0067
00972000 T 0067
START OF SEGMENT ***** 34
34 IS 3 LONG, NEXT SEG 31
00973000 T 0067
00974000 T 0067
00975000 T 0068
00976000 T 0068
00977000 T 0069
00978000 T 0069
00979000 T 0070
00980000 T 0071
00981000 T 0071
00982000 T 0073
00983000 T 0074
00984000 T 0074
START OF SEGMENT ***** 35
35 IS 3 LONG, NEXT SEG 31
00985000 T 0074

```

```
GO TO L1 END % EO=J4;
END;% CASES OF =J;
```

```
L1: IF EOL THEN COUNT:=0
ELSE COUNT:=COUNT+1;
GO TO L4;
L3: IF EOL THEN COUNT:=0
ELSE COUNT:=COUNT+1;
L2: END; % EO M1
```

```
BEGIN % M IS 2;
IL:=MAKE(TYPELOGIC,MODE[C,ADDR]);
MODE[ C,ADDR]:=CM,ADDR;
END % EO M2;
END % CASES OF M;
END % OWN BLOCK;
```

```
END;% PROCEDURE BLOCK *** END OF INLEX ***
```

```
REAL PROCEDURE LEXFIND;
BEGIN INTEGER T1,T2,T3;

DEFINE X=STACKPOSITION#;
P(P1)=POINTER(STACK[(P1.[18:7]),*])+P1.[11:12]#;
LABEL LEND;
CHR:=0;
REPLACE CP BY P(X) FOR 1;
T1:=CHR;
REPLACE CP=5 BY P(X) FOR (IF T1 LSS 6 THEN T1 ELSE 6);
T:=T3:=REF((T2:=AT(CHR MOD 127)+1),HASH);
WHILE T NEQ NULLIST DO
IF P(X) NEQ P((CAR(T)=2000)) FOR T1 THEN
T:=CDR(T) ELSE
BEGIN LEXFIND:=CAR(T); GO TO LEND END;
ASGN(T2,HASH,CONS(LEXFIND:=TOKEN(X+2000),T3));
X:=X+T1;
LEND:END LEXFIND;
```

```
PROCEDURE UREAD(FIL,BUF); VALUE FIL; REAL FIL; ARRAY BUF[0];
BEGIN
FILE IN DISKFIL DISK RANDOM (1,10,150,SAVE 999);

POINTER PX;
LABEL REM, L1,EOF;
REAL MFID,FID,N,X;
IF REF(AT6,FIL)=TOKEN(1010)THEN GO TO REM; %REMOTE DEVICE
```

```
00986000 T 0074
00987000 T 0075
START OF SEGMENT ***** 36
36 IS 5 LONG, NEXT SEG 31
00988000 T 0075
00989000 T 0076
00990000 T 0079
00991000 T 0079
00992000 T 0080
00993000 T 0083
31 IS 85 LONG, NEXT SEG 28
00994000 T 0012
00995000 T 0012
00996000 T 0015
00997000 T 0017
00998000 T 0017
00999000 T 0017
START OF SEGMENT ***** 37
37 IS 3 LONG, NEXT SEG 28
01000000 T 0017
28 IS 21 LONG, NEXT SEG 27
27 IS 6 LONG, NEXT SEG 2
```

```
01001000 T 0073
01002000 T 0073
START OF SEGMENT ***** 38
01003000 T 0000
01004000 T 0000
01005000 T 0000
01006000 T 0000
01007000 T 0001
01008000 T 0008
01009000 T 0009
01010000 T 0020
01011000 T 0025
01012000 T 0026
01013000 T 0039
01014000 T 0041
01015000 T 0045
01016000 T 0048
01017000 T 0050
38 IS 55 LONG, NEXT SEG 2
```

```
01018000 T 0073
01019000 T 0073
01020000 T 0073
START OF SEGMENT ***** 39
01021000 T 0006
01022000 T 0006
01023000 T 0006
01024000 T 0006
```

```

IF REF(AT6,FIL)=TOKEN(1011)THEN %DISK
BEGIN CHR:=0;
X:= REF(AT2,FIL)=2000;
PX:=POINTER(STACK[(X,[18:7]),*])+(X,[11:12]);
REPLACE CP BY PX:PX FOR 1;
N:=(IF CHR GTR 7 THEN 7 ELSE (CHR-1));
REPLACE CP-5 BY " " FOR 6;
REPLACE CP-6 BY PX:PX FOR N;
MFID:=CHR;
CHR:=0;
X:= REF(AT3,FIL)=2000;
PX:=POINTER(STACK[(X,[18:7]),*])+(X,[11:12]);
REPLACE CP BY PX:PX FOR 1;
N:=(IF CHR GTR 7 THEN 7 ELSE (CHR-1));
REPLACE CP-5 BY " " FOR 6;
REPLACE CP-6 BY PX:PX FOR N;
FID:=CHR;
FILL DISKFIL WITH MFID, FID;
READ(DISKFIL [REF(AT1,FIL),ADDR],10,BUF[*])[EOF];
WRITE(DATA,10,BUF[*]);
ASGN(AT1,FIL,REF(AT1,FIL)+1);
GO TO L1 END;
EOF: PRIMERROR("URED",FIL);
REM: READ(DATA,9,BUF[*]);
L1: END UREAD;

```

```

01025000 T 0008
01026000 T 0010
01027000 T 0012
01028000 T 0014
01029000 T 0019
01030000 T 0022
01031000 T 0026
01032000 T 0032
01033000 T 0037
01034000 T 0038
01035000 T 0039
01036000 T 0041
01037000 T 0046
01038000 T 0049
01039000 T 0053
01040000 T 0059
01041000 T 0063
01042000 T 0064
01043000 T 0068
01043010 T 0076
01044000 T 0080
01045000 T 0083
01046000 T 0086
01047000 T 0087
01048000 T 0091

```

39 IS 98 LONG, NEXT SEG 2

```

PROCEDURE MEMSAVE(X,N,Y);
VALUE N,Y;ARRAY X[0];REAL N,Y;
BEGIN INTEGER I,K;REAL MOO;

SAVE FILE MEMFL DISK SERIAL[20:25] (2,512,512,SAVE 999);
FILL MEMFL WITH Y,TIME(-1);
K:=LASTUSEDSPACE,[18:18]+1;
MEMORY[K,[17:9],K,[8:9]]:=MOO:=MEMORY[0,0];
MEMORY[0,0]:=N&LASTUSEDSPACE[45:22:23];
FOR I:=0 STEP 1 UNTIL N DO BEGIN
K:=K+1; MEMORY[K,[17:9],K,[8:9]]:=X[I] END;
FOR I:=0 STEP 1 UNTIL K,[17:9] DO
WRITE (MEMFL,512,MEMORY[I,*]);
STACK[0,0]:=STACKPOSITION; I:=-1;
WHILE I:=I+1 LEQ STACKPOSITION,[18:7] DO
WRITE (MEMFL,512,STACK[I,*]);
LOCK(MEMFL);
MEMORY[0,0]:=MOO;END OF SAVE;

```

```

01049000 T 0073
01050000 T 0073
01051000 T 0073
START OF SEGMENT ***** 40
01052000 T 0000
01053000 T 0006
01054000 T 0010
01055000 T 0012
01056000 T 0017
01057000 T 0020
01058000 T 0022
01059000 T 0028
01060000 T 0033
01061000 T 0038
01062000 T 0041
01063000 T 0043
01064000 T 0048
01065000 T 0050

```

40 IS 57 LONG, NEXT SEG 2

```

PROCEDURE UNSAVE(X,N,FID);
VALUE FID;ARRAY X[0];REAL N,FID;
BEGIN INTEGER I,K;

```

```

01066000 T 0073
01067000 T 0073
01068000 T 0073
START OF SEGMENT ***** 41

```



```

SAVE FILE MEMFL DISK SERIAL(1,512,512);
IF BOOLEAN(XX1) THEN FILL MEMFL WITH FID, TIME(-1) ELSE
FILL MEMFL WITH FID, XX2 & "B0005"[41:29:30];
READ (MEMFL, 512, MEMORY[0,*]);
LASTUSEDSPACE:=M[0], ADDR; N:=M[1], ADDR;
K:=LASTUSEDSPACE, [18:18]+N+2;
FOR I:=1 STEP 1 UNTIL K, [17:9] DO
READ (MEMFL, 512, MEMORY[I,*]); K:=K-N;
READ (MEMFL, 512, STACK[0,*]);
STACKPOSITION:=STACK[0,0]; I:=0;
WHILE I:=I+1 LEQ STACKPOSITION, [18:7] DO
READ (MEMFL, 512, STACK[I,*]);
MEMORY[0,0]:=MEMORY[(K-1), [17:9], (K-1), [8:9]
] END OF UNSAVE;

```

```

01069000 T 0000
01070000 T 0003
01071000 T 0008
01072000 T 0014
01073000 T 0018
01074000 T 0034
01075000 T 0036
01076000 T 0040
01077000 T 0048
01078000 T 0052
01079000 T 0055
01080000 T 0058
01083000 T 0063
01084000 T 0068
41 IS 73 LONG, NEXT SEG 2

```

```

REAL PROCEDURE BNFTREE(F);
REAL F;
BEGIN INTEGER I;

REAL X, Y, Y1, Z, Z1;
LABEL L1, L2, L3, L4, L5, L6, L7;
REAL ARRAY SS[0:5];
PROCEDURE ERR(I); INTEGER I;
BEGIN
FORMAT F1(X3, "$ ON LHS");

FORMAT F2(X3, "NON-TERMINAL IS TYPE LOGIC");

CASE I OF BEGIN;
WRITE(DATA, F1);
WRITE(DATA, F2);
END END OF ERR;

```

```

01085000 T 0073
01086000 T 0073
01087000 T 0073
START OF SEGMENT ***** 42
01088000 T 0000
01089000 T 0000
01090000 T 0000
01091000 T 0001
01092000 T 0001
01093000 T 0001
START OF SEGMENT ***** 43
START OF SEGMENT ***** 44
44 IS 6 LONG, NEXT SEG 43
01094000 T 0000
START OF SEGMENT ***** 45
45 IS 9 LONG, NEXT SEG 43
01095000 T 0000
01096000 T 0000
01097000 T 0004
01098000 T 0007
START OF SEGMENT ***** 46
46 IS 3 LONG, NEXT SEG 43
43 IS 8 LONG, NEXT SEG 42

```

```

SWITCH SW :=L5, L6, L4;
FILL SS[+] WITH "=", ";", ":", "s", ":", "e", "#";

FOR I:= 0 STEP 1 UNTIL 5 DO BEGIN
X:=LOADMODE(SS[I], 2);
X:=CONS(MAKE(TYPELOGIC, SS[I]), NULLIST);
SS[I]:=HANGON(X, X) END;
X:=LOADMODE("X", 68);
L6:Z:=CONS(MAKE(TYPECODE, 118), NULLIST);
Y1:=CONS(ATOMUNDEF, NULLIST);

```

```

01099000 T 0001
01100000 T 0007
START OF SEGMENT ***** 47
47 IS 6 LONG, NEXT SEG 42
01101000 T 0009
01102000 T 0011
01103000 T 0017
01104000 T 0020
01105000 T 0024
01106000 T 0029
01107000 T 0032

```

```

WHILE (X:=USERS(F)) NEQ SS[0] DO BEGIN
  IF X EQL SS[2] THEN BEGIN ERR(1);GO TO L4 END; %$ ON LHS
  IF X,TYPE EQL TYPELOGIC THEN BEGIN ERR(2); GO TO L4 END; % LOGIC ON L
  EQLIST(Y1,(Y1:=USERL(X))) END;
  IF(X:=USERS(F)) EQL SS[4] AND CAR(Y1) NEQ ATOMUNDEF
  THEN BEGIN EQLIST(Z,Y1); GO TO L5 END
  ELSE BEGIN EQLIST(Y1,Z); Y1:=Y:=CONS(MAKE(TYPECODE,119),NULLIST);
  GO TO L7 END;
  L5:Y1:=Y:=CONS(MAKE(TYPECODE,119),NULLIST);
  L2:Y1:=USERS(F);
  L7:IF X EQL SS[2] THEN BEGIN
  #SGN(ATOMCDR,Y1,(Y1:=CONS(USERS(F),NULLIST)));
  GO TO L2 END; % TERMINAL
  IF X EQL SS[4] THEN
  BEGIN I:=1; GO TO L3 END; % @
  IF X EQL SS[5] THEN
  BEGIN
  ASGN(ATOMCDR,Y1,(Y1:=CONS(USERL(X),NULLIST)));
  ASGN(ATOMCDR,Y1,FNS(X));
  GO TO L7 END;
  IF X EQL SS[1] THEN
  BEGIN I:=2; GO TO L3 END; % SEMICOLON
  IF X EQL SS[3] THEN
  BEGIN I:=3; GO TO L3 END; % COLON
  IF X,TYPE EQL TYPELOGIC THEN BEGIN ERR(2); GO TO L4 END; % NON-TERMINA
  ASGN(ATOMCDR,Y1,(Y1:=CONS(USERL(X),NULLIST)));
  GO TO L2;
  L3:ASGN(ATOMCDR,Z,CONS(CDR(Y),CDR(Z)));
  GO TO SW[I];
  L4:BNFTREE:=LABTAB
  END; % END OF BNFTREE

```

```

01108000 T 0034
01109000 T 0036
01110000 T 0040
01111000 T 0045
01112000 T 0047
01113000 T 0050
01114000 T 0053
01115000 T 0057
01116000 T 0058
01117000 T 0062
01118000 T 0063
01119000 T 0065
01120000 T 0068
01121000 T 0069
01122000 T 0070
01123000 T 0072
01124000 T 0073
01125000 T 0073
01126000 T 0076
01127000 T 0079
01128000 T 0080
01129000 T 0081
01130000 T 0083
01131000 T 0084
01132000 T 0085
01133000 T 0090
01134000 T 0093
01135000 T 0093
01136000 T 0097
01137000 T 0099
01138000 T 0099

```

42 IS 109 LONG, NEXT SEG 2

```

REAL PROCEDURE DA(X);VALUE X;REAL X;
BEGIN IF X,TYPE EQL TYPELOGIC THEN
DA:=X,ADDR ELSE
IF X,TYPE EQL TYPEAREA
THEN DA:=O&REF(AT1,X)[46:8:9]&REF(AT2,X)[37:18:19]
&REF(AT3,X)[18:18:19]ELSE PRIMERROR("DA",X) END;

```

```

01138010 T 0073
01138011 T 0073
01138012 T 0075
01138013 T 0077
01138014 T 0078
01138015 T 0083

```

```

REAL PROCEDURE AD(X);VALUE X;REAL X;IF X GTR 0 AND
X.[38:39] LSS 524288 THEN AD:=MAKE(TYPELOGIC,X)
ELSE BEGIN REAL AD1;AD1:=AD:=PMAKE(TYPEAREA,MAKE(TYPELOGIC,3));

```

```

01138016 T 0090
01138017 T 0091
01138018 T 0093
01138019 T 0003
01138020 T 0005
01138021 T 0008

```

START OF SEGMENT ***** 48

48 IS 12 LONG, NEXT SEG 2

```

REAL PROCEDURE AR(X,Y,Z);VALUE X,Y,Z;REAL X,Y,Z;
BEGIN REAL AR1,AR2;

REAL PROCEDURE FN(I,J,K);VALUE I,J,K;REAL I,J,K;
BEGIN REAL F;CASE I OF BEGIN
F:=J+K;F:=J-K;F:=J*K;F:=J/K;F:=J*K END; %END OF CASE

FN:=AD(F) END;

```

```

01138030 T 0100
01138035 T 0100
START OF SEGMENT ***** 49
01138040 T 0000
01138045 T 0000
START OF SEGMENT ***** 50
01138050 T 0000
START OF SEGMENT ***** 51
51 IS 5 LONG, NEXT SEG 50
01138055 T 0010
50 IS 15 LONG, NEXT SEG 49

```

```

IF Y.TYPE EQL TYPELIST THEN
BEGIN AR2:=AR1:=CONS(ATOMUNDEF,NULLIST);
IF Z.TYPE EQL TYPELIST THEN
BEGIN WHILE Y NEQ NULLIST AND Z NEQ NULLIST DO
BEGIN ASGN(ATO,AR1,AR1:=CONS(FN(X,DA(CAR(Y)),DA(CAR(Z))),
NULLIST));Y:=CDR(Y);Z:=CDR(Z) END;
IF Y NEQ Z THEN PRIMERRR("AR",Y) ELSE AR:=CDR(AR2) END
ELSE BEGIN Z:=DA(Z);
WHILE Y NEQ NULLIST DO BEGIN
ASGN(ATO,AR1,AR1:=CONS(FN(X,DA(CAR(Y)),Z),NULLIST));
Y:=CDR(Y) END; AR:=CDR(AR2) END
END ELSE IF Z.TYPE EQL TYPELIST THEN
BEGIN Y:=DA(Y);AR2:=AR1:=CONS(ATOMUNDEF,NULLIST);
WHILE Z NEQ NULLIST DO BEGIN
ASGN(ATO,AR1,AR1:=CONS(FN(X,Y,DA(CAR(Z))),NULLIST));
Z:=CDR(Z) END; AR:=CDR(AR2) END
ELSE AR:=FN(X,DA(Y),DA(Z)) END;

```

```

01138060 T 0000
01138065 T 0001
01138070 T 0003
01138075 T 0005
01138080 T 0008
01138085 T 0013
01138090 T 0017
01138100 T 0022
01138105 T 0024
01138110 T 0025
01138115 T 0030
01138120 T 0033
01138125 T 0035
01138130 T 0038
01138135 T 0040
01138140 T 0045
01138145 T 0048
49 IS 58 LONG, NEXT SEG 2

```

```

PROCEDURE RETURN(PROCESS,VAL);VALUE PROCESS,VAL;
REAL PROCESS,VAL;
BEGIN IF PROCESS.TYPE NEQ TYPEPROCESS THEN PRIMERRR("RTRN",
PROCESS);
ASGN(AT7,PROCESS,CONS(RC9,CONS(VAL,REF(AT7,PROCESS))));
PROCESS:=QQ(PROCESS)END RETURN;

```

```

01139000 T 0100
01140000 T 0100
01141000 T 0100
01142000 T 0103
01143000 T 0103
01144000 T 0107

```

```

PROCEDURE RESUME(PROCESS);VALUE PROCESS; REAL PROCESS;
BEGIN IF PROCESS.TYPE NEQ TYPEPROCESS THEN
PRIMERRR("RSM1",PROCESS);
IF REF(AT2,PROCESS) NEQ ATOMSUSPENDED THEN
PRIMERRR("RSM2",PROCESS);
ASGN(AT2,PROCESS,ATOMRESUMED);
ASGN(AT7,PROCESS,CONS(RC1,REF(AT7,PROCESS)));
PROCESS:=QQ(PROCESS) END RESUME;

```

```

01145000 T 0110
01146000 T 0110
01147000 T 0111
01148000 T 0112
01149000 T 0114
01150000 T 0115
01151000 T 0117
01152000 T 0119

```

```

REAL PROCEDURE CREATE(START,ENV,INTERPRETER,PROCESS);
VALUE START,ENV,INTERPRETER,PROCESS;
REAL START,ENV,INTERPRETER,PROCESS;
BEGIN REAL Y;

Y:=PMAKE(TYPEPROCESS,LOGIC(7));
ASGN(AT1,Y,REF(AT1,PROCESS));
ASGN(AT2,Y,ATOMSUSPENDED);
ASGN(AT3,Y,START);
ASGN(AT4,Y,ENV);
ASGN(AT5,Y,INTERPRETER);
ASGN(AT6,Y,PROCESS);
ASGN(AT7,Y,NULLIST);
CREATE:=Y END CREATE;

```

```

01153000 T 0124
01154000 T 0124
01155000 T 0124
01156000 T 0124
START OF SEGMENT ***** 52
01157000 T 0000
01158000 T 0002
01159000 T 0004
01160000 T 0005
01161000 T 0007
01162000 T 0008
01163000 T 0009
01164000 T 0010
01165000 T 0012
52 IS 16 LONG, NEXT SEG 2

```

```

PROCEDURE DEBUG;
BEGIN FORMAT F1(8U),F2(A4,I6),F3(I6,"!",A4,I6),

```

```

01166000 T 0124
01167000 T 0124
START OF SEGMENT ***** 53
START OF SEGMENT ***** 54

```

```

NOTE1("DEBUG?"),
NOTE2("INVALID PROCESS OPERATION"),
NOTE3("INVALID TYPE ON FIRST PARAMETER"),
NOTE4("INVALID TYPE ON SECOND PARAMETER"),
NOTE5("INVALID TYPE ON THIRD PARAMETER"),
NOTE7("SYSTEM SAVED AS FILE--",A6),
NOTE8("SYSTEM LOADED FROM FILE--",A6),
NOTE9("#"),
NOTE10("SYSTEM LOADED FROM FILE--",A6,X1,A2),
NOTE6("NO PREVIOUS EXECUTE");

FORMAT FMTLUSP("AMTSPACE=",I6,"",LASTUSEDSPACE=",I6);

```

```

01168000 T 0000
01169000 T 0000
01170000 T 0000
01171000 T 0000
01172000 T 0000
01173000 T 0000
01174000 T 0000
01175000 T 0000
01176000 T 0000
01177000 T 0000
54 IS 94 LONG, NEXT SEG 53
01178000 T 0000
START OF SEGMENT ***** 55
55 IS 10 LONG, NEXT SEG 53

```

```

REAL OP,T1,V1,T2,V2,T3,V3,I,JUNK,J;
REAL ENV,PROCESS,X;
BOOLEAN APLI;REAL APLENV,AP1,AP2,AP3,AP4;
LABEL L1,L2,L3,L4,L5,L6,LEND;
MONITOR ZERO;
DEFINE PP(PP1,PP2,PP3)=J:=0;WHILE TYPEARRAY[J] NEQ PP1 DO
IF J NEQ 15 THEN J:=J+1 ELSE
BEGIN WRITE(DATA,PP2);GO TO L2 END;
PP3:=MAKE(J,PP3)#;
P1:=BEGIN PP(T1,NOTE3,V1)END#;
P2:=BEGIN P1;PP(T2,NOTE4,V2)END#;
P3:=BEGIN P2;PP(T3,NOTE5,V3)END#;
ZERO:=L2;
L1:WRITE(DATA,NOTE1);
L2:T1:=T2:=T3:="SYMB";
V1:=V2:=V3:=524287;
READ(DATA[STOP],F1,OP,T1,V1,T2,V2,T3,V3,JUNK);
FOR I:=0 STEP 1 UNTIL NOOFPROCESSOPS DO
IF OP=PROCESSARRAY[I] THEN GO TO L3;
WRITE(DATA,NOTE2);GO TO L2;

```

```

01179000 T 0000
01180000 T 0000
01180010 T 0000
01181000 T 0000
01182000 T 0000
01183000 T 0002
01184000 T 0002
01185000 T 0002
01186000 T 0002
01187000 T 0002
01188000 T 0002
01189000 T 0002
01190000 T 0002
01191000 T 0004
01192000 T 0008
01193000 T 0009
01194000 T 0011
01195000 T 0030
01196000 T 0032
01197000 T 0035

```

```

L3:CASE I OF BEGIN
GO TO LEND; % E N D
BEGIN P3;ASGN(V1,V2,(V1:=V3))END; %ASGN
BEGIN P2; V1:=REF(V1,V2) END; %REF
BEGIN P1;V1:=QQ(V1);GO TO L2 END; %QQ
V1:=UQ; %UQ
BEGIN P1;V1:=CAR(V1) END; %CAR
BEGIN P1;V1:=CDR(V1) END; %CDR
BEGIN P2;V1:=CONS(V1,V2) END; %CONS
BEGIN P1;V1:=NEWSYMB(V1)END; %NEWSYMB
BEGIN WRITE(DATA,FMTLUSP,AMTSPACE,LASTUSEDSPACE);GO TO L2 END;%AMTSP
BEGIN P2;V1:=PMAKE(V1,TYPE,V2)END; %MAKE
BEGIN P1;PRINT(V1);WRITE(DATA,NOTE9);GO TO L2 END; %PRINT
;XBEGIN P3;V1:=LOADMODE(V1,V2,V3)END; %LDMODE
BEGIN GCL;GO TO L2 END;%GCL
BEGIN P1;IF V1 NEQ ATOMUNDEF THEN %INTERP
RESUME(V1);INTERP;GO TO L2 END;
V1:=LABTAB;
V1:=LEXTREE;
BEGIN P1;V1:=USERS(V1) END; % USERS
BEGIN SMEM(T1);WRITE(DATA,NOTE7,T1);GO TO L2 END;%SMEM
BEGIN XX1:=1;LMEM(T1); %LMEM
APLI:=FALSE;
WRITE(DATA,NOTE8,T1);
I:=41;T1:="LIST";V1:=REF(AT2,REF(TOKEN(2742),LABTAB)).ADDR;
T2:="AREA";V2:=REF(TOKEN(2608),LABTAB);GO TO L3 END;
BEGIN P1;V1:=STATS END;% PSCAN
BEGIN P1;V1:=BNFTREE(V1) END; % BNFTRE
BEGIN P2;V2:=USERS(V2);ASGN(V2,LABTAB,V1)END; % INCODE
BEGIN V1:=QU(5,0);WRITE(DATA,NOTE9);GO TO L2 END; %QDUMP
BEGIN FOR I:=T1 STEP 1 UNTIL V1 DO %DUMP
WRITE(DATA,F3,I,TYPEARRAY[M[I],TYPE],M[I],ADDR);
WRITE(DATA,NOTE9);GO TO L2 END;
BEGIN P1;PRINERROR(T2,V1);GO TO L2 END;
BEGIN P1;MASGN(T2,V1) END; %MASGN
BEGIN P1; XX1:=V1;XX2:=T2;V1:=QU(5,1);GO TO L2 END;%QASGN
V1:=M(T1); %M
BEGIN XX2:=T1;V1:=QU(5,2)END; %Q
BEGIN P1;ENV:=IF APLI THEN APLENV
ELSE PMAKE(TYPECONS,0); % EXE
LOADMODE("S",2);
PROCESS:= CREATE(NULLLIST,ENV,TOKEN(1004),CONS(AT(262144),NULLLIST));
ASGN(AT6,PROCESS,ATOMUNDEF);
L4:ASGN(AT2,PROCESS,ATOMSUSPENDED);
ASGN(AT3,PROCESS,(X:=V1));
WHILE CDR(X) NEQ NULLLIST DO X:=CDR(X);
ASGN(AT0,X,CONS(CONS(TOKEN(117),NULLLIST),NULLLIST));
RESUME (PROCESS);
INTERP;
ENV:=REF(AT4,PROCESS:=XX1);
ASGN(AT7,PROCESS,NULLLIST);
V1:=REF(TOKEN(1000),ENV);
IF APLI THEN BEGIN APLPNT(V1); APLENV:=ENV;
I:=36;T1:="LIST";T2:="AREA";V1:=REF(TOKEN(5003),LABTAB);
V2:=REF(AT2,V1).ADDR;V1:=REF(AT1,V1).ADDR;
GO TO L3 END ;
END;

```

```

01198000 T 0039
01199000 T 0040
01200000 T 0041
01201000 T 0076
01202000 T 0101
01203000 T 0114
01204000 T 0116
01205000 T 0130
01206000 T 0143
01207000 T 0167
01208000 T 0180
01209000 T 0193
01210000 T 0217
01211000 T 0234
01212000 T 0234
01213000 T 0236
01214000 T 0248
01215000 T 0250
01216000 T 0252
01217000 T 0253
01218000 T 0266
01219000 T 0277
01219010 T 0280
01220000 T 0281
01220010 T 0288
01220020 T 0293
01221000 T 0301
01222000 T 0317
01223000 T 0330
01224000 T 0355
01225000 T 0360
01226000 T 0362
01227000 T 0389
01228000 T 0393
01229000 T 0407
01230000 T 0419
01231000 T 0435
01232000 T 0443
01233000 T 0445
01233010 T 0458
01234000 T 0460
01235000 T 0465
01236000 T 0468
01237000 T 0470
01238000 T 0471
01239000 T 0473
01240000 T 0478
01241000 T 0481
01242000 T 0482
01242010 T 0482
01243000 T 0484
01244000 T 0485
01244010 T 0487
01244020 T 0490
01244025 T 0494
01244030 T 0498
01245000 T 0502

```

```

IF PROCESS.TYPE EQL TYPEPROCESS THEN GO TO L4 %CEXE
ELSE BEGIN WRITE(DATA,NOTE6);GO TO L2 END;
BEGIN XX1:=0;XX2:=""
LMEM(T1);WRITE(DATA,NOTE10,T1);GO TO L2 END;
BEGIN ARRAY X[0:9]; %READ

P1; UREAD(V1,X);
WRITE(DATA,9,X[*]); GO TO L2 END;

BEGIN POINTER SP,XP; % DSTACK

REAL N; ARRAY X[0:8];
XPI=POINTER(X); SPI=POINTER(STACK[0,*]);
CHR:=N:=8;
WHILE N LSS STACKPOSITION DO
  BEGIN REPLACE XP BY " " FOR 72;
  REPLACE CP BY SP+N FOR 1;
  REPLACE XP BY N+2000 FOR 4 DIGITS;
  REPLACE XP+6 BY SP+N+1 FOR CHR=1;
  N:=N+CHR;
  WRITE(DATA,9,X[*]) END;
  WRITE(DATA,NOTE9);GO TO L2 END DSTACK;

BEGIN P1;X:=V1; % MODEL
L6;V1:=ATOMUNDEF;T1:=3;T2:=T3:=0;
READ(DATA,F1,V1,T1,T2,T3,V2,V3,OP,JUNK);
IF V1=ATOMUNDEF THEN GO TO L2;
V1:=AT(V1+1);T1:=T1&T2[14:3:4]&T3[18:3:4];
ASGN(V1,X,AT(T1));GO TO L6 END;
BEGIN P2;ENV:=PMAKE(TYPECONS,0); %PARSE
ASGN(TOKEN(1018),ENV,NULLIST);
ASGN(TOKEN(1019),ENV,ATO);
ASGN(TOKEN(1006),ENV,CONS(USERS(V2),CONS(V2,NULLIST)));
PROCESS:=CREATE(NULLIST,ENV,TOKEN(1005),CONS(AT(262144),NULLIST));
ASGN(AT6,PROCESS,ATOMUNDEF);
ASGN(AT3,PROCESS,CONS(V1,CONS(TOKEN(117),NULLIST)));
ASGN(AT7,PROCESS,NULLIST);
RESUME(PROCESS); INTERP;
IF APLI THEN
  BEGIN T1:="PRCS";V1:=XX1.ADDR;
  I:=37;GO TO L3 END;
  V1:=XX1 END PARSE;
  BEGIN P1;WHILE PRIORITY,ADDR GTR 0 DO JUNK:=UQ;%TRANS
  ASGN(AT3,V1,CONS(CONS(TOKEN(143),NULLIST),CONS(
  CONS(TOKEN(117),NULLIST),NULLIST)));
  ASGN(AT7,V1,NULLIST);ASGN(AT5,V1,TOKEN(1017));
  RESUME(V1);
  INTERP;
  V1:=REF(AT1,REF(AT4,XX1));
  IF APLI THEN
    BEGIN T1:="LIST";V1:=V1.ADDR;
    I:=30;GO TO L3 END END TRANS ;
  BEGIN TRACE:=NOT TRACE; GO TO L2 END;%TRACE
  BEGIN PRINT(WASH);GO TO L2 END;%HASH
  BEGIN INTERP;V1:=XX1;END;%RSTART
  BEGIN APLI:=TRUE;I:=36;
  ASGN(TOKEN(5003),LABTAB,CONS(MAKE(TYPELIST,V1),

```

```

01247000 T 0502
01248000 T 0503
01249000 T 0508
01250000 T 0509
01252000 T 0521
START OF SEGMENT ***** 56
01253000 T 0002
01254000 T 0017
56 IS 27 LONG, NEXT SEG 53
01255000 T 0522
START OF SEGMENT ***** 57
01256000 T 0000
01257000 T 0002
01258000 T 0006
01259000 T 0008
01260000 T 0009
01261000 T 0013
01262000 T 0018
01263000 T 0022
01264000 T 0030
01265000 T 0032
01266000 T 0038
57 IS 47 LONG, NEXT SEG 53
01267000 T 0523
01268000 T 0536
01269000 T 0538
01270000 T 0555
01271000 T 0557
01272000 T 0561
01273000 T 0564
01274000 T 0587
01275000 T 0589
01276000 T 0591
01277000 T 0595
01278000 T 0598
01279000 T 0600
01280000 T 0603
01281000 T 0604
01281010 T 0605
01281020 T 0606
01281030 T 0608
01282000 T 0612
01283000 T 0613
01284000 T 0628
01284001 T 0631
01284050 T 0634
01284060 T 0637
01285000 T 0637
01286000 T 0638
01286010 T 0640
01286020 T 0640
01286030 T 0643
01287000 T 0646
01288000 T 0648
01288010 T 0650
01288020 T 0652
01288030 T 0653

```

```

CONS(MAKE(TYPEAREA,V2),NULLIST)))
APLENV:=PMAKE(TYPECONS,0); GO TO L3 END; %APL
BEGIN XX1:=1;LMEM(T1);APLI:=FALSE;
WRITE(DATA,NOTE8,T1);GO TO L2 END; % LSTRT
END CASE STATEMENT;

```

```

L5:WRITE(DATA,F2,TYPEARRAY[V1,TYPE],V1,ADDR);
GO TO L2;
LEND:END DEBUG;

```

```

PROCEDURE DOTRACE(PROCESS);VALUE PROCESS;REAL PROCESS;
BEGIN PRINT(PROCESS);
PRINT(REF(AT3,PROCESS));
PRINT(REF(AT4,PROCESS));
PRINT(REF(AT7,PROCESS));END DOTRACE;

```

```

PROCEDURE MASGN(A,V);VALUE A,V;REAL A,V;
BEGIN IF BOOLEAN(A) THEN
MEMORY[A.[18:9],A.[9:9]].[22:23]:=V ELSE
MEMORY[A.[18:9],A.[9:9]].[45:23]:=V END MASGN;

```

```

PROCEDURE PRINT(X);VALUE X;REAL X;
BEGIN REAL I;
LABEL L1,L2;
FORMAT F1(":",A4,I6);
F2(X3,A4,I6,"=",A4,I6);
F3(X3,I4,"=",A4,I6);
F4(X3,I4,"=NIL");
WRITE(DATA,F1,TYPEARRAY[X,TYPE],X,ADDR);
CASE X,TYPE OF BEGIN
; %GARBAGE
L1:BEGIN X:=M[X]; %CONS
WHILE X NEQ NULLIST DO
BEGIN WRITE(DATA,F2,TYPEARRAY[CAR(X),TYPE],CAR(X),ADDR,
TYPEARRAY[CAR((X:=CDR(X))),TYPE],CAR(X),ADDR);
X:=CDR(X)END END;
BEGIN I:=1; WHILE X NEQ NULLIST DO %LIST
BEGIN WRITE(DATA,F3,I,TYPEARRAY[CAR(X),TYPE],CAR(X),ADDR);
I:=I+1;X:=CDR(X) END;
WRITE(DATA,F4)END;
L2:FOR I:=0 STEP 1 UNTIL M[X],ADDR DO %AREA
WRITE(DATA,F3,I,TYPEARRAY[M[X-I],TYPE],M[X-I],ADDR);
;%FIELD

```

```

01288035 T 0656
01288040 T 0658
01288050 T 0662
01288060 T 0666
01289000 T 0675
START OF SEGMENT ***** 58
58 IS 43 LONG, NEXT SEG 53
01290000 T 0675
01291000 T 0686
01292000 T 0686
53 IS 698 LONG, NEXT SEG 2

```

```

01293000 T 0124
01294000 T 0124
01295000 T 0124
01296000 T 0126
01297000 T 0127

```

```

01298000 T 0129
01299000 T 0129
01300000 T 0130
01301000 T 0135

```

```

01302000 T 0140
01303000 T 0140
START OF SEGMENT ***** 59
01304000 T 0000
01305000 T 0000
START OF SEGMENT ***** 60
01306000 T 0000
01307000 T 0000
01308000 T 0000
60 IS 29 LONG, NEXT SEG 59
01309000 T 0000
01310000 T 0011
01311000 T 0012
01312000 T 0012
01313000 T 0020
01314000 T 0022
01315000 T 0032
01316000 T 0041
01317000 T 0043
01318000 T 0046
01319000 T 0060
01320000 T 0063
01321000 T 0066
01322000 T 0077
01323000 T 0109

```

```

;XSYMBOL
;XLOGIC
;XLINK
GO TO L2; XPROCESS
GO TO L2; XMONITOR
GO TO L2; XGENERAL
END CASE STATEMENT;

```

```
END PRINT;
```

```

PROCEDURE APLPNT(X);VALUE X;REAL X;
BEGIN REAL I;FORMAT F1(X2,R12.5),F2(A4,I6);

```

```

I:=0;
IF X,TYPE EQL TYPELIST THEN BEGIN
WHILE X NEQ NULLLIST DO BEGIN
IF NBR(CAR(X)) THEN
BEGIN IF I MOD 4 EQL 3 THEN WRITE(DATA,F1,DA(CAR(X)))
ELSE WRITE(DATA[STOP],F1,DA(CAR(X)));
I:=I+1;X:=CDR(X) END ELSE BEGIN
WRITE(DATA,F2,TYPEARRAY[X,TYPE],X,ADDR);
X:=NULLLIST END END;WRITE (DATA) END
ELSE IF NBR(X) THEN WRITE(DATA,F1,DA(X))
ELSE WRITE(DATA,F2,TYPEARRAY[X,TYPE],X,ADDR) END;

```

```

PROCEDURE SYSM;
BEGIN FORMAT F1(5U);

```

```

NOTE1("SYSTEM?"),
NOTE2("INVALID OPERATION"),
NOTE3("CANDE?");

```

```

ARRAY X[0:4];INTEGER I;
DEFINE LA=X[0],X[1],X[2],X[3],X[4];
LABEL L1,L2,L3,LEND;
LASTUSEDSPACE:=0;CURROW:= 0 ;
NEXTTOKEN:=TOKEN(2000);
TRACE:=FALSE;
CP:=POINTER(JUNKARRAY)+7;
HASH:=PMAKE(TYPEAREA,AT(127));
FOR I:=AT1 STEP 1 UNTIL AT(127)DO ASGN(I,HASH,NULLIST);
STACKPOSITIONI:=8;
MASGN (0,LINKTONIL);
LABTAB:=PMAKE(TYPECONS,0);
ING(HASH);

```

```

01324000 T 0109
01325000 T 0109
01326000 T 0109
01327000 T 0109
01328000 T 0109
01329000 T 0110
01330000 T 0110

```

```

START OF SEGMENT ***** 61
61 IS 11 LONG, NEXT SEG 59
01331000 T 0110
59 IS 113 LONG, NEXT SEG 2

```

```

01331010 T 0140
01331015 T 0140

```

```

START OF SEGMENT ***** 62
START OF SEGMENT ***** 63
63 IS 10 LONG, NEXT SEG 62

```

```

01331020 T 0000
01331025 T 0000
01331030 T 0002
01331035 T 0004
01331040 T 0013
01331045 T 0021
01331050 T 0035
01331060 T 0038
01331065 T 0050
01331070 T 0055
01331075 T 0068

```

```
62 IS 88 LONG, NEXT SEG 2
```

```

01332000 T 0140
01333000 T 0140

```

```

START OF SEGMENT ***** 64
START OF SEGMENT ***** 65

```

```

01334000 T 0000
01335000 T 0000
01336000 T 0000

```

```
65 IS 19 LONG, NEXT SEG 64
```

```

01337000 T 0000
01338000 T 0001
01339000 T 0001
01340000 T 0001
01341000 T 0003
01342000 T 0004
01343000 T 0005
01344000 T 0008
01345000 T 0010
01346000 T 0018
01347000 T 0018
01348000 T 0019
01349000 T 0021

```



```
INTLX;
DEBUG;
LEND;END SYSM;
```

```
01350000 T 0022
01358000 T 0027
01361000 T 0027
64 IS 33 LONG, NEXT SEG 2
```

```
REAL PROCEDURE USERS(F);
REAL F;
BEGIN REAL X,X1;

USERS:=INLEX(F)
END OF USERS;
```

```
01362000 T 0140
01363000 T 0140
01364000 T 0140
START OF SEGMENT ***** 66
01365000 T 0000
01366000 T 0000
66 IS 8 LONG, NEXT SEG 2
```

```
REAL PROCEDURE USERL(X);
REAL X;
BEGIN
IF (USERL:=REF(X,LABTAB)) EQL ATOMUNDEF THEN
ASGN(X,LABTAB,(USERL:=CONS(ATOMUNDEF,NULLIST)));
END OF USERL;
```

```
01367000 T 0140
01368000 T 0140
01369000 T 0140
01370000 T 0140
01371000 T 0142
01372000 T 0145
```

```
REAL PROCEDURE PSCAN(IJ,X,F);REAL X,F;INTEGER IJ;
BEGIN OWN REAL ARRAY SS[0:11];
```

```
REAL Y;
PROCEDURE ERR;
BEGIN
FORMAT F1(X3,"ARG MISSING");
```

```
WRITE(DATA,F1) END;
```

```
01373000 T 0147
01374000 T 0147
START OF SEGMENT ***** 67
01375000 T 0002
01376000 T 0002
01377000 T 0002
01378000 T 0002
START OF SEGMENT ***** 68
START OF SEGMENT ***** 69
69 IS 6 LONG, NEXT SEG 68
01379000 T 0000
68 IS 4 LONG, NEXT SEG 67
```

```
REAL PROCEDURE QT(X);VALUE X;REAL X;
IF REF(X,LABTAB) NEQ ATOMUNDEF THEN BEGIN
IF (X:=USERL(X)) EQL TOKEN(1007) THEN
QT:=CONS(TOKEN(122),CONS(QT(USERS(F)),NULLIST))
ELSE QT:=X END ELSE QT:=X;
```

```
01380000 T 0002
01381000 T 0002
01382000 T 0005
01383000 T 0007
01384000 T 0011
```

```
REAL PROCEDURE FSCAN(X,Y,F);VALUE X; REAL X,Y,F;
BEGIN REAL FSTK,F1,FN;
```

```
01385000 T 0017
01386000 T 0017
START OF SEGMENT ***** 70
```

```

LABEL L1,L2,L3,L4,L5,L6,LEND;
FSTK :=NULLIST;
L1: IF Y NEQ SS[0] THEN BEGIN FSCAN:=X; GO TO LEND END;
F1:=FN:=CONS(X,NULLIST);
L2: IF (X:=USERS(F)) EQL SS[4] THEN
BEGIN X:=ATOMUNDEF; GO TO L4 END; %DUMMY ARG
IF X EQL SS[1] THEN
BEGIN X:=ATOMUNDEF; GO TO L5 END; % DUMMY ARG LIST
L6: X:=QT(X);
IF (Y:=USERS(F)) EQL SS[4] THEN
L4: BEGIN ASGN (ATOMCDR,FN,(FN:=CONS(X,NULLIST)));
GO TO L2 END; % COMMA
IF Y EQL SS[1] THEN
BEGIN ASGN (ATOMCDR,FN,CONS(X,NULLIST));
L5: X:=FN:=F1;
IF CAR(FSTK) EQL ATOMUNDEF THEN
BEGIN Y:=USERS(F); GO TO L1 END;
; F1:=CAR(FSTK); FSTK:=CDR(FSTK);
FN:=CAR(FSTK);
FSTK:=CDR(FSTK);
GO TO L6 END; % RPAREN
IF X EQL QT(SS[8]) THEN
BEGIN X:=USERL(Y); GO TO L6 END; % LABEL
FSTK:=CONS(FN,FSTK);
FSTK:=CONS(F1,FSTK);
GO TO L1;
LEND: END OF FSCAN;

```

```

01387000 T 0000
01388000 T 0000
01389000 T 0000
01390000 T 0003
01391000 T 0005
01392000 T 0008
01393000 T 0010
01394000 T 0011
01395000 T 0012
01396000 T 0014
01397000 T 0017
01398000 T 0020
01399000 T 0021
01400000 T 0022
01401000 T 0024
01402000 T 0026
01403000 T 0027
01404000 T 0030
01405000 T 0032
01406000 T 0034
01407000 T 0035
01408000 T 0035
01409000 T 0037
01410000 T 0040
01411000 T 0041
01412000 T 0043
01413000 T 0043

```

70 IS 47 LONG, NEXT SEG 67

```

REAL PROCEDURE SSCAN(F);
REAL F;
BEGIN REAL X,Y,Z,S1,S2;

LABEL L1,L2,L3,L4,L5,LEND,L6;
SSCAN:=Z:=CONS(ATOMUNDEF,NULLIST);
GO TO L1;
L3: Z:=CONS(ATOMUNDEF,NULLIST);
L2: L1: IF (X:=USERS(F)) EQL SS[10] OR X EQL SS[1] THEN
L6: BEGIN
GO TO LEND END % END
ELSE L5: IF X EQL SS[5] THEN
GO TO L1; % SEMICOLON
IF (Y:=USERS(F)) EQL SS[6] THEN
BEGIN EQLIST(USERL(X),Z);
GO TO L1 END % COLON
ELSE ASGN(ATOMCAR,Z,FSCAN(QT(X),Y,F));
IF ATOM(CAR(Z)) THEN BEGIN ERR; GO TO L2 END;
IF (X:=Y) EQL SS[5] THEN
BEGIN ASGN(ATOMCDR,Z,(Z:=CONS(ATOMUNDEF,NULLIST)));
GO TO L2 END % NO SUCCESSOR GIVEN
; IF X EQL SS[10] THEN GO TO L6
; IF X EQL SS[1] THEN GO TO LEND
ELSE ASGN(ATOMCDR,Z,CONS(ATOMUNDEF,NULLIST)); EQLIST(REF(ATO,Z),USERL(
X));

```

```

01414000 T 0017
01415000 T 0017
01416000 T 0017
START OF SEGMENT ***** 71
01417000 T 0000
01418000 T 0000
01419000 T 0002
01420000 T 0002
01421000 T 0004
01422000 T 0008
01423000 T 0009
01424000 T 0009
01425000 T 0011
01426000 T 0011
01427000 T 0013
01428000 T 0015
01429000 T 0016
01430000 T 0020
01431000 T 0027
01432000 T 0028
01433000 T 0031
01434000 T 0032
01435000 T 0033
01436000 T 0034
01437000 T 0038

```

GO TO L3; % SUCCESSOR GIVEN
LEND; END OF SSCAN;

PROCEDURE LDSS; BEGIN INTEGER I,J,K,KK; REAL STK;
FILL SS[*] WITH "E","N","D","R","O","U","T","I","N","E";

KI=2;KK=0; FOR JI=0 STEP 1 UNTIL 1 DO
BEGIN STK:=NULLIST;
FOR II=KK STEP 1 UNTIL K DO
STK:=CONS(MAKE(TYPELOGIC,SS[II]),STK);
SS[J+10]:=HANGON(STK,STK);
KI=9;KK=3 END;
FILL SS[*] WITH "(",")"","[","]","<",">",";",";","=","#";

FOR II=0 STEP 1 UNTIL 8 DO
BEGIN KI:=LOADMODE(SS[II],2);
STK:=CONS(MAKE(TYPELOGIC,SS[II]),NULLIST);
SS[II]:=HANGON(STK,STK) END;
KI:=LOADMODE("%",68)
END OF LDSS;

LDSS;
CASE IJ OF BEGIN
PSCAN:=SSCAN(F);
IF (YI=QT(USERS(F))) EQL SS[0]
THEN BEGIN PSCAN:=SSCAN(F);
XI:=USERS(F) END ELSE
BEGIN XI:=USERS(F); PSCAN:=CONS(FSCAN(Y,X,F),NULLIST)
END END END OF PSCAN;

AT0:=LOGIC(0);
AT1:=LOGIC(1);
AT2:=LOGIC(2);
AT3:=LOGIC(3);
AT4:=LOGIC(4);
AT5:=LOGIC(5);
AT6:=LOGIC(6);
AT7:=LOGIC(7);
TKNO:=MAKE(TYPETOKEN,0);
RC1:=MAKE(TYPECODE,1);

01438000 T 0039
01439000 T 0039
71 IS 44 LONG, NEXT SEG 67

01440000 T 0017
START OF SEGMENT ***** 72
01441000 T 0000
START OF SEGMENT ***** 73
73 IS 10 LONG, NEXT SEG 72

01442000 T 0001
01443000 T 0004
01444000 T 0004
01445000 T 0006
01446000 T 0011
01447000 T 0013
01448000 T 0017

START OF SEGMENT ***** 74
74 IS 9 LONG, NEXT SEG 72

01449000 T 0019
01450000 T 0020
01451000 T 0026
01452000 T 0029
01453000 T 0033
01454000 T 0033
72 IS 43 LONG, NEXT SEG 67

01455000 T 0017
01456000 T 0019
01457000 T 0019
01458000 T 0022
01459000 T 0025
01460000 T 0028
01461000 T 0029
01462000 T 0034

START OF SEGMENT ***** 75
75 IS 2 LONG, NEXT SEG 67
67 IS 43 LONG, NEXT SEG 2

01463000 T 0147
01464000 T 0149
01465000 T 0151
01466000 T 0153
01467000 T 0155
01468000 T 0156
01469000 T 0158
01470000 T 0160
01471000 T 0162
01472000 T 0163

```

RC91=MAKE(TYPECODE,9);
ATOMUNDEF1=MAKE(TYPESYMB+1,0)=1;
LINKTONIL:=MAKE(TYPELINK+1,0)=1;
NULLLIST:=MAKE(TYPELIST+1,0)=1;
FILL SYSMARRAY[*] WITH "END","DEBUG";

```

```

FILL PROCESSARRAY[*] WITH "END","ASGN","REF","QQ","UQ","CAR",

```

```

"CDR","CONS","NEWSYM","AMTSP","MAKE","PRINT",
"LDMODE","GCL","INTERP","LABEL","NODES",
"USERS","SMEM","LMEM","PSCAN","BNFTRE","INCODE","QDUMP",
"DUMP","PERROR","MAASGN","QASGN","M","Q","EXE","CEXE","LOAD",
"READ","DSTACK","MODELD","PARSE","TRANS","TRACE","HASH","RSTART",
"APL","PARLD";

```

```

FILL TYPEARRAY[*] WITH "GRBG","CONS","LIST","AREA",

```

```

"BITS","SYMB","LGIC","LINK",
"PRCS","MNTR","GNRL","CODE",
"TP12","TP13","TP14","TP15";

```

```

SYSM;
END. END PROGRAM

```

NUMBER OF ERRORS DETECTED = 0. COMPILATION TIME = 247 SECONDS.

PRT SIZE = 264; TOTAL SEGMENT SIZE = 4578 WORDS; DISK SIZE = 270 SEGS; NO. PGM. SEGS = 92

ESTIMATED CORE STORAGE REQUIRED = 10093 WORDS.

ESTIMATED AUXILIARY MEMORY REQUIRED = 0 WORDS.

NUMBER OF CARD-IMAGES PROCESSED = 1607.

```

01473000 T 0165
01474000 T 0167
01475000 T 0170
01476000 T 0172
01477000 T 0175
START OF SEGMENT ***** 76
76 IS 2 LONG, NEXT SEG 2
01478000 T 0177
START OF SEGMENT ***** 77
01479000 T 0179
01480000 T 0179
01481000 T 0179
01482000 T 0179
01483000 T 0179
01483010 T 0179
77 IS 43 LONG, NEXT SEG 2
01484000 T 0179
START OF SEGMENT ***** 78
01485000 T 0180
01486000 T 0180
01487000 T 0180
78 IS 16 LONG, NEXT SEG 2
01488000 T 0180
01489000 T 0181
2 IS 184 LONG, NEXT SEG 1
1 IS 2 LONG, NEXT SEG 0
92 IS 69 LONG, NEXT SEG 0

```

LABEL 00000000LINE 00177116?COMPILE LISP/LISP XALGOL LIBRARY

XALGOL /LISP