

*Loc 0464R  
Must be changed from  
0C00 to 0A00 since  
the 300 printer has  
no rep of form*

## OS SOURCE UPDATER

### 1. INTRODUCTION

The O.S. Source Updater (Program Number 03-057) is an INTERDATA OS Utility which provides the ability to create and maintain a source file on a mass-storage type device. Commands may be entered by the operator on the teletype or may be prepared beforehand and read from cards or some other media. Directives allow copying, verifying, modifying, or displaying of source files. To facilitate positioning and accessing files, commands for positioning to a desired named file or to a specific file-mark are provided.

### 2. CONFIGURATION

The Source Updater requires the following configuration to run:

- Model 5, 50, 70, 74, 80 or 85 Processor
- 8KB Memory
- BOSS, DOS or RTOS
- Input, Output and List Device

The size of the Source Updater is 3,520 (X'0DC0') bytes.

### 3. FORMAT

In order for the Source Updater to function correctly, the volume must have an End of Volume mark (EOV) after the last file on the volume. An EOV consists of two file marks written back-to-back (see Figure 1).

The volume is made up of Source Files consisting of 80-byte records. Each record (Figure 2) can contain any ASCII source data in bytes 1 through 72. Bytes 73 through 80 are reserved for an 8-byte sequence number which is used for updating the source file. The end of a file is indicated by either a single file mark or an END statement as defined for the Assembler. If a file is terminated by an END statement, a file mark may optionally follow it for compatibility with other software systems. Figure 1 shows the format of a volume generated by the Source Updater.



BOT - Beginning of Tape  
 FM - File Mark  
 END - Assembler END Statement

Volume Format  
Figure 1

Free Formatted ASCII Data	AAANNNO
---------------------------	---------

Record Format  
Figure 2

#### 4. COMMANDS

##### 4.1 General Format

Each Source Updater directive has the following general format:

COMMAND arg1, arg2(,arg3). . .

where the command specifies the operation to be performed. It must not have any extraneous characters preceding it and only the first four characters of commands containing more than four characters need to be entered. If the command has any arguments to be input, it must be separated from the arguments by at least one space character. Otherwise, the command is terminated by a Carriage Return. The arguments are separated from each other by commas. If an optional argument (one enclosed in parentheses) is omitted, then the comma that would have separated it from the next argument must be present. If the last argument(s) is omitted, then the comma(s) separating the last argument(s) from the next previous argument may be omitted.

If an error is made in typing the first four characters of the command or if an illegal command is input, the error message:

ILLEGAL COMMAND

is printed on the console and a new directive is requested. The message:

OPERAND ERROR

is printed on the console when one of the following conditions is detected:

- a) A required argument was not entered
- b) An incorrect argument type
- c) An illegal character; i.e., a non-decimal character in a decimal argument or a non-hexadecimal character in a hexadecimal argument.

After the message has been printed, a new directive is requested.

## 4.2 FIND Command

This command searches the specified source file for the program indicated by the file name. After the file has been found, it is positioned immediately following the file identifier. The format of this command is:

```
FIND file (,lu)
```

where "file" is a six-character label preceded by \*\*, or a decimal number indicating the number of file marks between the beginning of tape and the desired file. LU specifies the logical unit number of the device on which the searching is to be performed. If a LU argument is not specified, logical unit 1 (the Old Master File) is assumed.

When the file has been located, the message:

```
FOUND
```

is output to logical unit 6 and a new command is requested. If the file name could not be found or more file-marks were specified to be skipped than were contained on the volume, the message:

```
NOT FOUND
```

is output to logical unit 6 and the next command is requested.

For example: to find a source file with the name '\*\*FILE7' contained on logical unit 4, the following command would be entered:

```
FIND **FILE7,4
```

Searching begins from where the volume is currently positioned. The command:

```
FIND 6
```

would cause the Old Master to be rewound, and the volume would then be positioned immediately following the sixth file mark on the volume. If the End of Volume is detected before the sixth file mark can be located, the message 'NOT FOUND' is output to LU6.

## 4.3 COPY Command

The COPY Command is used to copy a single source file containing a maximum of 80 bytes per record from one device to another. Before copying, the output device is searched for the End of Volume (EOV). Note that the EOV must exist on the output device before issuing this command. When the EOV is found, the output device is backspaced over the

EOV, a File-Mark is written, and the specified input file is copied. After the end of the file or END statement is detected, another EOV is written and backspaced over. The format of this command is:

```
COPY (file)(,LU)(,seqnbr)
```

where "file" optionally indicates a specific Source File which is to be copied. If a six-character file label preceded by \*\* is specified, the input device is searched starting at its current position. The file is rewound if a decimal number is entered indicating the number of file marks between the beginning of tape and the desired file. Unless otherwise indicated by 'LU', logical unit 1 is used as the input device. If the program is to be sequenced as it is copied, 'seqnbr' must be input; otherwise, no sequencing is done. The sequence number must be in the form AAANNNNN where AAA is any three character alpha-numeric identification and NNNNN is the initial five-numeric characters used as the sequence number for the first card. The sequence number is incremented by tens and wraps back to zero after 99990.

During the copying process, each source statement is scanned for a 'PAUSE' statement. When one is detected, it is copied on to the new Master Source Volume, the message:

```
** PAUSE STATEMENT COPIED **
```

is logged on the console and an SVC pause is executed. After the operator has entered a Continue command, copying resumes from where the program left off.

For example, to copy the source file '\*\*FILE6' from logical unit 4 to the new Master Source Volume (lu2) and sequence each record beginning with the sequence number 'SF600010', the following command is entered:

```
COPY **FILE6,4,SF600010
```

logical unit 4 is searched beginning from where the volume is currently positioned. The first record of \*\*FILE6 is sequenced with the number 'SF600010'.

The Command:

```
COPY,4
```

would cause copying to occur from logical unit 4 to the new Master Source Volume without resequencing (the original sequence numbers are copied). Copying begins starting a logical unit 4's current position.

#### 4.4 DUPE Command

This command causes files to be continuously copied from one device to another. As with the COPY command, the output device is searched for the end of volume and back-spaced before program duping begins. Note that the EOVS must exist on the output device before issuing this command. After all desired programs have been copied, a new end of volume is written on the output device (logical unit 2) and backspaced over.

The format of the DUPE command is:

```
DUPE (file)(,LU)
```

where 'file' optionally specifies the file label at which copying is to terminate. The indicated file is not copied. If a specific file is not indicated, the duping terminates at the end of the volume. Unless otherwise indicated by 'LU', logical unit 1 is used as the input device.

For example, the command:

```
DUPE **FILE9
```

causes files to be duplicated from the old Master Source Volume (lu1) on to the end of the new Master Source Volume (lu2). Duplicating starts at the current position of the old Master and proceeds up to, but not including, file '\*\*FILE9'.

#### 4.5 LIST Command

This command provides an 80-80 listing of a Source file on logical unit 3. Each page is numbered in the upper right hand corner; and contains up to 56 lines. The list operation is terminated after an END statement is printed or when an EOF or EOVS is detected. The format of this command is:

```
LIST (file)(,LU)
```

where the optional argument 'file' specifies a particular Source File that is to be listed. Otherwise, data is listed from the current position of the input device which is assumed to be logical unit 1 if not changed by the optional 'LU' argument.

The following command:

```
LIST 99,2
```

causes logical unit 2 to be rewound and 99 (decimal) file marks to be skipped. The file immediately following the 99th file mark is then listed on logical unit 3.

#### 4.6 TABLE Command

This command scans the specified logical unit after re-winding it and lists all program labels on logical unit 3. Thus, a table of contents of a Source File containing many labeled programs may be obtained. Program labels are indicated by two asterisk characters in the first two character positions of a record. The format of the command is:

```
TABLE LU
```

where 'LU' indicates the logical unit to be tabled.

#### 4.7 UPDATE Command

This command puts the system in the UPDATE mode and sets up various parameters for the following update commands:

```
INSERT -  
DELETE -  
MODIFY -  
END     - Exit UPDATE mode
```

These commands are legal only when in the UPDATE mode. The format of this command is:

```
UPDATE (LU)(,RESEQ)
```

where 'LU' specified the update logical unit from which new source statements are to be read. If 'LU' is not input, logical unit 5 is assumed to be in the input device. The 'reseq' option indicates whether or not resequencing is to be performed during updating. If an 'N' is input, no resequencing is performed and sequence numbers should then be input with any new statements to maintain sequencing. If anything other than 'N' is input, resequencing is performed. The initial sequence number is read from the first statement available on logical unit 1; therefore, logical unit 1 must be positioned to the desired file before issuing the Update command.

The following command:

```
UPDATE 4
```

puts the system in the UPDATE mode and reads the first statement from logical unit 1 to set the initial value of the sequence number of resequencing. New source statements are to be read from logical unit 4.

#### 4.8 INSERT Command

The INSERT Command is used to insert one or more new statements into an existing source program. The end of the insertion process is indicated by a "/" as the

first two characters of a record read from the update logical unit. The format of this command is:

```
INSERT seqnbr
```

where 'seqnbr' is the sequence number after which the new statements(s) is to be inserted. The new statements are read from the logical unit specified in the UPDATE Command. If the sequence number has not been found by the time an 'END' statement or an end of file is detected, the error message:

```
SEQUENCE NUMBER NOT FOUND
```

is printed on the console and the operation is aborted.

To insert two statements between XXX00150 and XXX00260, the following statements must be entered (the update logical unit is logical unit 5):

```
INSERT XXX00150
- statement #1 -
- statement #2 -
/%
```

#### 4.9 DELETE Command

The DELETE command provides the capability to delete a single statement or multiple statements from a source file contained on logical unit 1. The format of the command is:

```
DELETE seqnr1 (,seqnr2)
```

where 'seqnr1' specifies the sequence number of the first statement to be deleted. If 'seqnr2' is not input, only the statement specified by 'seqnr1' is deleted. If 'seqnr2' is specified, statements between 'seqnr1' and 'seqnr2' inclusively are deleted.

If 'seqnr1' or 'seqnr2' is not found by the time an 'END' statement or an end of file is detected, the error message:

```
FIRST SEQ NBR NOT FOUND
```

for 'seqnr1' or the error message:

```
SECOND SEQNBR NOT FOUND
```

for 'seqnr2' is printed on the console. The operation is aborted when a sequence number is not found.

To delete statements AAA01010 to and including AAA01500, the following command is entered:

```
DELETE AAA01010,AAA01500
```

#### 4.10 MODIFY Command

The MODIFY Command is used to delete a single statement and insert a new statement in its place. The format of the command is:

```
MODIFY seqnbr
```

where 'seqnbr' indicates the sequence number of the statement to be modified. If the sequence number can not be found before an 'END' statement or an end-of-file is detected, the error message:

```
SEQ NBR NOT FOUND
```

is printed on the console.

To use the MODIFY command to delete statement AAA01000 and replace it with a new statement, the following sequence is input (the update logical unit is 5):

```
MODIFY AAA01000  
- new statement -
```

#### 4.11 VERIFY Command

The VERIFY command is used to compare one source file with another. Verification is terminated normally when an "END" statement or end of file is detected. An abnormal termination occurs when a mismatch is detected. When this occurs, the message:

```
** VERIFICATION ERROR **  
- Record 1 -  
- Record 2 -
```

is printed on the List device (LU3) and then the program pauses. Record 1 is the source statement read from 'LU' or logical unit 1 and Record 2 is the statement read from logical unit 2. On continuing, the next records following the error records are read and compared, thus allowing all compare errors in the source records to be found. The format of this command is:

```
VERIFY (LU)(,80)
```

where 'LU' indicates which logical unit is to be used for the primary input. (Default is logical unit 1). The argument '80' indicates whether 72 or 80 character records are to be verified. If 'N' is specified, 72 character records are compared. If anything else is input or nothing in input, 80 character records are compared.

#### NOTE

Although 72- or 80- byte records are the only record sizes available, smaller records can be



compared since verification terminates, at the record level, on detection of a Carriage Return (X'0D') in either one of the input records or after successful verification on the 72nd or 80th character.

#### 4.12 END Command

This command causes the remainder of a file which is being updated to be copied to logical unit 2 and the Update Mode to be terminated. The input is read from logical unit 1 and resequencing is performed as indicated in the UPDATE command.

#### 4.13 WEOV Command

This command causes two file marks to be written on the specified logical unit and then backspaced over. The format of this command is:

WEOV LU

where 'LU' is the logical unit number of the device on which the file-marks are to be written.

#### 4.14 REWIND Command

This command rewinds logical unit 'LU' and is in the form:

REWIND LU

#### 4.15 PAUSE Command

This command causes the Source Updater program to relinquish system control to the Operating System via an SVC Pause.

#### 4.16 EOJ Command

This command causes the Source Updater program to go to End of Job via an SVC EOJ.

### 5. OPERATING INSTRUCTIONS

The following are instructions for loading and executing OS Source Updater:

- a. Load an operating system: BOSS, DOS or RTOS. If RTOS is used, OS Source Updater must be established as an RTOS task. (See RTOS Reference Manual Publication Number 29-240 for details).
- b. OS Source Updater is a relocatable program and may be loaded at any desired BIAS with the OS Resident Loader or the OS Library Loader.

c. Make the following logical unit assignments:

LU 1 - Old Master Source Volume  
LU 2 - New Master Source Volume  
LU 3 - List Device  
LU 5 - Operator Command Input  
LU 6 - Operator Message Output

NOTE: The devices used for maintaining the Source Volume must be able to support rewind, backspace a record, write file-mark, and skip forward to file-mark commands.

d. Start OS Source Updater at its origin. The restart address is also its origin.

# SOURCE UPDATER PROGRAM

## Table of Commands

FIND	file(,LU)
COPY	(file)(,LU)(,seqnbr)
DUPE	(file)(,LU)
VERIFY	(LU)(,80)
LIST	(file)(,LU)
TABLE	LU
UPDATE	(LU)(,reseq)
	(a) INSERT seqnbr
	(b) DELETE seqnr1(,seqnr2)
	(c) MODIFY seqnbr
	(d) END
WEOV	LU
REWIND	LU
PAUSE	
END	

**\*\*NOTE:** Optional arguments are enclosed in parentheses.

\*  
 \* COPYRIGHT INTERDATA, INC. JULY, 1973  
 \*  
 \* AUTHOR: WW HOGUF  
 \*  
 \* PROGRAM USES BASIC MODEL 74 INSTRUCTION SET  
 \*  
 \* THE OS SOURCE UPDATER PROVIDES THE ABILITY TO  
 \* MAINTAIN SOURCE FILES CONTAINED ON A MAGNETIC  
 \* TAPE TYPE DEVICE. COMMANDS ARE ENTERED INTER-  
 \* ACTIVELY THROUGH A CONSOLE DEVICE OR BATCHED  
 \* THROUGH A CARD READER, DISC FILE, ETC.. THE  
 \* FOLLOWING LOGICAL UNIT ASSIGNMENTS ARE REQUIRED:

- \* LU1 - OLD MASTER
- \* LU2 - NEW MASTER
- \* LU3 - LIST
- \* LU5 - OPR COMMAND IN
- \* LU6 - OPR MESSAGE OUT

\* PROGRAM LABEL: SRCUPD

\* REGISTER DEFINITIONS

0000  
 0001  
 0007  
 0008  
 0009  
 000A  
 000B  
 000C  
 000D  
 000E  
 000F

R0 EQU 0  
 SPC EQU 1  
 AC1 EQU 7  
 AC2 EQU 8  
 AC3 EQU 9  
 AC4 EQU 10  
 AC5 EQU 11  
 AC6 EQU 12  
 AC7 EQU 13  
 AC8 EQU 14  
 LNK EQU 15

0800  
 1000

EOF EQU X'0800'  
 EOM EQU X'1000'

0000R CA10  
 0020  
 0004R 0777  
 0006R 4070  
 0056R  
 000AR F110  
 0CA0R  
 000FR F110  
 0CA8R  
 0012R 4800

SRCUPD LHI SPC,x'20'  
 XHR AC1,AC1  
 STH AC1,MODE RESET UPDATE MODE  
 START SVC 1,READY LOG \*READY\* MSG  
 SVC 1,RDCMD READ IN A COMMAND  
 LH R0,RDCMD+2 ANY READ ERRORS ?

*LOC 0464R  
 must be changed  
 from 0C00  
 to 0A00 since  
 (the printer)  
 Centronics 306 has no  
 top of form*

SUD00020  
 SUD00030  
 SUD00040  
 SUD00050  
 SUD00060  
 SUD00070  
 SUD00080  
 SUD00090  
 SUD00100  
 SUD00110  
 SUD00120  
 SUD00130  
 SUD00140  
 SUD00150  
 SUD00160  
 SUD00170  
 SUD00180  
 SUD00190  
 SUD00200  
 SUD00210  
 SUD00220  
 SUD00230  
 SUD00240  
 SUD00250  
 SUD00260  
 SUD00270  
 SUD00280  
 SUD00290  
 SUD00300  
 SUD00310  
 SUD00320  
 SUD00330  
 SUD00340  
 SUD00350  
 SUD00360  
 SUD00370  
 SUD00380  
 SUD00390  
 SUD00400  
 SUD00410  
 SUD00420  
 SUD00430  
 SUD00440  
 SUD00450  
 SUD00460  
 SUD00470  
 SUD00480  
 SUD00490  
 SUD00500  
 SUD00510  
 SUD00520  
 SUD00530

0016R	0CAAR 4230		BNZ	INPERR		SUD00540	
001AR	006AR C870		LHI	AC1,C'		SUD00550	
001ER	2020 4070		STH	AC1,OPCODE	CLEAR COMMAND SAVE AREA	SUD00560	
0022R	0068R 4070		STH	AC1,OPCODE+2		SUD00570	
0026R	006AR 07AA		XHR	AC4,AC4		SUD00580	
0028R	037A	STRT00	LB	AC1,BFR(AC4)	MOVE 1ST 4 CHAR OF COMMAND	SUD00590	
002CR	006ER C570		CLHI	AC1,X'0D'	TO SAVE AREA AND SKIP TO	SUD00600	
0030R	000D 2339		BES	STRT02	FIRST ARGUMENT.	SUD00610	
0032R	C5A0		CLHI	AC4,4		SUD00620	
0036R	0004 2383		BNLS	STRT01		SUD00630	
0038R	027A		STB	AC1,OPCODE(AC4)		SUD00640	
003CR	0068R 26A1	STRT01	AIS	AC4,1		SUD00650	
003ER	0571		CLHR	AC1,SPC		SUD00660	
0040R	203C		BNES	STRT00		SUD00670	
0042R	4870	STRT02	LH	AC1,OPCODE	PICK UP COMMAND	SUD00680	
0046R	0068R 4880		LH	AC2,OPCODE+2		SUD00690	
004AR	006AP 0799		XHR	AC3,AC3		SUD00700	
004CR	4579	STRT03	CLH	AC1,CMDTAB(AC3)	SEARCH COMMAND TABLE	SUD00710	
0050R	0C46R 2135		BNES	STRT04		SUD00720	
0052R	4589		CLH	AC2,CMDTAB+2(AC3)		SUD00730	
0056R	0C48R 4330		BE	STRT05	IF EQUAL THIS IS IT	SUD00740	
005AR	007AR 2696	STRT04	AIS	AC3,6	INCR CMDTAB INDEX	SUD00750	
005CR	C590		CLHI	AC3,CTEND-CMDTAB	END OF TABLE ?	SUD00760	
0060R	0059 20AA		BLS	STRT03	NO. CONTINUE SEARCH	SUD00770	
0062R	F120	CMDERR	SVC	2,I LCMMSG	YES. LOG 'ILLFGAL COMMAND' MSG	SUD00780	
0066R	0CF0R 4300		B	START	THEN READ NEXT COMMAND	SUD00790	
	000AR					SUD00800	
						SUD00810	
006AR	F120		INPERR	SVC	2,CONV02	CONVERT ERROR STATUS TO ASCII	SUD00820
006ER	0D2ER F120		SVC	2,LOGE02	AND LOG ERROR MSG	SUD00830	
0072R	0D32R F120		SVC	2,PAUSE		SUD00840	
0076R	0CD8R 4300		B	START		SUD00850	
	000AR					SUD00860	
						SUD00870	
007AR	48F9	STRT05	LH	LNK,CMDTAB+4(AC3)	FETCH A(SERVICE ROUTINE	SUD00880	

007FR	0C4AR 48FF 0000	PARSER	LH	AC8.0(LNK)	FETCH A (CONTROL WORD)	SUD00890
0082R	26F2		AIS	LNK.2		SUD00900
0084R	C5F0 FFFF		CLHI	ACA.-1	-1 INDICATES NO ARGUMENTS	SUD00910
0088R	033F		RER	LNK		SUD00920
008AR	48DE 0000		LH	AC7.0(AC8)	PICK UP CONTROL WORD	SUD00930
008ER	C4D0 3FFF		NHI	AC7.X'3FFF'	RESET FLAGS	SUD00940
0092R	93CD		LBR	AC6.AC7	EXTRACT COUNT	SUD00950
0094R	C3D0 0200		THI	AC7.X'200'	HEXIDECIMAL ARGUMENT ?	SUD00960
0098R	4230		BNZ	HFX		SUD00970
00F8R	00F8R					
009CR	C3D0 0500		THI	AC7.X'500'	ASCII ARGUMENT ?	SUD00980
00A0R	4330 007ER		BZ	PARSER	IF NOT, IGNORE IT	SUD00990
		*				SUD01000
		*	ASCII ARGUMENT			SUD01010
		*				SUD01020
00A4R	C890 0020		LHI	AC3.C' '		SUD01030
00A8R	088E		LHR	AC2.AC8		SUD01040
00AAR	D298 0002	ASC01	STR	AC3.2(AC2)	FILL BFR WITH BLANKS	SUD01050
00AER	2681		AIS	AC2.1		SUD01060
00B0R	27C1		SIS	AC6.1	DECREMENT COUNT	SUD01070
00B2R	2034		BNZS	ASC01		SUD01080
00B4R	088E		LHR	AC2.AC8		SUD01090
00B6R	93CD		LBR	AC6.AC7		SUD01100
00BAR	C3D0 0400		THI	AC7.X'400'	RIGHT OR LEFT JUSTIFIED ?	SUD01110
00BCR	4330 00D4R		BZ	ASCRT		SUD01120
		*				SUD01130
		*	LEFT-JUSTIFIED ASCII			SUD01140
		*				SUD01150
00C0R	4170 0146R	ASCLFT	BAL	AC1.GETCHR	GRAB A CHAR FROM STRING	SUD01160
00C4R	C6D0 4000		OHI	AC7.X'4000'	ON RETURN, SET PARAM FND	SUD01170
00CAR	27C1		SIS	AC6.1	DECR COUNT	SUD01180
00CAR	2015		BMS	ASCLFT	IF MINUS, SKIP TO DELIMITER	SUD01190
00CCR	D298 0002		STR	AC3.2(AC2)	STORE CHAR IN BFR	SUD01200
00D0R	2681		AIS	AC2.1		SUD01210
00D2R	2209		BS	ASCLFT		SUD01220
		*				SUD01230
		*	RIGHT-JUSTIFIED ASCII			SUD01240
		*				SUD01250
00D4R	4170 0146R	ASCRT	BAL	AC1.GETCHR	FETCH A CHAR FROM STRING	SUD01260
00D8R	C6D0		OHI	AC7.X'4000'	SET FOUND FLAG	SUD01270

4000						
00DCR	CA8C		AHI	AC2,-1(AC6)		SUD01280
	FFFF					
00FOR	D378	ASCRT1	LB	AC1,2(AC2)	MOVF ARGUMENT LEFT AND	SUD01290
	0002					
00E4R	D298		STB	AC3,2(AC2)	INSERT NEW CHAR.	SUD01300
	0002					
00F8R	0897		LHR	AC3,AC1	MAKE OLD CHAR THE NEW CHAR	SUD01310
00EAR	2781		SIS	AC2,1		SUD01320
00ECR	27C1		SIS	AC6,1		SUD01330
00EER	2027		BPS	ASCRT1		SUD01340
00FOR	93CD		LBR	AC6,AC7	REINITIALIZE POINTERS	SUD01350
00F2R	088F		LHR	AC2,AC8		SUD01360
00F4R	4300		B	ASCRT	GO FETCH NEXT CHAR	SUD01370
	00D4R					
		*				SUD01380
		*	HEXIDECIMAL ARGUMENT			SUD01390
		*				SUD01400
00FAR	078B	HEX	XHR	AC5,AC5	CLEAR RESULT	SUD01410
00FAR	2581		LCS	AC2,1		SUD01420
00FCR	C5C0		CLHI	AC6,4	CALCULATE MASK VALUE	SUD01430
	0004					
0100R	2386		BNLS	HEX01		SUD01440
0102R	91C2		SLLS	AC6,2	MULTIPLY COUNT BY 4	SUD01450
0104R	CD8C		SLHL	AC2,0(AC6)		SUD01460
	0000					
0108R	C780		XHI	AC2,-1	COMPLEMENT MASK	SUD01470
	FFFF					
010CR	4170	HEX01	BAL	AC1,GETCHR	FETCH A CHAR FROM STRING	SUD01480
	0146R					
0110R	C590		CLHI	AC3,C*0'	INSURE CHAR IS A VALID HEX	SUD01490
	0030					
0114R	4280		BL	ERROR	CHAR. IF NOT, EXIT STAGE	SUD01500
	018AR					
0118R	C590		CLHI	AC3,C*G'	RIGHT.	SUD01510
	0047					
011CR	4380		BNL	ERROR		SUD01520
	018AR					
0120R	C590		CLHI	AC3,X*3A'		SUD01530
	003A					
0124R	2186		BLS	HEX02		SUD01540
0126R	C590		CLHI	AC3,C*A'		SUD01550
	0041					
012AR	4280		BL	ERROR		SUD01560
	018AR					
012ER	2797		SIS	AC3,7		SUD01570
0130R	C490	HEX02	NHI	AC3,X*0F'	MASK TO 4-BITS	SUD01580
	000F					
0134R	9184		SLLS	AC5,4	MAKE ROOM FOR NEXT DIGIT	SUD01590
0136R	0689		OHR	AC5,AC3	INSERT IT IN HOLE	SUD01600
0138R	0488		NHR	AC5,AC2	MASK TO DESIRED VALUE	SUD01610
013AR	C6D0		OHI	AC7,X*4000'	SET PARAMETER END FLAG	SUD01620
	4000					
013ER	408E		STH	AC5,2(AC8)	SAVE IN PARAMETER BLOCK	SUD01630
	0002					
0142R	4300		B	HEX01		SUD01640

010CR

\*  
 \* THIS ROUTINE FETCHES A CHAR FROM THE CHAR STRING  
 \* CHECKS IT. IF IT IS A DELIMITER, THE TERMINATION  
 \* ROUTINE IS ENTER. IF IT IS A SPACE, IT IS IGNORED.  
 \* OTHERWISE, THE CAR IS RETURNED TO THE CALLER.  
 \*

SUD01650  
 SUD01660  
 SUD01670  
 SUD01680  
 SUD01690  
 SUD01700  
 SUD01710

0146R D39A  
 0D6ER  
 014AR 26A1  
 014CR C590  
 0020  
 0150R 2235  
 0152R C590  
 002C  
 0156R 233R  
 0158R C590  
 000D  
 015CR 0237

GETCHR LB AC3,RFR(AC4) PICK UP A CHAR  
 AIS AC4.1  
 CLHI AC3,C' ' IS IT A SPACE  
 HES GETCHR IF SO, IGNORE IT  
 CLHI AC3,C',, IS IT A COMMA ?  
 HES END1 IF SO, GO TO TERMINATION  
 CLHI AC3,X'0D', HOW ABOUT A CAR RTN ?  
 BNER AC1 IF NOT, GIVE CHAR TO CALLER

SUD01720  
 SUD01730  
 SUD01740  
 SUD01750  
 SUD01760  
 SUD01770

\*  
 \*  
 \* TERMINATION  
 \*

SUD01780  
 SUD01790  
 SUD01800  
 SUD01810  
 SUD01820  
 SUD01830

015ER C3D0  
 3000  
 0162R 4330  
 018AR  
 0166R C6D0  
 8000  
 016AR 27A1  
 016CR 40DE  
 0000  
 0170R C3D0  
 0800  
 0174R 2135  
 0176R C3D0  
 4000  
 017AR 4330  
 018AR  
 017ER C3D0  
 2000  
 0182R 4330  
 007ER  
 0186R 08DD  
 0188R 021F  
 018AR F120  
 0CF4R  
 018ER 4300  
 000AR

THI AC7,X'3000' COULD THIS BE THE END ?  
 BZ ERROR  
 OHI AC7,X'8000' SET END-OF-STRING FLAG  
 END1 SIS AC4.1  
 STH AC7,0(ACA) BACKUP PTR TO CAR RTN  
 RESTORE CONTROL WORD  
 THI AC7,X'0800' IS THIS A OPTIONAL ARG ?  
 HN2S END2  
 THI AC7,X'4000' IF NOT, WAS IT FOUND ?  
 BZ ERROR NO, THEN IT IS F\*R\*R\*O\*R  
 END2 THI AC7,X'2000' END-OF-PARAMETER LIST ?  
 BZ PARSER NO, GO GET NEXT PARAMETER  
 LHR AC7,AC7 YES, SHOULD BE STRING END  
 RMR LNK IF MINUS, AOK  
 SVC 2,ARGERR LOG ERROR MSG  
 B START

SUD01840  
 SUD01850  
 SUD01860  
 SUD01870  
 SUD01880  
 SUD01890  
 SUD01900  
 SUD01910  
 SUD01920  
 SUD01930  
 SUD01940  
 SUD01950  
 SUD01960  
 SUD01970

\*  
 \* FIND LABEL(.LU)  
 \*

0192R 01F8R  
 0194R 0202R  
 0196R 4870  
 0056R

FNDLAB DC LABEL8  
 DC LABEL9  
 LH AC1,MODE

WHAT MODE ARE WE IN ?

SUD01980  
 SUD01990  
 SUD02000  
 SUD02010  
 SUD02020  
 SUD02030



019AR 4230		RNZ	CMDERR	IF UPDATF MODE. ERROR	SUD02040
0062R					
019FR 2481		LIS	AC2.1		SUD02050
01A0R 4870		LH	AC1,LAREL9		SUD02060
0202R					
01A4R 0A77		AHR	AC1,AC1		SUD02070
01A6R 2313		BNMS	LABEL		SUD02080
01A8R 0380		LB	AC2,LAREL9+3		SUD02090
0205R					
01ACR 0280	LABEL	STR	AC2,READ+1		SUD02100
0CB1R					
01B0R 0280		STR	AC2,BKSP+1		SUD02110
0CC9R					
01B4R 0280		STR	AC2,FNDEOF+1		SUD02120
0CC0R					
01B8R 0280		STB	AC2,REW+1		SUD02130
0CD5R					
01BCR 41F0		BAL	AC8,SEARCH	FIND DESIRED PROGRAM	SUD02140
0AD8R					
01C0R 01FAR		DC	LAREL8+2		SUD02150
01C2R 4300		R	NOTFND		SUD02160
01E4R					
01C6R 41F0		BAL	LNK,IOSURR		SUD02170
09C0R					
01CAR 0CC8R		DC	BKSP		SUD02180
01CCR E110		SVC	1.FNDMSG	LOG 'FOUND' MSG	SUD02190
0D06R					
01D0R 4800		LH	R0,FNDMSG+2	ANY PROBLEMS ?	SUD02200
0D08R					
01D4R 4330		BZ	START		SUD02210
000AR					
01D8R E120		SVC	2.CONV01	CONVERT STATUS TO ASCII	SUD02220
0D16R					
01DCR E120		SVC	2.LOGED01	LOG ERROR MSG	SUD02230
0D1AR					
01E0R 4300		R	START	GET NEXT COMMAND	SUD02240
000AR					
					SUD02250
	*				SUD02260
	*				SUD02270
01E4R 41F0	NOTFND	BAL	LNK,IOSURR	BACKSPACE OVER EOF MARKER	SUD02280
09C0R					
01F8R 0CC8R		DC	BKSP		SUD02290
01EAR 41F0		BAL	LNK,IOSURR		SUD02300
09C0R					
01EER 0CC8R		DC	BKSP		SUD02310
01FOR F120		SVC	2.NFMSG	LOG 'NOT FOUND' MSG	SUD02320
0D48R					
01F4R 4300		R	START		SUD02330
000AR					
	*				SUD02340
	*				SUD02350
01F8R 1408	LABEL8	DC	X'1408'		SUD02360
01FAR		DS	8		SUD02370
0202R 2A01	LABEL9	DC	X'2A01'.0		SUD02380
0000					
	*				

		* COPY (LABEL)(.IN)(.NEWSEQ)			
					SUD02390
		* COPY	DC	A(COPLAB)	SUD02400
0206R	020CR		DC	A(COPIN)	SUD02410
0208R	02C6R		DC	A(NEWSEQ)	SUD02420
020AR	02CAR		LH	AC1.MODE	SUD02430
020CR	4870			WHAT MODE ARE WF TN 7	SUD02440
	0036R				
0210R	4230		BN7	CMDERR	SUD02450
	0062R			IF UPDATE MODE, ERROR	
0214R	2472		LIS	AC1,2	SUD02460
0216R	0270		STR	AC1,BKSP+1	SUD02470
	0CC9R				
021AR	0270		STR	AC1,FNDEOF+1	SUD02480
	0CC0R				
021ER	0270		STR	AC1,READ+1	SUD02490
	0CB1R				
0222R	41F0	COPY00	BAL	LNK,IOSUR	SUD02500
	09C0R			SEARCH FOR END OF VOLUME	
0226R	0CCCR		DC	FNDEOF	SUD02510
0228R	41F0		BAL	LNK,IOSURR	SUD02520
	09C0R				
022CR	0CB0R		DC	READ	SUD02530
022ER	2236		BZS	COPY00	SUD02540
0230R	41F0		BAL	LNK,IOSURR	SUD02550
	09C0R			BACKSPACE OVER IT	
0234R	0CC8R		DC	BKSP	SUD02560
0236R	2481		LIS	AC2,1	SUD02570
0238R	4870		LH	AC1,COPIN	SUD02580
	02C6R				
023CR	0A77		AHR	AC1,AC1	SUD02590
023ER	2313		BNMS	COPY01	SUD02600
0240R	0380		LB	AC2,COPIN+3	SUD02610
	02C9R				
0244R	0280	COPY01	STB	AC2,READ+1	SUD02620
	0CB1R				
0248R	0280		STB	AC2,FNDEOF+1	SUD02630
	0CC0R				
024CR	0280		STB	AC2,REW+1	SUD02640
	0CD5R				
0250R	0280		STB	AC2,BKSP+1	SUD02650
	0CC9R				
0254R	4870		LH	AC1,COPLAB	SUD02660
	02BCR				
0258R	0A77		AHR	AC1,AC1	SUD02670
025AR	4310		BNM	COPY20	SUD02680
	026AR				
025ER	41F0		BAL	AC8,SEARCH	SUD02690
	0AD8R			SEARCH FOR DESIRED PROGRAM	
0262R	028ER		DC	A(COPLAB)+2	SUD02700
0264R	4300		B	NOTFND	SUD02710
	01F4R				
0268R	2306		BS	COPY21	SUD02720
					SUD02730
					SUD02740
					SUD02750
		* COPY20	BAL	LNK,IOSURR	
026AR	41F0				
	09C0R				

026ER	0CR0R		DC	READ		SUD02760
0270R	4230		BNZ	COPY50		SUD02770
	02R2R					
0274R	4870	COPY21	LH	AC1,NEWSFQ		SUD02780
	02CAR					
0278R	0A77		AHR	AC1,AC1	WAS A SEQUENCE ARG INPUT ?	SUD02790
027AR	4310		RNM	COPY40		SUD02800
	0288R					
027ER	41F0		BAL	LNK,RESEQ		SUD02810
	0AA0R					
0282R	41F0		BAL	LNK,DECINC	INCREMENT SEQ NBR BY TEN	SUD02820
	0C1AR					
0286R	0202R		DC	NEWSEQ+8		SUD02830
0288R	41F0	COPY40	BAL	LNK,IOSURR		SUD02840
	09C0R					
028CR	0CC0R		DC	WRITE		SUD02850
028FR	41F0		BAL	LNK,EXTRCT	IS THIS A 'PAUSE' STMT ?	SUD02860
	08B4R					
0292R	5041		DC	C'PAUSE '		SUD02870
	5553					
	4520					
0298R	4300		B	COPY45	NO. CHECK FOR AN 'END' STMT	SUD02880
	02A4R					
029CR	F120		SVC	2,PSMSG	YES. LOG OPERATOR MSG	SUD02890
	0204R					
02A0R	F120		SVC	2,PAUSE	AND PAUSE	SUD02900
	0CC0R					
02A4R	41F0	COPY45	BAL	LNK,EXTRCT		SUD02910
	08B4R					
02A8R	454E		DC	C'END '		SUD02920
	4420					
	2020					
02AER	4300		B	COPY20		SUD02930
	026AR					
02B2R	2472	COPY50	LIS	AC1,2	FETCH LU OF NEW FILE	SUD02940
02B4R	41E0		BAL	AC8,WFMS	WRITE AN 'EOF' AND BACKSPACE	SUD02950
	096AR					
02B8R	4300		B	START		SUD02960
	000AR					
						SUD02970
						SUD02980
02BCR	1C08	COPLAB	DC	X'1C08'		SUD02990
02BER			DS	8		SUD03000
02C6R	1A01	COPIN	DC	X'1A01',0		SUD03010
	0000					
02CAR	2908	NEWSEQ	DC	X'2908'		SUD03020
02CCR			DS	8		SUD03030
02D4R	0007	PSMSG	DC	7,28,C'*** PAUSE STATEMENT COPIED ***		SUD03040
	001C					
	2A2A					
	2050					
	4155					
	5345					
	2053					
	5441					
	5445					

4D45  
4E54  
2043  
4F50  
4945  
4420  
2A2A

		*	DUPE (LABEL)(.IN)			
		*				
		*				
02F4R	0394R	DUPF	DC	A(DUPLAB)		SUD03050
02F6R	039ER		DC	A(DUPIN)		SUD03060
02F8R	4870		LH	AC1,MODE	WHAT MODE ARE WE IN ?	SUD03070
	0056R					SUD03080
02FCR	4230		RNZ	CMOERR	IF UPDATE MODF. ERROR	SUD03090
	0062R					SUD03100
0300R	2472		LIS	AC1,2		SUD03110
0302R	0270		STB	AC1,READ+1		SUD03120
	0CR1R					SUD03130
0306R	0270		STB	AC1,WRTEOF+1		SUD03140
	0CD1R					SUD03150
030AR	0270		STB	AC1,BKSP+1		SUD03160
	0CC9R					SUD03170
030ER	0270		STB	AC1,FNDEOF+1		SUD03180
	0CCDR					SUD03190
0312R	41F0	DUP000	BAL	LNK,IOSUBR	SEARCH FOR END OF VOLUME	SUD03200
	09COR					SUD03210
0316R	0CCCR		DC	FNDEOF		SUD03220
0318R	41F0		BAL	LNK,IOSUBR		SUD03230
	09COR					SUD03240
031CR	0CR0R		DC	READ		SUD03250
031ER	2236		BZS	DUP000		SUD03260
0320R	41F0		BAL	LNK,IOSUBR	BACKSPACE OVER IT	SUD03270
	09COR					SUD03280
0324R	0CC8R		DC	BKSP		SUD03290
0326R	41F0		BAL	LNK,IOSUBR		SUD03300
	09COR					SUD03310
032AR	0CC8R		DC	BKSP		SUD03320
032CR	2481		LIS	AC2,1		SUD03330
032ER	4870		LH	AC1,DUPIN		SUD03340
	039ER					SUD03350
0332R	0A77		AHR	AC1,AC1		SUD03360
0334R	2313		BNMS	DUP005		SUD03370
0336R	0380		LB	AC2,DUPIN+3		SUD03380
	03A1R					SUD03390
033AR	0280	DUP005	STB	AC2,READ+1		SUD03400
	0CR1R					SUD03410
033ER	0280		STB	AC2,BKSP+1		SUD03420
	0CC9R					SUD03430
0342R	41F0	DUP010	BAL	LNK,IOSUBR		SUD03440
	09COR					SUD03450
0346R	0CR0R		DC	READ		SUD03460
0348R	2339		BZS	DUP015		SUD03470
034AR	41F0		BAL	LNK,IOSUBR		SUD03480
	09COR					SUD03490
034ER	0CD0R		DC	WRTEOF		SUD03500

0350R	41F0		BAL	LNK.IOSURR		SUD03380
	09C0R					
0354R	0CB0R		DC	READ		SUD03390
0356R	4230		BNZ	DUP030		SUD03400
	0384R					
035AR	4870	DUP015	LH	AC1.DUPLAB		SUD03410
	0394R					
035ER	0A77		AHR	AC1.AC1		SUD03420
0360R	4310		BNM	DUP020		SUD03430
	037AR					
0364R	C880		LHI	AC2.DUPLAB+2	FETCH ADDR OF DESIRED LABFL	SUD03440
	0396R					
0368R	41F0		BAL	LNK.LABCHK	DOES THIS RECORD EQUAL IT ?	SUD03450
	0B60R					
036CR	4300		B	DUP020	NOT FOUND ?	SUD03460
	037AR					
0370R	41F0		BAL	LNK.IOSURR	FOUND, BACKSPACE OVER IT	SUD03470
	09C0R					
0374R	0CC0R		DC	BKSP		SUD03480
0376R	4300		B	DUP099		SUD03490
	038AR					
		*				SUD03500
037AR	41F0	DUP020	BAL	LNK.IOSURR		SUD03510
	09C0R					
037ER	0CC0R		DC	WRITE		SUD03520
0380R	4300		B	DUP010		SUD03530
	0342R					
		*				SUD03540
		*				SUD03550
		*				SUD03560
0384R	41F0	DUP030	BAL	LNK.IOSURR	BACKSPACE OVER EOF MARK	SUD03570
	09C0R					
038AR	0CC0R		DC	BKSP		SUD03580
038AR	2472	DUP099	LIS	AC1.2		SUD03590
038CR	41F0		BAL	ACR.WFMS	WRITE AN EOF ON NEW MASTER	SUD03600
	096AR					
0390R	4300		B	START		SUD03610
	000AR					
		*				SUD03620
		*				SUD03630
		*				SUD03640
0394R	1C08	DUPLAB	DC	X'1C08'		SUD03650
0396R			DS	8		SUD03660
039ER	2A01	DUPIN	DC	X'2A01'.0		SUD03670
	0000					
		*				SUD03680
		*	TABLE LU			SUD03690
		*				SUD03700
03A2R	0436R	TABLE	DC	TABLU		SUD03710
03A4R	4870		LH	AC1.MODE	WHAT MODE ARE WE IN ?	SUD03720
	0D56R					
03A8R	4230		BNZ	CMDERR	IF UPDATE MODE. ERROR	SUD03730
	0062R					
03ACR	0370		LB	AC1.TABLU+3		SUD03740
	0439R					
03B0R	D270		STR	AC1.READ+1		SUD03750

0CB1R					
03B4R	D270	STB	AC1.REW+1		SUD03760
	0CD5R				
03B8R	D270	STB	AC1.BKSP+1		SUD03770
	0CC9R				
03BCR	41F0	BAL	LNK.IOSURR		SUD03780
	09C0R				
03C0R	0CD4R	DC	REW		SUD03790
03C2R	F110	SVC	1.PRTHDR		SUD03800
	043AR				
03C6R	4800	LH	RO.PRTHDR+2		SUD03810
	043CR				
03CAR	4230	BNZ	TAB045		SUD03820
	041AR				
03CFR	41F0	TAB010	BAL	LNK.IOSURR	SUD03830
	09C0R				
03D2R	0CR0R	DC	READ		SUD03840
03D4R	2336	BZS	TAB011		SUD03850
03D6R	41F0	BAL	LNK.IOSURR	IS THIS THE END OF VOLUME ?	SUD03860
	09C0R				
03DAR	0CB0R	DC	READ		SUD03870
03DCR	4230	BNZ	TAB050	YES. END OF OPERATION	SUD03880
	0426R				
03FOR	4870	TAB011	LH	AC1.BFR	IS THIS A LABEL RECORD ?
	0D6ER				SUD03890
03E4R	C570	CLHI	AC1.C***		SUD03900
	2A2A				
03E8R	4230	BNE	TAB010	NO. KEEP SEARCHING	SUD03910
	03CFR				
03ECR	2498	LIS	AC3.8	YES. PRINT IT	SUD03920
03EER	D219	TAB015	STB	SPC,LARSAV-1(AC3)	SUD03930
	0D5FR				
03F2R	2791	SIS	AC3.1		SUD03940
03F4R	2033	BNZS	TAB015		SUD03950
03F6R	D379	TAB030	LB	AC1.BFR(AC3)	SUD03960
	0D6ER				
03FAR	C570	CLHI	AC1.X'00'		SUD03970
	000D				
03FER	2338	BES	TAB040		SUD03980
0400R	D279	STB	AC1,LARSAV(AC3)		SUD03990
	0D60R				
0404R	2691	AIS	AC3.1		SUD04000
0406R	C590	CLHI	AC3.8	FINISHED MOVING THE LABEL ?	SUD04010
	0008				
040AR	4280	BL	TAB030		SUD04020
	03F6R				
040ER	F110	TAB040	SVC	1.PRTLAR	SUD04030
	0442R				
0412R	4800	LH	RO.PRTLAR+2		SUD04040
	0444R				
0416R	4330	BZ	TAB010		SUD04050
	03CER				
041AR	E120	TAB045	SVC	2.TAB910	SUD04060
	044AR				
041ER	E120	SVC	2.TAB911		SUD04070
	044ER				

0422R	4300		B	START		SUD04080
	000AR					
						SUD04090
						SUD04100
						SUD04110
0426R	41F0	TAB050	BAL	LNK.IOSURR		SUD04120
	09C0R					
042AR	0CC8R		DC	BKSP		SUD04130
042CR	41F0		BAL	LNK.IOSURR		SUD04140
	09C0R					
0430R	0CCAR		DC	BKSP		SUD04150
0432R	4300		B	START		SUD04160
	000AR					
						SUD04170
						SUD04180
0436R	2201	TABLU	DC	X'2201',0		SUD04190
	0000					
043AR	2803	PRTHDR	DC	X'2803',0.TABHDR,HDREND		SUD04200
	0000					
	0464R					
	0479R					
0442R	2803	PRTLAB	DC	X'2803',0.LABSAV,LARSAV+7		SUD04210
	0000					
	0D60R					
	0D67R					
044AR	0006	TAB910	DC	6,TAB911+18		SUD04220
	0460R					
044ER	0007	TAB911	DC	7.18.C'PRINT ERROR - XXXX'		SUD04230
	0012					
	5052					
	494E					
	5420					
	4552					
	524F					
	5220					
	2D20					
	5858					
	5858					
0464R	0C00	TABHDR	DC	X'0C00',0,0,0,0,C' PROGRAMS: '		SUD04240
	0000					
	0000					
	0000					
	0000					
	2020					
	5052					
	4F47					
	5241					
	4D53					
	3A20					
0479R		HDREND	EQU	*-1		SUD04250
						SUD04260
						SUD04270
						SUD04280
047AR	0518R	LIST	DC	LIST90		SUD04290
047CR	0522R		DC	LIST91		SUD04300
047FR	4870		LH	AC1,MODE	WHAT MODE ARE WE IN ?	SUD04310

0056R					
04A2R 4230		BNZ	CMDERR	IF UPDATE MODE. ERROR	SUD04320
0062R					
04A6R 24A1		LIS	AC2.1		SUD04330
04A8R 4A70		LH	AC1.1 LIST91		SUD04340
0522R					
048CR 0A77		AHR	AC1.AC1		SUD04350
04AFR 2313		BNMS	LIST01		SUD04360
0490R 0380		LB	AC2,LIST91+3		SUD04370
0525R					
0494R 0280	LIST01	STB	AC2.READ+1		SUD04380
0CB1R					
0498R 0280		STB	AC2,BKSP+1		SUD04390
0CC9R					
049CR C870		LHI	AC1.X'3030'		SUD04400
3030					
04A0R 4070		STH	AC1,PAGNBR		SUD04410
058ER					
04A4R 4070		STH	AC1,PAGNBR+2		SUD04420
0590R					
04A8R 078B		XHR	AC5.AC5		SUD04430
04AAR 4870		LH	AC1,LIST90		SUD04440
0518R					
04AER 0A77		AHR	AC1.AC1		SUD04450
04B0R 4310		BNM	LIST20		SUD04460
04C2R					
04R4R 41F0		BAL	AC8,SEARCH	FIND THE REQUESTED PROGRAM	SUD04470
0AD8R					
04R8R 051AR		DC	LIST90+2		SUD04480
04RAR 4300		B	NOTFND		SUD04490
01E4R					
04BER 4300		B	LIST30		SUD04500
04D2R					
	*				SUD04510
	*				SUD04520
04C2R 41F0	LIST20	BAL	LNK,IOSUBR		SUD04530
09C0R					
04C6R 0CB0R		DC	READ		SUD04540
04C8R 4230		BNZ	LIST50		SUD04550
0500R					
04CCR 27R1		SIS	AC5.1		SUD04560
04CER 4220		BP	LIST40		SUD04570
04E8R					
04D2R 41F0	LIST30	BAL	LNK,DECINC		SUD04580
0C1AR					
04D6R 0591R		DC	PAGNBR+3		SUD04590
04D8R 41F0		BAL	LNK,IOSUBR	OUTPUT TOP-OF-FORM SEQ AND	SUD04600
09C0R					
04DCR 0526R		DC	LIST93		SUD04610
04DER 41F0		BAL	LNK,IOSUBR	PRINT PAGE NBR AT TOP-OF-PAGE	SUD04620
09C0R					
04E2R 052ER		DC	LIST94		SUD04630
04E4R C8R0		LHI	AC5.56	INITIALIZE LINE COUNTER	SUD04640
0038					
04EAR 41F0	LIST40	BAL	LNK,IOSUBR		SUD04650
09C0R					



04ECR 0536R	DC	LIST95		SUD04660
04EER 41F0	BAL	LNK,EXTRCT		SUD04670
04F2R 454E	DC	C*END	'	SUD04680
4420				
2020				SUD04690
04F8R 4300	B	LIST20		SUD04700
04FCR 4300	B	START		SUD04710
000AR				SUD04720
				SUD04730
0500R 41F0	LIST50	BAL	LNK,IOSURR	SUD04740
09COR				SUD04750
0504R 0CR0R	DC	READ		SUD04760
0506R 2334	BZS	LIST51		
0508R 41F0	BAL	LNK,IOSURR		SUD04770
09COR				SUD04780
050CR 0CC8R	DC	BKSP		
050ER 41F0	LIST51	BAL	LNK,IOSURR	SUD04790
09COR				SUD04800
0512R 0CC8R	DC	BKSP		
0514R 4300	B	START		SUD04810
000AR				SUD04820
				SUD04830
0518R 1C08	LIST90	DC	X*1C08*	SUD04840
051AR		DS	8	SUD04850
0522R 2A01	LIST91	DC	X*2A01*.0	
0000				SUD04860
0526R 2803	LIST93	DC	X*2803*.0,LIST96,LIST97	
0000				
053ER 0547R	LIST94	DC	X*2803*.0,LIST98,LIST99	SUD04870
2803				
0000				
0548R 0593R	LIST95	DC	X*2803*.0,A(BFR),A(BFR+79)	SUD04880
2803				
0000				
053ER 0C00	LIST96	DC	X*0C00*.0,0,0,0	SUD04890
0000				
0000				
0000				
0000				SUD04900
0547R	LIST97	EQU	*-1	SUD04910
0548R	LIST98	DO	32	SUD04920
0548R 2020		DC	C* *	SUD04920
054AR 2020		DC	C* *	SUD04920
054CR 2020		DC	C* *	SUD04920
054ER 2020		DC	C* *	SUD04920
0550R 2020		DC	C* *	SUD04920
0552R 2020		DC	C* *	SUD04920
0554R 2020		DC	C* *	

0556R 2020	DC	C* *		SUD04920
0558R 2020	DC	C* *		SUD04920
055AR 2020	DC	C* *		SUD04920
055CR 2020	DC	C* *		SUD04920
055ER 2020	DC	C* *		SUD04920
0560R 2020	DC	C* *		SUD04920
0562R 2020	DC	C* *		SUD04920
0564R 2020	DC	C* *		SUD04920
0566R 2020	DC	C* *		SUD04920
0568R 2020	DC	C* *		SUD04920
056AR 2020	DC	C* *		SUD04920
056CR 2020	DC	C* *		SUD04920
056ER 2020	DC	C* *		SUD04920
0570R 2020	DC	C* *		SUD04920
0572R 2020	DC	C* *		SUD04920
0574R 2020	DC	C* *		SUD04920
0576R 2020	DC	C* *		SUD04920
0578R 2020	DC	C* *		SUD04920
057AR 2020	DC	C* *		SUD04920
057CR 2020	DC	C* *		SUD04920
057ER 2020	DC	C* *		SUD04920
0580R 2020	DC	C* *		SUD04920
0582R 2020	DC	C* *		SUD04920
0584R 2020	DC	C* *		SUD04920
0586R 2020	DC	C* *		SUD04920
0588R 2050	DC	C* PAGE *		SUD04930
4147				
4520				
058ER 3030	PAGNBR DC	C*0000*.X*0A0D*		SUD04940
3030				
0A0D				
0593R	LIST99 EQU	*-1		SUD04950
	*			SUD04960
	* VERIFY (LU)(.NBR)			SUD04970
	*			SUD04980
0594R 062CR	VERIFY DC	VER900		SUD04990
0596R 0630R		DC VER901		SUD05000
0598R 2481		LIS AC2.1		SUD05010
059AR 4870		LH AC1.VER900		SUD05020
062CR				
059ER 0A77	AHR	AC1.AC1	ALTERNATE LOGICAL UNIT INPUT	SUD05030
05A0R 2313	BNMS	VER010	NO. USE LU 1	SUD05040
05A2R 0380	LB	AC2.VER900+3	YES. PICK IT UP	SUD05050
062FR				
05A6R 0280	VER010 STB	AC2.READ+1	STORE LU IN SVC PAR BLK	SUD05060
0CB1R				
05AAR 0880	LHI	AC2.80		SUD05070
0050				
05AER 4870	LH	AC1.VER901	VERIFY LIMIT 80 OR 72 ?	SUD05080
0630R				
05B2R 0A77	AHR	AC1.AC1		SUD05090
05B4R 2318	BNMS	VER020		SUD05100
05B6R 0370	LB	AC1.VER901+2	CHECK INPUT CHAR	SUD05110
0632R				
05BAR 0570	CLHI	AC1.C*N*	WAS IT *NO* ?	SUD05120
004E				

05BER	2133		BNES	VER020		NO, ASSUME 'YES', USE 80	SUD05130
05COR	C880		LHI	AC2,72		IT WAS 'NO', USE 72	SUD05140
	0048						
05C4R	4080	VER020	STH	AC2,VERCNT		STORE LIMIT IN VERCNT	SUD05150
	0634R						
05C8R	41F0	VER030	BAL	LNK,IOSUBR		READ A RECORD TO BE VERIFIED	SUD05160
	09C0R						
05CCR	0636R		DC	VERINP			SUD05170
05CFR	4230		BNZ	START		ERROR IF FOM OR EOF FOUND	SUD05180
	000AR						
05D2R	41F0		BAL	LNK,IOSUBR		READ A MASTER RECORD	SUD05190
	09C0R						
05D6R	0C80R		DC	READ			SUD05200
05D8R	4230		BNZ	START		ERROR IF FOM OR EOF FOUND	SUD05210
	000AR						
05DCR	0799		XHR	AC3,AC3		CLEAR INDEX REG	SUD05220
05DER	D379	VER040	LB	AC1,BFR(AC3)		FETCH A CHAR FROM MASTER	SUD05230
	006ER						
05E2R	C570		CLHI	AC1,X'00'		IS IT A CAR RTN ?	SUD05240
	000D						
05E6R	4330		BE	VER050		IF SO, DONE, CHECK FOR END	SUD05250
	0606R						
05FAR	D389		LB	AC2,VERRFR(AC3)		FETCH A CHAR FROM DUPE	SUD05260
	066ER						
05EER	C580		CLHI	AC2,X'00'		IS IT A CAR RTN ?	SUD05270
	000D						
05F2R	4330		BE	VER050		IF SO, DONE CHECK FOR FND	SUD05280
	0606R						
05F6R	057A		CLHR	AC1,AC2		ARE THE TWO CHAR EQUAL ?	SUD05290
05F8R	4230		BNE	VER060		IF NOT, ERROR GO PRINT BFRS	SUD05300
	0618R						
05FCR	2691		AIS	AC3,1			SUD05310
05FER	4590		CLH	AC3,VERCNT		FINISHED ?	SUD05320
	0634R						
0602R	4280		BL	VER040		NO, KEEP CHECKING	SUD05330
	05DER						
0606R	41F0	VER050	BAL	LNK,EXTRCT		IS THIS AN 'END' CARD	SUD05340
	0884R						
060AR	454E		DC	C*END			SUD05350
	4420						
	2020						
0610R	4300		B	VER030		NO, GO GET NEXT TWO RECORDS	SUD05360
	05C8R						
0614R	4300		B	START		DONE, GET NEXT COMMAND	SUD05370
	000AR						
							SUD05380
							SUD05390
							SUD05400
0618R	E110	* VER060	SVC	1,VER950		PRINT VERIFY ERROR MSG	SUD05410
	063ER						
061CR	F110		SVC	1,VER951		LIST MASTER RECORD FIRST.	SUD05420
	0646R						
0620R	F110		SVC	1,VER952		THEN LISTED DUPED RECORD.	SUD05430
	064ER						
0624R	F120		SVC	2,PAUSE			SUD05440
	0C08R						
0628R	4300		B	VER030			SUD05440

```

05C8R
*
*
062CR 1A01 VER900 DC X'1A01'.0 SUD05450
0000 SUD05460
0630R 2C01 VER901 DC X'2C01'.C' SUD05470
2020 SUD05480
0634R 0000 VERCNT DC 0 SUD05490
0636R 4802 VERINP DC X'4802'.0,VERBFR,VERBFR+79 SUD05500
0000
066FR
06BDR
063ER 2803 VER950 DC X'2803'.0,VER960,VER961 SUD05510
0000
0656R
066DR
0646R 2803 VER951 DC X'2803'.0,BFR,BFR+79 SUD05520
0000
006ER
00RDR
064FR 2803 VER952 DC X'2803'.0,VERBFR,VERBFR+79 SUD05530
0000
066ER
06RDR
*
*
0656R 2A2A VER960 DC C'*** VERIFICATION ERROR *** SUD05540
2056 SUD05550
4552 SUD05560
4946
4943
4154
494F
4E20
4552
524F
5220
2A2A
066DR VER961 EQU *-1 SUD05570
066ER VERBFR DS 80 SUD05580
* SUD05590
* UPDATE (LU)(,RESEQ) SUD05600
* SUD05610
06BER 071AR UPDATE DC UPD900 SUD05620
06C0R 071FR DC UPD901 SUD05630
06C2R 2485 LIS AC2,5 SUD05640
06C4R 4870 LH AC1,UPD900 SUD05650
071AR
06C8R 0A77 AHR AC1,AC1 SUD05660
06CAR 2313 BNMS UPD010 SUD05670
06CCR 0380 LB AC2,UPD900+3 SUD05680
071DR
06D0R D280 UPD010 STR AC2,UPDINP+1 SUD05690
0CB9R
06D4R 2481 LIS AC2.1 SUD05700
06D6R D280 STB AC2,READ+1 SUD05710
    
```

06DAR	0CB1R D280		STR	AC2,BKSP+1		SUD05720
06DER	0CC9R 4870		LH	AC1,UPD901		SUD05730
06E2R	071ER 0A77		AHR	AC1,AC1		SUD05740
06E4R	2317		BNMS	UPD020		SUD05750
06E6R	0370		LB	AC1,UPD901+2		SUD05760
06EAR	0720R C570		CLHI	AC1,C*N*		SUD05770
06EER	004F 4330		BE	UPD030		SUD05780
06F2R	0710R 4080	UPD020	STH	AC2,MODE	SET UPDATE MODE	SUD05790
06F6R	0D56R 41F0		BAL	LNK,IOSURR	READ FIRST RECORD FOR SEQNR	SUD05800
06FAR	09C0R 0CB0R		DC	READ		SUD05810
06FCR	4230		BNZ	IORERR		SUD05820
0700R	09F6R 41F0		BAL	LNK,SEQSAV		SUD05830
0704R	0A5CR 02CCR		DC	NEWSEQ+2		SUD05840
0706R	41F0		BAL	LNK,IOSURR		SUD05850
070AR	09C0R 0CC8R		DC	BKSP		SUD05860
070CR	4300		B	START		SUD05870
	000AR					SUD05880
		*				SUD05890
0710R	2571	UPD030	LCS	AC1,1		SUD05900
0712R	4070		STH	AC1,MODE	SET MODE TO NO RESEQUENCING	SUD05910
	0D56R					
0716R	4300		B	START		SUD05920
	000AR					
		*				SUD05930
		*				SUD05940
		*				SUD05950
071AR	1A01	UPD900	DC	X'1A01'.0		SUD05960
	0000					
071ER	2C01	UPD901	DC	X'2C01'.C'		SUD05970
	2020					
		*				SUD05980
		*	INSERT	SEQNR		SUD05990
		*				SUD06000
0722R	079AR	INSERT	DC	INSRTX		SUD06010
0724R	4870		LH	AC1,MODE	WHAT MODE ARE WE IN ?	SUD06020
	0D56R					
0728R	4330		BZ	CMDERR	IF NOT IN UPDATE MODE, ERROR	SUD06030
	0062R					
072CR	41F0	INSRT1	BAL	LNK,IOSURR	READ A MASTER RECORD	SUD06040
	09C0R					
0730R	0CB0R		DC	READ		SUD06050
0732R	4230		BNZ	START		SUD06060
	000AR					
0736R	41F0		BAL	LNK,SEQSAV	SAVE OLD SEQNR	SUD06070

073AR 0A5CR					
073AR 0D58R		DC	SFQSR		SUD06080
073CR 4870		LH	AC1.MODE	RESEQUENCE ?	SUD06090
0740R 0D56R					
0740R 2116		BMS	INSRT2		SUD06100
0742R 41F0		BAL	LNK.RESEQ	YES. MOVE NEW SEQNR TO BFR	SUD06110
0746R 0AA0R					
0746R 41F0		BAL	LNK.DECINC	INCREMENT SEQ NBR	SUD06120
074AR 0C1AR					
074AR 02D2R		DC	NEWSEQ+8		SUD06130
074CR 41F0	INSRT2	BAL	LNK.IOSURR		SUD06140
0750R 09C0R					
0750R 0CC0R		DC	WRITE		SUD06150
0752R 41F0		BAL	LNK.EXTRCT	SEARCH FOR 'END' CARD	SUD06160
0756R 0BB4R					
0756R 454E		DC	C.END		SUD06170
0756R 4420					
0756R 2020					
075CR 4300		B	INSRT3		SUD06180
0760R 0764R					
0760R 4300		B	NOTFND	SEQ NBR WAS NOT FOUND	SUD06190
0760R 01E4R					
	*				SUD06200
	*				SUD06210
0764R 41F0	INSRT3	BAL	LNK.SEQCHK		SUD06220
0764R 0A84R					
0768R 079CR		DC	INSRTX+2		SUD06230
076AR 4300		B	INSRT1		SUD06240
076ER 072CR					
076ER 41F0	INSRT4	BAL	LNK.IOSURR		SUD06250
076ER 09C0R					
0772R 0CR8R		DC	UPDINP		SUD06260
0774R 4870		LH	AC1.BFR		SUD06270
0778R 0D6ER					
0778R C570		CLHI	AC1.C'/'*	END OF INSERT OPERATION ?	SUD06280
0778R 2F25					
077CR 4330		RE	START		SUD06290
0780R 000AR					
0780R 4870		LH	AC1.MODE		SUD06300
0780R 0D56R					
0784R 2116		BMS	INSRT5		SUD06310
0786R 41F0		BAL	LNK.RESEQ		SUD06320
078AR 0AA0R					
078AR 41F0		BAL	LNK.DECINC	INCREMENT SEQ NBR	SUD06330
078AR 0C1AR					
078ER 02D2R		DC	NEWSEQ+8		SUD06340
0790R 41F0	INSRT5	BAL	LNK.IOSURR		SUD06350
0794R 09C0R					
0794R 0CC0R		DC	WRITE		SUD06360
0796R 4300		B	INSRT4		SUD06370
0796R 076ER					
	*				SUD06380
	*				SUD06390
079AR 2108	INSRTX	DC	X'2108'		SUD06400
079CR		DS	8		SUD06410
	*				SUD06420

```

*   DELETE SEQNR1(,SEQNR2)
*
07A4R 0836R  DELETE DC   DLT900
07A6R 0840R          DC   DLT901
07A8R 4870          LH   AC1,MODE      WHAT MODE ARE WE IN ?
      0056R
07ACR 4330          RZ   CMDERR      IF NOT IN UPDATE MODE, ERROR
      0062R
07B0R 41F0  DLT010 BAL   LNK,IOSURR
      09C0R
07B4R 0CB0R          DC   READ
07B6R 2138          BNZS DLT015
07B8R 41F0          BAL   LNK,EXTRCT
      0BR4R
07BCR 454E          DC   C'END '
      4420
      2020
07C2R 4300          B    DLT030
      07CER
07C6R E120  DLT015 SVC   2,DLT910      LOG ERROR MSG
      084AR
07CAR 4300          B    START
      000AR

*
*   DLT030 BAL   LNK,SEQSAV
07CER 41F0
      0A5CR
07D2R 0D58R          DC   SEQSAV
07D4R 41F0          BAL   LNK,SEQCHK
      0A84R
07D8R 0838R          DC   DLT900+2
07DAR 4300          B    DLT060
      081CR
07DER 4870          LH   AC1,DLT901
      0840R
07E2R 0A77          AHR  AC1,AC1
07E4R 4310          BNM  START
      000AR
07E8R 41F0  DLT040 BAL   LNK,IOSURR
      09C0R
07ECR 0CB0R          DC   READ
07EER 4230          BNZ  DLT045
      0800R
07F2R 41F0          BAL   LNK,EXTRCT
      0BR4R
07F6R 454E          DC   C'END '
      4420
      2020
07FCR 4300          B    DLT050
      0808R
0800R E120  DLT045 SVC   2,DLT911      LOG ERROR MSG
      0866R
0804R 4300          B    START
      000AR

*
*
SUD06430
SUD06440
SUD06450
SUD06460
SUD06470
SUD06480
SUD06490
SUD06500
SUD06510
SUD06520
SUD06530
SUD06540
SUD06550
SUD06560
SUD06570
SUD06580
SUD06590
SUD06600
SUD06610
SUD06620
SUD06630
SUD06640
SUD06650
SUD06660
SUD06670
SUD06680
SUD06690
SUD06700
SUD06710
SUD06720
SUD06730
SUD06740
SUD06750
SUD06760

```

0808R	41F0	DLT050	BAL	LNK,SEQSAV	EXTRACT AND SAVE THE SEQNBR	SUD06770
	0A5CR					
080CR	0D58R		DC	SEQSAR		SUD06780
080ER	41F0		BAL	LNK,SEQCHK		SUD06790
	0A84R					
0812R	0842R		DC	DLT901+2		SUD06800
0814R	4300		B	DLT040		SUD06810
	07E8R					
0818R	4300		B	START		SUD06820
	000AR					
		*				SUD06830
		*				SUD06840
081CR	4870	DLT060	LH	AC1,MODE		SUD06850
	0D56R					
0820R	2116		BMS	DLT065		SUD06860
0822R	41F0		BAL	LNK,RESEQ		SUD06870
	0AAA0R					
0826R	41F0		BAL	LNK,DECINC	INCREMENT SEQ NBR	SUD06880
	0C1AR					
082AR	02D2R		DC	NEWSEQ+8		SUD06890
082CR	41F0	DLT065	BAL	LNK,IOSURR		SUD06900
	09C0R					
0830R	0CC0R		DC	WRITE		SUD06910
0832R	4300		B	DLT010		SUD06920
	07B0R					
		*				SUD06930
		*				SUD06940
0836R	1108	DLT900	DC	X'1108'		SUD06950
0838R			DS	8		SUD06960
0840R	2908	DLT901	DC	X'2908'		SUD06970
0842R			DS	8		SUD06980
084AR	0007	DLT910	DC	7,23,C*FIRST SEQ NBR NOT FOUND.		SUD06990
	0017					
	4649					
	5253					
	5420					
	5345					
	5120					
	4E42					
	5220					
	4E4F					
	5420					
	464F					
	554E					
	4420					
0866R	0007	DLT911	DC	7,24,C*SECOND SEQ NBR NOT FOUND.		SUD07000
	0018					
	5345					
	434F					
	4E44					
	2053					
	4551					
	204F					
	4252					
	204E					
	4F54					



2046  
4F55  
4F44

	*				SUD07010
	*	MODIFY	DC	MOD900	SUD07020
	*		LH	AC1,MODE	SUD07030
0882R	08F6R				SUD07040
0884R	4870			WHAT MODE ARE WF IN ?	SUD07050
	0056R				
0888R	4330		BZ	CMDERR	SUD07060
	0062R			IF NOT IN UPDATE MODE, ERROR	
088CR	41F0	MOD020	BAL	LNK,IOSURR	SUD07070
	09C0R				
0890R	0CR0R		DC	READ	SUD07080
0892R	4230		BNZ	MOD040	SUD07090
	08EER				
0896R	41F0		BAL	LNK,SEQSAV	SUD07100
	0A5CR			EXTRACT AND SAVE THE SEQNR	
089AR	0D58R		DC	SEQSAR	SUD07110
089CR	41F0		BAL	LNK,SEQCHK	SUD07120
	0A84R				
08A0R	08F8R		DC	MOD900+2	SUD07130
08A2R	4300		B	MOD030	SUD07140
	08CAR				
08A6R	41F0		BAL	LNK,IOSURR	SUD07150
	09C0R				
08AAR	0CR8R		DC	UPDINP	SUD07160
08ACR	4230		BNZ	IORERR	SUD07170
	09F6R				
08B0R	4870		LH	AC1,MODE	SUD07180
	0056R				
08B4R	2116		BMS	MOD025	SUD07190
08B6R	41F0		BAL	LNK,RESEQ	SUD07200
	0AA0R				
08BAR	41F0		BAL	LNK,DECINC	SUD07210
	0C1AR			INCREMENT SEQ NBR	
08BER	0202R		DC	NEWSEQ+8	SUD07220
08COR	41F0	MOD025	BAL	LNK,IOSURR	SUD07230
	09C0R				
08C4R	0CC0R		DC	WRITE	SUD07240
08C6R	4300		B	START	SUD07250
	000AR				
	*				SUD07260
	*				SUD07270
08CAR	4870	MOD030	LH	AC1,MODE	SUD07280
	0056R				
08CER	2116		BMS	MOD035	SUD07290
08D0R	41F0		BAL	LNK,RESEQ	SUD07300
	0AA0R				
08D4R	41F0		BAL	LNK,DECINC	SUD07310
	0C1AR			INCREMENT SEQ NBR	
08D8R	0202R		DC	NEWSEQ+8	SUD07320
08DAR	41F0	MOD035	BAL	LNK,IOSURR	SUD07330
	09C0R				
08DER	0CC0R		DC	WRITE	SUD07340
08E0R	41F0		BAL	LNK,EXTRCT	SUD07350

08F4R	08R4R 454E 4420 2020	DC	C*END		SUD07360
08EAR	4300 088CR	B	MoD020		SUD07370
08EER	E120 0900R	MOD040 SVC	2,MOD910	LOG ERROR MSG	SUD07380
08F2R	4300 000AR	B	START		SUD07390
	*				SUD07400
	*				SUD07410
	*				SUD07420
08F6R	2108	MOD900 DC	X*2108*		SUD07430
08F8R		DS	8		SUD07440
0900R	0007	MOD910 DC	7.17.C*SEQ NHR NOT FOUND*		SUD07450
	0011				
	5345				
	5120				
	4E42				
	5220				
	4E4F				
	5420				
	464F				
	554E				
	4420				
	*				SUD07460
	*	END			SUD07470
	*				SUD07480
0916R	FFFF	UPDEND DC	-1		SUD07490
0918R	4870	LH	AC1,MODE	WHAT MODE ARE WE IN ?	SUD07500
	0056R				SUD07510
091CR	4330	BZ	CMDERR	IF NOT IN UPDATE MODE, ERROR	SUD07520
	0062R				SUD07530
0920R	41F0	UPDE10 BAL	LNK.IOSUBR		SUD07540
	09C0R				SUD07550
0924R	0CR0R	DC	READ		SUD07560
0926R	4230	BNZ	UPDE30		SUD07570
	094ER				SUD07580
092AR	4870	LH	AC1,MODE		SUD07590
	0056R				SUD07600
092ER	2116	BMS	UPDE20		SUD07610
0930R	41F0	BAL	LNK.RESEQ		SUD07620
	0AA0R				SUD07630
0934R	41F0	BAL	LNK.DECINC	INCREMENT SEQ NHR	SUD07640
	0C1AR				SUD07650
0938R	02D2R	DC	NEWSEQ+8		SUD07660
093AR	41F0	UPDE20 BAL	LNK.IOSUBR		SUD07670
	09C0R				SUD07680
093ER	0CC0R	DC	WRITE		SUD07690
0940R	41F0	BAL	LNK.EXTRCT		SUD07700
	08B4R				SUD07710
0944R	454E	DC	C*END		SUD07720
	4420				SUD07730
	2020				SUD07740
094AR	4300	B	UPDE10		SUD07750

094FR	0920R	UPDE30	LIS	AC1.2		SUD07650
0950R	2472		BAL	ACB.WFMS	WRITE AN 'EOF' AND BACKSPACE	SUD07660
	41F0					
	096AR					
0954R	4300		B	SRCUPD	GO CLEAR MODE AND GET A CMD	SUD07670
	0000R					
						SUD07680
						SUD07690
						SUD07700
						SUD07710
						SUD07720
0958R	098CR	WE0V	DC	EOFLU		
095AR	4870		LH	AC1.MODE	WHAT MODE ARE WE IN ?	
	0056R					
095ER	4230		BNZ	CMDERR	IF UPDATE MODE. ERROR	SUD07730
	0062R					
0962R	0370		LB	AC1.EOFLU+3		SUD07740
	09AFR					
0966R	08F0		LHI	ACB.START	SET UP RETURN ADDRESS	SUD07750
	000AR					
096AR	0270	WFMS	STB	AC1.WRTEOF+1		SUD07760
	00D1R					
096ER	0270		STB	AC1.BKSP+1		SUD07770
	00C9R					
0972R	41F0		BAL	LNK.IOSUBR		SUD07780
	09C0R					
0976R	00D0R		DC	WRTEOF		SUD07790
0978R	41F0		BAL	LNK.IOSUBR		SUD07800
	09C0R					
097CR	00D0R		DC	WRTEOF		SUD07810
097ER	41F0		BAL	LNK.IOSUBR		SUD07820
	09C0R					
0982R	00C8R		DC	BKSP		SUD07830
0984R	41F0		BAL	LNK.IOSUBR		SUD07840
	09C0R					
0988R	00C8R		DC	BKSP		SUD07850
098AR	030E		BR	ACB		SUD07860
						SUD07870
						SUD07880
						SUD07890
098CR	2201	EOFLU	DC	X*2201*.0		SUD07900
	0000					SUD07910
						SUD07920
						SUD07930
						SUD07940
						SUD07950
						SUD07960
						SUD07970
						SUD07980
						SUD07990
						SUD08000
0990R	09ACR	REWIND	DC	REWLU		
0992R	4870		LH	AC1.MODE	WHAT MODE ARE WE IN ?	
	0056R					
0996R	4230		BNZ	CMDERR	IF UPDATE MODE. ERROR	SUD07950
	0062R					
099AR	0370		LB	AC1.REWLU+3		SUD07960
	09AFR					
099ER	0270		STB	AC1.REW+1		SUD07970
	00D5R					
09A2R	41F0		BAL	LNK.IOSUBR		SUD07980
	09C0R					
09A6R	00D4R		DC	REW		SUD07990
09A8R	4300		B	START		SUD08000
	000AR					

	*				SUD08010
	*				SUD08020
09ACR	2201	REWLU	DC	X'2201',0	SUD08030
	0000				
	*				SUD08040
	*	PAUSE			SUD08050
	*				SUD08060
09A0R	FFFF	STOP	DC	-1	SUD08070
09A2R	F120		SVC	2,PAUSE	SUD08080
	0CD8R				
09B6R	4300		R	START	SUD08090
	000AR				
	*				SUD08100
	*	EOJ			SUD08110
	*				SUD08120
09BAR	FFFF	EOJ	DC	-1	SUD08130
09BCR	E130		SVC	3,0	SUD08140
	0000				
	*				SUD08150
	*				SUD08160
09COR	487F	IOSURR	LH	AC1,0(LNK)	SUD08170
	0000				
09C4R	E117		SVC	1,0(AC1)	SUD08180
	0000				
09C8R	4807		LH	R0,2(AC1)	SUD08190
	0002				
09CCR	433F		HZ	2(LNK)	SUD08200
	0002				
09D0R	C300		THI	R0,EOF	SUD08210
	0800				
09D4R	423F		BNZ	2(LNK)	SUD08220
	0002				
09D8R	4877		LH	AC1,0(AC1)	SUD08230
	0000				
09DCR	2118		BMS	IOCERR	SUD08240
09DER	0A77		AHR	AC1,AC1	SUD08250
09E0R	2118		BMS	IORERR	SUD08260
09E2R	E120	IOWERR	SVC	2,IOS001	SUD08270
	0A06R				
09E6R	E120		SVC	2,IOS011	SUD08280
	0A12R				
09EAR	230A		BS	IOS999	SUD08290
09ECR	E120	IOCERR	SVC	2,IOS002	SUD08300
	0A0AR				
09F0R	E120		SVC	2,IOS012	SUD08310
	0A2AR				
09F4R	2305		BS	IOS999	SUD08320
09F6R	F120	IORERR	SVC	2,IOS003	SUD08330
	0A0ER				
09FAR	F120		SVC	2,IOS013	SUD08340
	0A46R				
09FER	F120	IOS999	SVC	2,PAUSE	SUD08350
	0CD8R				
0A02R	4300		R	IOSURR	SUD08360
	09COR				
0A06R	0006	IOS001	DC	6,IOS011+19	SUD08370

END OF FILE MARK FOUND ?

0A0AR	0A25R 0006	IOS002	DC	6,IOS012+24	SUD08380
0A0FR	0A42R 0006	IOS003	DC	6,IOS013+18	SUD08390
0A12R	0A58R 0007 0013 4F55 5450 5554 2045 5252 4F52 2020 2058 5858 5820	IOS011	DC	7,19,C*OUTPUT ERROR - XXXX*	SUD08400
0A2AR	0007 0018 492F 4F20 434F 4040 414E 4420 4552 524F 5220 2020 5858 5858	IOS012	DC	7,24,C*I/O COMMAND ERROR - XXXX*	SUD08410
0A46R	0007 0012 494E 5055 5420 4552 524F 5220 2020 5858 5858	IOS013	DC	7,18,C*INPUT ERROR - XXXX*	SUD08420
0A5CR	2571	SEQSAV	LCS	AC1,1	SUD08430
0A5ER	2671	SGSV10	AIS	AC1,1	SUD08440
0A60R	0387		LB	AC2,BFR(AC1)	SUD08450
0A64R	0D6ER C580		CLHI	AC2,X*00*	SUD08460
0A68R	2035		BNES	SGSV10	SUD08470
0A6AR	2497		LIS	AC3,7	SUD08480
0A6CR	48AF		LH	AC4,0(LNK)	SUD08490
0A70R	0000 2771	SGSV20	SIS	AC1,1	SUD08500
0A72R	0387		LB	AC2,BFR(AC1)	SUD08510
					SUD08520
					SUD08530

0A76R	0D6ER 028A 0007	STR	AC2.7(AC4)		SUD08540
0A7AR	27A1	SIS	AC4.1		SUD08550
0A7CR	2791	SIS	AC3.1		SUD08560
0A7ER	2217	BNMS	SQSV20		SUD08570
0A80R	430F 0002	B	2(LNK)		SUD08580
		*			SUD08590
		*			SUD08600
		*			SUD08610
0A84R	2486	SEQCHK	LIS AC2.6		SUD08620
0A86R	489F 0000	LH	AC3.0(LNK)		SUD08630
0A8AR	4878	SQCK10	LH AC1.SEQSAR(AC2)		SUD08640
	0D58R				
0A8ER	4579	CLH	AC1.6(AC3)		SUD08650
	0006				
0A92R	423F 0002	BNE	2(LNK)		SUD08660
0A96R	2792	SIS	AC3.2		SUD08670
0A98R	2782	SIS	AC2.2		SUD08680
0A9AR	2218	BNMS	SQCK10		SUD08690
0A9CR	430F 0006	B	6(LNK)		SUD08700
		*			SUD08710
		*			SUD08720
		*			SUD08730
0AA0R	0788	RESEQ	XHR AC2.AC2	CLEAR INDEX REGISTER	SUD08740
0AA2R	0799	XHR	AC3.AC3		SUD08750
0AA4R	D378	RESEQ1	LB AC1.BFR(AC2)	SEARCH FOR END-OF-LINE	SUD08760
	0D6ER				
0AA8R	C570 0000	CLHI	AC1.X'00'		SUD08770
0AACR	2335	BES	RESEQ2		SUD08780
0AAER	2681	AIS	AC2.1		SUD08790
0AB0R	C580 0048	CLHI	AC2.72		SUD08800
0AB4R	2088	BLS	RESEQ1		SUD08810
0AB6R	C580 0048	RESEQ2	CLHI AC2.72		SUD08820
0ABAR	2385	BNLS	RESEQ3		SUD08830
0ABCR	D218	STR	SPC.BFR(AC2)	FILL OUT LINE WITH SPACES	SUD08840
	0D6ER				
0AC0R	2681	AIS	AC2.1		SUD08850
0AC2R	2206	BS	RESEQ2		SUD08860
		*			SUD08870
		*			SUD08880
0AC4R	D379	RESEQ3	LB AC1.NEWSFQ+2(AC3)	MOVE SEQ NBR INTO BFR	SUD08890
	02CCR				
0AC8R	D278	STR	AC1.BFR(AC2)		SUD08900
	0D6ER				
0ACCR	2681	AIS	AC2.1		SUD08910
0ACFR	2691	AIS	AC3.1		SUD08920
0AD0R	C590 0008	CLHI	AC3.8	FINISHED YET ?	SUD08930

0AD4R 2088		BLS	RESEQ3		SUD08940
0AD6R 030F		BR	LNK		SUD08950
	*				SUD08960
	*				SUD08970
	*				SUD08980
	*				SUD08990
0AD8R 488E	SFARCH	LH	AC2.0(ACA)	FETCH A(LABEL)	SUD09000
0000					
0ADCR 4878		LH	AC1.0(AC2)		SUD09010
0000					
0AE0R C570		CLHI	AC1.C'***'	LABEL OR EOF COUNT ?	SUD09020
2A2A					
0AE4R 4330		BE	SRCH50	IF EQUAL. LABEL	SUD09030
0B4AR					
0AE8R 24A8		LIS	AC4.8	SET LENGTH	SUD09040
0AEAR 0799		XHR	AC3.AC3	CLEAR ACCUMULATOR	SUD09050
0AECR D378	SRCH01	LB	AC1.0(AC2)	FETCH A CHAR	SUD09060
0000					
0AF0R 0571		CLHR	AC1.SPC	IS IT A SPACE ?	SUD09070
0AF2R 4330		BE	SRCH02	IF SO. FINISHED CONVERTING	SUD09080
002CR					
0AF6R C570		CLHI	AC1.C'0'	LEGAL DECIMAL DIGIT ?	SUD09090
0030					
0AFAR 4280		BL	ERROR		SUD09100
018AR					
0AFER C570		CLHI	AC1.X'3A'		SUD09110
003A					
0R02R 4380		BNL	ERROR		SUD09120
018AR					
0R06R 2681		AIS	AC2.1		SUD09130
0R08R C470		NHI	AC1.X'0F'	MASK OFF ZONE BITS	SUD09140
000F					
0R0CR 08D9		LHR	AC7.AC3		SUD09150
0R0ER C390		THI	AC3.X'E000'		SUD09160
F000					
0R12R 4230		BNZ	ERROR		SUD09170
018AR					
0R16R 9193		SLLS	AC3.3		SUD09180
0R18R 0A0D		AHR	AC7.AC7		SUD09190
0R1AR 0A9D		AHR	AC3.AC7		SUD09200
0R1CR 4280		BC	ERROR		SUD09210
018AR					
0R20R 0A97		AHR	AC3.AC1		SUD09220
0R22R 4280		BC	ERROR		SUD09230
018AR					
0R26R 27A1		SIS	AC4.1	DECREMENT LENGHT CTR	SUD09240
0R28R 4230		BNZ	SRCH01	IF NOT ZERO. GET NEXT CHAR	SUD09250
0AECR					
0B2CR 41F0	SRCH02	BAL	LNK.IOSUBR	REWIND THE INPUT DEVICE	SUD09260
09COR					
0R30R 0CD4R		DC	REW		SUD09270
0R32R 41F0	SRCH03	BAL	LNK.IOSUBR		SUD09280
09COR					
0R36R 0CCCR		DC	FNDEOF		SUD09290
0R38R 41F0		BAL	LNK.IOSUBR	END OF VOLUME ?	SUD09300
09COR					

0B3CR	0CB0R	DC	READ		SUD09310
0B3ER	423E	BNZ	2(AC8)	YES, NOT FOUND	SUD09320
	0002				
0B42R	2791	SIS	AC3.1	DECREMENT EOF COUNT	SUD09330
0B44R	2039	BNZS	SRCH03		SUD09340
0B46R	430E	B	6(AC8)	FOUND, RETURN TO CALLER	SUD09350
	0006				
		*			SUD09360
		*			SUD09370
		*			SUD09380
0B4AR	41F0	SRCH50	BAL	LNK.IOSUBR	SUD09390
	09C0R			READ A RECORD	
0B4ER	0CB0R	DC	READ		SUD09400
0B50R	2336	BZS	SRCH51		SUD09410
0B52R	41F0	BAL	LNK.IOSUBR	END OF VOLUME ?	SUD09420
	09C0R				
0B56R	0CB0R	DC	READ		SUD09430
0B58R	423E	BNZ	2(AC8)	YES, TAKE 'NOT FOUND' EXIT	SUD09440
	0002				
0B5CR	41F0	SRCH51	BAL	LNK.LABCHK	SUD09450
	0B68R			COMPARE LABELS	
0B60R	4300	B	SRCH50	NOT IT, KEEP LOOKING	SUD09460
	0B4AR				
0B64R	430E	B	6(AC8)	FOUND, RETURN TO CALLER	SUD09470
	0006				
		*			SUD09480
		*			SUD09490
0B68R	C870	LABCHK	LHI	AC1.C***	SUD09500
	2A2A				
0B6CR	4570	CLH	AC1.BFR	POSSIBLE LABEL RECORD ?	SUD09510
	0D6ER				
0B70R	023F	BNER	LNK	IF NOT, KEEP READING	SUD09520
0B72R	2498	LIS	AC3.8	YES, CHECK IT	SUD09530
0B74R	0219	SRCH52	STR	SPC.LABSAV-1(AC3) CLEAR LABEL SAVE AREA	SUD09540
	0D5FR				
0B78R	2791	SIS	AC3.1		SUD09550
0B7AR	2033	BNZS	SRCH52		SUD09560
0B7CR	0379	SRCH53	LB	AC1.BFR(AC3) MOVE LABEL TO SAVE AREA	SUD09570
	0D6ER				
0B80R	C570	CLHI	AC1.X'0D'	CAR RTN ?	SUD09580
	000D				
0B84R	4330	BE	SRCH54	END OF LABEL	SUD09590
	0B96R				
0B88R	0279	STR	AC1.LABSAV(AC3)		SUD09600
	0D60R				
0B8CR	2691	AIS	AC3.1		SUD09610
0B8ER	C590	CLHI	AC3.8	FINISHED ?	SUD09620
	0008				
0B92R	4280	BL	SRCH53		SUD09630
	0B7CR				
0B96R	0799	SRCH54	XHR	AC3.AC3	SUD09640
0B98R	08A8	LHR	AC4.AC2		SUD09650
0B9AR	4879	SRCH55	LH	AC1.LABSAV(AC3)	SUD09660
	0D60R				
0B9ER	457A	CLH	AC1.0(AC4)		SUD09670
	0000				



0BA2R 023F		BNER	LNK	NOT IT. KEEP LOOKING	SUD09680
0BA4R 26A2		AIS	AC4,2		SUD09690
0BA6R 2692		AIS	AC3,2		SUD09700
0BA8R C590		CLHI	AC3,8		SUD09710
0008					
0BACR 4280		BL	SRCH55		SUD09720
0B9AR					
0BB0R 430F		B	4(LNK)	YEA. FOUND IT	SUD09730
0004					
	*				SUD09740
	*				SUD09750
0BB4R 2486	EXTRCT	LIS	AC2,6		SUD09760
0BB6R D218	LOOP0	STB	SPC.OPCODE-1(AC2)	CLEAR OP-CODE SAVE AREA	SUD09770
0067R					
0BBAR 2781		SIS	AC2,1		SUD09780
0BBCR 2033		BNZS	LOOP0		SUD09790
0BBER 0799		XHR	AC3,AC3		SUD09800
0BC0R D378		LB	AC1,BFR(AC2)		SUD09810
006ER					
0BC4R 2681		AIS	AC2,1		SUD09820
0BC6R C570		CLHI	AC1,C'&'	COMMENT CARD ?	SUD09830
002A					
0BCAR 433F		BE	6(LNK)	YES. EXIT	SUD09840
0006					
0BCEr 0571		CLHR	AC1,SPC	ANY LABEL ?	SUD09850
0BD0R 2336		BES	LOOP2	NO. FIND OP-CODE	SUD09860
0BD2R D378	LOOP1	LB	AC1,BFR(AC2)	YES. SKIP OVER THE LABEL	SUD09870
006ER					
0BD6R 2681		AIS	AC2,1		SUD09880
0BD8R 0571		CLHR	AC1,SPC	END OF LABEL ?	SUD09890
0BDAR 2034		BNES	LOOP1		SUD09900
0BDcR D378	LOOP2	LB	AC1,BFR(AC2)	SKIP OVER SPACES	SUD09910
006ER					
0BE0R 2681		AIS	AC2,1		SUD09920
0BE2R 0571		CLHR	AC1,SPC		SUD09930
0BE4R 2234		BES	LOOP2		SUD09940
0BE6R D279	LOOP3	STB	AC1.OPCODE(AC3)	MOVE OP-CODE TO SAVE AREA	SUD09950
0068R					
0BEAR 2691		AIS	AC3,1		SUD09960
0BECR D378		LB	AC1,BFR(AC2)		SUD09970
006ER					
0BF0R 2681		AIS	AC2,1		SUD09980
0BF2R 0571		CLHR	AC1,SPC	END OF OP-CODE ?	SUD09990
0BF4R 2334		BES	LOOP4		SUD10000
0BF6R C570		CLHI	AC1,X'00'		SUD10010
000D					
0BFAR 203A		BNES	LOOP3		SUD10020
0BFcR 0799	LOOP4	XHR	AC3,AC3		SUD10030
0BFER 08AF		LHR	AC4,LNK		SUD10040
0C00R 4879	LOOP5	LH	AC1.OPCODE(AC3)	IS THIS THE DESIRED OP-CODE ?	SUD10050
0068R					
0C04R 457A		CLH	AC1,0(AC4)		SUD10060
0000					
0C08R 423F		BNE	6(LNK)	NO. TAKE NOT FOUND EXIT	SUD10070
0006					
0C0cR 26A2		AIS	AC4,2		SUD10080

0C0ER	2692	AIS	AC3,2		SUD10090
0C10R	C590	CLHI	AC3,6	IS THIS REALLY IT ?	SUD10100
	0006				
0C14R	208A	BLS	LOOPS	NOT YET, KEEP CHECKIN'	SUD10110
0C16R	430F	B	10(LNK)	HOORAH, THIS IS IT	SUD10120
	000A				
		*			SUD10130
		*			SUD10140
0C1AR	48RF	DECINC	LH	AC5,0(LNK)	SUD10150
	0000			FETCH A(NBR TO BE INCR)	
0C1ER	2483	LIS	AC2,3		SUD10160
0C20R	D37B	DILOOP	LB	AC1,0(AC5)	SUD10170
	0000				
0C24R	2671	AIS	AC1,1		SUD10180
0C26R	D27B	STB	AC1,0(AC5)		SUD10190
	0000				
0C2AR	C570	CLHI	AC1,X'3A'		SUD10200
	003A				
0C2ER	428F	BL	2(LNK)		SUD10210
	0002				
0C32R	C870	LHI	AC1,X'30'		SUD10220
	0030				
0C36R	D27B	STB	AC1,0(AC5)		SUD10230
	0000				
0C3AR	27B1	SIS	AC5,1		SUD10240
0C3CR	2781	SIS	AC2,1		SUD10250
0C3ER	4310	RNM	DILOOP		SUD10260
	0C20R				
0C42R	430F	B	2(LNK)		SUD10270
	0002				
		*			SUD10280
		*	OPERATOR COMMAND TABLE		SUD10290
		*			SUD10300
0C46R	4649	CMDTAB	DC	C'FIND',A(FNDLAB)	SUD10310
	4E44				
	0192R				
0C4CR	434F		DC	C'COPY',A(COPY)	SUD10320
	5059				
	0206R				
0C52R	4455		DC	C'DUPE',A(DUPE)	SUD10330
	5045				
	02F4R				
0C58R	5645		DC	C'VERI',A(VERIFY)	SUD10340
	5249				
	0594R				
0C5ER	5441		DC	C'TABL',A(TABLE)	SUD10350
	424C				
	03A2R				
0C64R	4C49		DC	C'LIST',A(LIST)	SUD10360
	5354				
	047AR				
0C6AR	5550		DC	C'UPDA',A(UPDATE)	SUD10370
	4441				
	06B6R				
0C70R	494E		DC	C'INSE',A(INSERT)	SUD10380
	5345				

0C76R	0722R 4445 4C45 07A4R	DC	C'DELE',A(DELETE)	SUD10390
0C7CR	4D4F 4449 0882R	DC	C'MODI',A(MODIFY)	SUD10400
0C82R	454E 4420 0916R	DC	C'END ',A(UPDEND)	SUD10410
0C88R	5745 4F56 0958R	DC	C'WEOV',A(WEOV)	SUD10420
0C8ER	5245 5749 0990R	DC	C'REWI',A(REWIND)	SUD10430
0C94R	5041 5553 0980R	DC	C'PAUS',A(STOP)	SUD10440
0C9AR	454F 4A20 09BAR	DC	C'EOJ ',A(EOJ)	SUD10450
0C9FR		CTEND EQU	*-1	SUD10460
		*		SUD10470
		*	I/O SVC PARAMETER BLOCKS	SUD10480
		*		SUD10490
0CA0R	2806 0000 0CDAR	RDYDY DC	X'2806',0,RDYMSG,RDYMSG+5	SUD10500
0CA8R	0CDFR 4805 0000 0D6ER	RDCMD DC	X'4805',0,A(BFR),A(BFR)+79	SUD10510
0CB0R	0DBDR 4801 0000 0D6ER	READ DC	X'4801',0,A(BFR),A(BFR)+79	SUD10520
0CB8R	0DBDR 4805 0000 0D6ER	UPDINP DC	X'4805',0,A(BFR),A(BFR)+79	SUD10530
0CC0R	0DBDR 2802 0000 0D6ER	WRITE DC	X'2802',0,A(BFR),A(BFR)+79	SUD10540
0CC8R	0DBDR A001 0000	BKSP DC	X'A001',0	SUD10550
0CCCR	8402 0000	FNDEOF DC	X'8402',0	SUD10560
0CD0R	8801 0000	WRTEOF DC	X'8801',0	SUD10570
0CD4R	C001 0000	REW DC	X'C001',0	SUD10580
0CD8R	0001	PAUSE DC	1	SUD10590
		*		SUD10600

## \* MESSAGES

00DAR	2052	RDYMSG	DC	C' READY'	SUD10610
	4541				SUD10620
	4459				SUD10630
00EOR	0007	ILCMG	DC	7.15.C'ILLEGAL COMMAND'	SUD10640
	000F				
	494C				
	4C45				
	4741				
	4C20				
	434F				
	4D4D				
	414E				
	4420				
0CF4R	0007	ARGERR	DC	7.13.C'OPERAND FRROR'	SUD10650
	000D				
	4F50				
	4552				
	414E				
	4420				
	4552				
	524F				
	5220				
0D06R	2806	FNDMSG	DC	X'2806'.0.FONDIT.FONDIT+7	SUD10660
	0000				
	0D0ER				
	0D15R				
0D0FR	2020	FONDIT	DC	C' FOUND'	SUD10670
	464F				
	554E				
	4420				
0D16R	0006	CONV01	DC	6.LOGE01+16	SUD10680
	0D2AR				
0D1AR	0007	LOGE01	DC	7.16.C'I/O ERROR - XXXX'	SUD10690
	0010				
	492F				
	4F20				
	4552				
	524F				
	5220				
	2D20				
	5858				
	5858				
0D2ER	0006	CONV02	DC	6.LOGE02+18	SUD10700
	0D44R				
0D32R	0007	LOGE02	DC	7.18.C'INPUT ERROR - XXXX'	SUD10710
	0012				
	494E				
	5055				
	5420				
	4552				
	524F				
	5220				
	2D20				
	5858				

5858						
0048R 0007	NFMSG	DC	7,9,C	NOT FOUND		SUD10720
0009						
4E4F						
5420						
464F						
554E						
4420						
0056R 0000	MODE	DC	0	UPDATE MODE FLAG		SUD10730
0058R 2020	SEQSAR	DC	C			SUD10740
2020						
2020						
2020						
0060R	LARSAV	DS	8			SUD10750
0068R 2020	OPCODE	DC	C	OP-CODE SAVE AREA		SUD10760
2020						
2020						
006ER	BFR	DS	80			SUD10770
00BER 000A		DC	X'000A'			SUD10780
00COR		END	SRCUPD			SUD10790

## NO ERRORS

AC1	0007
AC2	0008
AC3	0009
AC4	000A
AC5	000B
AC6	000C
AC7	000D
AC8	000E
ARGERR	0CF4R
ASC01	00AAR
ASCLFT	00COR
ASCRT	00D4R
ASCRT1	00E0R
BFR	0D6ER
BKSP	0CC8R
CMDERR	0062R
CMDTAB	0C46R
CONV01	0D16R
CONV02	0D2ER
COPIN	02C6R
COPLAR	02BCR
COPY	0206R
COPY00	0222R
COPY01	0244R
COPY20	026AR
COPY21	0274R
COPY40	0288R
COPY45	02A4R
COPY50	02B2R
CTEND	0C9FR
DECINC	0C1AR
DELETE	07A4R
DILOOP	0C20R
DLT010	07B0R
DLT015	07C6R
DLT030	07CER
DLT040	07E8R
DLT045	0800R
DLT050	0808R
DLT060	081CR
DLT065	082CR
DLT900	0836R
DLT901	0840R
DLT910	084AR
DLT911	0866R
DUP000	0312R
DUP005	033AR
DUP010	0342R
DUP015	035AR
DUP020	037AR
DUP030	0384R
DUP099	038AR
DUPE	02F4R
DUPIN	039ER
DUPLAR	0394R

END1	016CR
END2	017ER
EOF	0800
EOFLU	098CR
EOJ	098AR
EOM	1000
ERROR	018AR
EXTRCT	0884R
FNDEOF	0CCCR
FNDLAB	0192R
FNDMSG	0D06R
FONDIT	0D0ER
GETCHR	0146R
HOREND	0479R
HEX	00F8R
HEX01	010CR
HEX02	0130R
ILCMG	0CE0R
INPERR	006AR
INSERT	0722R
INSRT1	072CR
INSRT2	074CR
INSRT3	0764R
INSRT4	076ER
INSRT5	0790R
INSRTX	079AR
IOCERR	09ECR
IORERR	09F6R
IOS001	0A06R
IOS002	0A0AR
IOS003	0A0ER
IOS011	0A12R
IOS012	0A2AR
IOS013	0A46R
IOS999	09FER
IOWERR	09C0R
IOWERR	09E2R
LABCHK	0868R
LABEL	01ACR
LABEL8	01F8R
LABEL9	0202R
LABSAV	0D60R
LIST	047AR
LIST01	0494R
LIST20	04C2R
LIST30	04D2R
LIST40	04E8R
LIST50	0500R
LIST51	050ER
LIST90	0518R
LIST91	0522R
LIST93	0526R
LIST94	052ER
LIST95	0536R
LIST96	053ER
LIST97	0547R

LIST98	0548R
LIST99	0593R
LNK	000F
LOGE01	0D1AR
LOGE02	0D32R
LOOP0	0BB6R
LOOP1	0BD2R
LOOP2	0B0CR
LOOP3	0BE6R
LOOP4	0RFCR
LOOP5	0C00R
MOD020	088CR
MOD025	08C0R
MOD030	08CAR
MOD035	08DAR
MOD040	08EER
MOD900	08F6R
MOD910	0900R
MODE	0D56R
MODIFY	0882R
NEWSEQ	02CAR
NFMSG	0D48R
NOTFND	01E4R
OPCODE	0D6BR
PAGNR	058ER
PARSER	007ER
PAUSE	0CD8R
PRTHDR	043AR
PRTLAR	0442R
PSMSG	02D4R
RD	0000
RDCMD	0CA8R
RDYMSG	0CDAR
READ	0CB0R
READY	0CA0R
RESEQ	0AA0R
RESEQ1	0AA4R
RESEQ2	0AB6R
RESEQ3	0AC4R
REW	0CD4R
REWIND	0990R
REWLU	09ACR
SEARCH	0AD8R
SEQCHK	0A84R
SEQSAR	0D58R
SEQSAV	0A5CR
SPC	0001
SQCK10	0A8AR
SQSV10	0A5ER
SQSV20	0A70R
SRCH01	0AFCR
SRCH02	0F2CR
SRCH03	0F32R
SRCH50	0B4AR
SRCH51	0B5CR
SRCH52	0B74R



SRCH53	0R7CH
SRCH54	0R96R
SRCH55	0R9AR
SRCUPD	0000R
START	000AR
STOP	09B0R
STRTO0	0028R
STRTO1	003CR
STRTO2	0042R
STRTO3	004CR
STRTO4	005AR
STRTO5	007AR
TAB010	03CER
TAB011	03E0R
TAB015	03EER
TAB030	03F6R
TAB040	040ER
TAB045	041AR
TAB050	0426R
TAB910	044AR
TAB911	044ER
TABHDR	0464R
TABLE	03A2R
TABLU	0436R
UPD010	06D0R
UPD020	06F2R
UPD030	0710R
UPD900	071AR
UPD901	071ER
UPDATE	06BER
UPDE10	0920R
UPDE20	093AR
UPDE30	094ER
UPDEND	0916R
UPDINP	0CB8R
VER010	05A6R
VER020	05C4R
VER030	05C8R
VER040	05DER
VER050	0606R
VER060	0618R
VER900	062CR
VER901	0630R
VER950	063ER
VER951	0646R
VER952	064ER
VER960	0656R
VER961	0660R
VERBFR	066ER
VERCNT	0634R
VERIFY	0594R
VERINP	0636R
WFOV	0958R
WFMS	096AR
WRITE	0CC0R
WRTEOF	0CD0R

OS COPY PROGRAM DESCRIPTION

B03-056 A15

## Table of Contents

1. General
  2. Concepts and Conventions
  3. Commands
    - 3.1 Device Commands
      - 3.1.1 Backspace
      - 3.1.2 Write File Mark
      - 3.1.3 Skip Forward File Mark
      - 3.1.4 Rewind
    - 3.2 Function Commands
      - 3.2.1 Copy
      - 3.2.2 Verify
      - 3.2.3 Display
      - 3.2.4 Label
      - 3.2.5 Find
      - 3.2.6 List
      - 3.2.7 Continue
      - 3.2.8 Variable
      - 3.2.9 Fixed
      - 3.2.10 Pause
      - 3.2.11 End
  4. Operating Instructions
- 
- APPENDIX A      Summary of Messages
- APPENDIX B      Sample Copy Procedures
- APPENDIX C      Sample Display Printout

1. General

OS Copy (program number 03-056) is an INTERDATA OS Utility which provides basic file duplication capabilities required by the OS user. Directives, entered by the operator on the teletype allow copying, verifying or displaying of individual files or entire volumes to and from any device supported by the OS. Commands for writing and positioning to file marks and rewinding are provided to facilitate positioning and accessing files. OS COPY requires the following configuration:

- a) A Model 5, 50, 70, 74, 80 or 85 processor
- b) 8KB Memory
- c) a teletype
- d) an input, output and list device (may be the teletype or any sequential device)
- e) an operating system (BOSS, DOS or RTOS)

## 2. Concepts and Conventions

OS COPY communicates to I/O devices by way of logical units. The following is a list of logical units used by OS COPY. Logical units should be assigned to a physical unit by the OS before using the commands of OS COPY.

<u>Logical Unit</u>	<u>Function</u>
1	Acts as the input device for all COPY, VERIFY and DISPLAY operations.
2	Acts as the output device for all COPY and LABEL commands, and as a secondary input device for the VERIFY command.
3	Acts as the list device for DISPLAY operations.
5	Acts as the operator command input device (normally assigned to the teletype).

When executed under R01 of BOSS or DOS, OS Copy can read and write variable length records for copying, verifying or displaying. A directive is supplied to specify whether fixed or variable length records are being processed.

Understanding OS COPY operation with respect to various types of files and formats requires definition of terminology. The following is a list of terms referenced in this document.

- a) File                      a sequence of records terminated by an end-of-file indication (EOF)
- b) ASCII file                a file of ASCII characters, notably source information or program data.
- c) Binary file                a file of binary characters, notably object programs or core image data.  
Example: a program library on cassette.
- d) Volume                    a sequence of files terminated by a double EOF indicator. Two consecutive EOF's denote end of volume (EOV).

TABLE OF INDICATORS

MAG TAPE, DISC, DRUM  
CASSETTE

CARDS OR PAPER TAPE

End of File		End of Volume	
ASCII	BINARY	ASCII	BINARY
file mark	file mark	2 file marks or EOM	2 file marks or EOM
/*	None	/&	ENDVOL Label

### 3. Commands

There are two categories of directives for OS COPY - function commands and device commands. The device commands refer only to bulk storage devices and allow writing and positioning of file marks and rewinding of files. The function commands perform copying, verifying and displaying of ASCII or binary files to and from any device. All commands are terminated with a carriage return. Commands may contain operands which further define the operation to be performed. If a command or operand cannot be recognized by OS COPY, the message CMD-ERR is printed on the teletype and the command is ignored. All decimal number operands must be less than or equal to 65,535.

#### 3.1 Device Commands

The device commands are ignored for non-bulk storage devices. Note that there is no message in this case to the operator to indicate that the command was ignored.

##### 3.1.1 Backspace File

BSP LU,N

where: LU is the logical unit of the device to be backspaced.  
N is the decimal number of file marks to be backspaced.

This command backspaces over N file marks. N is optional and defaults to 1.

Examples: BSP 4,6 Backspace 6 files on LU 4  
          BSP 4 Backspace 1 file on LU 4

##### 3.1.2 Write Filemark

WFM LU

This command writes one file mark on the device specified by LU.

##### 3.1.3 Skip Forward Filemark

SKP LU,N  
SKP LU,ALL

where: LU is the logical unit of the device to be skipped forward.  
N is the decimal number of file marks to be skipped.  
ALL is the literal 'ALL'.

This command skips forward N file marks on device LU, or when 'ALL' is specified the device is positioned forward until end of volume (2 consecutive file marks are encountered). The

device is backspaced over the second file mark in anticipation of file appending. N is optional and defaults to one.

Examples: SKP 2,7 skip forward 7 file marks on LU2  
SKP 2,ALL skip to EOv on LU2  
SKP 2 skip forward 1 file mark on LU2

#### 3.1.4 Rewind

RWD LU

This command rewinds device LU.

### 3.2 Function Commands

#### 3.2.1 Copy

CPYA FN or \*\*NNNNNN,R,N or ALL  
CPYB FN or LLLLLL,R,N or ALL

where: FN is a decimal file number indicating where copying is to begin.

\*\*NNNNNN is an ASCII label where copying is to begin.

R is the record length of the file being copied.

N is the decimal number of files to be copied.

ALL is the literal 'ALL'.

LLLLLL is a binary file label where copying is to begin.

The CPYA command is used to copy ASCII files, while the CPYB command is used to copy binary files. In either case copying is done from LU1 to LU2. In CPYA or CPYB the user may specify a file number (FN) which causes OS COPY to skip FN file marks on LU1 before copying. If labels are specified (\*\*NNNNNN or LLLLLL) then LU1 is searched until the label is found. If the label or file number is not found before EOv is encountered, the message FILE END is printed on the console and the command is aborted. FN is optional and defaults to zero, i.e., first file on the volume.

After positioning via FN, the LU1 remains positioned after the last file mark skipped. When positioning to a label, LU1 is backspaced over the label to include it in the copy operation. In either case, searching begins from where the file is currently positioned when the CPY command is issued. The user should issue a rewind command to LU1 prior to the CPY command if he desires positioning or searching from the beginning of the volume.



If an ASCII file or files are being copied and the 'ALL' option is not specified, OS COPY will pause if a record ~~PAUSE~~ is read. See Section 3.2.7 for instructions to continue.

R is the record length of the file. R is optional and defaults to 80 for ASCII files and 108 for binary files. N determines the number of files to be copied. N is optional and defaults to 1. 'ALL' indicates copying until EOF is encountered. NOTE: See the VARI command for an explanation of variable length record copying.

Examples:

```
CPYA  **SOURCE,80,2   Copy 2 files beginning at label **SOURCE
CPYA  **SOURCE,,ALL  Copy until EOF beginning at label **SOURCE
                        (record length defaults to 80)
CPYB  3,,2           Copy 2 files after skipping 3 file marks
CPYB  ASMBLR         Copy the ASMBLR program
CPYB  ,,ALL          Copy an entire binary volume
CPYB                   Copy one program
```

NOTE: The first or second operand may be defaulted by entering just the comma delimiter.

### 3.2.2 Verify

```
VERA  FN or **NNNNNN,R,N or ALL
VERB  FN or LLLLLL,R,N or ALL
```

The verify command is used to compare files (LU1 to LU2). The operands of the verify command are equivalent to the Copy command and the positioning of LU1 is identical. After positioning, LU1 is compared to LU2. When a discrepancy occurs, the input and output file records are displayed on the list device as described in Section 3.2.3, and OS COPY pauses with the message 'NON VERIFY'.

### 3.2.3 Display

```
DSPA  FN or **NNNNNN,R,N or ALL
DSPB  FN or LLLLLL,R,N or ALL
```

The display command displays files or entire volumes (LU1) onto the list device (LU3). The display format consists of a record number followed by the hexadecimal representation of the record. The line immediately below the hexadecimal representation contains the printer graphic characters associated with the hexadecimal characters. Non-printing characters are represented by a space.

Operands, positioning, record size defaults are as described in Section 3.2.1. N or ALL determine the number of files to be displayed.

Examples:

DSPA 0,80,3	List 3 ASCII files starting at file 0
DSPA **FILE2	List **FILE2
DSPB OBJECT	List the file OBJECT
DSPA ,,ALL	List an entire ASCII volume of files OBJECT

See Appendix 1 for example of Display printout.

### 3.2.4 Label

```
LBLA **NNNNNN
LBLB LLLLLL
```

where: \*\*NNNNNN is an ASCII file label  
LLLLLL is a binary file label

The label command writes a label record for ASCII files (LBLA) or binary files (LBLB). When used, this command should precede the actual copying of a file. The Label command writes the label on LU2. An ASCII label record is 80 bytes long with \*\*NNNNNN as the leading bytes and the remaining bytes undefined. A binary label record is a 108 byte loader format record with a single (label) loader item. If the name specified contains less than 6 characters, blanks are used for right fill.

```
Examples: LBLA **F2
          LBLB PROG
```

### 3.2.5 Find

```
FNDA **NNNNNN,LU
FNDB LLLLLL,LU
```

where: \*\*NNNNNN is an ASCII file label  
LLLLLL is a binary file label

This command will locate an ASCII file label (FNDA) or binary file label (FNDB) on the logical unit specified by LU. The file is backspaced in front of the label, if the label is found. If EOF is encountered before the label is found, the message 'FILE END' is logged on the console. Searching begins from where the file is currently positioned.

```
Examples: FNDA **NAME,1
          FNDB LABEL,1
```

### 3.2.6 List

LSTA LU  
LSTB LU

The List command lists all ASCII label names (LSTA) or binary label names (LSTB) contained on the device specified by LU. Listing begins from where the file is presently positioned. This operation terminates on an End of Volume indicator.

### 3.2.7 Continue

CONT

The Continue directive is used to restart an interrupted OS COPY operation. The following are conditions that may require the Continue command.

- a) A verify operation may terminate with a NON-VERIFY message. The CONT command can be used to continue verifying.
- b) OS COPY pauses and prints the message 'HALT' when a record containing the characters 'PAUSE' preceded by a single blank is read from an ASCII file. The CONT allows the paused operation to continue.
- c) Abnormal device status results in the message I/O-ERR:XXDD where XX is the abnormal status of the device, and DD is the device number. After the operator takes appropriate action, the CONT command can be used to continue the interrupted operation.

Device status considered abnormal by OS COPY may be device unavailable (XX=A0) or unrecoverable error (XX=84).

### 3.2.8 Variable

VARI

OS COPY is initialized to process fixed length records. The Variable command is used to indicate that variable length records are to be copied. The VARI command acts as a mode setting. All OS COPY operations after the VARI command are variable length operations. When copying variable length records OS COPY writes the exact number of bytes for each record. This feature is necessary only when copying programs with bootstrap front ends. Verify and Display operations are also handled in variable mode. The VARI command may only be

used under BOSS R01 and above, or DOS R01 and above. Variable operations are valid only on Mag tape, cassette, high speed paper tape reader or Teletype. Variable binary operations are not allowed on paper tape devices.

In using R01 of BOSS to process variable length records the following patches are necessary:

MTR1+2 should be X'2178'  
SKIP should be X'0200'

Refer to the symbol table on a BOSS listing (03-019R01A13) to determine the actual locations.

### 3.2.9 Fixed

FIXD

The FIXD command resets OS COPY to process fixed length records.

### 3.2.10 Pause

P

The P command pauses OS COPY and returns to the Operating System. The applicable Operating System command may be issued to continue.

### 3.2.11 End

END

The END command causes OS COPY to return to the Operating System.

#### 4. Operating Instructions

The following are instructions for loading and executing OS COPY.

- a) Load an operating system: BOSS, DOS or RTOS. If RTOS is used, OS COPY must be established as an RTOS task. (See RTOS Reference Manual for details).
- b) OS COPY is a relocatable program, and may be loaded at any desired BIAS with the OS Resident Loader or the OS Library Loader.
- c) Make the desired device assignments. LU5 should be assigned to the Teletype for command input processing.
- d) Start OS COPY at the origin. The restart address is also the origin.

OS COPY requires approximately 3.5KB of memory, not including buffer area for copying. The buffer area required is equal to the record length specified in the copy operation. For verify operations, a buffer of twice the record length is required. The maximum record length allowed is limited only by the amount of memory available on the processor. If the specified record length exceeds the amount of memory available, the message MEM-FULL will result, indicating not enough memory is available

#### NOTE

When loading any other program into memory above OS COPY, the loader bias should be adjusted to leave enough room for the necessary buffer area.

When establishing OS COPY as a task under RTOS, a GET Storage command should be issued to obtain the required buffer space for the program.

APPENDIX A  
SUMMARY OF MESSAGES

CMD-ERR	Illegal command entered
PAUSE	∅ PAUSE record encountered
NON VERIFY	verify operation mismatch
FILE END	EOV encountered while searching (LU1)
I/O ERR XXDD	Device unavailable or unrecoverable error on device DD.
MEM-FULL	Record length exceeds available memory.

APPENDIX B  
SAMPLE COPY PROCEDURES

- 1) To copy a binary paper tape to cassette under BOSS, make the following assignments:

AS 1,13	Assign LU1 to paper tape
AS 2,45	Assign LU2 to cassette

and enter the following OS COPY command:

CPYB

This command defaults to:

- a) no skipping prior to copying
- b) 108 bytes loader format records
- c) one file to be copied

This command will be terminated by an ENDEVOL label record or EOM on paper tape.

- 2) To copy a source disc volume to mag tape under DOS, make the following assignments:

AC FILE,1	Assign file name 'FILE' to LU1
AS 285	Assign LU2 to mag tape

and enter the following OS COPY command:

CPYA ,,ALL

This command defaults to a) no skipping prior to copying  
b) 80 byte ASCII records

This command will be terminated by a /& record, EOVS, or EOM on disc.

- 3) To copy a binary cassette to another cassette under BOSS make the following assignments:

```
AS 1,45    assign LU1 to cassette input
AS 2,55    assign LU2 to cassette output
```

and enter the following OS COPY command

```
CPYB , ,ALL
```

This command defaults to a) no skipping prior to copying  
b) 108 byte loader format records

This command terminates on EOVS or EOM on cassette.

- 4) To copy the source of DOS from a mag tape file to cassette, make the following assignment:

```
AS 185,245
```

and enter the following OS COPY command

```
CPYA **DOS
```

This command defaults to a) 80 byte ASCII records  
b) 1 file to be copied

This command searches for file \*\*DOS and terminates on EOF.

- 5) To verify the previous example issue the following OS COPY command:

```
BSP 1,2
BSP 2,2
VERA
```

APPENDIX C

RECNO 0 1 2 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F

00001 31202020202052454D205553494E47205445535420312D2D535452494E472044  
I R E M U S I N G T E S T I - - S T R I N G D  
41544120  
A T A  
20202020202020202020202020202020

00002 35202020202044494D41242831292020202020202020202020202020202020  
5 D I M A \$ ( 1 )  
20  
20202020202020202020202020202020

00003 31302020202044494D20422428352920202020202020202020202020202020  
1 0 D I M B \$ ( 5 )  
20  
20202020202020202020202020202020

00004 31352020202044494D43242831302920202020202020202020202020202020  
1 5 D I M C \$ ( 1 0 )  
20  
20202020202020202020202020202020

00005 32302020202044494D44242832302920202020202020202020202020202020  
2 0 D I M D \$ ( 2 0 )  
20  
20202020202020202020202020202020

00006 32352020202044494D46312428322920202020202020202020202020202020  
2 5 D I M F 1 \$ ( 2 )  
20  
20202020202020202020202020202020



\*  
 \* COPYRIGHT INTERDATA, INC. JUNE, 1973  
 \*  
 \* AUTHOR: J. PRATT  
 \*

CPY00030  
 CPY00040  
 CPY00050  
 CPY00060  
 CPY00070  
 CPY00080  
 CPY00090  
 CPY00100  
 CPY00110  
 CPY00120  
 CPY00130  
 CPY00140  
 CPY00150  
 CPY00160  
 CPY00170  
 CPY00180  
 CPY00190  
 CPY00200  
 CPY00210  
 CPY00220  
 CPY00230  
 CPY00240  
 CPY00250  
 CPY00260  
 CPY00270  
 CPY00280  
 CPY00290

\* REGISTER EQUATIONS

0000 R0 EQU 0  
 0001 LINK EQU 1  
 0002 RECRDS EQU 2  
 0003 R3 EQU 3  
 0004 LASTAD EQU 4  
 0005 COMAND EQU 5  
 0006 ERROR EQU 6  
 0007 CMDER EQU 7  
 0008 RTN EQU 8  
 0009 R9 EQU 9  
 000A R10 EQU 10  
 000B R11 EQU 11  
 000C R12 EQU 12  
 000D R13 EQU 13  
 000E NUMBER EQU 13  
 000F FROM EQU 14  
 000F TO EQU 15

-----\*

0000R C8D0 OSCOPY LHI NUMBER,CLREND-CLRBEG  
 0024  
 0004R C8F0 LHI TO,CLRBEG  
 0F34R  
 0008R 07EE XHR FROM, FROM CLEAR VALUE  
 000AR 4110 BAL LINK,CLEAR CLEAR PARAMETERS ETC.  
 092AR  
 000ER C8E0 LHI FROM,X'2020' CLEARING VALUE  
 2020  
 0012R C8F0 LHI TO,BUFFER START ADDRESS  
 0EC8R  
 0016R C8D0 LHI NUMBER,SPCEND-BUFFER  
 0058  
 001AR 4110 BAL LINK,CLEAR CLEAR BUFFER  
 092AR  
 001ER D100 NAME LM R0,REGSAV LOAD REGS  
 0F22R  
 0022R E120 SVC 2,NAMBLK  
 0152R

-----\*

\* COMMAND PROCESSOR  
 0026R E110 COMD SVC 1,CMDBLK READ A COMMAND  
 00C0R  
 002AR 4800 LH R0,CMDBLK+2 TEST STATUS  
 00C2R  
 002ER 0216 BMR ERROR ERROR  
 0030R D1E0 LM FROM,BUFFER GET DIRECTIVE  
 0EC8R  
 0034R C8B0 LHI R11,CMDEND-CMDLST LIST LENGTH  
 008A

CPY00390  
 CPY00400  
 CPY00410  
 CPY00420  
 CPY00430  
 CPY00440  
 CPY00450

0038R	45E8	CMD1	CLH	FROM,CMDLST-4(R11)	CHECK ENTRY	CPY00460
	00C4R					
003CR	2334		HES	CMD3	LOCATED	CPY00470
003ER	27B6	CMD2	SIS	R11,6	DECREMENT INDEX	CPY00480
0040R	2214		BNMS	CMD1	CONTINUE SEARCH	CPY00490
0042R	0307		BR	CMDER	COMMAND ERROR	CPY00500
0044R	45FB	CMD3	CLH	TO,CMDLST-2(R11)	CHECK NEXT TWO CHARACTERS	CPY00510
	00C6R					
0048R	2035		BNES	CMD2		CPY00520
004AR	481B		LH	LINK,CMDLST-6(R11)	PICK UP ADDRESS	CPY00530
	00C2R					
004ER	C310		THI	LINK,1	CHECK PARSING FLAG	CPY00540
	0001					
0052R	0231		BNZR	LINK	EXIT TO HANDLER	CPY00550
0054R	C890		LHI	R9,-1	CLEAR INDEX	CPY00560
	FFFF					
0058R	C8A0	CMD4	LHI	R10,CMD5	SET RETURN ADDR	CPY00570
	005CR					
005CR	2691	CMD5	AIS	R9,1	INCREMENT INDEX	CPY00580
005ER	D309		LB	R0,BUFFER(R9)	GET A CHARACTER	CPY00590
	0EC8R					
0062R	C590		CLHI	R9,BUFEND-BUFFER	CHECK FOR BUFFER END	CPY00600
	0050					
0066R	0331		BER	LINK	YES - EXIT	CPY00610
0068R	C500		CLHI	R0,X'0D'	CHECK FOR CR	CPY00620
	000D					
006CR	0331		BER	LINK	YES - EXIT	CPY00630
006ER	C500		CLHI	R0,X'20'	CHECK FOR SPACE	CPY00640
	0020					
0072R	023A		BNER	R10	NO - CONTINUE SCAN	CPY00650
0074R	C8A0		LHI	R10,PARSER	YES - SET SCAN FOR FIRST CHARACTER	CPY00660
	007CR					
0078R	4300		H	CMD5	CONTINUE	CPY00670
	005CR					
007CR	07FF	PARSER	XHR	TO,TO	OPERAND INDEX	CPY00680
007ER	07DD		XHR	NUMBER,NUMBER	LENGTH OF OPERAND	CPY00690
0080R	D309	CMD6	LB	R0,BUFFER(R9)	GET A CHARACTER	CPY00700
	0EC8R					
0084R	2691		AIS	R9,1	BUMP INDEX	CPY00710
0086R	C590		CLHI	R9,BUFEND-BUFFER	INDEX = END ?	CPY00720
	0050					
008AR	4380		BNL	CMD7	YES - FINISH	CPY00730
	00ACR					
008ER	C500		CLHI	R0,C','	CHECK FOR SPACE	CPY00740
	0020					
0092R	2239		HES	CMD6	YES - IGNORE	CPY00750
0094R	C500		CLHI	R0,C','	CHECK FOR COMMA	CPY00760
	002C					
0098R	233A		BES	CMD7		CPY00770
009AR	C500		CLHI	R0,X'0D'	CHECK FOR CR	CPY00780
	000D					
009ER	2337		HES	CMD7		CPY00790
00A0R	26F1		AIS	TO,1	INCREMENT	CPY00800
00A2R	D20F		STB	R0,OPRAND(TO)	STORE CHARACTER	CPY00810
	0F34R					
00A6R	26D1		AIS	NUMBER,1	BUMP LENGTH ATTRIBUTE	CPY00A20

00A8R	4300 0080R		H	CMD6	CONTINUE		CPY00830
00ACK	08EF	CMD7	LHR	FROM,TO			CPY00840
00AER	0BED		SHR	FROM,NUMBER	CALCULATE ATTRIBUTE DISPLACEMENT		CPY00850
00B0R	02DE		STB	NUMBER,OPRAND(FROM)	STORE LENGTH ATTRIBUTE		CPY00860
	0F34R						
00B4R	26F1		AIS	TO,1	BUMP OPERAND INDEX		CPY00870
00B6R	C590		CLHI	R9,BUFEND-BUFFER	CHECK FOR END		CPY00880
	0050						
00BAR	0381		BNLR	LINK	YES - EXIT		CPY00890
00RCR	4300		B	PARSER+2	CONTINUE PARSING		CPY00900
	007ER						
00C0R	4805	CMDBLK	DC	X*4805*,0,BUFFER,BUFFER+79			CPY00910
	0000						
	0EC8R						
	0F17R						
00C8R	015ER	CMDLST	DC	CPYA,C*CPYA*			CPY00920
	4350						
	5941						
00CER	016CR		DC	CPYB,C*CPYB*			CPY00930
	4350						
	5942						
00D4R	02A4R		DC	VERA,C*VERA*			CPY00940
	5645						
	5241						
00DAR	02BAR		DC	VERB,C*VERB*			CPY00950
	5645						
	5242						
00E0R	0404R		DC	DSPA,C*DSPA*			CPY00960
	4453						
	5041						
00F6R	0412R		DC	DSPB,C*DSPB*			CPY00970
	4453						
	5042						
00ECR	0C52R		DC	LBLA,C*LBLA*			CPY00980
	4C42						
	4C41						
00F2R	0C96R		DC	LBLB,C*LBLB*			CPY00990
	4C42						
	4C42						
00F8R	0A3ER		DC	LSTA,C*LSTA*			CPY01000
	4C53						
	5441						
00FER	0AFER		DC	LSTB,C*LSTB*			CPY01010
	4C53						
	5442						
0104R	07A4R		DC	NVR,C*FIXD*			CPY01020
	4649						
	5844						
010AR	0B96R		DC	FNDA,C*FNDA*			CPY01030
	464E						
	4441						
0110R	0BF8R		DC	FNDB,C*FNDB*			CPY01040
	464E						
	4442						
0116R	05B8R		DC	RWD,C*RWD*			CPY01050

5257					
4420					
011CR	05DCR	DC	WFM,C'WFM'		CPY01060
5746					
4020					
0122R	0678R	DC	BSP,C'BSP'		CPY01070
4253					
5020					
0128R	07CFR	DC	END+1,C'END'		CPY01080
454E					
4420					
012ER	07CFR	DC	END+1,X'454E',X'440D'		CPY01090
454E					
4400					
0134R	07AER	DC	VAR,C'VARI'		CPY01100
5641					
5249					
013AR	0077R	DC	CONT+1,C'CONT'		CPY01110
434F					
4E54					
0140R	0600K	DC	SKP,C'SKP'		CPY01120
534B					
5020					
0146R	07C9R	DC	PAWS+1,C'P'		CPY01121
5020					
2020					
014CR	07C9R	DC	PAWS+1,X'500D'		CPY01122
500D					
0150H	2020	DC	X'2020'		CPY01123
0152R		CMDEND EQU	*		CPY01130
0152R	0007	NAMBLK DC	7,NAMEND-NAMST		CPY01140
0008					
0156R	4F53	NAMST DC	C'OS COPY'		CPY01150
2043					
4F50					
5920					
015ER		NAMEND EQU	*		CPY01160
					CPY01170
					CPY01180
					CPY01190
015ER	C890	CPYA	LHI R9,X'2020'		
2020					
0162R	4090		STH R9,CLRVAL		CPY01200
0F4ER					
0166R	C890		LHI R9,X'2848'	WRITE/READ ASCII	CPY01210
2848					
016AR	2303		BS COPY		CPY01220
016CR	C890	CPYB	LHI R9,X'3858'	WRITE/READ BINARY	CPY01230
3858					
0170R	D290	COPY	STB R9,RDBLK	PRIME READ BLOCK	CPY01240
0832R					
0174R	9098		SRLS R9,8		CPY01250
0176R	D290		STB R9,WRBLK	PRIME WRITE BLOCK	CPY01260
084ER					
017AR	0799		XHR R9,R9		CPY01270
017CR	4090		STH R9,FMKFLG	CLEAR FILEMARK FLAG	CPY01280
0F50R					

0180R	4180		BAL	R11,PROC	PROCESS COMMAND	CPY01290
	007CR					
0184R	CAED	CP2	AHI	FROM,1(NUMBER)	BUMP TO OPERAND	CPY01300
	0001					
0188R	D3DE		LB	NUMBER,-1(FROM)	PICK UP NEXT LENGTH	CPY01310
	FFFF					
018CR	07BR		XHR	R11,R11	CLEAR FILE COUNTER	CPY01320
018ER	E120		SVC	2,FETCH	FETCH POINTER	CPY01330
	02A0R					
0192R	4834		LH	R3,0(LASTAD)	CTOP	CPY01331
	0000					
0196R	4830		SH	R3,RECSIZ	RECORD SIZE	CPY01332
	029ER					
019AR	C430		NHI	R3,X'FFFE'	HALFWORD BOUNDARY	CPY01333
	FFFF					
019ER	4534		CLH	R3,4(LASTAD)	COMPARE TO UTOP	CPY01334
	0004					
01A2R	4280		BL	MEMERR	LESS = MEMORY FULL	CPY01335
	0540R					
01A6R	0893		LHR	R9,R3		CPY01340
01A8R	4090		STH	R9,RDBLK+4	PRIME I/O BLOCK	CPY01350
	0836R					
01ACR	4090		STH	R9,WRBLK+4	PRIME I/O BLOCK	CPY01360
	0852R					
01B0R	4A90		AH	R9,RECSIZ	CALC REC END	CPY01370
	029ER					
01B4R	2791		SIS	R9,1		CPY01380
01B6R	4090		STH	R9,RDBLK+6	PRIME I/O BLOCK	CPY01390
	0838R					
01BAR	4090		STH	R9,WRBLK+6	PRIME I/O BLOCK	CPY01400
	0854R					
01BER	0722		XHR	RECRDS,RECRDS	CLEAR RECORDS COUNTER	CPY01410
01C0R	08DD		LHR	NUMBER,NUMBER	DEFAULT ?	CPY01430
01C2R	233F		BZS	CP4	YES	CPY01440
01C4R	C8F0		LHI	TO,QUANT	SET RECEIVER	CPY01450
	0F56R					
01C8R	D3BE		LB	R11,0(FROM)	GET OPERAND	CPY01460
	0000					
01CCR	C5B0		CLHI	R11,X'41'	CHECK FOR 'ALL'	CPY01470
	0041					
01D0R	2334		BES	CP3		CPY01480
01D2R	4110		BAL	LINK,XLATDB	TRANSLATE DECIMAL TO BINARY	CPY01490
	0896R					
01D6R	2305		RS	CP4		CPY01500
01D8R	C8B0	CP3	LHI	R11,-1	SET FLAG FOR 'ALL'	CPY01510
	FFFF					
01DCR	40B0		STH	R11,QUANT	SAVE NO. FILES	CPY01520
	0F56R					
01E0R	E120	CP4	SVC	2,FETCH	FETCH CONTAB POINTER	CPY01530
	02A0R					
01E4R	48B0		LH	R11,VARFLG	CHECK VARIABLE LENGTH FLAG	CPY01540
	0F20R					
01E8R	2333		BZS	CP5		CPY01550
01EAR	C840		LHI	LASTAD,RDBLK+8	FIXED LENGTH OPNS	CPY01560
	083AR					
01EER	48E0	CP5	LH	FROM,CLRVAL	CLEAR VALUE	CPY01570

01F2R	0F4CR 48DD		LH	NUMBER,RECSIZ	RECORD SIZE	CPY01580
01F6R	029FK 48F0		LH	TO,RDBLK+4	SET RECEIVER ADDRESS	CPY01590
01FAR	0836R 4110		BAL	LINK,CLEAR	CLEAR BUFFER APPROPRIATELY	CPY01600
01FER	092AR 4110		BAL	LINK,READ	READ A RECORD	CPY01610
0202R	07F2R 48B0		LH	R11,FMKFLG	CHECK FILEMARK FLAG	CPY01620
0206R	0F50R 4210		BM	CPEOV1	MINUS = TWO FILEMARKS	CPY01630
020AR	0292R 4230		BNZ	CPEOF	NOT ZERO = ONE FILEMARK	CPY01640
020ER	0272R 48B0		LH	R11,CLRVAL		CPY01650
0212R	0F4ER 4330		BZ	CP8	BINARY	CPY01660
0216R	0266R 48BF	CP6	LH	R11,0(TO)	GET HALFWORD FROM BUFFER	CPY01670
021AR	0000 C5B0		CLHI	R11,C'/*'	EOF ?	CPY01680
021ER	2F2A 4330		BE	CPEOF		CPY01690
0222R	0272R C5B0		CLHI	R11,C'/*'	EOF ?	CPY01700
0226R	2F26 4330		BE	CPEOV		CPY01710
022AR	028CR 2621	CP7	AIS	RECRDS,1	INCREMENT RECORD COUNTER	CPY01720
022CR	48B4		LH	R11,-2(LASTAD)	PICK UP RECORD SIZE	CPY01730
0230R	FFFE 40B0		STH	R11,WRBLK+6	PRIME I/O BLOCK	CPY01740
0234R	0854R 4110		BAL	LINK,WRITE	WRITE A RECORD	CPY01750
0238R	083AR 48BF		LH	R11,0(TO)	GET HALFWORD FROM BUFFER	CPY01760
023CR	0000 C5B0		CLHI	R11,C' P'	PAUSE ?	CPY01770
0240R	2050 4230	CP7A	BNE	CP5	SKIP PAUSE HALT	CPY01780
0244R	01EER 48BF		LH	R11,2(TO)		CPY01790
0248R	0002 C5B0		CLHI	R11,C'AU'		CPY01800
024CR	4155 2036		BNES	CP7A		CPY01810
024ER	48BF		LH	R11,4(TO)		CPY01820
0252R	0004 C5B0		CLHI	R11,C'SE'		CPY01830
0256R	5345 203B		BNES	CP7A		CPY01840
0258R	4890		LH	R9,QUANT	CHECK QUANTITY	CPY01850
025CR	0F56R 2113		BMS	CP7B	SKIP PAUSE HALT	CPY01860
025ER	4110		BAL	LINK,HALT	HALT	CPY01870

0262R	0D64R 4300	CP7B	B	CP5	CONTINUE	CPY01880
0266R	01EER 4180	CP8	BAL	RTN,VOLCHK	CHECK FOR ENDVOL	CPY01890
026AR	0E36R 4300		B	CPEOV	END OF VOLUME RETURN	CPY01900
026FR	028CR 4300		B	CP7	CONTINUE	CPY01910
0272R	022AR 2432	CPEOF	LIS	R3,2	SET LOGICAL UNIT	CPY01920
0274R	4110		BAL	LINK,WFMK	WRITE A FILEMARK	CPY01930
0278R	05ECR 48B0		LH	R11,QUANT	PICK UP NO. FILES	CPY01940
027CR	0F56R 0335	CP10	BZR	COMAND	DONE	CPY01950
027ER	2115		BMS	CP11	MINUS = ALL	CPY01960
0280R	27B1		SIS	R11,1	DECREMENT COUNTER	CPY01970
0282R	0335		BZR	COMAND	DONE	CPY01980
0284R	40B0		STH	R11,QUANT	NO	CPY01990
0288R	0F56R 4300	CP11	B	CP5	CONTINUE	CPY02000
028CR	01EER 2432	CPEOV	LIS	R3,2	SET LOGICAL UNIT	CPY02010
028ER	4110		BAL	LINK,WFMK	WRITE A FILEMARK	CPY02020
0292R	05ECR 2432	CPEOV1	LIS	R3,2	SET LOGICAL UNIT FOR ALT. ENTRY	CPY02025
0294R	4110		BAL	LINK,WFMK		CPY02030
0298R	05ECR 0305		BR	COMAND		CPY02040
029AR	0002	GETCOR	DC	2,R3		CPY02050
029ER	0003					
029CR	0000	RECSIZ	DC	0		CPY02060
02A0R	0005	RELCOR	EQU	GETCOR+2		CPY02070
	0004	FETCH	DC	5, LASTAD		CPY02080
*-----*						
				VERIFY		CPY02090
02A4R	C890	VERA	LHI	R9,X'48'	ASCII READ	CPY02100
	0048					CPY02110
02A8R	D290		STB	R9,RDBLK		CPY02120
02ACR	0832R D290		STB	R9,RDBLK2		CPY02130
02B0R	03FCR C8A0		LHI	R10,X'2020'	CLEARING VALUE	CPY02140
02B4R	2020 40A0		STH	R10,CLRVAL	SAVE CLEAR VALUE / FLAG	CPY02150
02B8R	0F4ER 2307		BS	VERIFY		CPY02160
02BAR	C890	VERB	LHI	R9,X'58'	BINARY READ	CPY02170
02BER	0058 D290		STB	R9,RDBLK	BINARY READ PROCEED	CPY02180
02C2R	0832R D290		STB	R9,RDBLK2		CPY02190
02C6R	03FCR 0799	VERIFY	XHR	R9,R9		CPY02200
02C8R	4090		STH	R9,FMKFLG	CLEAR FILEMARK FLAG	CPY02210

02CCR	0F50R 4180	BAL	RTN,PROC	PROCESS COMMAND	CPY02220
02D0R	0D7CR CAED	VR2 AHI	FROM,1(NUMBER)	BUMP TO NEXT OPERAND	CPY02230
02D4R	0001 D3DE	LB	NUMBER,-1(FROM)	PICK UP LENGTH	CPY02240
02D8R	FFFF E120	SVC	2,FETCH	FETCH POINTER	CPY02250
02DCR	02A0R 4834	LH	R3,0(LASTAD)	CTOP	CPY02251
02E0R	0000 4B30	SH	R3,RECSIZ	SUB TRACT RECORD SIZE	CPY02252
02E4R	029ER C430	NHI	R3,X'FFFE'	HALF WORD BOUNBARY	CPY02253
02E8R	FFFE 4534	CLH	R3,4(LASTAD)	COMPARE TO UTOP	CPY02254
02ECR	0004 4280	BL	MEMERR	LESS = MEMORY FULL	CPY02255
02F0R	0540R 4030	STH	R3,RDBLK+4	PRIME I/O BLOCK	CPY02260
02F4R	0836R 4A30	AH	R3,RECSIZ	CALCULATE BUFFER END	CPY02270
02F8R	029ER 2731	SIS	R3,1	END ADDRESS	CPY02280
02FAR	4030	STH	R3,RDBLK+6	END ADDRESS	CPY02290
02FER	0838R E120	SVC	2,FETCH	FETCH POINTER	CPY02300
0302R	02A0R 4834	LH	R3,0(LASTAD)	CTOP	CPY02301
0306R	0000 4B30	SH	R3,RECSIZ	SUB TRACT RECORD SIZE	CPY02302
030AR	029ER 4B30	SH	R3,RECSIZ	FIRST RECORD	CPY02303
030ER	029ER C430	NHI	R3,X'FFFE'	HALF WORD BOUNBARY	CPY02304
0312R	FFFE 4534	CLH	R3,4(LASTAD)	COMPARE TO UTOP	CPY02305
0316R	0004 4280	BL	MEMERR	LESS = MEMORY FULL	CPY02306
031AR	0540R 4030	STH	R3,RDBLK2+4	PRIME I/O BLOCK	CPY02310
031ER	0400R 4A30	AH	R3,RECSIZ	CALCULATE BUFFER END	CPY02320
0322R	029ER 2731	SIS	R3,1	END ADDRESS	CPY02330
0324R	4030	STH	R3,RDBLK2+6	END ADDRESS	CPY02340
0328R	0402R 0722	XHR	RECRDS,RECRDS	CLEAR RECORD COUNTER	CPY02370
032AR	080D	LHR	NUMBER,NUMBER	DEFAULT?	CPY02380
032CR	233F	BZS	VR5	YES	CPY02390
032ER	03RE	LB	R11,0(FROM)	PICK UP OPERAND BYTE	CPY02400
0332R	0000 C5B0	CLHI	R11,X'41'	'ALL' ?	CPY02410
0336R	0041 2336	BES	VR3	YES	CPY02420
0338R	C8F0	LHI	TO,QUANT	SET RECEIVER	CPY02430



033CR	0F56R 4110		BAL	LINK,XLATDB	TRANSLATE DECIMAL TO BINARY	CPY02440
	0896R					
0340R	2305		BS	VR5		CPY02450
0342R	C8B0	VR3	LHI	R11,-1		CPY02460
	FFFF					
0346R	40B0	VR4	STH	R11,QUANT	SET QUANTITY	CPY02470
	0F56R					
034AR	48F0	VR5	LH	TO,RDBLK+4	SET RECEIVER	CPY02480
	0836R					
034ER	48D0		LH	NUMBER,RECSIZ	GET RECORD SIZE	CPY02490
	029ER					
0352R	48E0		LH	FROM,CLRVAL	CLEARING VALUE	CPY02500
	0F4ER					
0356R	4110		BAL	LINK,CLEAR	CLEAR BUFFER	CPY02510
	092AR					
035AR	48F0		LH	TO,RDBLK2+4	SET RECEIVER	CPY02520
	0400R					
035ER	4110		BAL	LINK,CLEAR	CLEAR BUFFER	CPY02530
	092AR					
0362R	4110		BAL	LINK,READ	READ A RECORD	CPY02540
	07F2R					
0366R	48B0		LH	R11,FMKFLG	CHECK FILEMARK FLAG	CPY02550
	0F50R					
036AR	4210		BM	NOFND	END OF VOLUME	CPY02560
	0884R					
036ER	4230		HNZ	VREOF	END OF FILE	CPY02570
	03DCR					
0372R	48B0		LH	R11,CLRVAL	TEST CLEARING VALUE	CPY02580
	0F4ER					
0376R	4330		BZ	VR10		CPY02590
	03F0R					
037AR	48B0		LH	R11,RDBLK+4	PICK UP BUFFER ADDRESS	CPY02600
	0836R					
037ER	48AR		LH	R10,0(R11)		CPY02610
	0000					
0382R	C5A0		CLHI	R10,C'/*'	UNIT RECORD EOF ?	CPY02620
	2F2A					
0386R	4330		BE	RDEOF	SET EOF FLAG	CPY02630
	0828R					
038AR	C5A0	VR6	CLHI	R10,C'/*'	UNIT RECORD EOF ?	CPY02640
	2F26					
038ER	2133		BNES	VR7		CPY02650
0390R	4300		R	RDEOV	SET EOF FLAG	CPY02660
	081ER					
0394R	2621	VR7	AIS	RECRDS,1	INCREMENT RECORD COUNT	CPY02670
0396R	E110		SVC	1,RDBLK2	READ A RECORD	CPY02680
	03FCR					
039AR	4800		LH	R0,RDBLK2+2	CHECK STATUS	CPY02690
	03FER					
039ER	2330		BZS	VR8	OK - VERIFY	CPY02700
03A0R	C300		THI	R0,X'800'	CHECK FOR EOF	CPY02710
	0800					
03A4R	2134		BNZS	FLGCHK	YES - CHECK FLAG	CPY02720
03A6R	C300		THI	R0,X'1000'	CHECK FOR EOM	CPY02730
	1000					

03AAR	0336		BZR	ERROR	NO - ERROR	CPY02740
03ACR	48B0	FLGCHK	LH	R11,FMKFLG		CPY02750
	0F50R					
03BOR	4330		BZ	NONVER		CPY02760
	09A2R					
03B4R	4300		B	VR5	CONTINUE	CPY02770
	034AR					
03B8R	48D0	VR8	LH	NUMBER,RDBLK+4	ADDRESS OF BUFFER ONE	CPY02780
	0836R					
03BCR	24E1		LIS	FROM,1	INCREMENT	CPY02790
03BER	48F0		LH	TO,RDBLK+6	END OF BUFFER ONE	CPY02800
	0838R					
03C2R	48A0		LH	R10,RDBLK2+4	ADDRESS OF BUFFER TWO	CPY02810
	0400R					
03C6R	03RA	VR9	LB	R11,0(R10)		CPY02820
	0000					
03CAR	04B0		CLB	R11,0(NUMBER)	COMPARE CONTENTS	CPY02830
	0000					
03CER	4230		BNE	NONVER	NOT EQUAL	CPY02840
	09A2R					
03D2R	26A1		AIS	R10,1		CPY02850
03D4R	C1D0		RXLE	NUMBER,VR9	CONTINUE COMPARING	CPY02860
	03C6R					
03D8R	4300		B	VR5		CPY02870
	034AR					
03DCR	48B0	VRFOF	LH	R11,QUANT	NO. OF FILES	CPY02880
	0F56R					
03E0R	0335	VRFOF2	BZR	COMAND	FINISHED	CPY02890
03E2R	2115		BMS	VRFOF3	ALL = CONTINUE	CPY02900
03E4R	27B1		SIS	R11,1	DECREMENT COUNT	CPY02910
03E6R	0335		BZR	COMAND	DONE	CPY02920
03E8R	40B0		STH	R11,QUANT		CPY02930
	0F56R					
03ECR	4300	VRFOF3	B	VR7	CONTINUE	CPY02940
	0394R					
03FOR	4180	VR10	BAL	RTN,VOLCHK	CHECK FOR ENDVOL	CPY02950
	0E36R					
03F4R	4305		B	0(COMAND)	DONE	CPY02960
	0000					
03F8R	4300		B	VR7	CONTINUE	CPY02970
	0394R					
03FCR	0002	RDBLK2	DC	2,0,0,0		CPY02980
	0000					
	0000					
	0000					
*-----*						CPY02990
* DISPLAY						CPY03000
0404R	C890	DSPA	LHI	R9,X'48'	ASCII READ	CPY03010
	0048					
0408R	D290		STB	R9,RDBLK		CPY03020
	0832R					
040CR	C890		LHI	R9,X'2020'	CLEARING VALUE	CPY03030
	2020					
0410R	2306		BS	DS1		CPY03040
0412R	C890	DSPB	LHI	R9,X'58'	BINARY READ	CPY03050
	0058					

0416R	D290		STB	R9,RDBLK		CPY03060
	0832R					
041AR	0799		XHR	R9,R9		CPY03070
041CR	4090	DS1	STH	R9,CLRVAL		CPY03080
	0F4ER					
0420R	0799		XHR	R9,R9		CPY03090
0422R	4090		STH	R9,FMKFLG	CLEAR FILEMARK FLAG	CPY03100
	0F50R					
0426R	0722		XHR	RECRDS,RECRDS	CLEAR RECORDS COUNT	CPY03110
0428R	4180		BAL	RTN,PROC	PROCESS COMMAND	CPY03120
	0D7CR					
042CR	E120		SVC	2,FETCH	FETCH POINTER	CPY03130
	02A0R					
0430R	4834		LH	R3,0(LASTAD)	CTOP	CPY03131
	0000					
0434R	4830		SH	R3,RECSIZ	SUB TRACT RECORD SIZE	CPY03132
	029ER					
0438R	C430		NHI	R3,X'FFFE'	HALF WORD BOUNDARY	CPY03133
	FFFE					
043CR	4534		CLH	R3,4(LASTAD)	COMPARE TO UTOP	CPY03134
	0004					
0440R	4280		BL	MEMERR	LESS = MEMORY FULL	CPY03135
	0540R					
0444R	4890		LH	R9,RECSIZ	GET STORAGE	CPY03150
	029ER					
0448R	4030		STH	R3,RDBLK+4	PRIME I/O BLOCK	CPY03160
	0836R					
044CR	CA39		AHI	R3,-1(R9)	BUMP FOR BUFFER END	CPY03170
	FFFF					
0450R	4030		STH	R3,RDBLK+6	PRIME END ADDRESS	CPY03180
	0838R					
0454R	0788		XHR	R11,R11		CPY03190
0456R	CAED		AHI	FROM,1(NUMBER)	BUMP TO NEXT OPERAND	CPY03200
	0001					
045AR	D3DE		LB	NUMBER,-1(FROM)	PICK UP LENGTH ATTRIBUTE	CPY03210
	FFFF					
045ER	08DD		LHR	NUMBER,NUMBER	CHECK FOR DEFAULT	CPY03220
0460R	233D		BZS	DS7		CPY03230
0462R	D38E		LB	R11,0(FROM)	CHECK FOR 'ALL'	CPY03240
	0000					
0466R	C8F0		LHI	TO,QUANT	SET RECEIVER	CPY03250
	0F56R					
046AR	C5B0		CLHI	R11,X'41'		CPY03260
	0041					
046ER	2334		BES	DS6	YES	CPY03270
0470R	4110		BAL	LINK,XLATDB	TRANSLATE	CPY03280
	0896R					
0474R	2305		BS	DS7+4	GET CONTAB POINTER	CPY03290
0476R	C8B0	DS6	LHI	R11,-1	SET VALUE NEGATIVE	CPY03300
	FFFF					
047AR	40B0	DS7	STH	R11,QUANT	SAVE VALUE	CPY03310
	0F56R					
047ER	E120		SVC	2,FETCH	GET CONTAB POINTER	CPY03320
	02A0R					
0482R	48A0		LH	R10,VARFLG	CHECK VAR - FIXED FLAG	CPY03330
	0F20R					

0486R 2333		BZS DS8	VARIABLE	CPY03340
0488R C840		LHI LASTAD,RDBLK+8	SET FOR FIXED LENGTH RECORDS	CPY03350
083AR				
048CR 4110	DS8	BAL LINK,HEDPRT	PRINT HEADERS	CPY03360
0552R				
0490R E110	DS9	SVC 1,SKIPLN	SKIP A LINE	CPY03370
0766R				
0494R 4110		BAL LINK,READ	READ A RECORD	CPY03380
07F2R				
0498R 4880		LH R11,FMKFLG	CHECK FILEMARK FLAG	CPY03390
0F50R				
049CR 4230		BNZ DSCHEK	DETERMINE IF FINISHED	CPY03400
0516R				
04A0R 4880		LH R11,CLRVAL	CLEARING VALUE	CPY03410
0F4ER				
04A4R 4330		BZ DS15	CHECK FOR ENDEVOL	CPY03420
0534R				
04A8R 4880		LH R11,RDBLK+4	ADDRESS BUFFER	CPY03430
0836R				
04ACR 48AB		LH R10,0(R11)		CPY03440
0000				
04B0R C5A0		CLHI R10,C'/*'	UNIT RECORD EOF	CPY03450
2F2A				
04B4R 4330		BE RDEOF		CPY03460
0828R				
04B8R C5A0		CLHI R10,C'/*'	UNIT RECORD EOF	CPY03470
2F26				
04BCR 4330		BE RDEOV		CPY03480
081ER				
04C0R 2621		AIS RECRDS,1	BUMP RECORD COUNT	CPY03490
04C2R 4894	DS10	LH R9,-2(LASTAD)	PICK UP LAST BYTE	CPY03500
FFFE				
04C6R 4890		SH R9,RDBLK+4	CALCULATE LENGTH	CPY03510
0836R				
04CAR 2691		AIS R9,1	ZERO RELATIVE	CPY03520
04CCR 4110		BAL LINK,CLRLNE	CLEAR PRINTLINE	CPY03530
07D2R				
04D0R C8E0		LHI FROM,WORK	SET SENDER	CPY03540
0F48R				
04D4R C8F0		LHI TO,PRTBUF	SET RECEIVER	CPY03550
06F8R				
04D8R 4020		STH RECRDS,WORK	RECORD NUMBER	CPY03560
0F48R				
04DCR 4110		BAL LINK,XLATBD	TRANSLATE TO DECIMAL	CPY03570
096CR				
04E0R 48E0		LH FROM,RDBLK+4	SENDER ADDRESS	CPY03580
0836R				
04E4R C8A0	DS11	LHI R10,X'20'	PRINT LINE LENGTH	CPY03590
0020				
04E8R 059A		CLHR R9,R10	CHECK RECORD LENGTH	CPY03600
04EAR 238B		BNLS DS13	PRINT A FULL LINE	CPY03610
04ECR 08A9		LHR R10,R9	SET INDEX FOR SMALL LINE	CPY03620
04EER 4110		BAL LINK,LINE	PRINT LINE	CPY03630
077ER				
04F2R C530		CLHI R3,X'38'	CHECK LINE COUNT	CPY03640
0038				

04F6R 2183		BLS	DS12			
04F8R 4110		BAL	LINK,HEDPRT	NEW PAGE		CPY03650
0552R						CPY03660
04FCR 4300	DS12	B	DS9+4	CONTINUE		CPY03670
0494R						
0500R 089A	DS13	SHR	R9,R10	DECREMENT RECORD SIZE		CPY03680
0502R 2233		BZS	DS12	DONE		CPY03690
0504R 4110		BAL	LINK,LINE	PRINT FULL LINE		CPY03700
077ER						
0508R C530		CLHI	R3,X'38'	PAGE CHECK		CPY03710
0038						
050CR 2183		BLS	DS13A	CONTINUE		CPY03720
050ER 4110		BAL	LINK,HEDPRT	PRINT HEADERS, TOF ETC.		CPY03730
0552R						
0512R 4300	DS13A	B	DS11	CONTINUE		CPY03740
04E4R						
0516R 4890	DSCHEK	LH	R9,QUANT	CHECK NUMBER OF FILES		CPY03750
0F56R						
051AR 2116		BMS	DS14	ALL -		CPY03760
051CR 0335		HZR	COMAND	END WITH ONE FILE		CPY03770
051ER 2791		SIS	R9,1	DECREMENT COUNT		CPY03780
0520R 0335		HZR	COMAND	DONE		CPY03790
0522R 4090		STH	R9,QUANT			CPY03800
0F56R						
0526R 088B	DS14	LHR	R11,R11	CHECK FILEMARK FLAG		CPY03810
0528R 0215		BMR	COMAND	END		CPY03820
052AR 4110		BAL	LINK,HEDPRT	PRINT HEADERS		CPY03830
0552R						
052ER 0722		XHR	RECRDS,RECRDS	CLEAR RECORD COUNT		CPY03840
0530R 4300	DS14A	B	DS9+4	CONTINUE		CPY03850
0494R						
0534R 4180	DS15	BAL	RTN,VOLCHK	CHECK FOR ENDOVOL		CPY03860
0E36R						
0538R 4305		B	0(COMAND)			CPY03870
0000						
053CR 4300		B	DS10-2	CONTINUE		CPY03880
04C0R						
0540R E120	MEMERR	SVC	2,MEMMSG			CPY03881
0546R						
0544R 0305		BR	COMAND			CPY03882
0546R 0007	MEMMSG	DC	7,MEMEND-MEM			CPY03883
0008						
054AR 4045	MEM	DC	C'MEM-FULL'			CPY03884
4020						
4655						
4C4C						
0552R	MEMEND	EQU	*			CPY03885
						CPY03890
						CPY03900
0552R 0090	HEDPRT	STM	R9,SAVE4	SAVE REGS		CPY03910
0EA8R						
0556R 4010		STH	LINK,HEDRTN	SAVE RETURN		CPY03920
058ER						
055AR 4110		BAL	LINK,CLRLNE			CPY03930
07D2R						
055ER C8F0		LHI	TO,PRTRUF	SET RECEIVER		CPY03940

0562R	06F8R	C8E0	LHI	FROM,HEADER					CPY03950													
		0590R																				
0566R	C8D0		LHI	NUMBER,HEDEND-HEADER					CPY03960													
	0028																					
056AR	4110		BAL	LINK,MOVE	MOVE	HEADER			CPY03970													
	090ER																					
056ER	27D8		SIS	NUMBER,8					CPY03980													
0570R	26E8		AIS	FROM,8					CPY03990													
0572R	CAF0		AHI	TO,HEDEND-HEADER					CPY04000													
	0028																					
0576R	4110		BAL	LINK,MOVE	MOVE	SECOND HALF			CPY04010													
	090ER																					
057AR	E110		SVC	1,FORM	FORM	FEED			CPY04020													
	076ER																					
057ER	2434		LIS	R3,4	LINE	COUNTER			CPY04030													
0580R	4110		BAL	LINK,PRINT	PRINT	HEADER			CPY04040													
	068AR																					
0584R	E110		SVC	1,SKIPLN	SKIP	A LINE			CPY04050													
	0766R																					
0588R	D190		LM	R9,SAVE4	SAVE	REGS			CPY04060													
	0EA8R																					
058CR	4300		DC	X'4300'	RETURN				CPY04070													
058ER	0000	HE DRTN	DC	0					CPY04080													
0590R	5245	HE ADER	DC	C'RECNO	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	'	CPY04090

0588R	434F								
	4F20								
	2020								
	3020								
	3120								
	3220								
	3320								
	3420								
	3520								
	3620								
	3720								
	3820								
	3920								
	4120								
	4220								
	4320								
	4420								
	4520								
	4620								

0588R	HE DEND	EQU	*						CPY04100
	*-----*								
	* REWIND								
	RWD	LHR	LINK,COMAND	SET	RETURN				CPY04120
058AR	D330	LB	R3,OPRAND+1	GET	LOGICAL	UNIT			CPY04130
	0F35R								CPY04140
05BER	C730	XHI	R3,C'0'	DE-	ASCII-IZE				CPY04150
	0030								
05C2R	C530	CLHI	R3,5	VALIDATE					CPY04160
	0005								
05C6R	0387	BNLR	CMDER						CPY04170
05C8R	D230	RWD	STB	R3,REWIND+1	SET	LOGICAL	UNIT		CPY04180

0509R					
05CCR	E110	SVC	1,REWIND	REWIND	CPY04190
	05D8R				
05D0R	4800	LH	R0,REWIND+2	CHECK STATUS	CPY04200
	05DAR				
05D4R	0311	BNMR	LINK	RETURN	CPY04210
05D6R	0306	BR	ERROR		CPY04220
05D8R	C000	REWIND DC	X'C000',0		CPY04230
	0000				

\*-----\*

			WRITE FILE MARK		CPY04240
		WFM	LHR LINK,COMAND	SET RETURN	CPY04250
05DCR	0815		LB R3,OPRAND+1	GET LOGICAL UNIT	CPY04260
05DER	D330				CPY04270
	0F35R				
05E2R	C730	XHI	R3,C'0'	DE-ASCII-IZE	CPY04280
	0030				
05E6R	C530	CLHI	R3,5	VALIDATE	CPY04290
	0005				
05EAR	0387	BNLR	CMDER		CPY04300
05ECR	D230	WFMK STB	R3,WRTFMK+1	SET LOGICAL UNIT	CPY04310
	05FDR				
05FOR	E110	SVC	1,WRTFMK	WRITE FILEMARK	CPY04320
	05FCR				
05F4R	4800	LH	R0,WRTFMK+2	CHECK STATUS	CPY04330
	05FER				
05F8R	0311	BNMR	LINK	RETURN	CPY04340
05FAR	0306	BR	ERROR		CPY04350
05FCR	8800	WRTFMK DC	X'8800',0		CPY04360
	0000				

\*-----\*

			SKIP FILEMARK		CPY04370
		SKP	LB R3,OPRAND+1	GET LOGICAL UNIT	CPY04380
0600R	D330				CPY04390
	0F35R				
0604R	C730	XHI	R3,C'0'	CLEAR ASCII BITS	CPY04400
	0030				
0608R	C530	CLHI	R3,5	CHECK VALIDITY	CPY04410
	0005				
060CR	0387	BNLR	CMDER		CPY04420
060ER	D3D0	LB	NUMBER,OPRAND+2	GET NO. FILES	CPY04430
	0F36R				
0612R	0815	LHR	LINK,COMAND	SET RETURN	CPY04440
0614R	08D0	LHR	NUMBER,NUMBER	CHECK FOR DEFAULT	CPY04450
0616R	4330	BZ	SKIP		CPY04460
	063ER				
061AR	C8E0	LHI	FROM,OPRAND+3	SET SENDER	CPY04470
	0F37R				
061ER	C8F0	LHI	TO,SKPNO	SET RECEIVER	CPY04480
	0F52R				
0622R	D39E	LB	R9,0(FROM)	GET A CHARACTER	CPY04490
	0000				
0626R	C590	CLHI	R9,X'41'	CHECK FOR ALL	CPY04500
	0041				
062AR	2134	BNES	SKP1	NO - TRANSLATE	CPY04510
062CR	C810	LHI	LINK,SKPALL	YES - SET RETURN	CPY04520
	0652R				
0630R	2307	BS	SKIP		CPY04530

0632R	4110	SKP1	BAL	LINK,XLATDB	TRANSLATE DECIMAL TO BINARY	CPY04540
	0896R					
0636R	4890		LH	R9,SKPNO	GET SKIP COUNT	CPY04550
	0F52R					
063AR	C810		LHI	LINK,SKP2		CPY04560
	0648R					
063ER	D230	SKIP	STB	R3,SKIPFM+1	SET LOGICAL UNIT	CPY04570
	0675R					
0642R	E110		SVC	1,SKIPFM	SKIP FM	CPY04580
	0674R					
0646R	0301		BR	LINK	RETURN	CPY04590
0648R	0899	SKP2	LHR	R9,R9	CHECK SKIP NO.	CPY04600
064AR	0335		BZR	COMAND		CPY04610
064CR	2791		SIS	R9,1	DECREMENT COUNTER	CPY04620
064ER	0335		BZR	COMAND	DONE	CPY04630
0650R	2209		BS	SKIP	CONTINUE SKIPPING	CPY04640
0652R	D230	SKPALL	STB	R3,SKPRD+1	SET LOGICAL UNIT	CPY04650
	066DR					
0656R	E110		SVC	1,SKPRD	READ A RECORD	CPY04660
	066CR					
065AR	4800		LH	R0,SKPRD+2		CPY04670
	066ER					
065ER	C300		THI	R0,X'0800'	CHECK FOR FM	CPY04680
	0800					
0662R	4330		BZ	SKIP	NO - CONTINUE	CPY04690
	063ER					
0666R	4110		BAL	LINK,BKSP	BACKSPACE	CPY04700
	06A4R					
066AR	0305		BR	COMAND		CPY04710
066CR	4800	SKPRD	DC	X'4800',0,WORK,WORK+5		CPY04720
	0000					
	0F48R					
	0F4DR					
0674R	8400	SKIPFM	DC	X'8400',0		CPY04730
	0000					
						CPY04740
						CPY04750
						CPY04760
0678R	D330	BSP	LB	R3,OPRAND+1	GET LOGICAL UNIT	CPY04770
	0F35R					
067CR	C730		XHI	R3,C'0'	DE-ASCII-IZE	CPY04780
	0030					
0680R	C530		CLHI	R3,5	VALIDATE	CPY04790
	0005					
0684R	0387		RNLR	CMDER		CPY04800
0686R	0815		LHR	LINK,COMAND	SET RETURN	CPY04810
068AR	D3D0		LB	NUMBER,OPRAND+2	GET NO. FILEMARKS	CPY04820
	0F36R					
068CR	08DD		LHR	NUMBER,NUMBER	DEFAULT ?	CPY04830
068ER	233B		BZS	BKSP	YES	CPY04840
0690R	C8E0		LHI	FROM,OPRAND+3	SET SENDER	CPY04850
	0F37R					
0694R	C8F0		LHI	TO,BKNO	SET RECEIVER	CPY04860
	0F54R					
0698R	4110		BAL	LINK,XLATDB	TRANSLATE DECIMAL TO BINARY	CPY04870
	0896R					
069CR	C810		LHI	LINK,BSP2		CPY04880



06AER	06AER					
06A0R	4890		LH	R9,BKNO		
	0F54R					CPY04880
06A4R	D230	BKSP	STB	R3,BACKSP+1	SET LOGICAL UNIT	
	06B7R					CPY04890
06A8R	E110		SVC	1.BACKSP	BACKSPACE	
	06B6R					CPY04900
06ACR	0301		BR	LINK	RETURN	
06AER	0899	BSP2	LHR	R9,R9	CHECK NO. FILE MARKS	CPY04910
06B0R	0335		BZR	COMAND	DONE	CPY04920
06B2R	2791		SIS	R9,1	DECREMENT	CPY04930
06B4R	2208		BS	BKSP	CONTINUE	CPY04940
06B6R	8200	BACKSP	DC	X'8200',0		CPY04950
	0000					CPY04960
*-----*						
				PRINT LINE		
06BAR	D090	PRINT	STM	R9,SAVE5		CPY04970
	0EB8R					CPY04980
06BER	C890		LHI	R9,X'0020'		CPY04990
	0020					CPY05000
06C2R	C8A0		LHI	R10,PRTEND		CPY05010
	0764R					CPY05020
06C6R	D3BA	PRNT1	LB	R11,0(R10)		CPY05030
	0000					CPY05040
06CAR	27A1		SIS	R10,1		CPY05050
06CCR	059B		CLHR	R9,R11		CPY05060
06CER	2234		BES	PRNT1		CPY05070
06D0R	26A2		AIS	R10,2		CPY05080
06D2R	40A0		STH	R10,PRTBLK+6		CPY05090
	06F6R					CPY05100
06D6R	E110		SVC	1,PRTBLK		CPY05110
	06F0R					CPY05120
06DAR	4800		LH	R0,PRTBLK+2		CPY05130
	06F2R					CPY05140
06DER	0216		BMR	ERROR		CPY05150
06E0R	4010		STH	LINK,PRTN		CPY05160
	06EER					CPY05170
06E4R	4110		BAL	LINK,CLRLNE		CPY05180
	07D2R					CPY05190
06E8R	D190		LM	R9,SAVE5		CPY05200
	0EB8R					CPY05210
06ECR	4300		DC	X'4300'		CPY05140
06EER	0000	PRTN	DC	0		CPY05150
06FOR	2803	PRTBLK	DC	X'2803',0,PRTBUF,PRTEND		CPY05160
	0000					CPY05170
	06F8R					CPY05180
	0764R					CPY05190
06F8R		PRTBUF	DS	108		CPY05200
0764R		PRTEND	EQU	*		CPY05170
0764R	2020		DC	X'2020'		CPY05180
0766R	2803	SKIPLN	DC	X'2803',0,PRT1,PRT1+1		CPY05190
	0000					CPY05200
	0776R					CPY05210
	0777R					CPY05220
076ER	2803	FORM	DC	X'2803',0,PTOF,PTOF+5		CPY05210
	0000					CPY05220

```

0778R
077DR
0776R 0D20 PRT1 DC X'0D20'
0778R 0C00 PTOF DC X'0C00',X'0000',X'0000'
0000
000D
*-----*
077ER C8F0 * LINE GENERATOR SET INDEX CPY05240
0700R LHI TO,PRTRUF+8 CPY05250
0782R 4010 STH LINK,LNRTN SAVE RETURN LINK CPY05260
07A2R 07A2R
0786R 2633 AIS R3,3 BUMP LINE COUNTER CPY05280
0788R 08DA LHR NUMBER,R10 COUNT CPY05290
078AR 4110 BAL LINK,XLATEH TRANSLATE TO ASCII HEX CPY05300
08CCR
078ER 4110 BAL LINK,PRINT PRINT LINE CPY05310
06BAR
0792R 4110 BAL LINK,XLATEC TRANSLATE TO CHARACTER CPY05320
093ER
0796R 4110 BAL LINK,PRINT PRINT LINE CPY05330
06BAR
079AR E110 SVC 1,SKIPLN SKIP A LINE CPY05340
0766R
079ER 0AEA AHR FROM,R10 BUMP POINTER CPY05350
07A0R 4300 DC X'4300' RETURN CPY05360
07A2R 0000 LNRTN DC 0 CPY05370
*-----*
07A4R C800 * HALT VARIABLE FUNCTION CPY05380
FFFF NVR LHI R0,-1 CPY05390
07A8R 4000 STH R0,VARFLG SET FLAG CPY05410
0F20R
07ACR 0305 BR COMAND CPY05420
*-----*
07AER 0700 * SET VARIABLE FUNCTION CPY05430
07B0R 4000 VAR XHR R0,R0 CPY05440
0F20R STH R0,VARFLG UNSET FLAG CPY05450
07B4R 0305 BR COMAND CPY05460
*-----*
07B6R E120 * COMMAND ERROR CPY05470
07BCR 0305 CMDERR SVC 2,CMDSMG ERROR MESSAGE CPY05480
07BCR 0007 CMDSMG DC 7,CMEND-CMSG CPY05490
0008 CPY05500
07C0R 434D CMSG DC C'CMD-ERR' CPY05530
442D
4552
5220
07C8R CMEND EQU * CPY05540
*-----*
07C8R E120 * PAUSE CPY05550
0894R PAWS SVC 2,PAUSE PAUSE CPY05560
CPY05570

```

Job ID	BR	COMAND	RETURN	CPY
07CCR 0305				CPY05590
*-----*				CPY05591
07CER E130		TERMINATION		CPY05600
0000	END	SVC 3,0		CPY05610
*-----*				CPY05620
07D2R D090	CLRLNE	CLEAR PRINTLINE		CPY05630
0E78R	STM	R9,SAVE	SAVE REGS	CPY05640
07D6R C8D0		LHI	NUMBER,PRTEND-PRTBUF	CPY05650
006C				
07DAR C8E0		LHI	FROM,X'2020'	CPY05660
2020			CLEARING VALUE	
07DER C8F0		LHI	TO,PRTBUF	CPY05670
06F8R			RECEIVER	
07E2R 4010		STH	LINK,CLROUT	CPY05680
07FOR			SAVE RETURN	
07E6R 4110		BAL	LINK,CLEAR	CPY05690
092AR			CLEAR BUFFER	
07EAR D190		LM	R9,SAVE	CPY05700
0E78R			RESTORE REGS	
07EER 4300		DC	X'4300'	CPY05710
07FOR 0000	CLROUT	DC	0	CPY05720
*-----*				CPY05730
07F2R D090	READ	READ ROUTINE		CPY05740
0E98R	STM	R9,SAVE3	SAVE REGS	CPY05750
07F6R C890		LHI	R9,X'7FFF'	CPY05760
7FFF				
07FAR 4090		STH	R9,RDBLK+2	CPY05770
0834R			SET STATUS VALUE	
07FFR E110		SVC	1,RDBLK	CPY05780
0832R			READ	
0802R 4800	RD1	LH	R0,RDBLK+2	CPY05790
0834P			CHECK STATUS	
0806R 2116		BMS	RDCHK	CPY05800
0808R 4000	RDEXIT	STH	R0,FMKFLG	CPY05810
0F50R			ERROR	
080CR D190	RD3	LM	R9,SAVE3	CPY05820
0E98R			RESTORE REGS	
0810R 0301		BR	LINK	CPY05830
0812R C300	RDCHK	THI	R0,X'800'	CPY05840
0800			RETURN	
0816R 2139		BNZS	RDEOF	CPY05850
0818R C300		THI	R0,X'1000'	CPY05860
1000			YES	
081CR 0336		BZR	ERROR	CPY05870
081ER C890	RDEOV	LHI	R9,-1	CPY05880
FFFF				
0822R 4090	RD2	STH	R9,FMKFLG	CPY05890
0F50R			SET EOY	
0826R 2200		RS	RD3	CPY05900
0828R 4890	RDEOF	LH	R9,FMKFLG	CPY05910
0F50R			EXIT	
082CR 2037		BNZS	RDEOV	CPY05920
082ER 249F	LIS	LIS	R9,15	CPY05930
			LAST = FILEMARK	
			NO, SET FLAG	

CPY05940  
CPY05950

0830R 2207  
0832R 0001  
0000  
0000  
0000

RDHLK BS RD2  
DC X'0001',0,0,0

\*-----\* CPY05960  
\* WRITE ROUTINE CPY05970  
\* CPY05980

083AR D090  
0E98R  
083ER E110  
084ER  
4800  
0842R 0850R  
0846R 0236  
0848R D190  
0E98R  
084CR 0301  
084ER 0002  
0000  
0000  
0000

WRITE STM R9,SAVE3 SAVE REGS  
SVC 1,WRBLK WRITE  
LH R0,WRBLK+2 CHECK STATUS  
RNZR ERROR  
LM R9,SAVE3 RESTORE REGS  
BR LINK RETURN  
WRBLK DC 2,0,0,0

CPY05990  
CPY06000  
CPY06010  
CPY06020  
CPY06030  
CPY06040

\*-----\* CPY06050  
\* ERROR ROUTINE CPY06060  
\* CPY06070  
\* CPY06080

0856R 24D2  
0858R 4000  
0F48R  
085CR C8E0  
0F48R  
0860R C8F0  
088ER  
0864R 0891  
0866R 4110  
08CCR  
086AR E120  
0880R  
086ER E120  
0894R  
0872R 2794  
0874R 4090  
087ER  
0878R D190  
0E98R  
087CR 4300  
087ER 0000  
0880R 0007  
0010  
0884R 492F  
4F20  
4552  
5220  
3A20  
2020  
2020  
2020

ERR LIS NUMBER,2  
STH R0,WORK  
LHI FROM,WORK  
LHI TO,ERRMSG+10  
LHR R9,LINK  
BAL LINK,XLATEH TRANSLATE HEX TO ASCII  
SVC 2,FERRR  
SVC 2,PAUSE  
SIS R9,4  
STH R9,ERXIT  
LM R9,SAVE3  
DC X'4300'  
ERXIT DC 0  
ERRR DC 7,ERMEND-ERRMSG  
ERRMSG DC C'I/O ERR !

CPY06090  
CPY06100  
CPY06110  
CPY06120  
CPY06130  
CPY06140  
CPY06150  
CPY06160  
CPY06170  
CPY06180  
CPY06190  
CPY06200  
CPY06210

0894R  
0894R 0001

ERMEND EQU \*  
PAUSE DC 1

CPY06220  
CPY06230

```

*-----*
*          TRANSLATE - DECIMAL ASCII TO INTERNAL          * CPY06240
0896R D0A0  XLATDB STM   R10,SAVE          SAVE REGISTERS          CPY06250
      0E78R
0899R C5D0          CLHI  NUMBER,6          CHECK OPERAND LENGTH 5 = MAX    CPY06270
      0006
089ER 2389          BNLS  XB2
08A0R 078R          XHR   R11,R11          CLEAR MULTIPLICAND          CPY06280
08A2R 24CA          LIS   R12,10          MULTIPLIER                  CPY06290
08A4R D30E  XB1    LB    R0,0(FROM)        GET A CHARACTER              CPY06300
      0000
08A8R CB00          SHI   R0,X'30'          INTERNALIZE                  CPY06310
      0030
08ACR 0217          BMR   CMDER          ILLEGAL CHARACTER          CPY06330
08AER 090C          CHR   R0,R12          HIGH LIMIT CHECK           CPY06340
08B0R 0387  XB2    BNLR  CMDER          ILLEGAL CHARACTER          CPY06350
08B2R 0CAC          MHR   R10,R12          DECIMALIZE                  CPY06360
08B4R 0AB0          AHR   R11,R0
08B6R 26E1          AIS   FROM,1          BUMP SENDER FIELD          CPY06370
08B8R 27D1          SIS   NUMBER,1        DECREMENT LENGTH COUNTER    CPY06380
08BAR 203B          BNZS  XB1
08BCR 40BF  XB3    STB   R11,0(TO)
      0000
08C0R D1A0          LM    R10,SAVE          RESTORE REGISTERS          CPY06420
      0E78R
08C4R 0301          HR    LINK          RETURN                      CPY06430
08C6R 08DD          LHR  NUMBER,NUMBER  IF LENGTH IS ZERO - RETURN  CPY06440
08C8R 4330          HZ   XB3
      08BCR
*-----*
*          TRANSLATE - INTERNAL TO ASCII HEX              * CPY06460
08CCR D090  XLATEH STM   R9,SAVE          SAVE REGISTERS          CPY06470
      0E78R
08D0R C890          LHI   R9,X'3A'
      003A
08D4R D3AE  XH1    LB    R10,0(FROM)        GET A BYTE                  CPY06490
      0000
08D8R D3BE          LB    R11,0(FROM)
      0000
08DCR ECA0          SRL  R10,4          SPLIT BYTE                  CPY06500
      0004
08E0R 90BC          SRLS R11,12
08E2R CAA0          AHI   R10,X'30'        ASCII-IZE                   CPY06530
      0030
08E6R CAB0          AHI   R11,X'30'        ASCII-IZE                   CPY06540
      0030
08EAR 05A9          CLHR  R10,R9          GREATER THAN 9 ?          CPY06560
08ECR 2182          BLS  XH2          NO                          CPY06570
08EER 26A7          AIS  R10,7          YES - ADJUST ASCII VALUE  CPY06580
08FOR 05B9  XH2    CLHR  R11,R9          SECOND CHAR GREATER THAN 9 ? CPY06590
08F2R 2182          BLS  XH3          NO                          CPY06600
08F4R 26B7          AIS  R11,7          YE - ADJUST                CPY06610
08F6R D2AF  XH3    STB   R10,0(TO)        STORE CHARACTER            CPY06620
      0000
08FAR D2BF          STB   R11,1(TO)        STORE SECOND CHARACTER      CPY06630
      0001

```

08FER 26F2		AIS	TO,2	BUMP INDEX	
0900R 26E1		AIS	FROM,1	BUMP SENDER	CPY06640
0902R 27D1		SIS	NUMBER,1	DECREMENT COUNT	CPY06650
0904R 4230		BNZ	XH1	CONTINUE	CPY06660
08D4R					CPY06670
0908R D190		LM	R9,SAVE	RESTORE REGISTERS	
0E78R					CPY06680
090CR 0301		BR	LINK	RETURN	
*-----*					
			MOVE		
090ER D0C0		MOVE	STM R12,SAVE2	SAVE REGISTERS	CPY06690
0E88R					CPY06700
0912R 27D1	M1	SIS	NUMBER,1	DECREMENT COUNT	CPY06710
0914R 2118		BMS	M2	DONE	CPY06720
0916R D3CE		LB	R12,0(FROM)	GET BYTE	CPY06730
0000					CPY06740
091AR D2CF		STB	R12,0(TO)	STORE BYTE	CPY06750
0000					CPY06760
091ER 26E1		AIS	FROM,1	BUMP SENDER	CPY06770
0920R 26F1		AIS	TO,1	BUMP RECEIVER	CPY06780
0922R 2208		BS	M1	CONTINUE	CPY06790
0924R D1C0	M2	LM	R12,SAVE2	RESTORE REGISTERS	CPY06800
0E88R					CPY06810
0928R 0301		BR	LINK	RETURN	CPY06820
*-----*					
			CLEAR		
092AR D0D0		CLEAR	STM NUMBER,SAVE2	SAVE REGISTERS	CPY06830
0E88R					CPY06840
092ER D2EF	C1	STB	FROM,0(TO)	STORE BYTE	CPY06850
0000					CPY06860
0932R 26F1		AIS	TO,1	BUMP RECEIVER FIELD	CPY06870
0934R 27D1		SIS	NUMBER,1	DECREMENT COUNTER	CPY06880
0936R 2034		BNZS	C1	CONTINUE	CPY06890
0938R D1D0		LM	NUMBER,SAVE2	RESTORE REGS	CPY06900
0E88R					CPY06910
093CR 0301		BR	LINK	RETURN	CPY06920
*-----*					
			TRANSLATE - INTERNAL	TO CHARACTER	
093ER D0A0		XLATEC	STM R10,SAVE	SAVE REGISTERS	CPY06930
0E78R					CPY06940
0942R D3BE	XC1	LB	R11,0(FROM)	GET A CHARACTER	CPY06950
0000					CPY06960
0946R C5B0		CLHI	R11,X'20'	GRAPHIC ?	CPY06970
0020					CPY06980
094AR 2383		BNLS	XC3	YES	CPY06990
094CR C8B0	XC2	LHI	R11,X'20'	NO - REPLACE WITH A ' '	CPY07000
0020					CPY07010
0950R C5B0	XC3	CLHI	R11,X'60'	GRAPHIC ?	CPY07020
0060					CPY07030
0954R 2182		BLS	XC4	YES	CPY07040
0956R 2205		BS	XC2	NO	CPY07050
0958R D2BF	XC4	STB	R11,1(TO)	STORE CHARACTER	CPY07060
0001					CPY07070
095CR 26F2		AIS	TO,2		CPY07080
095ER 26E1		AIS	FROM,1	BUMP SENDER	CPY07090
0960R 27D1		SIS	NUMBER,1	DECREMENT COUNT	CPY07100

0962R	4230		BNZ	XC1	CONTINUE		CPY07050
	0942R						
0966R	01A0		LM	R10,SAVE	RESTORE REGISTERS		CPY07060
	0E78R						
096AR	0301		BR	LINK	RETURN		CPY07070
*-----*							
					TRANSLATE - INTERNAL TO DECIMAL ASCII		CPY07080
096CR	00A0		XLATBD	STM	R10,SAVE	SAVE REGISTERS	CPY07090
	0E78R						CPY07100
0970R	24C8		LIS	R12,8	DIVISOR INDEX		CPY07110
0972R	48BE		LH	R11,0(FROM)	DIVIDEND		CPY07120
	0000						
0976R	07AA	XD1	XHR	R10,R10			CPY07130
0978R	48DC		LH	R13,XD3(R12)	PICK UP POWER OF TEN		CPY07140
	0998R						
097CR	0DAD		DHR	R10,R13			CPY07150
097ER	C7B0	XD2	XHI	R11,X*30	ASCII-IZE QUOTIENT		CPY07160
	0030						
0982R	D2BF		STB	R11,0(TO)	STORE		CPY07170
	0000						
0986R	26F1		AIS	T0,1	BUMP STORAGE INDEX		CPY07180
0988R	27C2		SIS	R12,2	DECREMENT SENDER INDEX		CPY07190
098AR	08BA		LHR	R11,R10	SET REMAINDER AS QUOTIENT		CPY07200
098CR	203B		BNZS	XD1	CONTINUE		CPY07210
098ER	08CC		LHR	R12,R12	FIVE PLACES ?		CPY07220
0990R	2219		BNMS	XD2	PAD ZEROES		CPY07230
0992R	D1A0		LM	R10,SAVE	RESTORE REGISTERS		CPY07240
	0E78R						
0996R	0301		BR	LINK	RETURN		CPY07250
0998R	0001	XD3	DC	1,10,100,1000,1000			CPY07260
	000A						
	0064						
	03E8						
	03E8						
*-----*							
					NON-VERIFY		CPY07270
09A2R	E120		NONVER	SVC	2,VERMSG	ISSUE NON VERIFY MESSAGE	CPY07280
	0A14R						CPY07290
09A6R	4110		BAL	LINK,HEDPRT			CPY07300
	0552R						
09AAR	D190		LM	R9,INREC	INPUT RECORD		CPY07310
	0A22R						
09AER	D090		STM	R9,PRTBUF	MESSAGE		CPY07320
	06F8R						
09B2R	4110		BAL	LINK,PRINT	PRINT INPUT RECORD		CPY07330
	06BAR						
09B6R	C8B0		LHI	R11,NV3			CPY07340
	09ECR						
09BAR	48E0		LH	FROM,RDBLK+4			CPY07350
	0836R						
09BER	4890		LH	R9,RDBLK+6	END ADDRESS		CPY07360
	0838R						
09C2R	4B90		SH	R9,RDBLK+4	CALCULATE LENGTH		CPY07370
	0836R						
09C6R	2691		AIS	R9,1	ZERO RELATIVE-IZE		CPY07371
09C8R	C8A0	NV	LHI	R10,X*20	PRINT LINE LENGTH		CPY07380

09CCR	0020					
09CER	059A	NV0	CLHR	R9,R10	COMPARE FOR SHORT RECORD	CPY07390
09D0R	2385		BNLS	NV2	NO	CPY07400
09D0R	08A9		LHR	R10,R9	YES SET SIZE	CPY07410
09D2R	4110		BAL	LINK,LINE	FORMAT & PRINT A LINE	CPY07420
	077ER					
09D6R	0308	NV1	BR	R11	GET OUTPUT RECORD	CPY07430
09D8R	0B9A	NV2	SHR	R9,R10	DECREMENT LENGTH	CPY07440
09DAR	033B		BZR	R11	DONE	CPY07450
09DCR	4110		BAL	LINK,LINE	PRINT A LINE	CPY07460
	077ER					
09E0R	C530		CLHI	R3,X'38'	LINE COUNTER CHECK	CPY07470
	0038					
09E4R	208C		BLS	NV0	CONTINUE	CPY07480
09E6R	4110		BAL	LINK,HEDPRT	NEW PAGE	CPY07490
	0552R					
09EAR	220F		BS	NV0	CONTINUE	CPY07500
09ECR	4110	NV3	BAL	LINK,HEDPRT	PRINT HEADER	CPY07510
	0552R					
09F0R	D190		LM	R9,OUTREC	OUTPUT RECORD	CPY07520
	0A30R					
09F4R	D090		STM	R9,PRTBUF	SET RECORD MSG	CPY07530
	06F8R					
09F8R	4110		BAL	LINK,PRINT	PRINT RECORD	CPY07540
	06BAR					
09FCR	4890		LH	R9,RDBLK2+6		CPY07550
	0402R					
0A00R	48E0		LH	FROM,RDBLK2+4	BEGIN ADDRESS	CPY07560
	0400R					
0A04R	0B9E		SHR	R9,FROM	RECORD LENGTH	CPY07570
0A06R	2691		AIS	R9,1	ZERO RELATIVE-IZE	CPY07571
0A08R	41B0		BAL	R11,NV	CONTINUE	CPY07580
	09C8R					
0A0CR	4110	NV4	BAL	LINK,HALT	HALT PROCESSING	CPY07590
	0D64R					
0A10R	4300		B	VR5	CONTINUE	CPY07600
	034AR					
0A14R	0007	VERMSG	DC	7,VERMND-VERM		CPY07610
	000A					
0A18R	4E4F	VERM	DC	C'NON-VERIFY'		CPY07620
	4E2D					
	5645					
	5249					
	4659					
0A22R		VERMND	EQU	*		CPY07630
0A22R	494E	INREC	DC	C'INPUT RECORD'		CPY07640
	5055					
	5420					
	5245					
	434F					
	5244					
0A2ER	2020		DC	C' .		CPY07650
0A30R	4F55	OUTREC	DC	C'OUTPUT RECORD'		CPY07660
	5450					
	5554					
	2052					



4543  
4F52  
4420

*-----*				CPY07670
* LIST LABELS				CPY07680
0A3ER	C890	LSTA	LHI R9,X'48'	CPY07690
	0048		ASCII READ	
0A42R	D1E0		LM FROM,LSTBUF	CPY07700
	0B1C		BUFFER LIMITS	
0A46R	D0E0	LST	STM FROM,RDBLK+4	CPY07710
	0836R		PRIME I/O BLOCK	
0A4AR	4110		BAL LINK,CLRLNE	CPY07711
	07D2R		CLEAR PRINT LINE	
0A4ER	D1C0		LM R12,LABELS	CPY07720
	0B24R		LABEL HEADER	
0A52R	D0C0		STM R12,PRTBUF	CPY07730
	06F8R		SET PRINT LINE	
0A56R	4110		BAL LINK,PRINT	CPY07740
	06BAR		PRINT LINE	
0A5AR	D290		STB R9,RDBLK	CPY07750
	0832R		PRIME I/O BLOCK	
0A5ER	D3D0		LB NUMBER,OPRAND	CPY07760
	0F34R		LU SPECIFIED ?	
0A62R	08DD		LHR NUMBER,NUMBER	CPY07770
0A64R	0337		HZR CMDER	CPY07780
0A66R	D390		LB R9,OPRAND+1	CPY07790
	0F35R		GET VALUE	
0A6AR	C490		NHI R9,3	CPY07800
	0003		MAX VALUE	
0A6ER	D290		STB R9,RDBLK+1	CPY07810
	0833R		SET I/O BLOCK	
0A72R	0799		XHR R9,R9	CPY07820
0A74R	4090		STH R9,FMKFLG	CPY07830
	0F50R		FILEMARK FLAG	
0A78R	4110	LST1	BAL LINK,READ	CPY07840
	07F2R		READ A RECORD	
0A7CR	4890		LH R9,FMKFLG	CPY07850
	0F50R		CHECK FLAG	
0A80R	4210		BM NOFND	CPY07860
	0B84R		END	
0A84R	D390		LB R9,RDBLK	CPY07870
	0832R		ASCII OR BINARY ?	
0A88R	C590		CLHI R9,X'48'	CPY07880
	0048			
0A8CR	4230		BNE LST3	CPY07890
	0AB0R		BINARY	
0A90R	4890		LH R9,PRTBUF	CPY07900
	06F8R		ASCII - CHECK FOR '***'	
0A94R	C590		CLHI R9,X'2A2A'	CPY07910
	2A2A			
0A98R	4230	LST2	BNE LST1	CPY07920
	0A78R			
0A9CR	D1C0		LM R12,PRTBUF	CPY07930
	06F8R		LABEL	
0AA0R	4110		BAL LINK,CLRLNE	CPY07940
	07D2R		CLEAR PRINT LINE	

0AA4R	DOC0 06F8R	STM	R12,PRTBUF	RESTORE LABEL	CPY07950
0AA8R	4110 06BAR	BAL	LINK,PRINT	PRINT LABEL	CPY07960
0AACR	4300 0A78R	B	LST1	CONTINUE	CPY07970
0AB0R	4890 06F8R	LST3 LH	R9,PRTBUF		CPY07980
0AB4R	C590 FFFF	CLHI	R9,-1	CHECK SEQNUM FOR - 1	CPY07990
0AB8R	4230 0A78R	LST3A BNE	LST1	NO - CONTINUE	CPY08000
0ABCR	24DC	LIS	NUMBER,12		CPY08010
0ABER	48E0 06FCR	LH	FROM,PRTBUF+4	SET FOR SUBROUTINE	CPY08020
0AC2R	C8F0 06FCR	LHI	T0,PRTBUF+4	SET FOR SUBROUTINE	CPY08030
0AC6R	4110 0D3ER	BAL	LINK,XTRACT	GET LOADER ITEM	CPY08040
0ACAR	27EF	SIS	FROM,15	LABEL ITEM ?	CPY08050
0ACCR	203A	BNZS	LST3A		CPY08060
0ACER	07CC	XHR	R12,R12		CPY08070
0AD0R	0733	XHR	R3,R3	CLEAR COUNTER	CPY08080
0AD2R	41B0 0D50R	LST4 BAL	R11,LORWD	EXTRACT A HALFWORD	CPY08090
0AD6R	40A3 0F18R	STH	R10,LBLSAV(R3)	SAVE LABEL	CPY08100
0ADAR	2632	AIS	R3,2	BUMP INDEX	CPY08110
0ADCR	C530 0006	CLHI	R3,6	SIX CHARACTERS ?	CPY08120
0AE0R	2037	BNES	LST4	NO - CONTINUE	CPY08130
0AE2R	4180 0E36R	BAL	RTN,VOLCHK	CHECK FOR ENDVOL	CPY08131
0AE6R	4300 0B0AR	B	LST5		CPY08132
0AEAR	4110 07D2R	BAL	LINK,CLRLNE	CLEAR PRINTLINE	CPY08140
0AEER	D1D0 0F18R	LM	NUMBER,LBLSAV	LABEL	CPY08150
0AF2R	D0D0 06F8R	STM	NUMBER,PRTBUF		CPY08160
0AF6R	4110 06BAR	BAL	LINK,PRINT	PRINT LABEL	CPY08170
0AFAR	4300 0A78R	B	LST1	CONTINUE	CPY08200
0AFER	C890 0058	LSTB LHI	R9,X'58'	BINARY READ	CPY08210
0B02R	D1E0 0B20R	LM	FROM,LSTBFB	BUFFER EXTENTS	CPY08220
0B06R	4300 0A46R	B	LST	CONTINUE	CPY08230
0B0AR	4110 07D2R	LST5 BAL	LINK,CLRLNE	CLEAR PRINTLINE	CPY08231
0B0ER	D1D0 0F18R	LM	NUMBER,LBLSAV	LABEL	CPY08232
0R12R	D0D0	STM	NUMBER,PRTBUF		CPY08233

0B16R	06F8R 4110 06BAR	BAL	LINK,PRINT	PRINT LINE	CPY08234
0B1AR	0305	BR	COMAND	RETURN	CPY08235
0B1CR	06F8R 0747R	LSTBUF DC	PRTBUF,PRTBUF+79		CPY08240
0B20R	06F8R 0763R	LSTBFB DC	PRTBUF,PRTBUF+107		CPY08250
0B24R	204C 4142 454C 533A	LABELS DC	C LABELS:*		CPY08260
0R2CR	4010 0B82R	FIND	STH LINK,FNDRTN	SAVE RETURN LINK	CPY08270 CPY08280
0B30R	24D8	LIS	NUMBER,8	LABEL LENGTH	CPY08290
0B32R	C8E0 2020	LHI	FROM,X*2020*	SPACES	CPY08300
0B36R	C8F0 0F18R	LHI	TO,LBLSAV	CLEAR LABEL STORAGE AREA	CPY08310
0B3AR	4110 092AR	BAL	LINK,CLEAR	CLEAR BUFFER	CPY08320
0B3ER	D3D0 0F34R	LB	NUMBER,OPRAND	PICK UP LABEL	CPY08330
0B42R	08DD	LHR	NUMBER,NUMBER	LABEL ENTERED ?	CPY08340
0B44R	0337	BZR	CMDER	NO - ERROR	CPY08350
0B46R	C5D0 0009	CLHI	NUMBER,9	LENGTH CHECK	CPY08360
0B4AR	0387	BNLR	CMDER	NOT LESS = ERROR	CPY08370
0B4CR	C8E0 0F35R	LHI	FROM,OPRAND+1	SENDER ADDRESS	CPY08380
0B50R	C8F0 0F18R	LHI	TO,LBLSAV	RECEIVER ADDRESS	CPY08390
0B54R	4110 090ER	BAL	LINK,MOVE	MOVE LABEL	CPY08400
0B58R	CAED 0001	AHI	FROM,1(NUMBER)	BUMP TO LU	CPY08410
0B5CR	D3DE FFFF	LB	NUMBER,-1(FROM)		CPY08420
0B60R	08DD	LHR	NUMBER,NUMBER	LU ENTERED ?	CPY08430
0B62R	0337	BZR	CMDER	NO - ERROR	CPY08440
0B64R	D39E 0000	LB	R9,0(FROM)	GET LU	CPY08450
0B68R	C490 0003	NHI	R9,X*03*	MAX VALUE	CPY08460
0B6CR	D290 0833R	STB	R9,RDBLK+1	PRIME LU	CPY08470
0B70R	0799	XHR	R9,R9		CPY08480
0B72R	4090 0F50R	STH	R9,FMKFLG	CLEAR FILEMARK FLAG	CPY08490
0B76R	4110 07F2R	FND1 BAL	LINK,READ	READ A RECORD	CPY08500
0B7AR	4890 0F50R	LH	R9,FMKFLG	CHECK FLAG	CPY08510
0B7ER	2113	BMS	NOFND	MINUS = END OF VOLUME	CPY08520
0B80R	4300	DC	X*4300*	RETURN	CPY08530

0B82R	0000	FNDRTN	DC	0			CPY08540
0B84R	E120	NOFND	SVC	2,NFNDMG	FILE END MESSAGE		CPY08550
	0B8AR						
0B88R	0305		BR	COMAND			CPY08560
0B8AR	0007	NFNDMG	DC	7,NFEND-NFND			CPY08570
	0008						
0B8ER	4649	NFND	DC	C*FILE END*			CPY08580
	4C45						
	2045						
	4E44						
0B96R		NFEND	EQU	*			CPY08590
		*					CPY08600
		*	FIND	ASCII FILE NAME			CPY08610
0B96R	D1E0	FNDA	LM	FROM,LSTBUF	BUFFER EXTENTS		CPY08620
	0B1CR						
0B9AR	4050		STH	COMAND,FNDRNA	SET RETURN ADDRESS		CPY08630
	0BE8R						
0B9ER	D0E0		STM	FROM,RDBLK+4	PRIME I/O BLOCK		CPY08640
	0836R						
0BA2R	D390		LB	R9,OPRAND+1	LU		CPY08650
	0F35R						
0BA6R	C590		CLHI	R9,X*2A*	CHECK FOR STAR		CPY08660
	002A						
0BAAR	0237		BNER	CMDER			CPY08670
0BACR	C890		LHI	R9,X*48*	ASCII READ		CPY08680
	0048						
0BB0R	D290		STB	R9,RDBLK	PRIME I/O BLOCK		CPY08690
	0832R						
0BB4R	4110		BAL	LINK,FIND	READ A RECORD		CPY08700
	0B2CR						
0BB8R	0799	FNDA1	XHR	R9,R9			CPY08710
0BBAR	D3E9		LB	FROM,PRTBUF(R9)	COMPARE RECORDS		CPY08720
	06F8R						
0BBER	D3F9		LB	TO,LBLSAV(R9)			CPY08730
	0F18R						
0BC2R	05EF		CLHR	FROM,TO	IF NOT EQUAL - CONTINUE		CPY08740
0BC4R	2136		BNES	FNDA2			CPY08750
0BC6R	2691		AIS	R9,1			CPY08760
0BC8R	C590		CLHI	R9,8	COMPARE		CPY08770
	0008						
0BCCR	2335		BES	FNDA3			CPY08780
0RCER	220A		BS	FNDA1+2	CONTINUE CHECKING		CPY08790
0BD0R	4110	FNDA2	BAL	LINK,FND	CONTINUE LOOKING		CPY08800
	0C4AR						
0BD4R	220E		BS	FNDA1	COMPARE		CPY08810
0BD6R	E120	FNDA3	SVC	2,PGMFND	PGM FOUND - MESSAGE		CPY08820
	0BEAR						
0BDAR	D390		LB	R9,RDBLK+1	LU		CPY08821
	0833R						
0BDER	D290		STB	R9,BAKREC+1	SET LU FOR BACKSPACE		CPY08822
	0E21R						
0BE2R	E110		SVC	1,BAKREC	BACKSPACE ONE RECORD		CPY08823
	0E20R						
0BE6R	4300		DC	X*4300*			CPY08830
0BE8R	0000	FNDRNA	DC	0			CPY08840
0BEAR	0007	PGMFND	DC	7,FNDEND-FNDMSG			CPY08850

000A						
0BEER	5047	FNDMSG DC	C*PGM FOUND*			CPY08860
	4D20					
	464F					
	554E					
	4420					
0BF8R		FNDEND EQU *				CPY08870
		*-----*				CPY08880
		* FIND	BINARY FILE NAME			CPY08890
0BF8R	D1E0	FNDB	LM	FROM,LSTBFB	BUFFER EXTENTS	CPY08900
	0B20R					
0BF8R	D0E0		STM	FROM,RDBLK+4	PRIME I/O BLOCK	CPY08910
	0836R					
0C00R	C890		LHI	R9,X*58*	BINARY READ	CPY08920
	0058					
0C04R	D290		STB	R9,RDBLK		CPY08930
	0832R					
0C08R	4050		STH	COMAND,FNDRNA	SET RETURN	CPY08940
	0BE8R					
0C0CR	4110		BAL	LINK,FIND	READ A RECORD	CPY08950
	0B2CR					
0C10R	48E0	FNDB0	LH	FROM,PRTBUF+4	SET FOR SUBROUTINE	CPY08960
	06FCR					
0C14R	C8F0		LHI	TO,PRTBUF+4	SET FOR SUBROUTINE	CPY08970
	06FCR					
0C18R	24DC		LIS	NUMBER,12		CPY08980
0C1AR	4110		BAL	LINK,XTRACT	EXTRACT LOADER ITEM	CPY08990
	0D3ER					
0C1ER	27EF		SIS	FROM,15	LABEL ITEM 7	CPY09000
0C20R	213D		BNZS	FNDB2	NO	CPY09010
0C22R	07CC		XHR	R12,R12	YES	CPY09020
0C24R	41B0	FNDB1	BAL	R11,LDRWD	EXTRACT A HALFWORD	CPY09030
	0D50R					
0C28R	45AC		CLH	R10,LBSAV(R12)	COMPARE WITH ENTERED NAME	CPY09040
	0F18R					
0C2CR	2137		BNES	FNDB2	NOT EQUAL	CPY09050
0C2ER	26C2		AIS	R12,2	BUMP INDEX	CPY09060
0C30R	C5C0		CLHI	R12,6	DONE ?	CPY09070
	0006					
0C34R	2038		BNES	FNDB1	NO - GET ANOTHER HALFWORD	CPY09080
0C36R	4300		B	FNDAS	FOUND	CPY09090
	0BD6R					
0C3AR	4110	FNDB2	BAL	LINK,FND	CONTINUE	CPY09100
	0C4AR					
0C3ER	4180		BAL	RTN,VOLCHK	CHECK FOR ENDEVOL	CPY09110
	0E36R					
0C42R	4300		B	NOFND	NOT FOUND	CPY09120
	0B84R					
0C46R	4300		B	FNDB0		CPY09130
	0C10R					
0C4AR	4010	FND	STH	LINK,FNDRTN		CPY09140
	0B82K					
0C4ER	4300		B	FND1		CPY09150
	0B76R					
		*-----*				CPY09160
		* LABEL				CPY09170

0C52R	4110	LBLA	BAL	LINK,CLRLNE	CLEAR BUFFER	CPY09180
	07D2R					
0C56R	03D0		LB	NUMBER,OPRAND		CPY09190
	0F34R					
0C5AR	C5D0		CLHI	NUMBER,9	LENGTH CHECK	CPY09200
	0009					
0C5ER	0387		BNLR	CMDER	LESS = OK	CPY09210
0C60R	C8E0		LHI	FROM,OPRAND+1	SET SENDER ADDRESS	CPY09220
	0F35R					
0C64R	C8F0		LHI	TO,PRIBUF	SET RECEIVER ADDRESS	CPY09230
	06F8R					
0C68R	D39E		LB	R9,0(FROM)	GET A CHARACTER	CPY09240
	0000					
0C6CR	C590		CLHI	R9,X'2A'	STAR ?	CPY09250
	002A					
0C70R	0237		BNER	CMDER	ERROR IF NOT	CPY09260
0C72R	D39E		LB	R9,1(FROM)	GET SECOND CHARACTER	CPY09270
	0001					
0C76R	C590		CLHI	R9,X'2A'	STAR ?	CPY09280
	002A					
0C7AR	0237		BNER	CMDER	NO = ERROR	CPY09290
0C7CR	4110		BAL	LINK,MOVE	MOVE FILE NAME	CPY09300
	090ER					
0C80R	D1E0		LM	FROM,LSTBUF	BUFFER EXTENTS	CPY09310
	0B1CR					
0C84R	D0E0		STM	FROM,WRBLK+4	PRIME I/O BLOCK	CPY09320
	0A52K					
0C88R	C890		LHI	R9,X'2802'	WRITE ASCII	CPY09330
	2802					
0C8CR	4090		STH	R9,WRBLK	PRIME BLOCK	CPY09340
	084ER					
0C90R	4110	LBLA1	BAL	LINK,WRITE	WRITE LABEL RECORD	CPY09350
	083AR					
0C94R	0305		BR	COMAND		CPY09360
0C96R	C8D0	LBLB	LHI	NUMBER,108	BUFFER LENGTH	CPY09370
	006C					
0C9AR	C8F0		LHI	TO,PRIBUF	SET RECEIVER	CPY09380
	06F8R					
0C9ER	C8E0		LHI	FROM,X'2020'	CLEAR TO SPACES	CPY09390
	2020					
0CA2R	4110		BAL	LINK,CLEAR	CLEAR BUFFER	CPY09400
	092AR					
0CA6R	07EE		XHR	FROM,FROM	CLEAR VALUE	CPY09410
0CA8R	26FC		AIS	TO,12	BUMP RECEIVER	CPY09420
0CAAR	27DC		SIS	NUMBER,12	DECREMENT COUNT	CPY09430
0CACR	4110		BAL	LINK,CLEAR	CLEAR BUFFER TO LOW VALUES	CPY09440
	092AR					
0CB0R	D1E0		LM	FROM,LSTBFB	BUFFER LIMITS	CPY09450
	0B20R					
0CB4R	D0E0		STM	FROM,WRBLK+4	PRIME I/O BLOCK	CPY09460
	0A52K					
0CB8R	C890		LHI	R9,X'3802'	WRITE BINARY	CPY09470
	3802					
0CBCR	4090		STH	R9,WRBLK	PRIME BLOCK	CPY09480
	084ER					
0CC0R	C890		LHI	R9,-1	SEQUENCE NUMBER	CPY09490

0CC4R	FFFF 4090	STH	R9,PRTBUF		
0CC8R	06F8R C8F0	LHI	T0,PRTBUF+5	SET RECEIVER ADDRESS	CPY09500
0CCC	06F0P C8E0	LHI	FROM,OPRAND+1	LABEL ADDRESS	CPY09510
0CD0R	0F35R D3D0	LB	NUMBER,OPRAND	OPERAND LENGTH	CPY09520
0CD4R	0F34P 08DD	LHR	NUMBER,NUMBER	ZERO ?	
0CD6R	0337	RZR	CMDER	YES - ERROR	CPY09540
0CD8R	C5D0	CLHI	NUMBER,7	SIZE CHECK	CPY09550
0CDCR	0007				CPY09560
0CDER	0387	RNLR	CMDER	NOT LESS = ERROR	CPY09570
0CE2R	4110	BAL	LINK,MOVE	MOVE LABEL	CPY09580
0CE4R	090ER 2486	LIS	R11,6		
0CE8R	D3CF	LB	R12,0(T0)	PICK FIRST BYTE	CPY09590
0CEAR	0000	SRLS	R12,4	SAVE ZONE	CPY09600
0CEER	90C4	STB	R12,-1(T0)	SAVE FOUR BITS IN BUFFER	CPY09610
0CF2R	D2CF	LHLB1	LB R12,0(T0)	GET ANOTHER BYTE	CPY09620
0CF4R	FFFF	SLLS	R12,12	SHIFT OFF NUMERIC	
0CF8R	D3CF	LB	R10,1(T0)	GET ANOTHER BYTE	CPY09640
0CFAR	0000	SLLS	R10,4		CPY09650
0CFCR	91CC	OMR	R12,R10	FORM A BYTE	CPY09660
0CFER	D3AF	EXBR	R12,R12	EXCHANGE BYTES	CPY09670
0D02R	0001	STB	R12,0(T0)	STORE BYTE	CPY09680
0D04R	91A4	AIS	T0,1	BUMP SENDER/RECEIVER	CPY09700
0D06R	06CA	SIS	R11,1	DECREMENT COUNT	CPY09710
0D08R	94CC	BNZS	LHLB1	*	CPY09720
0D0CR	D2CF	LB	R10,PRTBUF+4		CPY09730
0D10R	0000	OMI	R10,X'F0'	SET LOADER ITEM	CPY09740
0D14R	26F1	STB	R10,PRTBUF+4		
0D18R	27B1	LB	R10,PRTBUF+10	GET LAST HALF BYTE	CPY09750
0D1CR	203C	NHI	R10,X'F0'	SET LOADER ITEM TO ZERO	CPY09770
0D1ER	D3A0	AIS	R10,14	SET LOADER ITEM	
0D22R	06FCR	STB	R10,PRTBUF+10	SET BACK IN BUFFER	CPY09775
0D26R	C6A0	LHI	R10,X'40'	'E4' ITEM	CPY09780
0D2AR	00F0	STB	R10,PRTBUF+11	CLEAR NEXT BYTE	CPY09790
0D2ER	00F0	LHI	NUMBER,108	LENGTH	CPY09800
	26AE	LHI	FROM,PRTBUF	START ADDRESS	CPY09810
	D2A0				CPY09820
	0702R				
	C8A0				
	0040				
	D2A0				
	0703R				
	C8D0				
	006C				
	C8E0				
	06F8R				

0D32R	4110	BAL	LINK,CKSUM	COMPUTE CHECKSUM	CPY09830
	0E24R				
0D36R	40F0	STH	TO,PRTBUF+2	STORE IN RECORD	CPY09840
	06FAR				
0D3AR	4300	B	LBLA1	CONTINUE	CPY09850
	0C90R				
*-----*					CPY09860
			EXTRACT A LOADER ITEM		CPY09870
0D3ER	CCED	XTRACT	SRHL FROM,0(NUMBER)	SHIFT X BITS	CPY09880
	0000				
0D42R	C4E0	NHI	FROM,X*0F*	GET ITEM	CPY09890
	000F				
0D46R	27D4	SIS	NUMBER,4	DECREMENT BIT COUNT	CPY09900
0D48R	0311	BNMR	LINK	CONTINUE	CPY09910
0D4AR	24DC	LIS	NUMBER,12	RESET BIT COUNT	CPY09920
0D4CR	26F2	AIS	TO,2	BUMP SENDER	CPY09930
0D4ER	0301	BR	LINK	RETURN	CPY09940
0D50R	2494	LDRWD	LIS R9,4		CPY09950
0D52R	48EF	LDRWD1	LH FROM,0(TO)	GET A HALFWORD	CPY09960
	0000				
0D56R	4110	BAL	LINK,XTRACT	EXTRACT AN ITEM	CPY09970
	0D3ER				
0D5AR	91A4	SLLS	R10,4		CPY09980
0D5CR	06AE	OHR	R10,FROM	PACK IT	CPY09990
0D5ER	2791	SIS	R9,1	DECREMENT COUNTER	CPY10000
0D60R	2037	BNZS	LDRWD1	CONTINUE	CPY10010
0D62R	030B	BR	R11	RETURN	CPY10020
*-----*					CPY10030
			HALT		CPY10040
0D64R	D000	HALT	STM R0,SAVE	SAVE REGS	CPY10050
	0E78R				
0D68R	E120	SVC	2,HLTMSG	HALT MESSAGE	CPY10060
	0D6ER				
0D6CR	0305	BR	COMAND		CPY10070
0D6ER	0007	HLTMSG	DC 7,HLTEND=HALTM		CPY10080
	0004				
0D72R	4841	HALTM	DC C'HALT'	HALT MESSAGE	CPY10090
	4C54				
0D76R		HLTEND	EQU *		CPY10100
*-----*					CPY10110
			CONTINUE		CPY10120
0D76R	D100	CONT	LM R0,SAVE	RESTORE REGS	CPY10130
	0E78R				
0D7AR	0301	BR	LINK		CPY10140
*-----*					CPY10151
0D7CR	D3D0	PROC	LB NUMBER,OPRAND	FILE NUMBER / LABEL	CPY10160
	0F34R				
0D80R	C8E0	LHI	FROM,OPRAND+1	SENDER ADDRESS	CPY10170
	0F35R				
0D84R	2431	LIS	R3,1	SET LU	CPY10180
0D86R	D230	STB	R3,RDBLK+1	SET READ LU	CPY10190
	0833P				
0D8AR	08DD	LHR	NUMBER,NUMBER	DEFAULT ?	CPY10200
0D8CR	2133	BNZS	PR1	NO	CPY10210
0D8ER	4300	B	PR6	CONTINUE	CPY10220
	0DF4R				



0D92R	C8A0 0DF0R	PR1	LHI	R10,PR5A	SET RETURN ADDRESS	CPY10230
0D96R	40A0 0BE8R		STH	R10,FNDRNA	SET RETURN	CPY10240
0D9AR	D090 0E78R		STM	R9,SAVE	SAVE REGISTERS	CPY10250
0D9ER	C8F0 0F18R		LHI	T0,LBLSAV	RECEIVER ADDRESS	CPY10260
0DA2R	4110 090ER		BAL	LINK,MOVE	MOVE LABEL	CPY10270
0DA6R	D39E 0000		LB	R9,0(FROM)	FIRST CHARACTER	CPY10280
0DAAR	C590 0030		CLHI	R9,X'30'	STAR ?	CPY10290
0DAER	2389		BNLS	PR2	NO	CPY10300
0DB0R	D1E0 0B1CR		LM	FROM,LSTBUF	I/O AREA	CPY10310
0DB4R	D0E0 0836R		STM	FROM,RDBLK+4	SET I/O BLOCK	CPY10320
0DB8R	4110 0C4AR		BAL	LINK,FND		CPY10330
0DBCR	4110 0BB8R		BAL	LINK,FNDA1	FIND FILE	CPY10340
0DC0R	C590 0040	PR2	CLHI	R9,X'40'	LABEL ?	CPY10350
0DC4R	218R		BLS	PR4	NO - NO. FILES	CPY10360
0DC6R	D1E0 0B20R		LM	FROM,LSTBFB	I/O AREA	CPY10370
0DCAR	D0E0 0836R		STM	FROM,RDBLK+4	PRIME I/O BLOCK	CPY10380
0DCER	4110 0C4AR		BAL	LINK,FND		CPY10390
0DD2R	4110 0C10R		BAL	LINK,FNDB0	FIND FILE	CPY10400
0DD6R	4300 0DF4R	PR3	B	PR6	CONTINUE	CPY10410
0DDAR	C8F0 0F48R	PR4	LHI	T0,WORK	SET RECEIVER	CPY10420
0DDER	4110 0896R		BAL	LINK,XLATDB	TRANSLATE TO HEX	CPY10430
0DE2R	48B0 0F48R		LH	R11,WORK		CPY10440
0DE6R	4110 063ER	PR5	BAL	LINK,SKIP	SKIP A FILE	CPY10450
0DEAR	27B1		SIS	R11,1	DECREMENT COUNTER	CPY10460
0DECR	2033		BNZS	PR5	CONTINUE	CPY10470
0DEER	2303		BS	PR6		CPY10480
0DF0R	D190 0E78R	PR5A	LM	R9,SAVE	RESTORE REGISTERS	CPY10490
0DF4R	CAED 0001	PR6	AHI	FROM,1(NUMBER)	SET POINTER	CPY10510
0DF8R	D3DE FFFF		LR	NUMBER,-1(FROM)	PICK UP LENGTH	CPY10520
0DFCR	080D		LHR	NUMBER,NUMBER	CHECK FOR DEFAULT	CPY10530
0DFER	213C		BNZS	PR9	NO	CPY10540
0E00R	48A0		LH	R10,CLRVAL	A OR B ?	CPY10550

0E04R	0F4ER		BZS	PR7	B		CPY10560
0E06R	C8A0		LHI	R10,80	ASCII = 80		CPY10570
	0050						
0E0AR	2303		BS	PR8			CPY10580
0E0CR	C8A0	PR7	LHI	R10,108	BINARY = 108		CPY10590
	006C						
0E10R	40A0	PR8	STH	R10,RECSIZ	RECEIVER		CPY10600
	029ER						
0E14R	0308		BR	RTN	RETURN		CPY10610
0E16R	C8F0	PR9	LHI	TO,RECSIZ	SET RECEIVER		CPY10620
	029ER						
0E1AR	4110		BAL	LINK,XLATDB	TRANSLATE		CPY10630
	0896R						
0E1ER	0308	PR10	BR	RTN	RETURN		CPY10640
0E20R	A001	HAKREC	DC	X'A001',0			CPY10650
	0000						
*-----*							CPY10660
			* CHECKSUM				CPY10670
0E24R	0AED	CKSUM	AHR	FROM,NUMBER			CPY10680
0E26R	C8F0		LHI	TO,-1	PRESET RECEIVER		CPY10690
	FFFF						
0E2AR	47FE	CK1	XH	TO,0(FROM)	X-OR HALFWORD		CPY10700
	0000						
0E2ER	27E2		SIS	FROM,2	DECREMENT SENDER		CPY10710
0E30R	27D2		SIS	NUMBER,2	DECREMENT COUNT		CPY10720
0E32R	2214		BNMS	CK1	CONTINUE		CPY10730
0E34R	0301		BR	LINK	RETURN		CPY10740
*-----*							CPY10750
			* ENDVOL CHECK				CPY10760
0E36R	D090	VOLCHK	STM	R9,SAVE	SAVE REGS		CPY10770
	0E78R						
0E3AR	48E0		LH	FROM,ROBLK+4	INPUT BUFFER		CPY10780
	0836R						
0E3ER	C8FE		LHI	TO,4(FROM)	SET FOR SUBROUTINE		CPY10790
	0004						
0E42R	48EE		LH	FROM,4(FROM)	SET FOR SUBROUTINE		CPY10800
	0004						
0E46R	24DC		LIS	NUMBER,12	SET SHIFT MASK		CPY10810
0E48R	4110		BAL	LINK,XTRACT	EXTRACT ITEM		CPY10820
	003ER						
0E4CR	27EF		SIS	FROM,15	LABEL ?		CPY10830
0E4ER	213E		BNZS	ENEXIT	NO		CPY10840
0E50R	07CC		XHR	R12,R12	CLEAR INDEX		CPY10850
0E52R	41B0	EN1	BAL	R11,LDRWD	GET A WORD		CPY10860
	0D50R						
0E56R	45AC		CLH	R10,ENDVOL(R12)	CHECK FOR ENDVOL		CPY10870
	0E72R						
0E5AR	2138		BNES	ENEXIT			CPY10880
0E5CR	26C2		AIS	R12,2	BUMP INDEX		CPY10890
0E5ER	C5C0		CLHI	R12,6	DONE ?		CPY10900
	0006						
0E62R	2038		BNES	EN1	NO		CPY10910
0E64R	D190		LM	R9,SAVE			CPY10920
	0E78R						
0E68R	0308		BR	RTN	RETURN		CPY10930

Address	Label	Operation	Mode	Value	Destination
0E6AR	D190	ENEXIT	LM	R9.SAVE	RESTORE REGS
0E78R	0E78R				CPY10940
0E6ER	2684		AIS	RTN,4	
0E70R	0308		BR	RTN	CPY10950
0E72R	454E	ENDVOL	DC	C*ENDVOL*	CPY10960
	4456				CPY10970
	4F4C				
0E78R		SAVE	DS	16	
0E88R		SAVE2	DS	16	CPY10980
0E98R		SAVE3	DS	16	CPY10990
0EA8R		SAVE4	DS	16	CPY11000
0EB8R		SAVE5	DS	16	CPY11010
0EC8R		BUFFER	EQU	*	CPY11020
0EC8R			00	40	CPY11030
0EC8R	2020		DC	X'2020'	CPY11040
0ECAR	2020		DC	X'2020'	CPY11050
0ECCR	2020		DC	X'2020'	CPY11050
0ECER	2020		DC	X'2020'	CPY11050
0ED0R	2020		DC	X'2020'	CPY11050
0ED2R	2020		DC	X'2020'	CPY11050
0ED4R	2020		DC	X'2020'	CPY11050
0ED6R	2020		DC	X'2020'	CPY11050
0ED8R	2020		DC	X'2020'	CPY11050
0EDAR	2020		DC	X'2020'	CPY11050
0EDCR	2020		DC	X'2020'	CPY11050
0EDER	2020		DC	X'2020'	CPY11050
0EE0R	2020		DC	X'2020'	CPY11050
0EE2R	2020		DC	X'2020'	CPY11050
0EE4R	2020		DC	X'2020'	CPY11050
0EE6R	2020		DC	X'2020'	CPY11050
0EE8R	2020		DC	X'2020'	CPY11050
0EEAR	2020		DC	X'2020'	CPY11050
0EECR	2020		DC	X'2020'	CPY11050
0EEER	2020		DC	X'2020'	CPY11050
0EF0R	2020		DC	X'2020'	CPY11050
0EF2R	2020		DC	X'2020'	CPY11050
0EF4R	2020		DC	X'2020'	CPY11050
0EF6R	2020		DC	X'2020'	CPY11050
0EF8R	2020		DC	X'2020'	CPY11050
0EFAR	2020		DC	X'2020'	CPY11050
0EFER	2020		DC	X'2020'	CPY11050
0F00R	2020		DC	X'2020'	CPY11050
0F02R	2020		DC	X'2020'	CPY11050
0F04R	2020		DC	X'2020'	CPY11050
0F06R	2020		DC	X'2020'	CPY11050
0F08R	2020		DC	X'2020'	CPY11050
0F0AR	2020		DC	X'2020'	CPY11050
0F0CR	2020		DC	X'2020'	CPY11050
0F0ER	2020		DC	X'2020'	CPY11050
0F10R	2020		DC	X'2020'	CPY11050
0F12R	2020		DC	X'2020'	CPY11050
0F14R	2020		DC	X'2020'	CPY11050
0F16R	2020		DC	X'2020'	CPY11050
0F18R		BUFEND	EQU	*	CPY11050
0F18R		LBL SAV	DS	8	CPY11060
					CPY11070

0F20R		SPCEND	EQU	*		CPY11080
0F20R	FFFF	VARFLG	DC	-1		CPY11090
0F22R	0000	REGSAV	DC	0,0,0,0,PRINT,OSCOPY,ERR,CMDERR,0		CPY11100
	0000					
	0000					
	0000					
	068AR					
	0Q00R					
	0856R					
	07B6R					
	0000					
0F34R		CLRREG	EQU	*		CPY11110
0F34R		OPRAND	DS	20		CPY11120
0F48R	0000	WORK	DC	0		CPY11130
0F4AR		WORK2	DS	4		CPY11140
0F4ER	0000	CLRVAL	DC	0		CPY11150
0F50R	0000	FMKFLG	DC	0		CPY11160
0F52R	0000	SKPNO	DC	0		CPY11170
0F54R	0000	BKNO	DC	0		CPY11180
0F56R	0000	QUANT	DC	0		CPY11190
0F58R		CLREND	EQU	*		CPY11200
0F58R		END		OSCOPY		CPY11210

NO ERRORS

BACKSP	06B6R
BAKRFC	0E20R
BKNO	0F54R
BKSP	06A4R
BSP	0678R
RSP2	06AEH
BUFEND	0F18R
BUFFER	0EC8R
C1	092ER
CK1	0E2AR
CKSUM	0E24R
CLEAR	092AR
CLRREG	0F34R
CLREND	0F58R
CLRLNE	07D2R
CLROUT	07F0R
CLRVAL	0F4ER
CMD1	0038R
CMD2	003ER
CMD3	0044R
CMD4	0058R
CMD5	005CR
CMD6	0080R
CMD7	00ACR
CMDBLK	00C0R
CMDEND	0152R
CMDER	0007
CMDERR	07B6R
CMDLST	00C8R
CMDMSG	07BCR
CMEND	07C8R
CMSG	07C0R
COMAND	0005
COMD	0026R
CONT	0D76R
COPY	0170R
CP10	027CR
CP11	0288R
CP2	0184R
CP3	01D8R
CP4	01E0R
CP5	01EER
CP6	0216R
CP7	022AR
CP7A	0240R
CP7B	0262R
CP8	0266R
CPEOF	0272R
CPEOV	028CR
CPEOV1	0292R
CPYA	015ER
CPYB	016CR
DS1	041CR
DS10	04C2R
DS11	04F4R

DS12	04FCR
DS13	0500R
DS13A	0512R
DS14	0526R
DS14A	0530R
DS15	0534R
DS6	0476R
DS7	047AR
DS8	048CR
DS9	0490R
DSCHEK	0516R
DSPA	0404R
DSPR	0412R
EN1	0E52R
END	07CER
ENDVOL	0E72R
ENEXIT	0E6AR
ERMEND	0894R
ERR	0856R
ERRMSG	0884R
ERROR	0006
ERRR	0880R
ERXIT	087ER
FETCH	02A0R
FIND	0R2CR
FLGCHK	03ACR
FMKFLG	0F50R
FND	0C4AR
FND1	0B76R
FNDA	0B96R
FNDA1	0BB8R
FNDA2	0BD0R
FNDA3	0BD6R
FNDB	0BF8R
FNDB0	0C10R
FNDB1	0C24R
FNDB2	0C3AR
FNDEND	0BF8R
FNDMSG	0BEER
FNORNA	0RE8R
FNDRTN	0B82R
FORM	076ER
FROM	000E
GETCOR	029AR
HALT	0D64R
HALTM	0D72R
HEADER	0590R
HEDEND	05B8R
HEDPRT	0552R
HEDRTN	058ER
HLTEND	0D76R
HLTMSG	0D6ER
INREC	0A22R
LABELS	0B24R
LASTAD	0004
LBLA	0C52R

LBLA1 0C90R  
 LBLB 0C96R  
 LBLC1 6CFER  
 LBLSAV 0F18R  
 LDRWD 0D50R  
 LDRWD1 0D52R  
 LINE 077ER  
 LINK 0001  
 LNRTN 07A2R  
 LST 0A46R  
 LST1 0A78R  
 LST2 0A98R  
 LST3 0AB0R  
 LST3A 0AB8R  
 LST4 0AD2R  
 LST5 0B0AR  
 LSTA 0A3ER  
 LSTB 0AFER  
 LSTBFB 0B20R  
 LSTBUF 0B1CR  
 M1 0912R  
 M2 0924R  
 MEM 054AR  
 MEMEND 0552R  
 MEMERR 0540R  
 MEMMSG 0546R  
 MOVE 090ER  
 NAMBLK 0152R  
 NAME 001ER  
 NAMEND 015ER  
 NAMST 0156R  
 NFEND 0D96R  
 NFND 0B8ER  
 NFNDMG 0B8AR  
 NOFND 0B84R  
 NONVER 09A2R  
 NUMBER 000D  
 NV 09C8R  
 NV0 09CCR  
 NV1 09D6R  
 NV2 09D8R  
 NV3 09ECR  
 NV4 0A0CR  
 NVR 07A4R  
 OPRAND 0F34R  
 OSCOPY 0000R  
 OUTREC 0A30R  
 PARSER 007CR  
 PAUSE 0B94R  
 PAWS 07C8R  
 PGMEND 0BEAR  
 PR1 0D92R  
 PR16 0E1ER  
 PR2 0DC0R  
 PR3 0DD6R  
 PR4 0DDAR

PR5	0DE6R
PR5A	0DF0R
PR6	0DF4R
PR7	0E0CR
PR8	0E10R
PR9	0E16R
PRINT	06BAR
PRNT1	06C6R
PROC	0D7CR
PRT1	0776R
PRTBLK	06F0R
PRTBUF	06F8R
PRTEND	0764R
PRTN	06EER
PTOF	0778R
QUANT	0F56R
R0	0000
R10	000A
R11	000B
R12	000C
R13	000D
R3	0003
R9	0009
RD1	0802R
RD2	0822R
RD3	080CR
RDBLK	0832R
ROBLK2	03FCR
RDCHK	0812R
RDEOF	0828R
RDEOV	081ER
RDEXIT	0808R
READ	07F2R
RECRDS	0002
RECSIZ	029ER
REGSAV	0F22R
RELCOR	029CR
REWIND	05D8R
RTN	0008
RWD	05B8R
RWND	05C8R
SAVE	0E78R
SAVE2	0E88R
SAVE3	0E98R
SAVE4	0EA8R
SAVE5	0EB8R
SKIP	063ER
SKIPFM	0674R
SKIPLN	0766R
SKP	0600R
SKP1	0632R
SKP2	0648R
SKPALL	0652R
SKPNO	0F52R
SKPRD	066CR
SPCEND	0F20R



TO	000F
VAR	07AER
VAR1G	0F20R
VFRA	02A4R
VERH	02BAR
VERIFY	02C6R
VERM	0A1BR
VERMND	0A22R
VERMSG	0A14R
VOLCHK	0E36R
VR10	03F0R
VR2	02D0R
VR3	0342R
VR4	0346R
VR5	034AR
VR6	038AR
VR7	0394R
VR8	03B8R
VR9	03C6R
VREOF	03D0R
VREOF2	03E0R
VREOF3	03E0R
WFM	05D0R
WFMK	05E0R
WORK	0F48R
WORK2	0F4AR
WRBLK	084ER
WRITE	083AR
WRTFMK	05FCR
XB1	08A4R
XB2	08B0R
XB3	08BCR
XC1	0942R
XC2	094CR
XC3	0950R
XC4	0958R
XD1	0976R
XD2	097ER
XD3	0998R
XH1	08D4R
XH2	08F0R
XH3	08F6R
XLATBD	096CR
XLATDR	0896R
XLATEC	093ER
XLATEH	08CCR
XTRACT	0D3ER

**OS/16 AIDS  
(AUTOMATIC INTERACTIVE  
DEBUGGING SYSTEM)  
PROGRAM MANUAL**

**INTERDATA®**

2 Crescent Place, Oceanport, New Jersey 07757

© INTERDATA INC., 1973  
All Rights Reserved  
Printed In U.S.A.  
March 1974

OS/16 AIDS

Table of Contents

	Page
1. Introduction-----	1
1.1 Configuration Requirements-----	1
1.2 Configuration Options-----	1
2. Terminology-----	2
3. Description of Operations-----	4
3.1 Environment Control-----	4
3.2 Cell/Register Examination and Modification-----	6
3.3 Program Utilities-----	11
4. Operating Procedures-----	16
4.1 Loading Procedures-----	16
4.2 Starting Location-----	16
4.3 Program Size-----	16
4.4 Device Selection-----	16
4.5 Error Conditions-----	17

Appendices/Illustrations

1. Summary of Directives-----	A1-1
2. Assembly Format Dump-----	A2-1/A2-2
3. Biased Assembly Format Dump-----	A3-1/A3-2
4. Decimal Format Dump-----	A4-1/A4-2
5. Floating Point Format Dump-----	A5-1/A5-2
6. Hexadecimal with Character Translation Dump-----	A6-1/A6-2
7. Sentry Location Dump-----	A7-1/A7-2
8. OS/16 AIDS Memory Map-----	A8-1/A8-2

OS/16 AIDS  
PROGRAM MANUAL

1. INTRODUCTION

OS/16 AIDS (Automatic Interactive Debugging System) is a comprehensive debugging program design to provide the user with a variety of debugging tools. Operating under one of the INTERDATA operating systems (BOSS or DOS), OS/16 AIDS accepts directives to provide cell/register manipulation in hexadecimal, decimal, character, floating point and assembly formats. Additionally, OS/16 AIDS provides the user with eight snapshot, trace and break-points to allow a total of twenty-four sentinel points for execution checkpointing. The built-in interpretive processor allows the user to protect up to eight cells and registers. When the contents of the protected cells or registers is altered, OS AIDS notifies the user of the instruction responsible for alteration. The dump formats (decimal hexadecimal with character translation, binary, floating point, disassembly and loader format) are available individually through directives and may also be specified under the snapshot option.

1.1 Configuration Requirements

OS/16 AIDS requires the following minimum configuration:

- any 16-bit Processor
- BOSS or DOS
- 8KB memory over and above the OS
- a console device (CRT or TTY)

1.2 Configuration Options

OS/16 AIDS requires a floating-point instruction set or an OS with software traps for floating-point format operations. Record-oriented devices supported by the OS may be used for list and dump operations.

## 2. Terminology

The term cell refers to a halfword (sixteen bit) memory location or register.

An open cell or register is one that is currently available for modification. To operate on a cell or register in memory, it must be made the current open cell. Floating point, assembly and character options may operate on multiples of cells.

Directives are instructions to OS/16 AIDS and consist of one or more characters that may be followed by operands. The characters, in most cases, represent the first letter of the type of operation or option desired; ex. O = Open, H = Hexadecimal, D = Dump, etc. Where possible the meaning of each character remains constant for each of the directives.

The operands may be addresses (relative or absolute), registers, data, instructions or special functions (i.e., increments or counters). Leading zeroes are not required and defaults are permitted in some cases. An illegal combination of directives or operands causes the response 'CMD-ERR' to be typed on the operator's console. OS/16 AIDS ignores an illegal directive and waits for another.

The delimiters used in the directive statements are flexible - with the exception of a required 'space' separating the directive and its operand(s). OS/16 AIDS recognizes a hyphen, open parenthesis, close parenthesis, comma, period, quote or space as separators (delimiters) between operands. This feature has most significance when entering data in assembly format. A normal assembly statement might be entered as RA LH 7,300(2). A faster version, eliminating the parentheses and upper case characters, could be entered as RA<sub>Δ</sub>LH<sub>Δ</sub>7<sub>Δ</sub>300<sub>Δ</sub>2 or RA<sub>Δ</sub>LH<sub>Δ</sub>7,300,2 etc. Multiple delimiters are treated as single delimiters when entered contiguously.

The relative addressing feature provided by the Bias (BI) command and the assembly directives permits the user to reference program relative addresses and instructions when using the Cell Examination and Modification directives. Program Utility directives require absolute address operands.

References to the next 'logical' cell etc. are based upon the length of the current open cell(s). The length of the data in the cell as determined by the display format is responsible for the logical length. For example, floating point values are full words, instructions may be full words or half words and character strings are variable in bytes.

Note that data may be displayed and replaced in decimal, hexadecimal, character and floating point formats. The formats are similar to INTERDATA Assembly constants except the 'DC' and single quotes are not required.

A Floating Point number consists of the following:

1. An optional leading plus or minus sign.
2. One or more decimal digits that may include a decimal point.
3. An optional 'E' character followed by an optional plus or minus sign and one or two decimal digits denoting a power of ten.

### 3. Description of Operations

There are three classes of directives:

1. Environment Control
2. Cell Examination and Modification
3. Program Utilities

The directive characters are always separated from the operands by one or more spaces. Environment control commands and utility dump operations respond with the message OS AIDS to indicate completion of the operation. In the examples, the user entered directive statement is underlined and the OS AIDS response is not. When directive statements are entered via the operator's console keyboard, the normal operating system control characters '@', '#', and '←' provide (respectively) the ability to pause, (under DOS), ignore a line and ignore a character. The carriage return (CR) is utilized only for operator console command input.

#### 3.1 Environment Control

LU Logical Unit Selection

OS AIDS is initialized with the command input device, list device and binary output device accessed through logical units 1, 3 and 5, respectively. Frequently, the user program being debugged utilizes one or more of these logical units. To prevent such conflicts, OS AIDS can change its logical units before debugging begins through the use of the 'LU' directive followed by the new logical units.

Example: LU 6,4,A (CR)  
OS AIDS

In this example, commands are read from the device assigned to logical unit 6, listed data written to logical unit 4 and binary data written to logical unit A. A pause (directive 'P') at this point returns the user to the operating system where the user should assign the desired physical devices to the defined logical units. The normal operating system CONTINUE command returns the user to OS AIDS.

BI Bias Definition

The relative addressing feature permits the user to reference addresses indicated in his program listing when debugging relocatable programs. The bias is usually set relative to the first location within a program, i.e., a program loaded at X'4500' would

set the bias to X'4500' to achieve memory and listing address correspondence. Cells opened after the establishment of a bias are now relative to the established bias regardless of the display format:

Example:  $\frac{BI_{\Delta}5100}{OS_{\Delta}AIDS}$  (CR) Sets the bias to X'5100'

$\frac{BI_{\Delta}0}{OS_{\Delta}AIDS}$  (CR) Sets the bias to zero

Note: The bias is initially set to zero.

P Pause

The pause directive causes an exit to the operating system. The operating system CONTINUE command can be used to return to OS AIDS. All sentries, as well as the bias and open cell/reg, remain in effect.

Example:  $\frac{P}{PAUSE}$  (CR) Pause OS AIDS

EN End

The end directive terminates OS AIDS, restores special SVC pointers and restores all sentry points.

Example:  $\frac{EN}{EOJ}$  (CR) Ends OS AIDS

Note: Prior to dumping a user program it is recommended that all Snapshot, Trace and Breakpoint sentries be removed. A quick method of accomplishing this is to utilize the end (EN) directive, restart OS AIDS and dump the program in the desired format. During normal debugging this is not required - provided the user recognizes inserted X'E1E0', X'E1E2', or X'E1E4' as OS AIDS sentries.

### 3.2 Cell examination and Modification

The directives described in this section provide memory cell and register examination and modification in the following formats:

- 'H' - Hexadecimal Format
- 'D' - Decimal Format
- 'C' - Character Format
- 'F' - Floating Point Format
- 'A' - Assembly Statement Format
- 'B' - Biased Assembly Statement Format



Care should be exercised when replacing data with the Character, Floating Point and Assembly formats. Floating point format generates a fullword, Assembly format may generate a fullword and character format allows string replacement (variable length). Judicious use will avoid inadvertent overlaying of adjoining cells or registers. The length of the data being accessed will be used to calculate the next logical cell except where the preceding cell is requested (defaults to the preceding halfword always).

## O Open Cell/Register

OS/16 AIDS provides the user with an extremely flexible system for cell modification and display. The 'O' directive is immediately followed by a second character that determines the format for displaying specified cells or registers. All addresses specified are relative to the BIAS.

Hexadecimal format causes two bytes to be opened and displayed in four hex characters.

Example:  $\text{OH}_{\Delta 3000} \text{ (CR)}$       Open cell X'3000'  
                    4300                      contents in hex format

Decimal format causes the contents of the specified memory location to be converted to decimal and displayed in 5 digits.

Example:  $\text{OD}_{\Delta 3000} \text{ (CR)}$       Open cell X'3000'  
                    17152                      contents in decimal format  
  X'4300' = 17152<sub>10</sub>

Character format causes the cell contents to be displayed in printer graphics. When the contents of the cell are non-printing characters, periods (.) are substituted.

Example:  $\text{OC}_{\Delta 3000} \text{ (CR)}$       Open cell X'3000'  
                    C.                              contents in printer graphics

Floating point format displays a fullword in hex format followed by the corresponding value in floating point ASCII format. The Floating point registers (0,2,4,6,8,10,12 and 14) must be opened as memory locations X'0', X'4', X'8', X'C', X'10', X'14', X'18', and X'1C'.

Example:  $\text{OF}_{\Delta 3000} \text{ (CR)}$       Open fullword at X'3000'  
                    43004000 4                      Contents in floating point format

The Assembly statement format causes the first byte of the cell to be converted into its corresponding mnemonic and either a halfword or fullword statement created. When an invalid OP code is encountered, the halfword is displayed as a defined constant with its character translation in the comment field.

Example: OA $\Delta$ 3000 (CR) Open cell X'3000'  
3000 43004000 BFC 0,4000

OA $\Delta$ 4000 (CR) Open cell X'4000'  
4000 5148 DC X'5148' C=PH

When a bias has been set, a given cell may be displayed as a biased assembly statement.

Example: BI $\Delta$ 3000 (CR) Set bias to X'3000'  
OS AIDS

OB  $\Delta$  0 (CR) Open Biased Assembly cell X'3000'  
0000R 43001000R BFC 0,1000

OA  $\Delta$  0 (CR)  
3000 43004000 BFC 0,4000

Note: When opening cells, an option is available to open consecutive logical cells. To open 'n' cells follow the address with a ',n' where 'n' is a decimal value.

Example: OH $\Delta$ 3000,4 (CR)  
4300  
4000  
D71E  
5000

The current open cell is the last cell opened.

#### OR Open Register

The 'R' directive allows user registers to be opened. The display format is forced to hexadecimal. After opening, the register may be modified in the same manner as a cell.

Example: OR  $\Delta$  A (CR) Open register 10  
0100 contents in Hex

Note: Use caution when modifying Registers in Floating Point, character or assembly formats due to the possibility of operand lengths in excess of 16 bits.

#### R Replace

The replace directive, 'R', permits the user to change an opened cell or register's contents with a specified value. The 'R' directive is immediately followed by a second character, H,D,C,F,A or B, which describes the format of the operand portion of the replace statement. Replacement in character format may be done with strings of 60 or less characters.

Example: RH $\Delta$ 48D0	(CR)	Replace Cell/Reg with X'48D0'
RD $\Delta$ 10000	(CR)	Replace Cell/Reg with X'2710'
RC $\Delta$ TEXT MESSAGE	(CR)	Replace Cells with text string
RF $\Delta$ 12.1E-45	(CR)	Replace Cells with Floating Point
RA $\Delta$ LHI $\Delta$ C,4000(9)	(CR)	Replace with instruction
RB $\Delta$ LHI $\Delta$ C,1000(9)	(CR)	Replace biased instruction

Note: In all cases the appropriate internal data, as specified in the operands, is placed into memory. Fullwords and variable length data are generated where applicable. (Floating Point, assembly statement and character strings).

### N Next Cell

The next logical cell is opened, but not displayed, when the 'N' directive is entered. The location counter is incremented by the logical length of the last open cell.

Example: N (CR) The current open Cell/Reg is closed and the next logical Cell/Reg is opened.

### (CR) or (SPACE) Open Next Cell

The use of a carriage return, (CR), when entering directives through the operator's console keyboard or a space when entering from another device is interpreted as the directive requesting display of the next logical Cell/Reg. The secondary character may be omitted and the Cell/Reg is opened in the format last specified. If the default option is not desired, any valid secondary character used with the open directive may be applied. When a secondary character is specified, the primary directive must be a space.

Example: <u><math>\Delta</math></u> (CR)	Opens next Cell/Reg in previously established format
<u>(CR)</u>	Opens next Cell/Reg in previously established format
<u><math>\Delta</math>H</u> (CR)	Opens next Cell/Reg in Hex format
<u><math>\Delta</math>D</u> (CR)	Opens next Cell/Reg in Decimal format
<u><math>\Delta</math>C</u> (CR)	Opens next Cell/Reg in Character format
<u><math>\Delta</math>F</u> (CR)	Opens next Cell/Reg in Floating Point format

- ΔA (CR) Opens next Cell/Reg in Assembly Statement format
- ΔB (CR) Opens next Cell/Reg in Biased Assembly Statement format

O Reshow

The ability to reshow an open cell/register in a different display format is provided by the Open 'O' directive with a default operand. By specifying the Open 'O' directive in conjunction with one of the secondary characters without an operand address, the current open Cell/Reg is re-displayed in the specified format. The operand address is defaulted. The secondary character may also be defaulted to reshow the open Cell/Reg in the previously established format.

- Preceding Cell

The Cell/Reg immediately preceding the current open Cell/Reg may be displayed using the '-' directive. The secondary character may be defaulted to the previous format or may specify any of the secondary characters that are valid for the Open, 'O' directive.

- Example:
- (CR) Open preceding in established format
  - H (CR) Open preceding in Hex format
  - D (CR) Open preceding in Decimal format
  - C (CR) Open preceding in Character format
  - F (CR) Open preceding in Floating Point format
  - A (CR) Open preceding in Assembly Statement format
  - B (CR) Open preceding in Biased Assembly Statement format

Note: In some instances, with floating point and assembly statement formats, two Preceding Cell directives may be required for logical fullword alignment. The Preceding Cell directive decrements the current open cell address by the length of one halfword cell only.

J Jump to Address Contained in Open Cell/Reg

OS AIDS opens the Cell/Reg specified by the contents of the current open cell when the Jump, 'J' directive is entered. The inclusion of a secondary character (H,D,C,F,A,B or R) causes the

Cell/Reg to be displayed in a given format. When the secondary character is not entered, OS AIDS defaults to the previously established format.

Example: If the current open cell contains X'4100', location X'4100' is opened and displayed by the following directives:

J	(CR)	Defaults to previous format
JH	(CR)	Opens in Hex format
JD	(CR)	Opens in Decimal format
JC	(CR)	Opens in Character format
JF	(CR)	Opens in Floating Point format
JA	(CR)	Opens in Assembly Statement format
JB	(CR)	Opens in Biased Assembly Statement format
JR	(CR)	Opens register in Hex format

If the current open cell contains a value less than X'10', the user may wish to interpret the value as a register. In this case the 'JR CR' directive would open the appropriate register.

#### A Add to Contents

The Add directive, 'A', permits the user to increment an open cell/reg with a given value. The secondary directives (H, D & F) all use values expressed in hexadecimal, decimal or Floating Point.

Example:

<u>OH<sub>Δ</sub>4000</u> CR	Open cell as Hex
2A18	Open cell content = X'2A18'
<u>AH<sub>Δ</sub>1A00</u> CR	Add Hex value
	Open cell contents = X'4418'
<u>OD<sub>Δ</sub>4000</u> CR	Open cell as Decimal
17432	Open cell contents = X'4418'
<u>AD<sub>Δ</sub>256</u> CR	Add decimal value
	Open cell contents = X'4518'
<u>OF</u> CR	Open cell as Floating Point
4000 45180000 98304	Open cell contents = X'45180000'
<u>AF<sub>Δ</sub>100000</u> CR	Add Floating Point value
	Open cell contents = X'451A7100'
<u>AF<sub>Δ</sub>1E5</u> CR	Add Floating Point Value
	Open cell contents = X'451CE200'

### S Subtract from Contents

The Subtract directive, 'S', provides the capability of decrementing an open cell/reg with a given value. The secondary directives (H, D & F) allow use of values expressed in hexadecimal, decimal or Floating Point.

Example:

<u>OH 4000 (CR)</u> 701C	Open Cell as Hex Open cell contents = X'701C'
<u>SH 1A00 (CR)</u>	Subtract Hex value Open cell contents = X'461C'
<u>OD (CR)</u> 17948	Open Cell as Decimal
<u>SD 256 (CR)</u>	Subtract Decimal value Open cell contents = X'451C'
<u>OF (CR)</u> 4000 451CE200 118304	Open Cell as Floating Point Open cell contents = X'451CE200'
<u>SF 1E4 (CR)</u>	Subtract Floating Point value Open cell contents = X'451A7100'
<u>SF 10000 (CR)</u>	Subtract Floating Point value Open cell contents = X'45180000'

### 3.3 Program Utilities

The directives contained in this section provide a two-fold purpose. First, they provide a means of viewing an executing program as well as allowing changes during a logical halt. Secondly, they provide general purpose utilities for creating assembly type listings, absolute loader format program files and binary program files.

The bias defined in Environment Control applies only to the biased assembly output 'DB'. All applicable operands are assumed to be absolute addresses (i.e., the user must use absolute addresses as operands for OS AIDS Utility directives).

#### D Dump

The 'D' directive causes OS AIDS to output the contents of a section of memory defined by the specified low and high limit operands. The format of the dump and the device dumped to are dictated by the required secondary characters. The directive conforms to the following format.

D(+option) (Low Addr), (High Addr)

The only exception to the above is the utility dump (DU) that displays the location of Snapshot, Trace, and Breakpoint Sentries as well as the cells and registers that are currently protected.

Example: DH $\Delta$ 3000,3A90 (CR) Binary dump to binary output device

DD $\Delta$ 3000,3A90 (CR) Decimal dump to list device  
(See Appendix 4)

DC $\Delta$ 3000,3A90 (CR) Hex character dump to list device  
(See Appendix 6)

*copies a prog from core to disc!* → DL $\Delta$ 3000,3A90 (CR) Loader format dump to binary device

DF $\Delta$ 3000,3A90 (CR) Floating Point format dump to list device  
(See Appendix 5)

DA $\Delta$ 3000,3A90 (CR) Assembly type dump to list device  
(See Appendix 2)

DB $\Delta$ 3000,3A90 (CR) Biased Assembly type dump to list device  
(See Appendix 3)

DU (CR) Sentry locations and protected cell/registers to list device  
(See Appendix 7) (no address operands)

Note: All dumps return to the command mode and log 'OS AIDS' on the operator's console upon completion.

'Final' versions of Assembly format dumps and Loader format dumps should reflect only the program modifications made by the user. The existence of active sentinels within user programs causes erroneous dumps. Therefore, sentries should be eliminated by the user through the ZAP directives before dumping.

The 'Loader' format dump is an absolute program capable of loading and executing under an operating system. The 'Binary' dump may be loaded by the '50 Sequence' or an SVC 1.

## I Insert Sentry

Sentries may be one of 4 program logic control mechanisms:

- 1) X - BREAKPOINTS
- 2) T - TRACE POINTS
- 3) S - SNAPSHOT POINTS
- 4) P - PROTECT POINTS

The 'I' directive, when modified by a secondary character (X,T,S or P) causes OS AIDS to insert a sentry (SVC 14) into the user program and retain the original instruction, etc., within an internal table. The sentry should always reside in a user execution path and although it is a legal instruction, its presence is logically transparent to the executing program. In every case, the instruction replaced by the sentry is executed interpretively by OS AIDS before the function is performed.

The insertion of the Breakpoint (X) and Trace (T) sentries is at the operand addresses specified. The insertion of the Snapshot (S) sentry is accomplished at the current open cell and is not dependent upon the operands. The Protection (P) sentry allows cells and registers specified by the operand address to be protected. If a protected cell or register's contents change as a result of user instructions, OS AIDS notifies the user of the offending instruction's location. When protection is specified, the entire user program is executed interpretively until all protected cells or registers are released (Zapped).

Example: IX<sub>A</sub>32F0 (CR)  
OS AIDS

Inserts Breakpoint sentry at X'32F0'

BRK: 32F0

Execution of the sentry logs this message to the operator's console. The current open cell is set to the next logical instruction. OS AIDS is now in the command mode. The 'GO' directive with no operand address causes processing to continue in line.

Example: IT<sub>A</sub>3100,316A (CR)  
OS AIDS

Inserts a Trace sentry at X'3100' and lists the execution thread through X'316A'

TRC: 3100  
3102  
3104  
3108  
310A  
etc.

Execution of the sentry will cause listing of instruction addresses after they are executed. When execution exceeds the high address, tracing stops and execution continues in line without interpretation.

Example: IP<sub>A</sub>375E  
OS AIDS

Initiates protection of cell X'375E'

IPR<sub>A</sub>  
OS AIDS

Initiates protection of register 10.



PRO: 375E 36F2

After restarting the user program through the 'GO' directive, a modification of the protected cell or register will cause this message indicating the cell or register affected and the effecting instruction location. OS AIDS is in the command mode.

Example: OH<sub>A</sub>3F1C (CR)  
          48C0

Open the cell where a Snapshot is to be inserted.

IS\* <sub>A</sub>3000,3100,10 (CR)

When the sentry inserted at the current open cell is executed 10 times, a dump of memory between X'3000' and X'3100' occurs. The dump format is specified by a third character. The executions between dumps (third operand) is specified in decimal and may not exceed 255. The third operand may be defaulted in which case the dump occurs upon every execution of the sentry. See Appendices for dump formats.

\* = CHOICE OF:  
H - Binary dump - Binary device  
C - Hex with Character - List device  
D - Decimal dump - List device  
F - Floating Point - list device  
L - Loader format dump - Binary Device  
A - Assembly statement dump - list device  
B - Biased Assembly Statement dump - list device

Note: Multiple sentries should not be placed in the same memory location. i.e., a snapshot sentry and a breakpoint sentry should not be placed in the same cell. Sentries may be placed in a 'Trace' path but should not be entered in the first cell of the path.

Nested (overlapping) Trace extents would present a confusing picture of the execution path. To preclude this possibility, OS AIDS rejects a Trace insertion with overlapping extents.

The 'Protect' capability is effective for user registers only after the user program has established the initial register contents. It is recommended that Protection be established after the execution of a Breakpoint sentry. At this time, the user will have his registers loaded.

The protection feature is effective when the user issues the 'GO' directive. Protection is pre-empted by Snapshot, Tracing and Breakpoints and is not active during the execution of sentries or trace extents.

## Z Zap Sentry

Breakpointing, Tracing, Snapshotting, and Protecting require entries in OS AIDS tables and instruction changes within user programs. When individual sentries or entire sentry operations

are no longer required, they may be eliminated with the ZAP instruction which in turn restores the original cell contents. The use of the 'Dump Utility Locations' directive (DU) provides a list of all individual sentries which may be useful prior to 'Zapping' them. All sentries within the same function may be 'Zapped' by defaulting the address operand.

Examples: <u>ZX<math>\Delta</math>3000</u> (CR)	Zap Breakpoint at X'3000'
OS AIDS	
<u>ZX</u> (CR)	Zap all Breakpoints
OS AIDS	
<u>ZT<math>\Delta</math>3FA0</u> (CR)	Zap Trace at X'3FA0'
OS AIDS	
<u>ZT</u> (CR)	Zap all Tracing
OS AIDS	
<u>ZS<math>\Delta</math>41B2</u> (CR)	Zap Snapshot at X'41B2'
OS AIDS	
<u>ZS</u> (CR)	Zap all Snapshots
OS AIDS	
<u>ZP<math>\Delta</math>3806</u> (CR)	Zap protection of cell X'3806'
OS AIDS	
<u>ZPR<math>\Delta</math>E</u> (CR)	Zap protection of register 14
OS AIDS	
<u>ZP</u> (CR)	Zap all protection
OS AIDS	

GO GO Execute

The 'GO' directive may be used to start and return processing to the user program with the status and condition code restored to its in-line or initial value. The current open cell is used as the execution address when the address operand is defaulted.

Example: <u>GO<math>\Delta</math>3000</u> (CR)	Begin execution at X'3000'
<u>OH<math>\Delta</math>31CE</u> (CR)	Open cell established as X'31CE'
4800	
<u>GO</u> (CR)	Execution begins at X'31CE'

Note: When a Breakpoint or Protected Cell/Reg causes an operator message, the current open cell has automatically updated to the next executable instruction in the user's logic path. Use of the 'GO' directive with a defaulted address will return processing in line.

## 4. Operating Procedures

### 4.1 Loading Procedures

OS AIDS is a relocatable program in Loader format and can be loaded using the 'LOAD' directive in the operating system or via the OS Library Loader.

### 4.2 Starting Location

The starting location for OS/16 AIDS is the first instruction in the program. If OS/16 AIDS were loaded at X'3000' the starting address would be X'3000'.

Restarting OS/16 AIDS does not affect any of the established sentries or table entries. The restart address is also the first instruction.

OS/16 AIDS has a transfer address for operating systems (DOS, BOSS) that utilize the 'START' command with a defaulted address.

### 4.3 Program Size

OS/16 AIDS occupies approximately 6.5KB of memory. When memory is critical, functions of OS AIDS may be overlaid. The directives that pertain to the overlaid functions should not be used. See Appendix 8 for OS AIDS Memory Map.

When functions are overlaid, the End 'EN' directive should not be utilized to terminate OS AIDS. The overlaying of OS AIDS lists would cause erroneous instruction/sentry restoration. The sentries still intact, should be individually 'Zapped' and the Pause 'P' directive used to return to the operating system.

### 4.4 Device Selection

OS AIDS is initialized to use logical units 1, 3 and 5 as the command input, list and binary output devices (respectively). Batch operation is provided by assigning the command input device to other than the operator's console keyboard. Additionally, the list device may be assigned to a disc, tape etc. and be printed at a later date through the facilities of OS COPY or the DOS copy directive. All cell/register examination and modification output is forced to the operator's console in an assumed interactive mode.

#### 4.5 Error Conditions

Format errors or illegal functions presented as directives are rejected with the message 'CMD-ERR' logged to the operator's console. OS AIDS returns to the command mode and awaits further communication.

Device errors cause the message 'I/O - ERR: XXPU' to be logged to the operator's console (XX = status, PU = physical unit device address). OS AIDS then enters the command mode in anticipation of operator correction.

Illegal SVC 14's (OS/16 AIDS Sentries) will be brought to the user's attention by logging the message 'ILL: NNNN' on the operator's console (NNNN = address of illegal SVC 14). OS AIDS returns to the command mode.

APPENDIX 1

OS/16 AIDS - SUMMARY OF DIRECTIVES

ENVIRONMENT CONTROL

LU	Δ	( CMD INPUT LU )	,	( ASCII LIST LU )	,	( BINARY LU )	(CR)	Set logical units
P		(CR)						Pause
EN		(CR)						END
BI	Δ	(ADDR)		(CR)				Set bias

CELL/REGISTER EXAMINATION & MODIFICATION

OH	Δ	(ADDR)	(CR)	OPEN CELL AS HEXADECIMAL
OD	Δ	(ADDR)	(CR)	OPEN CELL AS DECIMAL
OC	Δ	(ADDR)	(CR)	OPEN CELL AS CHARACTER
OA	Δ	(ADDR)	(CR)	OPEN CELL AS ASSEMBLY STATEMENT
OB	Δ	(ADDR)	(CR)	OPEN CELL AS BIASED ASSEMBLY STATEMENT
OF	Δ	(ADDR)	(CR)	OPEN CELL AS FLOATING POINT
OR	Δ	(REG)	(CR)	OPEN REGISTER (HEX FORMAT)
N		(CR)		NEXT LOGICAL CELL/REG (NO DISPLAY)
		(CR)		OPEN NEXT CELL/REG (PREVIOUS FORMAT)
Δ H		(CR)		OPEN NEXT CELL/REG AS HEX
Δ D		(CR)		OPEN NEXT CELL/REG AS DECIMAL
Δ C		(CR)		OPEN NEXT CELL/REG AS CHARACTER
Δ A		(CR)		OPEN NEXT CELL/REG AS ASSEMBLY
Δ F		(CR)		OPEN NEXT CELL/REG AS FLOATING POINT
Δ B		(CR)		OPEN NEXT CELL/REG AS BIASED ASSEMBLY STATEMENT

-	(CR)	OPEN PRECEDING CELL (IN ESTABLISHED FORMAT)
-H	(CR)	OPEN PRECEDING CELL AS HEX
-D	(CR)	OPEN PRECEDING CELL AS DECIMAL
-C	(CR)	OPEN PRECEDING CELL AS CHARACTER
-A	(CR)	OPEN PRECEDING CELL AS ASSEMBLY STATEMENT
-B	(CR)	OPEN PRECEDING CELL AS BIASED ASSEMBLY STATEMENT
-F	(CR)	OPEN PRECEDING CELL AS FLOATING POINT
OH	(CR)	RESHOW CELL/REG AS HEX
OD	(CR)	RESHOW CELL/REG AS DECIMAL
OC	(CR)	RESHOW CELL/REG AS CHARACTER
OA	(CR)	RESHOW CELL AS ASSEMBLY STATEMENT
OB	(CR)	RESHOW CELL AS BIASED ASSEMBLY STATEMENT
OF	(CR)	RESHOW CELL/REG AS FLOATING POINT
O	(CR)	RESHOW CELL/REG IN PREVIOUS FORMAT
RH <sub>Δ</sub> (HEX VALUE)	(CR)	REPLACE CELL/REG WITH HEX VALUE
RD <sub>Δ</sub> (DEC VALUE)	(CR)	REPLACE CELL/REG WITH DECIMAL VALUE
RC <sub>Δ</sub> (CHAR STRING)	(CR)	REPLACE CELL/REG WITH CHARACTER STRING
RA <sub>Δ</sub> (ASSEMBLY STATEMENT)	(CR)	REPLACE CELL/REG WITH ASSEMBLED STATEMENT
RB <sub>Δ</sub> (ASSEMBLY STATEMENT)	(CR)	REPLACE CELL/REG WITH BIASED, ASSEMBLED STATEMENT
RF <sub>Δ</sub> (FLOATING POINT NO.)	(CR)	REPLACE CELL/REG WITH FLOATING POINT NO.
JH	(CR)	JUMP TO ADDRESS IN OPEN CELL/REG & OPEN AS HEX
HD	(CR)	JUMP TO ADDRESS IN OPEN CELL/REG & OPEN AS DECIMAL
JC	(CR)	JUMP TO ADDRESS IN OPEN CELL/REG & OPEN AS CHARACTER

JA (CR) JUMP TO ADDRESS IN OPEN CELL/REG & OPEN AS ASSEMBLY STATEMENT  
 JB (CR) JUMP TO ADDRESS IN OPEN CELL/REG & OPEN AS BIASED ASSEMBLY STATEMENT  
 JF (CR) JUMP TO ADDRESS IN OPEN CELL/REG & OPEN AS FLOATING POINT  
 J (CR) JUMP TO ADDRESS IN OPEN CELL/REG & OPEN IN PREVIOUS MODE  
 JR (CR) OPEN REGISTER SPECIFIED IN OPEN CELL/REG  
  
 AH<sub>Δ</sub>(HEX VALUE) (CR) ADD VALUE (HEX) TO OPEN CELL/REG  
 AD<sub>Δ</sub>(DEC VALUE) (CR) ADD VALUE (DEC) TO OPEN CELL/REG  
 AF<sub>Δ</sub>(FLOATING POINT NO.) (CR) ADD VALUE (FLOATING POINT) TO OPEN CELL/REG  
 SH<sub>Δ</sub>(HEX VALUE) (CR) SUBTRACT VALUE (HEX) FROM OPEN CELL/REG  
 SD<sub>Δ</sub>(DEC VALUE) (CR) SUBTRACT VALUE (DEC) FROM OPEN CELL/REG  
 SF<sub>Δ</sub>(FLOATING POINT NO.) (CR) SUBTRACT VALUE (FLOATING POINT) FROM OPEN CELL/REG

PROGRAM UTILITIES

DU (CR)	DUMP UTILITY SENTINEL & PROTECTION LOCATIONS
DH <sub>Δ</sub> (low addr), (high addr) (CR)	DUMP IN BINARY FORMAT (BINARY DEVICE)
DB <sub>Δ</sub> (low addr), (high addr) (CR)	DUMP IN BIASED ASSEMBLY FORMAT (LIST DEVICE)
DA <sub>Δ</sub> (low addr), (high addr) (CR)	DUMP IN ASSEMBLY FORMAT (LIST DEVICE)
DC <sub>Δ</sub> (low addr), (high addr) (CR)	DUMP IN HEX WITH CHARACTER TRANSLATION (LIST DEVICE)
DL <sub>Δ</sub> (low addr), (high addr) (CR)	DUMP IN LOADER FORMAT (BINARY DEVICE)
DD <sub>Δ</sub> (low addr), (high addr) (CR)	DUMP IN DECIMAL FORMAT (LIST DEVICE)
DF <sub>Δ</sub> (low addr), (high addr) (CR)	DUMP IN FLT. PT. FORMAT (LIST DEVICE)
IX <sub>Δ</sub> (ADDR) (CR)	INSERT BREAKPOINT
IT <sub>Δ</sub> (low addr), (high addr) (CR)	INSERT TRACT
ISH <sub>Δ</sub> (low addr), (high addr), (count) (CR)	INSERT SNAPSHOT - BINARY DUMP
ISA <sub>Δ</sub> (low addr), (high addr), (count) (CR)	INSERT SNAPSHOT - ASSEMBLY DUMP
ISB <sub>Δ</sub> (low addr), (high addr), (count) (CR)	INSERT SNAPSHOT - BIASED ASSEMBLY DUMP
ISC <sub>Δ</sub> (low addr), (high addr), (count) (CR)	INSERT SNAPSHOT - HEX and CHAR DUMP
ISL <sub>Δ</sub> (low addr), (high addr), (count) (CR)	INSERT SNAPSHOT - LOADER DUMP
ISD <sub>Δ</sub> (low addr), (high addr), (count) (CR)	INSERT SNAPSHOT - DECIMAL DUMP
ISF <sub>Δ</sub> (low addr), (high addr), (count) (CR)	INSERT SNAPSHOT - FLT. PT. DUMP
IP <sub>Δ</sub> (ADDR)	INITIATE PROTECTION (CELL)
IPR <sub>Δ</sub> (REG)	INITIATE PROTECTION (REG)



ZX	(CR)	ZAP ALL BREAKPOINTS
ZX	$\Delta$ (ADDR)	(CR) ZAP BREAKPOINT AT ADDR
ZT	(CR)	ZAP ALL TRACE
ZT	$\Delta$ (ADDR)	(CR) ZAP TRACE AT ADDR
ZS	(CR)	ZAP ALL SNAPSHOTS
ZS	$\Delta$ (ADDR)	(CR) ZAP SNAPSHOT AT ADDR
ZP	(CR)	ZAP ALL PROTECTION
ZPR	$\Delta$ (REG)	(CR) ZAP PROTECTION OF REG
ZP	$\Delta$ (ADDR)	(CR) ZAP PROTECTION OF ADDR
GO	$\Delta$ (ADDR)	(CR) GO EXECUTE AT ADDR
GO	(CR)	GO EXECUTE AT CURRENT OPEN CELL

APPENDIX 2  
ASSEMBLY FORMAT DUMP

5000	C8100020	LHI	1,0020
5004	0777	XHR	7,7
5006	40705C9A	STH	7,5C9A
500A	E1105BD4	SVC	1,5BD4
500F	E1105BDC	SVC	1,5BDC
5012	48005BDE	LH	0,5BDE
5016	4230506A	BTC	3,506A
501A	C8702020	LHI	7,2020
501E	40705CAA	STH	7,5CAA
5022	40705CAC	STH	7,5CAC
5026	07AA	XHR	A,A
5028	D37A5CB0	LB	7,5CB0(A)
502C	C570000D	CLHI	7,000D
5030	2339	BFFS	3,9
5032	C5A00004	CLHI	A,0004
5036	23A3	BFFS	8,3
5038	D27A5CAA	STB	7,5CAA(A)
503C	26A1	AIS	A,1
503E	0571	CLHR	7,1
5040	203C	RTBS	3,C

APPENDIX 3  
BIASED ASSEMBLY DUMP

0000R	C8100020	LHI	1,0020
0004R	0777	XHR	7,7
0006R	40700C9AR	STH	7,0C9A
000AR	E1100BD4R	SVC	1,0BD4
000ER	E1100BDCK	SVC	1,0BDC
0012R	48000BDEK	LH	0,0BDE
0016R	4230006AR	BTC	3,006A
001AR	C8702020	LHI	7,2020
001ER	40700CAAR	STH	7,0CAA
0022R	40700CACR	STH	7,0CAC
0026R	07AA	XHR	A,A
0028R	D37A0CB0R	LB	7,0CB0(A)
002CR	C570000D	CLHI	7,000D
0030R	2339	BFFS	3,9
0032R	C5A00004	CLHI	A,0004
0036R	23P3	BFFS	8,3
0038R	D27A0CAAR	STB	7,0CAA(A)
003CR	26A1	AIS	A,1
003ER	0571	CLHR	7,1
0040R	203C	RTBS	3,C

APPENDIX 4  
DECIMAL FORMAT DUMP

5000	C810=	51216	0020=	00032	0777=	01911	4070=	16496
5008	5C9A=	23706	E110=	57616	5BD4=	23508	F110=	57616
5010	5B0C=	23516	4800=	18432	5BDE=	23518	4230=	16944
5018	506A=	20586	C870=	51312	2020=	08224	4070=	16496
5020	5CAA=	23722	4070=	16496	5CAC=	23724	07AA=	01962
5028	037A=	54138	5CB0=	23728	C570=	50544	000D=	00013
5030	2339=	09017	C5A0=	50592	0004=	00004	2383=	09091
5038	027A=	53882	5CAA=	23722	26A1=	09889	0571=	01393
5040	203C=	08252	4870=	18544	5CAA=	23722	4880=	18560

APPENDIX 5  
FLOATING POINT FORMAT DUMP

5000	C8100020	-.26844E1	5004	07774070	.10799E-68
5008	5C9AE110	.31413E35	500C	5B04E110	.26985E34
5010	5B0C4800	.2792E34	5014	5BDE4230	.28174E34
5018	506AC870	.76945E2	501C	20204070	.37023E-39
5020	5CAA4070	.34531E35	5024	5CAC07AA	.34891E35
5028	D37A5CB0	-.36114E24	502C	C570000D	-458753
5030	2339C5A0	.27164E-35	5034	00042383	4

APPENDIX 6  
HEXADECIMAL WITH CHARACTER TRANSLATION DUMP

```
5000 C810 0020 0777 4070 5C9A E110 5BD4 E110 * ... ..@.\...[... *
5010 5BDC 4800 5BDE 4230 506A C870 2020 4070 * [.H.[.BOP... @. *
5020 5CAA 4070 5CAC 07AA D37A 5CH0 C570 000D * \.@.\.....\..... *
5030 2339 C5A0 0004 2383 D27A 5CAA 26A1 0571 * #9.....#...\&... *
5040 203C 4870 5CAA 4880 5CAC 0799 4579 5B7A * <H.\.H.\...E.[. *
```

APPENDIX 7

SENTRY POINT DUMP

DU

BRK: 7000 7AF0 6884 6888 721C

TRC: 8100

SNP: 9ADC A20F

PRO: A41C A42E A504

OS AIDS

Appendix 8

OS/16 AIDS Memory Map

OSAIDS

DUMP

UTIL

TRACE

SNAP

PROTEK

XLATEU

MSDEND

<p>Directives:  Open, Replace, Next, Preceding,  Jump, Add, Subtract, LU, Bias,  Pause, &amp; GO</p> <p>Formats:  Hexadecimal, Decimal, Character  &amp; Floating Point.</p>
<p>DUMP DIRECTIVES</p>
<p>BREAKPOINT DIRECTIVES</p>
<p>TRACE DIRECTIVES</p>
<p>SNAPSHOT DIRECTIVES</p>
<p>PROTECTION DIRECTIVES</p>
<p>ASSEMBLY DIRECTIVES</p>

Note: User should reference the OS/16 AIDS Assembly listing to correspond symbolic locations with relative addresses to calculate the amount of overlaying required and its resultant limitations.



PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From \_\_\_\_\_ Date \_\_\_\_\_

Title \_\_\_\_\_ Publication Title \_\_\_\_\_

Company \_\_\_\_\_ Publication Number \_\_\_\_\_

Address \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

FOLD

FOLD

Check the appropriate item.

Error (Page No. \_\_\_\_, Drawing No. \_\_\_\_\_)

Addition (Page No. \_\_\_\_, Drawing No. \_\_\_\_\_)

Other (Page No. \_\_\_\_, Drawing No. \_\_\_\_\_)

Explanation:

FOLD

FOLD

CUT ALONG LINE

Fold and Staple  
No postage necessary if mailed in U. S. A.

STAPLE

STAPLE

FOLD

FOLD

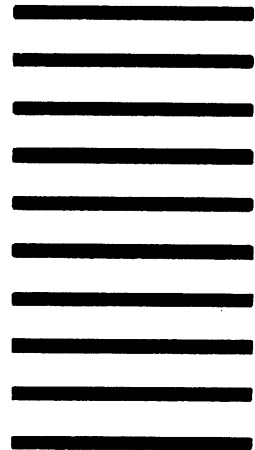
**BUSINESS REPLY MAIL**  
NO POSTAGE NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY:

  
**INTERDATA®**

2 Crescent Place, Oceanport, New Jersey 07757

FIRST CLASS  
PERMIT No. 22  
OCEANPORT, N.J.



TECH PUBLICATIONS DEPT. MS15

FOLD

FOLD

STAPLE

STAPLE

\* COPYRIGHT INTERDATA, INC. MAY, 1973  
 \* AUTHOR: J. PRATT  
 \*

0000R		ENTRY	OSAIDS		AID00030
0001	LINK	EQU	1		AID00040
0002	LINKS	EQU	2		AID00050
0003	BIAS	EQU	3		AID00060
0004	OPEN	EQU	4		AID00070
0005	VALUE	EQU	5		AID00080
0006	CMDER	EQU	6		AID00090
0007	LENGTH	EQU	7		AID00100
0008	MODE	EQU	8		AID00110
0009	OUTPUT	EQU	9		AID00120
000A	INPUT	EQU	10		AID00130
000D	NUMBER	EQU	13		AID00140
000E	FROM	EQU	14		AID00150
000F	TO	EQU	15		AID00160
0000	R0	EQU	0		AID00170
0001	R1	EQU	1		AID00180
0002	R2	EQU	2		AID00190
0003	R3	EQU	3		AID00200
0004	R4	EQU	4		AID00210
0005	R5	EQU	5		AID00220
0006	R6	EQU	6		AID00230
0007	R7	EQU	7		AID00240
0008	R8	EQU	8		AID00250
0009	R9	EQU	9		AID00260
000A	R10	EQU	10		AID00270
000B	R11	EQU	11		AID00280
000C	R12	EQU	12		AID00290
000D	R13	EQU	13		AID00300
000E	R14	EQU	14		AID00310
000F	R15	EQU	15		AID00320

-----\*  
 \* SET UP VALUES AND OUTPUT NAME

0000R	4200	OSAIDS	NOP	OS	ONE TIME SWITCH	AID00360
	001CR					AID00370
0004R	C800		LHI	R0,256	OP CODE TOGGLE	AID00380
	0100					
0008R	6100		AHM	R0,OSAIDS	THROW SWITCH	AID00390
	0000R					AID00400
000CR	D1E0		LM	FROM,X'B6'	SVC POINTERS	AID00410
	00B6					
0010R	D0E0		STM	FROM,PSWSAV	SAVE PSWS	AID00420
	0A4AR					
0014R	C800		LHI	R0,SVC14	SVC 14 HANDLER ADDR	AID00430
	1150R					
0018R	4000		STH	R0,X'B8'	STORE IN PSW FOR SVC	AID00440
	00B8					
001CR	C800	OS	LHI	R0,OS1	SET SVC RETURN ADDRESS	AID00450
	0026R					
0020R	4000		STH	R0,X'B6'		AID00460
	00B6					
0024R	E1DF		DC	X'E1DF'	SVC 13	AID00470
0026R	4800	OS1	LH	R0,X'96'	GET OLD STATUS / CC	AID00480
	0096					

002AR	C400		NHI	R0,X'7FFF'	RESET WAIT BIT	AID00490
	7FFF					
002ER	4000		STH	R0,CCSAVE	RETAIN OLD CC / STATUS	AID00500
	0FC0R					
0032R	C400		NHI	R0,X'7E00'	REMOVE PROTECTION	AID00510
	7E00					
0036R	4000		STH	R0,OS2	PRIME PSW LOCATION	AID00520
	0042R					
003AR	4000		STH	R0,EXIT	SET PSW FOR INTERPRETER EXIT	AID00530
	0FB2R					
003ER	C200		LPSW	OS2		AID00540
	0042R					
0042R	0000	OS2	DC	0,LDREG		AID00550
	0046R					
0046R	D100	LDREG	LM	R0,AIDSAV	LOAD UTILITY REGISTERS	AID00560
	0AE0R					
004AR	E120	NAME	SVC	2,TITLE	OUTPUT NAME	AID00570
	0B20R					
*-----*						
					GET A PARAMETER / DIRECTIVE AND PROCESS	AID00580
004ER	4110	IN	BAL	LINK,RDIN	READ INPUT	AID00590
	07CAR					AID00600
0052R	D100		LM	R0,AIDSAV	RESTORE REGISTERS	AID00610
	0AE0R					
0056R	D300		LB	R0,BUFFER	CHECK CMD	AID00620
	0A4ER					
005AR	C8C0		LHI	R12,CMCREN-CMDCHR	FIRST CMD CHARACTER TABLE	AID00630
	0011					
005ER	D40C	CHK1	CLB	R0,CMDCHR(R12)	CMD CHARACTER IN TABLE ?	AID00640
	0A06R					
0062R	2334		BES	CHK2	YES	AID00650
	0064R		SIS	R12,1	NO, CHECK ANOTHER ENTRY	AID00660
0066R	2214		BNMS	CHK1		AID00670
	0068R		BR	CMDEP	LAST ENTRY = ERROR	AID00680
006AR	0ACC	CHK2	AHR	R12,R12	CREATE INDEX FOR VECTOR TABLE	AID00690
	006CR		LH	LINK,CMDTBL(R12)	LOAD VECTOR ADDRESS	AID00700
	0A18R					
0070R	C310		THI	LINK,1	BREAK DOWN OPERANDS ?	AID00710
	0001					
0074R	0331		BZR	LINK	NO - BRANCH TO HANDLER ROUTINE	AID00720
	0076R	CHK2A	LHI	R11,CHK3	SET EXIT	AID00730
	007CR					
007AR	07CC		XHR	R12,R12	RESET INDEX	AID00740
	007CR	CHK3	AIS	R12,1		AID00750
007ER	D30C		LB	R0,BUFFER-1(R12)	GET A CMD CHAR	AID00760
	0A4DR					
0082R	C5C0		CLHI	R12,BUFEND-BUFFER	BLANK OPERAND ?	AID00770
	0028					
0086R	0331		BER	LINK	YES - GO TO HANDLER	AID00780
	0088R		CLHI	R0,X'000D'	CRLF ?	AID00790
	000D					
008CR	0331		BER	LINK	YES - GO TO HANDLER	AID00800
	008ER		CLHI	R0,X'20'	SPACE ?	AID00810
	0020					
0092R	023B		BNER	R11	NO	AID00820
	0094R	CHK3A	LHI	R11,PARSER	YES - SET TO FIND VALID CHARS	AID00830

009AR	0098R	220E	BS	CHK3	CONTINUE SCAN	AID00840	
	009AR	07FF	PARSER	XHR	TO,TO	OPERAND INDEX	AID00850
	009CR	0777		XHR	LENGTH,LENGTH	RESET OPERAND LENGTH	AID00860
	009ER	D30C	CHK4	LB	R0,BUFFER-1(R12)	GET A CHARACTER	AID00870
		0A4DR					
	00A2R	26C1		AIS	R12,1		AID00880
	00A4R	C5C0		CLHI	R12,BUFEND-BUFFER-1	CHECK FOR END OF BUFFER	AID00890
		0027					
	00A8R	4380		BNL	CHK6	INSERT FINAL LENGTH ATTRIBUTE	AID00900
		00C6R					
	00ACR	C8B0		LHI	R11,DELEND-DELIM	SET DELIMITER TABLE EXTENTS	AID00910
		0008					
	00B0R	D40B	CHK5	CLB	R0,DELIM(R11)	COMPARE TO DELIMITERS	AID00920
		0A3AR					
	00B4R	2339		BES	CHK6	IF EQUAL - CHECK FOR END	AID00930
	00B6R	27B1		SIS	R11,1		AID00940
	00B8R	2214		BNMS	CHK5	SCAN ENTIRE TABLE	AID00950
	00BAR	26F1		AIS	TO,1	BUMP OPERAND POINTER	AID00960
	00BCR	D20F		STB	R0,OPRAND(TO)	SAVE CHARACTER	AID00970
		0A78R					
	00C0R	2671		AIS	LENGTH,1	INCREMENT LENGTH	AID00980
	00C2R	4300		B	CHK4	CONTINUE	AID00990
		009ER					
	00C6R	08EF	CHK6	LHR	FROM,TO	GET CURRENT POINTER	AID01000
	00C8R	0877		LHR	LENGTH,LENGTH	CHECK FOR CLEAR	AID01010
	00CAR	2335		BZS	CHK7		AID01020
	00CCR	0BE7		SHR	FROM,LENGTH	SUBTRACG OPERAND LENGTH	AID01030
	00CER	D27E		STB	LENGTH,OPRAND(FROM)	STORE LENGTH ATTRIBUTE	AID01040
		0A78R					
	00D2R	26F1		AIS	TO,1	BUMP OPERAND INDEX	AID01050
	00D4R	C5C0	CHK7	CLHI	R12,BUFEND-BUFFER-1	CHECK FOR END	AID01060
		0027					
	00D8R	0381		BNLR	LINK	BRANCH TO HANDLER	AID01070
	00DAR	4300		B	CHK4-2	CONTINUE	AID01080
		009CR					
*-----*							AID01090
			*	LOGICAL UNIT PRIMER			AID01100
	00DER	D3D0	LU	LB	NUMBER,OPRAND	OPERAND LENGTH	AID01110
		0A78R					
	00E2R	08DD		LHR	NUMBER,NUMBER	IF ZERO - ERROR	AID01120
	00E4R	0336		BZR	CMDER		AID01130
	00E6R	C8E0		LHI	FROM,OPRAND+1	SET SENDER FIELD	AID01140
		0A79R					
	00EAR	C8F0		LHI	TO,CMDINP	SET RECEIVER ADDRESS	AID01150
		080AR					
	00EER	0700		XHR	R0,R0	CLEAR REG	AID01160
	00FOR	D20F		STB	R0,1(TO)	CLEAR LU	AID01170
		0001					
	00F4R	4120		BAL	LINKS,XLATHB	TRANSLATE TO BINARY	AID01180
		044CR					
	00F8R	0AED		AHR	FROM,NUMBER	BUMP OPERAND POINTER	AID01190
	00FAR	D3DE		LB	NUMBER,0(FROM)	OPERAND LENGTH	AID01200
		0000					
	00FER	08DD		LHR	NUMBER,NUMBER	IF ZERO - ERROR	AID01210
	0100R	0336		BZR	CMDER		AID01220

0102R	26E1	AIS	FROM,1	BUMP TO SENDER FIELD	AID001230
0104R	C8F0	LHI	TO,LSTBLK	SET RECEIVER ADDRESS	AID001240
	090CR				
0108R	D20F	STB	RO,1(TO)	CLEAR LU	AID001250
	0001				
010CR	4120	BAL	LINKS,XLATHB	TRANSLATE TO BINARY	AID001260
	044CR				
0110R	0AED	AHR	FROM,NUMBER	BUMP OPERAND POINTER	AID001270
0112R	D3DE	LB	NUMBER,0(FROM)	OPERAND LENGTH	AID001280
	0000				
0116R	08DD	LHR	NUMBER,NUMBER	IF ZERO - ERROR	AID001290
0118R	0336	BZR	CMDER		AID001300
011AR	C8F0	LHI	TO,BINBLK	SET RECEIVER ADDRESS	AID001310
	08BCR				
011ER	D20F	STB	RO,1(TO)	CLEAR LU	AID001320
	0001				
0122R	26E1	AIS	FROM,1	BUMP TO SENDER FIELD	AID001330
0124R	4120	BAL	LINKS,XLATHB	TRANSLATE TO BINARY	AID001340
	044CR				
0128R	4300	B	NAME	OUTPUT NAME AND GET ANOTHER DIRECTIVE	AID001350
	004AR				
*-----*					AID001360
		* BIAS (BI) HANDLER			AID001370
012CR	D3B0	BYAS LB	R11,BUFFER+1	SECOND CHARACTER	AID001380
	0A4FR				
0130R	C5B0	CLHI	R11,C'I'	CHECK FOR 'I'	AID001390
	0049				
0134R	0236	BNER	CMDER	NO - ERROR	AID001400
0136R	D3D0	LB	NUMBER,OPRAND	OPERAND LENGTH	AID001410
	0A78R				
013AR	08DD	LHR	NUMBER,NUMBER	CHECK FOR VALUE	AID001420
013CR	0336	BZR	CMDER	BLANK - ERROR	AID001430
013ER	C8E0	LHI	FROM,OPRAND+1	SENDER ADDRESS	AID001440
	0A79R				
0142R	40D0	STH	NUMBER,OPSAVE	CLEAR WORK AREA	AID001450
	0A96R				
0146R	C8F0	LHI	TO,OPSAVE	RECEIVER ADDRESS	AID001460
	0A96R				
014AR	4120	BAL	LINKS,XLATHB	TRANSLATE HEX ASCII TO BINARY	AID001470
	044CR				
014ER	4830	LH	BIAS,OPSAVE	SET BIAS REGISTER	AID001480
	0A96R				
0152R	4300	B	NAME	ISSUE NAME & READ DIRECTIVE	AID001490
	004AR				
*-----*					AID001500
		* END (EN) HANDLER			AID001510
0156R	D3B0	TERM LB	R11,BUFFER+1	SECOND CHARACTER	AID001520
	0A4FR				
015AR	C5B0	CLHI	R11,C'N'	CHARACTER 'N' ?	AID001530
	004E				
015ER	0236	BNER	CMDER	NO - ERROR	AID001540
0160R	D1E0	LM	FROM,PSWSAV	RESTORE PSWS	AID001550
	0A4AR				
0164R	D0E0	STM	FROM,X'B6'		AID001560
	00B6				
0168R	C800	LHI	RO,-256	OP CODE UN TOGGLE	AID001570

016CR	FF00 6100	AHM	R0,OSAIDS	RESET SWITCH	AID01580
0170R	0000F 07DD	XHR	NUMBER,NUMBER	SET ALL INDICATOR	AID01590
0172R	4120	BAL	LINKS,ZAPBRK	ZAP BREAKPOINTS	AID01600
0176R	1228R 4120	BAL	LINKS,ZAPSNP	ZAP SNAPSHOT POINTS	AID01610
017AR	145CR 4120	BAL	LINKS,ZAPPRO	ZAP PROTECTION	AID01620
017ER	1506R 4120	BAL	LINKS,ZAPTRC	ZAP TRACE POINTS	AID01630
0182R	1352R E130 0000	SVC	3,0	EOJ	AID01640
*-----*					
* OPEN (0) HANDLER					
0186R	D3D0 0A78R	OPIN	LB	NUMBER,OPRAND	GET OPERAND LENGTH
018AR	C8E0 0A79R	LHI	FROM,OPRAND+1	ADDR OF OPERAND	AID01680
018ER	08DD	LHR	NUMBER,NUMBER	IF ZERO - RESHOW CELL / REG	AID01690
0190R	4330 01C6R	BZ	OP1	YES	AID01700
0194R	40D0 0A96R	STH	NUMBER,OPSAVE	CLEAR WORK AREA	AID01710
0198R	2472	LIS	LENGTH,2	SET CELL SIZE	AID01720
019AR	C8F0 0A96R	LHI	TO,OPSAVE	ADDR FOR RESULTS	AID01730
019ER	4120 044CR	BAL	LINKS,XLATHB	TRANSLATE TO BINARY	AID01740
01A2R	4840 0A96R	LH	OPEN,OPSAVE	LOAD CURRENT CELL / REG	AID01750
01A6R	D3D0 0A78R	LB	NUMBER,OPRAND	OPERAND LENGTH	AID01760
01AAR	C8E0 0A79R	LHI	FROM,OPRAND+1(NUMBER)		AID01770
01AER	D3DE 0000	LB	NUMBER,0(FROM)	OPERAND LENGTH	AID01780
01B2R	C8F0 0220R	LHI	TO,OPCONT	COUNTER ADDRESS	AID01790
01B6R	26E1	AIS	FROM,1	BUMP SENDER ADDRESS	AID01800
01B8R	08DD	LHR	NUMBER,NUMBER	BYPASS ?	AID01810
01BAR	2336	BZS	OP1		AID01820
01BCR	4120 04B2R	BAL	LINKS,XLATDB	TRANSLATE TO HEX	AID01830
01COR	48D0 0220R	LH	NUMBER,OPCONT	GET VALUE	AID01840
01C4R	27D1	SIS	NUMBER,1	ZERO RELATIVE	AID01850
01C6R	40D0 0220R	OP1	STH	NUMBER,OPCONT	STORE IN OPEN COUNTER
01CAR	4040 0A96R	OP1A	STH	OPEN,OPSAVE	STORE OPEN CELL ADDR
01CER	D300 0A4FR	LB	R0,BUFFER+1	SECONDARY MODIFIER	AID01880
01D2R	C500 0052	CLHI	R0,C'R	REGISTER ?	AID01890

01D6R 2139		BNES	OP2	NO	AID01900
01D8R 0A44		AHR	OPEN,OPEN	INDEX = 2	AID01910
01DAR CA40		AHI	OPEN,REGSAV	INDEX REGISTER SAVE AREA	AID01920
0A98R					
01DER C540		CLHI	OPEN,REGSAV+32	REGISTER LIMIT CHECK	AID01930
0A88R					
01E2R 0386		BNLR	CMDER	LESS = OK	AID01940
01E4R 0843		SHR	OPEN,BIAS	SET SAVE AREA AS OPEN CELL	AID01950
01E6R 230B		BS	OP4		AID01960
01E8R C500	OP2	CLHI	R0,X'20'	SECOND OPTION = DEFAULT ?	AID01970
0020					
01ECR 2337		BES	OP3	YES	AID01980
01EER C500		CLHI	R0,0	CHECK FOR DEFAULT	AID01990
0000					
01F2R 2334		BES	OP3		AID02000
01F4R C500		CLHI	R0,X'0D'	CR = DEFAULT ALSO	AID02010
000D					
01F8R 2132		BNES	OP4		AID02020
01FAR 0808	OP3	LHR	R0,MODE	USE PREVIOUS MODE	AID02030
01FCR 08E3	OP4	LHR	FROM,BIAS	ADD BIAS - IF ANY	AID02040
01FER 0AE4		AHR	FROM,OPEN	SET CELL AS ADDR OF OPERAND	AID02050
0200R C4E0		NHI	FROM,X'FFFE'	EVEN BOUNDARY	AID02060
FFFE					
0204R 24D2		LIS	NUMBER,2	SET COUNT FOR TRANSLATION	AID02070
0206R C8F0		LHI	TO,OUTBUF+2	SET OUTPUT	AID02080
0844R					
020AR 4120		BAL	LINKS,TABA	LOOK UP OPERATION & PERFORM	AID02090
097AR					
020ER 0129		BALR	LINKS,OUTPUT	PRINT RESULTS	AID02100
0210R 48D0		LH	NUMBER,OPCONT	CHECK COUNTER	AID02110
0220R					
0214R 033A		BZR	INPUT	SKIP MULTIPLE OPEN	AID02120
0216R 27D1		SIS	NUMBER,1	DECREMENT COUNTER	AID02130
0218R 40D0		STH	NUMBER,OPCONT		AID02140
0220R					
021CR 4300		B	NEXT	OPEN NEXT LOGICAL CELL	AID02150
0284R					
0220R 0000		OPCONT	DC	0	AID02160
					AID02170
*-----*					
* NEXT (N) HANDLER					
0222R 0A47	NEKST	AHR	OPEN,LENGTH	BUMP OPEN CELL	AID02180
0224R 030A		BR	INPUT	GET ANOTHER COMMAND	AID02190
*-----*					
* REPLACE (R) HANDLER					
0226R D380	RPLACE	LB	R11,BUFFER+1	GET SECOND CHARACTER MODIFIER	AID02200
0A4FR					AID02210
022AR C580		CLHI	R11,C'F'	CHECK FOR FLOATING POINT	AID02220
0046					
022ER 2338		BES	RP1	YES - DO NOT PARSE	AID02250
0230R C580		CLHI	R11,C'C'	CHARACTER FORMAT	AID02260
0043					
0234R 2335		BES	RP1	YES	AID02270
0236R C810		LHI	LINK,RP4	SET RETURN ADDRESS	AID02280
0264R					
023AR 4300		B	CHK2A	PROCESS INPUT LINE	AID02290
0076R					



023ER	C870	RP1	LHI	LENGTH,BUFEND-1	BUFFER END ADDRESS	AID02300
	0A75R					
0242R	07CC		XHR	R12,R12	BUFFER FILL CHARACTER	AID02310
0244R	D4C7	RP2	CLB	R12,0(LENGTH)	CHECK FOR LAST CHARACTER	AID02320
	0000					
0248R	2133		BNES	RP3	HIT VALID CHARACTER	AID02330
024AR	2771		SIS	LENGTH,1	DECREMENT POINTER	AID02340
024CR	2204		BS	RP2	CONTINUE SCAN	AID02350
024ER	24CD	RP3	LIS	R12,13	CARRIAGE RETURN	AID02360
0250R	45C7		CLH	R12,0(LENGTH)	CHECK LAST CHARACTER	AID02370
	0000					
0254R	2132		BNES	RP3A	NO - SKIP	AID02380
0256R	2771		SIS	LENGTH,1	DECREMENT LENGTH	AID02390
0258R	CB70	RP3A	SHI	LENGTH,BUFFER+3	GET TRUE LENGTH	AID02400
	0A51R					
025CR	08D7		LHR	NUMBER,LENGTH		AID02410
025ER	C8E0		LHI	FROM,BUFFER+3	START ADDRESS	AID02420
	0A51R					
0262R	230A		BS	RP5	PROCESS CHARACTERS	AID02430
0264R	D3D0	RP4	LB	NUMBER,OPRAND	OPRAND SIZE	AID02440
	0A78R					
0268R	C8E0		LHI	FROM,OPRAND+1	SET SENDER FIELD	AID02450
	0A79R					
026CR	2472		LIS	LENGTH,2	SET OPERATION LENGTH	AID02460
026ER	0854		LHR	VALUE,OPEN	GET OPEN CELL ADDRESS	AID02470
0270R	0A53		AHR	VALUE,BIAS	ADD BIAS	AID02480
0272R	40D5		STH	NUMBER,0(VALUE)	CLEAR CONTENTS	AID02490
	0000					
0276R	08F4	RP5	LHR	TO,OPEN	SET RECEIVER FIELD	AID02500
0278R	0AF3		AHR	TO,BIAS	CALCULATE ABSOLUTE ADDRESS	AID02510
027AR	D300		LB	R0,BUFFER+1	SET SECOND CHARACTER	AID02520
	0A4FR					
027ER	4120		BAL	LINKS,TABB	TRANSLATE & PROCESS	AID02530
	0974R					
0282R	030A		BR	INPUT	READ INPUT	AID02540
						AID02550
						AID02560
0284R	0A47	NEXT	AHR	OPEN,LENGTH	BUMP OPEN CELL/REGISTER	AID02570
0286R	4300		B	OP1A	PERFORM OPERATION	AID02580
	01CAR					
						AID02590
						AID02600
028AR	2742	PRECED	SIS	OPEN,2	DECREMENT OPEN CELL BY A HALFWORD	AID02610
028CR	4300		B	OP1A	PERFORM OPERATION	AID02620
	01CAR					
						AID02630
						AID02640
0290R	4844	JUMP	LH	OPEN,0(OPEN)	GET CONTENTS OF OPEN CELL/REG	AID02650
	0000					
0294R	D300		LB	R0,BUFFER+1	GET SECONDARY DIRECTIVE	AID02660
	0A4FR					
0298R	C500		CLHI	R0,X'0052'	REGISTER ?	AID02670
	0052					
029CR	2332		BES	J1	YES - NO BIAS	AID02680
029ER	0B43		SHR	OPEN,BIAS	UNBIAS ADDRESS	AID02690
02A0R	4300	J1	B	OP1A	PERFORM OPERATION	AID02700

01CAR		*-----*			AID02710
		* ARITHMETIC HANDLER			AID02720
02A4R	D3D0 0A78R	ARITH	LB	NUMBER,OPRAND	GET OPERAND SIZE AID02730
02A8R	08DD		LHR	NUMBER,NUMBER	CHECK FOR MISSING OPERAND AID02740
02AAR	0336		BZR	CMDBR	YES - ERROR AID02750
02ACR	C8E0 0A79R		LHI	FROM,OPRAND+1	GET SENDER ADDR AID02760
02B0R	C8F0 0AB8R		LHI	TO,DBLWK	GET RECEIVER ADDR AID02770
02B4R	40DF 0000		STH	NUMBER,0(TO)	CLEAR RECEIVER AID02780
02B8R	D300 0A4FR		LB	R0,BUFFER+1	SET SECOND CHARACTER AID02790
02BCR	0854		LHR	VALUE,OPEN	AID02800
02BER	0A53		AHR	VALUE,BIAS	GET OPEN CELL/REG ADDRESS AID02810
02COR	C8C0 0053		LHI	R12,C'S'	S = SUBTRACT AID02820
02C4R	C500 0046		CLHI	R0,X'46'	FLOATING POINT ? AID02830
02C8R	4330 02F2R		HE	FLOT	FLOATING POINT OPERATION AID02840
02CCR	4120 0974R		BAL	LINKS,TABB	TRANSLATE OPERAND AID02850
02D0R	C8C0 0053		LHI	R12,C'S'	SUBTRACT CONSTANT AID02860
02D4R	D4C0 0A4ER		CLB	R12,BUFFER	ADD OR SUBTRACT ? AID02870
02D8R	2138		BNES	ADD	ADD AID02880
02DAR	48C5 0000		LH	R12,0(VALUE)	LOAD CONTENTS AID02890
02DER	4BC0 0AB8R		SH	R12,DBLWK	SUBTRACT VALUE AID02900
02E2R	40C5 0000		STH	R12,0(VALUE)	RESTORE REMAINDER AID02910
02E6R	030A		BR	INPUT	AID02920
02E8R	4800 0AB8R	ADD	LH	R0,DBLWK	NO - LOAD VALUE AID02930
02ECR	6105 0000		AHM	R0,0(VALUE)	ADD VALUE AID02940
02FOR	030A		BR	INPUT	AID02950
02F2R	6000 0ABCR	FLOT	STE	R0,DBLSAV	SAVE FLT.PT. REG 0 AID02960
02F6R	6805 0000		LE	R0,0(VALUE)	LOAD INTO REG ZERO AID02970
02FAR	C8F0 0AB8R		LHI	TO,DBLWK	SET RECEIVER ADDRESS AID02980
02FER	4120 0672R		BAL	LINKS,XLATEL	TRANSLATE TO FLOATING POINT AID02990
0302R	D4C0 0A4ER		CLB	R12,BUFFER	ADD OR SUBTRACT ? AID03000
0306R	2138		BNES	ADFLOT	ADD AID03010
0308R	6800 0AB8R	SBFLOT	SE	R0,DBLWK	SUBTRACT VALUE AID03020
030CR	0855		LHR	VALUE,VALUE	CHECK ADDRESS AID03030

030ER	033A		HZR	INPUT	NO RESTORATION REQUIRED	AID03040
0310R	6005		STE	R0,0(VALUE)	STORE BACK IN MEMORY	AID03050
	0000					
0314R	030A		BR	INPUT	RETURN	AID03060
0316R	6A00	ADFLOT	AE	R0,DBLWK	ADD VALUE	AID03070
	0A88R					
031AR	0855		LHR	VALUE,VALUE	CHECK ADDRESS	AID03080
031CR	033A		BZR	INPUT	NO RESTORATION	AID03090
031ER	6005		STE	R0,0(VALUE)	STORE BACK IN MEMORY	AID03100
	0000					
0322R	030A		BR	INPUT	RETURN	AID03110
						AID03120
						AID03130
						AID03140
0324R	C800		GO	GO (GO) HANDLER	CHARACTER '0'	
	004F		LHI	R0,C'0'		
0328R	D400		CLB	R0,BUFFER+1	CHECK SECOND DIRECTIVE	AID03150
	0A4FR					
032CR	0236		BNER	CMDER	NOT EQUAL = ERROR	AID03160
032ER	D3D0		LB	NUMBER,OPRAND	OPERAND LENGTH	AID03170
	0A78R					
0332R	08DD		LHR	NUMBER,NUMBER	DEFAULT ?	AID03180
0334R	2136		BNZS	G1	NO	AID03190
0336R	0804		LHR	R0,OPEN	YES - USE OPEN CELL	AID03200
0338R	0A03		AHR	R0,BIAS		AID03210
033AR	4000		STH	R0,G3	SET RETURN ADDRESS	AID03220
	040AR					
033ER	2309		BS	G1B	CHECK PROTECTION	AID03230
0340R	C8E0	G1	LHI	FROM,OPRAND+1	SET FROM ADDR	AID03240
	0A79R					
0344R	C8F0		LHI	TO,G3	SET TO ADDR	AID03250
	040AR					
0348R	40D0		STH	NUMBER,G3	CLEAR WORK AREA	AID03260
	040AR					
034CR	4120		BAL	LINKS,XLATHB	TRANSLATE ADDRESS	AID03270
	044CR					
0350R	4800	G1B	LH	R0,TRCFLG	TRACE MODE STE ?	AID03280
	0A94R					
0354R	4230		BNZ	SVCE03	YES	AID03281
	1308R					
0358R	4800		LH	R0,PROFLG	PROTECT MODE SET ?	AID03282
	0A48R					
035CR	4330		BZ	G10	NO PROTECTION - EXIT TO USER	AID03290
	03F8R					
0360R	48C0		LH	R12,G3	GET EX ADDR	AID03300
	040AR					
0364R	4840	G1A	LH	OPEN,G3	SET OPEN CELL	AID03310
	040AR					
0368R	4040		STH	OPEN,G12	SAVE OLD OPEN CELL	AID03320
	040CR					
036CR	4120		BAL	LINKS,EX	YES - EXECUTE AN INSTRUCTION	AID03330
	0DF6R					
0370R	4040	PROCHK	STH	OPEN,G3	SET ADDRESS	AID03340
	040AR					
0374R	240F		LIS	R0,15	BIT COUNTER	AID03350
0376R	48B0	G1C	LH	R11,PROREG-2	POSITIONAL REG COUNTER	AID03360
	110AR					

037AR 4330		BZ	G7	SKIP NO REGS PROTECTED	AID03370
03AAR 03AAR					
037ER 07CC		XHR	R12,R12	ZERO-IZE REG 12	AID03380
0380R 90B1	G5	SRLS	R11,1	CHECK A BIT (REG)	AID03390
0382R 2388		BNCS	G6		AID03400
0384R 08EC	G5B	LHR	FROM,R12	SAVE BIT NUMBER	AID03410
0386R 0AEE		AHR	FROM,FROM	INDEX OF REG	AID03420
0388R 48FE		LH	TO,SAVREG(FROM)	PROTECTED REG CONTENTS	AID03430
0A98R					
038CR 45FE		CLH	TO,PROREG(FROM)	COMPARE TO PROTECTED REG	AID03440
110CR					
0390R 2139		BNES	G6A	REG VALUE CHANGED	AID03450
0392R 26C1	G6	AIS	R12,1	INCREMENT INDEX	AID03460
0394R 2701		SIS	R0,1	DECREMENT COUNTER	AID03470
0396R 221B		BNMS	G5	CONTINUE CHECKING	AID03480
0398R 48B0		LH	R11,PROR0	CHECK REG ZERO	AID03490
0A46R					
039CR 2337		BZS	G7	SKIP	AID03500
039ER 07CC		XHR	R12,R12	CLEAR REG NO.	AID03510
03A0R 220E		BS	G5B		AID03520
03A2R 40FE	G6A	STH	TO,PROREG(FROM)	SWITCH REG VALUES	AID03521
110CR					
03A6R 4300		B	G9	ISSUE MESSAGE	AID03522
03D2R					
03AAR C8B0	G7	LHI	R11,PROCLN-PROCEL	LIST LENGTH FOR INDEXING	AID03530
0020					
03AER 48EB		LH	FROM,PROCEL-4(R11)	PICK UP ADDRESS	AID03540
112AR					
03B2R C5E0		CLHI	FROM,-1	DELETED ?	AID03550
FFFF					
03B6R 2336		BES	G8	YES IGNORE	AID03560
03B8R 480E		LH	R0,0(FROM)	GET CELL CONTENTS	AID03570
0000					
03BCR 450B		CLH	R0,PROCEL-2(R11)	COMPARE TO PROTECTED CELL	AID03580
112CR					
03C0R 2135		BNES	G6B	NOT EQUAL = PROTECTION VIOLATION	AID03590
03C2R 27B4	G8	SIS	R11,4	DECREMENT INDEX	AID03600
03C4R 203B		BNZS	G7+4	CONTINUE CHECKING	AID03610
03C6R 4300		B	G1B	CONTINUE EXECUTING	AID03620
0350R					
03CAR 400B	G6B	STH	R0,PROCEL-2(R11)	SWITCH CONTENTS	AID03621
112CR					
03CER 48CB	G8A	LH	R12,PROCEL-4(R11)	VIOLATED CELL	AID03630
112AR					
03D2R D1E0	G9	LM	FROM,PROMSG		AID03640
0A90R					
03D6R D0E0		STM	FROM,OUTBUF+2	SET UP PROTECT MESSAGE	AID03650
0844R					
03DAR 24D2		LIS	NUMBER,2	SET BYTE COUNT	AID03660
03DCR C8E0		LHI	FROM,G12	ADDRESS OF SENDER	AID03670
040CR					
03E0R C8F0		LHI	TO,OUTBUF+8		AID03680
084AR					
03E4R 4120		BAL	LINKS,XLATEH	TRANSLATE ADDRESS OF PROT VIOLATOR	AID03690
040ER					
03E8R 40C0		STH	R12,G12	SAVE VIOLATED CELL/REG	AID03700

040CR							
03ECR	26F6		AIS	TO,6		SET RECEIVER ADDRESS	AID03710
03EER	4120		BAL	LINKS,XLATEH		TRANSLATE TO ASCII	AID03720
	040ER						
03F2R	0129		BALR	LINKS,OUTPUT			AID03730
03F4R	4300		B	NAME			AID03740
	004AR						
03F8R	4800	G10	LH	R0,CCSAVE		CC/STATUS	AID03770
	0FCOR						
03FCR	4000		STH	R0,G11		SET CONDITION CODE / STATUS	AID03780
	0408R						
0400R	D100	G4	LM	R0,SAVREG		RESTORE USER REGS	AID03790
	0A98R						
0404R	C200		LPSW	G11			AID03800
	0408R						
0408R	0000	G11	DC	0			AID03810
040AR	0000	G3	DC	0			AID03820
040CR	0000	G12	DC	0			AID03830
							AID03840
*-----*							
* TRANSLATE - BINARY TO ASCII HEX							
040ER	D090		XLATEH	STM	R9,SAVE		AID03850
	0800R						AID03860
0412R	C890		LHI	R9,X'3A'			AID03870
	003A						
0416R	D3AE	XH1	LB	R10,0(FROM)		GET A BYTE	AID03880
	0000						
041AR	24BF		LIS	R11,15		MASK	AID03890
041CR	04BA		NHR	R11,R10		SPLIT BYTE	AID03895
041ER	90A4		SRLS	R10,4			AID03900
0420R	CAA0		AHI	R10,X'30'		ASCII-IZE	AID03910
	0030						
0424R	CAB0		AHI	R11,X'30'			AID03920
	0030						
0428R	05A9		CLHR	R10,R9		GREATER THAN 9 ?	AID03930
042AR	2182		BLS	XH2		NO	AID03940
042CR	26A7		AIS	R10,7		YES - ADJUST ASCII VALUE	AID03950
042ER	05B9	XH2	CLHR	R11,R9		SECOND CHAR GREATER THAN 9	AID03960
0430R	2182		BLS	XH3		NO	AID03970
0432R	26B7		AIS	R11,7		YES - ADJUST ASCII VALUE	AID03980
0434R	D2AF	XH3	STB	R10,0(TO)		STORE CHARACTERS	AID03990
	0000						
0438R	D2BF		STB	R11,1(TO)			AID04000
	0001						
043CR	26F2		AIS	TO,2		BUMP INDEX	AID04010
043ER	26E1		AIS	FROM,1			AID04020
0440R	27D1		SIS	NUMBER,1		DECREMENT COUNT	AID04030
0442R	4230		BNZ	XH1		CONTINUE	AID04040
	0416R						
0446R	D190		LM	R9,SAVE		RETURN	AID04050
	0800R						
044AR	0302		BR	LINKS			AID04060
							AID04070
*-----*							
* TRANSLATE - HEX ASCII TO INTERNAL							
044CR	D0A0		XLATHB	STM	R10,SAVE	SAVE REGISTERS	AID04080
	0800R						AID04090
0450R	0AED		AHR	FROM,NUMBER		POINT TO LAST CHARACTER	AID04100

0452R	C5D0		CLHI	NUMBER,5	RANG CHECK	AID04110
	0005					
0456R	0386		BNLR	CMDER	ERROR	AID04120
0458R	24A2	XHB1	LIS	R10,2	SWITCH	AID04130
045AR	D38E	XHB2	LB	R11,-1(FROM)	GET ASCII CHARACTER	AID04140
	FFFF					
045ER	C5B0	XHB2A	CLHI	R11,X'3A'	*	AID04150
	003A					
0462R	2182		BLS	XHB3	* INTERNALIZE LSD	AID04160
0464R	26B9		AIS	R11,9	*	AID04170
0466R	27D1	XHB3	SIS	NUMBER,1	DECREMENT COUNT	AID04180
0468R	4210		BM	XHB5	DONE	AID04190
	04ACR					
046CR	C8B0		SHI	R11,C'0'	LIMIT CHECK	AID04200
	0030					
0470R	0216		BMR	CMDER		AID04210
0472R	C5B0		CLHI	R11,X'20'	HIGH LIMIT	AID04220
	0020					
0476R	0386		BNLR	CMDER		AID04230
0478R	27A1		SIS	R10,1	DECREMENT SWITCH	AID04240
047AR	213D		BNZS	XHB4		AID04250
047CR	91B4		SLLS	R11,4	FORM MSD	AID04260
047ER	D3CF		LB	R12,1(TO)	GET BYTE	AID04270
	0001					
0482R	C4C0		NHI	R12,X'0F'	CLEAR HALFBYTE	AID04280
	000F					
0486R	06CB		OHR	R12,R11	COMBINE VALUES	AID04290
0488R	D2CF		STB	R12,1(TO)	SAVE BYTE	AID04300
	0001					
048CR	27F1		SIS	TO,1	DECREMENT RECEIVER	AID04310
048ER	27E1		SIS	FROM,1	DECREMENT SENDER	AID04320
0490R	4300		B	XHB1	CONTINUE	AID04330
	0458R					
0494R	C4B0	XHB4	NHI	R11,X'0F'	FORM LSD	AID04340
	000F					
0498R	D3CF		LB	R12,1(TO)		AID04350
	0001					
049CR	C4C0		NHI	R12,X'F0'		AID04360
	00F0					
04A0R	06BC		OHR	R11,R12		AID04370
04A2R	D2BF		STB	R11,1(TO)	STORE IN RECEIVER FIELD	AID04380
	0001					
04A6R	27E1		SIS	FROM,1		AID04390
04A8R	4300		B	XHB2		AID04400
	045AR					
04ACR	D1A0	XHB5	LM	R10,SAVE	RESTORE REGS	AID04410
	0B00R					
04B0R	0302		BR	LINKS	RETURN	AID04420
						AID04430
						AID04440
						AID04450
						AID04460
						AID04470
						AID04480
04B2R	D0A0	XLATDB	STM	R10,SAVE	SAVE REGS	AID04490
	0H00R					
04B6R	C5D0		CLHI	NUMBER,6	5 DIGITS = MAXIMUM	AID04460
	0006					
04BAR	2389		BNLS	XB2		AID04470
04BCR	07BB		XHR	R11,R11	CLEAR MULTIPLICAND	AID04480

04BER 24CA		LIS	R12,10	MULTIPLIER		AID04490
04COR D30E	XB1	LB	R0,0(FROM)	GET A BYTE		AID04500
0000						
04C4R CB00		SHI	R0,X'30'	INTERNALIZE		AID04510
0030						
04C8R 0216		BMR	CMDER			AID04520
04CAR 090C		CHR	R0,R12	RANGE CHECK		AID04530
04CCR 0386	XB2	BNLR	CMDER			AID04540
04CER 0CAC		MHR	R10,R12	DECIMALIZE		AID04550
04D0R 0AB0		AHR	R11,R0			AID04560
04D2R 26E1		AIS	FROM,1	BUMP SENDER FIELD		AID04570
04D4R 27D1		SIS	NUMBER,1	DECREMENT COUNTER		AID04580
04D6R 203B		BNZS	XB1	NOT DONE		AID04590
04D8R 40BF		STH	R11,0(TO)			AID04600
0000						
04DCR D1A0		LM	R10,SAVE	RESTORE REGS		AID04610
0B00R						
04E0R 0302		BR	LINKS	RETURN		AID04620
*-----*						
* TRANSLATE - INTERNAL TO DECIMAL ASCII						
04E2R D0A0	XLATED	STM	R10,SAVE	SAVE REGS		AID04630
0R00R						AID04640
04E6R 24C8		LIS	R12,8	DIVISOR INDEX		AID04660
04E8R 48BE		LH	R11,0(FROM)	DIVIDEND		AID04670
0000						
04ECR 07AA	XD1	XHR	R10,R10			AID04680
04EER 48DC		LH	R13,XD3(R12)	GET POWER OF TEN		AID04690
050ER						
04F2R 0DAD		DHR	R10,R13	DIVIDE		AID04700
04F4R C7B0	XD2	XHI	R11,X'30'	ASCII-IZE QUOTIENT		AID04710
0030						
04F8R 02BF		STB	R11,0(TO)	STORE		AID04720
0000						
04FCR 26F1		AIS	TO,1	BUMP STORAGE INDEX		AID04730
04FER 27C2		SIS	R12,2	DECREMENT INDEX		AID04740
0500R 08BA		LHR	R11,R10	SET REMAINDER AS QUOTIENT		AID04750
0502R 203B		BNZS	XD1	CONTINUE		AID04760
0504R 08CC		LHR	R12,R12	FIVE PLACES ?		AID04770
0506R 2219		HNMS	XD2	PAD ZEROES		AID04780
0508R D1A0		LM	R10,SAVE			AID04790
0B00R						
050CR 0302		BR	LINKS	RETURN		AID04800
050ER 0001	XD3	DC	1,10,100,1000,10000			AID04810
000A						
0064						
03E8						
2710						
*-----*						
* TRANSLATE - INTERNAL TO FLOATING DECIMAL						
0518R D000	XLATEF	STM	R0,SAVE2	SAVE REGS		AID04820
0AC0R						AID04830
051CR D1C0		LM	R12,0	SAVE FLOATING POINT REGS		AID04850
0000						
0520R D0C0		STM	R12,XFSAVE			AID04860
065ER						
0524R D100		LM	R0,SAVE2			AID04870

0528R	0AC0R 40E0		STH	FROM,DECD2	SET ADDRESS	AID04880
052CR	0A44R C8E0		LHI	FROM,DECD2		AID04890
0530R	0A44R 2402		LIS	NUMBER,2	PRESET LENGTH	AID04900
0532R	4120		BAL	LINKS,XLATEH	TRANSLATE ADDRESS TO ASCII	AID04910
0536R	040ER 2404		LIS	NUMBER,4	SET BYTE COUNT	AID04920
0538R	48E0		LH	FROM,DECD2	RESTORE ADDRESS	AID04930
053CR	0A44R 26F6		AIS	T0,6	BUMP LINE POINTER	AID04940
053ER	4120		BAL	LINKS,XLATEH	TRANSLATE FULLWORD TO ASCII	AID04950
0542R	040ER 26FC		AIS	T0,12	BUMP LINE POINTER	AID04960
0544R	08BF		LHR	R11,T0		AID04970
0546R	D1EE		LM	FROM,0(FROM)		AID04980
054AR	0000 0733		XHR	R3,R3	EXPONENT FLAG	AID04990
054CR	080E		LHR	R0,FROM	COPY EXPONENT & HI MANTISSA	AID05000
054ER	C400		NHI	R0,X'7FF0'	CHECK FOR ZERO	AID05010
0552R	7FF0 4330		HZ	XZERO	IF ZERO - IT'S ZERO	AID05020
0556R	063CR 08EE		LHR	FROM,FROM	CHECK SIGN	AID05030
0558R	2128		BPS	XF1	POSITIVE MANTISSA	AID05040
055AR	C4E0		NHI	FROM,X'7FFF'	SET SIGN PLUS	AID05050
055ER	7FFF C800		LHI	R0,C'-'	MINUS SIGN	AID05060
0562R	002D D20B		STB	R0,0(R11)	PLACE IN PRINT LINE	AID05070
0566R	0000 26B1		AIS	R11,1	BUMP POINTER	AID05080
0568R	D0E0	XF1	STM	FROM,0	LOAD FLT PT REG 0	AID05090
056CR	0000 2800		LER	0,0	NORMALIZE	AID05100
056ER	6820		LE	2,D10	LOAD TEN ON FLT PT REG 2	AID05110
0572R	066ER 24A1		LIS	R10,1	SET EXP TO ONE	AID05120
0574R	4110		BAL	R1,XREDUZ	REDUCE NUMBER	AID05130
0578R	0654R 6900	XF2	CE	0,FC0N1		AID05140
057CR	066AR 2384		BNLS	XF3		AID05150
057ER	2C02		MER	0,2	MULT BY TEN	AID05160
0580R	27A1		SIS	R10,1	DECREMENT EXP COUNTER	AID05170
0582R	2205		BS	XF2	CHECK	AID05180
0584R	6A00	XF3	AE	0,D3C		AID05190
0588R	0666R 4110		BAL	R1,XREDUZ	REDUCE TO TEN OR LESS	AID05200
058CR	0654R 0788		XHR	R8,R8	SET DIGIT COUNTER	AID05210
058ER	08AA		LHR	R10,R10	TEST EXPONENT	AID05220
0590R	2117		BMS	XF5	IF MINUS, USE 'E' FORMAT	AID05230
0592R	C5A0		CLHI	R10,6	IF GRTR 6 OR LT 0	AID05240
	0006					



0596R	2124		BPS	XF5	USE 'E' FORMAT	AID05250
0598R	058A	XF4	CLHR	R8,R10	COMPARE DIGIT COUNT & EXPONENT	AID05260
059AR	2139		BNES	XF6	NOT EQUAL, -GENERATE A DIGIT	AID05270
059CR	08AA		SHR	R10,R10	CLEAR EXPONENT	AID05280
059ER	C800	XF5	LHI	R0,C'.'	PUT A DECIMAL PT	AID05290
	002E					
05A2R	0833		LHR	R3,R3	CHECK EXP FLAG	AID05300
05A4R	2134		BNZS	XF6	SKIP DEC PT	AID05310
05A6R	D20B		STB	R0,0(R11)	IN PRINTLINE	AID05320
	0000					
05AAR	26B1		AIS	R11,1	ADJUST INDEX	AID05330
05ACR	C580	XF6	CLHI	R8,6	COMPARE DIGIT COUNT TO 6	AID05340
	0006					
05B0R	4330		BE	XF7	STOP	AID05350
	05EAR					
05B4R	2681		AIS	R8,1	BUMP COUNT	AID05360
05B6R	D1E0		LM	FROM,0	LOAD VALUE	AID05370
	0000					
05BAR	93EE		LBR	FROM,FROM	GET HI MANTISSA	AID05380
05BCR	D3C0		LB	R12,0	GET EXP	AID05390
	0000					
05C0R	C5C0	XFR0	CLHI	R12,X'41'	ONE ?	AID05400
	0041					
05C4R	2185		BLS	XFR2		AID05410
05C6R	EDE0		SLL	FROM,4		AID05420
	0004					
05CAR	27C1		SIS	R12,1	DECREMENT VALUE	AID05430
05CCR	2206		BS	XFR0		AID05440
05CER	D0E0	XFR2	STM	FROM,0	PLACE IN FLT PT REG 0	AID05450
	0000					
05D2R	D2C0		STB	R12,0	SET EXP	AID05460
	0000					
05D6R	90E8		SRLS	FROM,8		AID05470
05D8R	C6E0		OHI	FROM,X'30'	ASCIIIZE	AID05480
	0030					
05DCR	D2EB		STB	FROM,0(R11)	STORE DIGIT	AID05490
	0000					
05E0R	6C00		ME	0,D10	MULT BY TEN	AID05500
	066ER					
05E4R	26B1		AIS	R11,1	BUMP INDEX	AID05510
05E6R	4300		B	XF4	CONTINUE GENERATING DIGITS	AID05520
	0598R					
05EAR	C800	XF7	LHI	R0,X'20'	SPACE	AID05530
	0020					
05EER	D20B		STB	R0,0(R11)	BLANK OUT ZERO	AID05540
	0000					
05F2R	27B1		SIS	R11,1	DECREMENT PRINT INDEX	AID05550
05F4R	D30B		LB	R0,0(R11)		AID05560
	0000					
05F8R	C500		CLHI	R0,X'30'	CHECK FOR TRAILING ZERO	AID05570
	0030					
05FCR	2239		HES	XF7	YES - KEEP CHECKING	AID05580
05FER	C500		CLHI	R0,C'.'	HIT A '. ' ?	AID05590
	002E					
0602R	2135		BNES	XF8		AID05600
0604R	270E		SIS	R0,14	CONVERT TO SPACE	AID05610



0000						
067AR	DOC0	STM	R12,XFSAVE		AID05990	
065ER	D100	LM	R0,SAVE2	RESTORE REGS	AID06000	
0682R	081F	LHR	R1,TO	SAVE RECEIVER ADDRESS	AID06010	
0684R	0766	XHR	R6,R6	DIGIT COUNT	AID06020	
0686R	0788	XHR	R8,R8	EXP SIGN	AID06030	
0688R	0799	XHR	R9,R9	EXP COUNTER	AID06040	
068AR	07AA	XHR	R10,R10		AID06050	
068CR	2B00	SER	0,0	CLEAR FLT PT REG 0	AID06060	
068ER	0733	XHR	R3,R3	MINUS MANTISSA	AID06070	
0690R	082E	LHR	R2,FROM		AID06080	
0692R	D372	LB	R7,0(R2)	GET CHARACTER	AID06090	
0000						
0696R	C570	CLHI	R7,C'-'	CHECK FOR MINUS	AID06100	
0020						
069AR	2133	BNES	XLFT1	SKIP IF NOT	AID06110	
069CR	C830	LHI	R3,X'8000'	SET R3 TO MINUS	AID06120	
8000						
06A0R	D372	XLFT1	LB	R7,0(R2)	PICK UP CHARACTER	AID06130
0000						
06A4R	24CA	LIS	R12,10	CONSTANT OF TEN	AID06140	
06A6R	C570	CLHI	R7,C'0'	CHECK FOR NUMERIC	AID06150	
0030						
06AAR	2188	BLS	XLDEC	LESS - CHECK FOR DECIMAL POINT	AID06160	
06ACR	C570	CLHI	R7,X'3A'	HIGH NUMERIC LIMIT	AID06170	
003A						
06BOR	218F	BLS	XL2	OK - DIGIT	AID06180	
06B2R	C570	CLHI	R7,C'E'	CHAR = 'E'	AID06190	
0045						
06B6R	4330	BE	XLXPT	YES - EXPONENT BEGINNING	AID06200	
06FOR						
06BAR	C570	XLDEC	CLHI	R7,C',.'	DECIMAL POINT ?	AID06210
002E						
06BER	4230	BNE	XLDONE	FINISHED	AID06220	
0716R						
06C2R	08AA	LHR	R10,R10	CHECK EXP	AID06230	
06C4R	4230	BNZ	XLDONE	DONE	AID06240	
0716P						
06C8R	24A1	XLDEC2	LIS	R10,1	SET DEC PT FLAG	AID06250
06CAR	4300	B	XLTF		AID06260	
06EAR						
06CER	08AA	XL2	LHR	R10,R10	COUNT DIGITS TO RT OF ',.'	AID06270
06D0R	2332	BZS	XL3	NO DEC PT - DON'T COUNT	AID06280	
06D2R	2661	AIS	R6,1		AID06290	
06D4R	6C00	XL3	ME	0,D10	MULT BY TEN	AID06300
066ER						
06D8R	C470	NHI	R7,15		AID06310	
000F						
06DCR	C8E7	LHI	FROM,X'4200'(R7)	LOAD HI MANTISSA	AID06320	
4200						
06E0R	07FF	XHR	TO,TO		AID06330	
06E2R	D0E0	STM	FROM,4	PRIME FLT PT REG 2	AID06340	
0004						
06E6R	2822	LER	2,2	NORMALIZE	AID06350	

06E8R	2A02		AER	0,2	ADD IN VALUE	AID06360
06EAR	2621	XLTF	AIS	R2,1	BUMP BUFFER POINTER	AID06370
06ECR	4300		B	XLFT1	GET ANOTHER DIGIT	AID06380
	06A0R					
06FOR	2621	XLEXPT	AIS	R2,1	BUMP INDEX	AID06390
06F2R	D3A2		LB	R10,0(R2)	GET CHARACTER	AID06400
	0000					
06F6R	C5A0		CLHI	R10,C'+'	POSITIVE EXP ?	AID06410
	002B					
06FAR	2334		BES	XLFT2	YES	AID06420
06FCR	CBA0		SHI	R10,C'-'	NEG EXP ?	AID06430
	002D					
0700R	2132		BNES	XLFT3	NO	AID06440
0702R	2621	XLFT2	AIS	R2,1	BUMP POINTER	AID06450
0704R	D372	XLFT3	LB	R7,0(R2)	GET A CHARACTER	AID06460
	0000					
0708R	C570		CLHI	R7,C'0'	* NUMERIC	AID06470
	0030					
070CR	2185		BLS	XLDONE	*	AID06480
070ER	C570		CLHI	R7,X'3A'	* CHECK	AID06490
	003A					
0712R	4280		BL	XLEXDG		AID06500
	0738R					
0716R	08AA	XLDONE	LHR	R10,R10		AID06510
0718R	2134		BNZS	XLDCNT		AID06520
071AR	C790		XHI	R9,-1	COMPLEMENT EXP COUNTER	AID06530
	FFFF					
071ER	2691		AIS	R9,1	ADD ONE	AID06540
0720R	0B96	XLDCNT	SHR	R9,R6	SUBTRACT DECIMAL PLACRS	AID06550
0722R	4330		BZ	XLOUT		AID06560
	0746R					
0726R	2115		BMS	XLCMEX		AID06570
0728R	2461		LIS	R6,1		AID06580
072AR	6C00		ME	0,D10		AID06590
	066ER					
072ER	2207		BS	XLDCNT		AID06600
0730R	2561	XLCMEX	LCS	R6,1		AID06610
0732R	6D00		DE	0,D10		AID06620
	066ER					
0736R	220B		BS	XLDCNT		AID06630
0738R	2621	XLEXDG	AIS	R2,1		AID06640
073AR	C470		NHI	R7,15		AID06650
	000F					
073ER	0C8C		MHR	R8,R12		AID06660
0740R	0A97		AHR	R9,R7		AID06670
0742R	4300		B	XLFT3		AID06680
	0704R					
0746R	D1E0	XLOUT	LM	FROM,0	GET RESULT	AID06690
	0000					
074AR	D0E0		STM	FROM,XLSAVE	SAVE RESULT	AID06700
	0768R					
074ER	D1C0		LM	R12,XFSAVE		AID06710
	065ER					
0752R	D0C0		STM	R12,0		AID06720
	0000					
0756R	D1E0		LM	FROM,XLSAVE	LOAD VALUE	AID06730

075AR	06E3	OHR	FROM,R3	SET SIGN	AID06740
075CR	00E1	STM	FROM,0(R1)	STORE IN MEMORY	AID06750
0760R	D100	LM	R0,SAVE2		AID06760
0764R	2474	LIS	LENGTH,4	SET LENGTH	AID06770
0766R	0302	BR	LINKS	RETURN	AID06780
0768R		XLSAVE	DS	4	AID06790
*-----*					
* TRANSLATE - INTERNAL TO CHARACTER					
076CR	D0A0	XLATEC	STM	R10,SAVE	AID06800
0770R	D38E	XC1	LB	R11,0(FROM)	AID06830
0774R	C9B0	CHI	R11,X'20'	GRAPHIC ?	AID06840
0778R	2383	BNLS	XC3	YES	AID06850
077AR	C8B0	XC2	LHI	R11,X'2E'	AID06860
077ER	C9B0	XC3	CHI	R11,X'60'	AID06870
0782R	2182	BLS	XC4	YES	AID06880
0784R	2205	BS	XC2	NO	AID06890
0786R	D2BF	XC4	STB	R11,0(TO)	AID06900
078AR	26F1	AIS	TO,1	BUMP INDEX	AID06910
078CR	26E1	AIS	FROM,1		AID06920
078ER	27D1	SIS	NUMBER,1	DECREMENT COUNTER	AID06930
0790R	4230	BNZ	XC1		AID06940
0794R	D1A0	LM	R10,SAVE		AID06950
0798R	0302	BR	LINKS	RETURN	AID06960
*-----*					
* MOVE CHARACTERS					
079AR	D0C0	MOVE	STM	R12,SAVE+10	AID06980
079ER	27D1	M1	SIS	NUMBER,1	AID06990
07A0R	2118	BMS	M2	DECREMENT COUNT	AID07000
07A2R	D3CE	LB	R12,0(FROM)	DONE	AID07010
07A6R	D2CF	STB	R12,0(TO)	GET BYTE	AID07020
07AAR	26E1	AIS	FROM,1	STORE BYTE	AID07030
07ACR	26F1	AIS	TO,1	BUMP SENDER	AID07040
07AER	2208	BS	M1	BUMP RECEIVER	AID07050
07B0R	D1C0	M2	LM	CONTINUE	AID07060
07B4R	0302	BR	R12,SAVE+10	RESTORE REGS	AID07070
*-----*					
* CLEAR FIELD					
07B6R	D0D0	CLEAR	STM	NUMBER,SAVE2	AID07080
07BAR	D2EF	STB	FROM,0(TO)	SAVE REGS	AID07090
	0000			STORE BYTE	AID07100
					AID07110
					AID07120

07BER	26F1	AIS	TO,1	BUMP RECEIVER FIELD	AID07130
07C0R	27D1	SIS	NUMBER,1	DECREMENT COUNTER	AID07140
07C2R	2034	BNZS	CLEAR+4	CONTINUE	AID07150
07C4R	D1D0	LM	NUMBER,SAVE2	RESTORE REGS	AID07160
	0AC0R				
07C8R	0302	BR	LINKS	RETURN	AID07170
*-----*					
		*	COMMAND INPUT		
		RDIN	LHI	NUMBER,BUFEND-BUFFER BUFFER SIZE	AID07180
07CAR	C8D0				AID07190
	0028				AID07200
07CER	D000	STM	R0,AIDSAV	SAVE AID REGS	AID07210
	0AE0R				
07D2R	24E0	LIS	FROM,0	CLEARING VALUE	AID07220
07D4R	C8F0	LHI	TO,BUFFER	BUFFER START	AID07230
	0A4ER				
07D8R	4120	BAL	LINKS,CLEAR	CLEAR DATA AREA	AID07240
	07B6R				
07DCR	C8D0	LHI	NUMBER,OPREND-OPRAND	OPERAND SIZE	AID07250
	0014				
07E0R	C8F0	LHI	TO,OPRAND	OPERAND STORAGE START	AID07260
	0A78R				
07E4R	4120	BAL	LINKS,CLEAR	CLEAR OPERANDS	AID07270
	07B6R				
07E8R	C8E0	LHI	FROM,X'2020'	SPACES	AID07280
	2020				
07ECR	C8F0	LHI	TO,OUTBUF	SET RECEIVER ADDRESS	AID07290
	0842R				
07FOR	C8D0	LHI	NUMBER,OUTEND-OUTBUF		AID07300
	006C				
07F4R	4120	BAL	LINKS,CLEAR	CLEAR OUTPUT BUFFER	AID07310
	07B6R				
07F8R	E110	READ	SVC	1,CMDINP	READ A DIRECTIVE
	080AR				
07FCR	4800	LH	R0,CMDINP+2	CHECK STATUS	AID07330
	080CR				
0800R	4230	BNZ	IOERR		AID07340
	092AR				
0804R	D100	LM	R0,AIDSAV	LOAD AID REGS	AID07350
	0AE0R				
0808R	0301	BR	LINK	RETURN	AID07360
080AR	4801	CMDINP	DC	X'4801',0,BUFFER,BUFEND	AID07370
	0000				
	0A4ER				
	0A76R				
*-----*					
		*	INTERACTIVE OUTPUT		
		OUT	LHI	R11,OUTEND-OUTBUF	BUFFER SIZE
0812R	C8B0				AID07380
	006C				AID07390
0816R	C8C0	LHI	R12,X'20'	CHECK FOR SPACES	AID07400
	0020				
081AR	D4CB	OT1	CLB	R12,OUTBUF-1(R11)	SPACE ?
	0841R				
081ER	2134	BNES	OT2	NO - USE LENGTH	AID07430
0820R	27B1	SIS	R11,1	DECREMENT POINTER	AID07440
0822R	0332	BZR	LINKS	RETURN IF BLANK	AID07450
0824R	2205	BS	OT1	CHECK AGAIN	AID07460

0826R	40B0	OT2	STH	R11,OUTBLK+2	PRIME BLOCK	AID07470
	0840R					
082AR	E120		SVC	2,OUTBLK	ISSUE I/O	AID07480
	083ER					
082ER	C8D0		LHI	NUMBER,OUTEND-OUTBUF	BUFFER SIZE	AID07490
	006C					
0832R	C8E0		LHI	FROM,X*20'	CLEARING VALUE	AID07500
	0020					
0836R	C8F0		LHI	TO,OUTBUF	BUFFER START	AID07510
	0842R					
083AR	4300		B	CLEAR	CLEAR DATA AREA	AID07520
	07B6R					
083ER	0007	OUTBLK	DC	7		AID07530
0840R	0000		DC	0		AID07540
0842R		OUTBUF	EQU	*		AID07550
0842R			DO	54		AID07560
0842R	2020		DC	C' '		AID07570
0844R	2020		DC	C' '		AID07570
0846R	2020		DC	C' '		AID07570
0848R	2020		DC	C' '		AID07570
084AR	2020		DC	C' '		AID07570
084CR	2020		DC	C' '		AID07570
084ER	2020		DC	C' '		AID07570
0850R	2020		DC	C' '		AID07570
0852R	2020		DC	C' '		AID07570
0854R	2020		DC	C' '		AID07570
0856R	2020		DC	C' '		AID07570
0858R	2020		DC	C' '		AID07570
085AR	2020		DC	C' '		AID07570
085CR	2020		DC	C' '		AID07570
085ER	2020		DC	C' '		AID07570
0860R	2020		DC	C' '		AID07570
0862R	2020		DC	C' '		AID07570
0864R	2020		DC	C' '		AID07570
0866R	2020		DC	C' '		AID07570
0868R	2020		DC	C' '		AID07570
086AR	2020		DC	C' '		AID07570
086CR	2020		DC	C' '		AID07570
086ER	2020		DC	C' '		AID07570
0870R	2020		DC	C' '		AID07570
0872R	2020		DC	C' '		AID07570
0874R	2020		DC	C' '		AID07570
0876R	2020		DC	C' '		AID07570
0878R	2020		DC	C' '		AID07570
087AR	2020		DC	C' '		AID07570
087CR	2020		DC	C' '		AID07570
087ER	2020		DC	C' '		AID07570
0880R	2020		DC	C' '		AID07570
0882R	2020		DC	C' '		AID07570
0884R	2020		DC	C' '		AID07570
0886R	2020		DC	C' '		AID07570
0888R	2020		DC	C' '		AID07570
088AR	2020		DC	C' '		AID07570
088CR	2020		DC	C' '		AID07570
088ER	2020		DC	C' '		AID07570
0890R	2020		DC	C' '		AID07570

0892R	2020	DC	C'	'		AID07570
0894R	2020	DC	C'	'		AID07570
0896R	2020	DC	C'	'		AID07570
0898R	2020	DC	C'	'		AID07570
089AR	2020	DC	C'	'		AID07570
089CR	2020	DC	C'	'		AID07570
089ER	2020	DC	C'	'		AID07570
08A0R	2020	DC	C'	'		AID07570
08A2R	2020	DC	C'	'		AID07570
08A4R	2020	DC	C'	'		AID07570
08A6R	2020	DC	C'	'		AID07570
08A8R	2020	DC	C'	'		AID07570
08AAR	2020	DC	C'	'		AID07570
08ACR	2020	DC	C'	'		AID07570
08AER		OUTEND	EQU	*		AID07580
*-----*						
* BINARY OUTPUT						
08AER	E110	BINOUT	SVC	1,BINBLK	ISSUE I/O	AID07600
	08BCR					AID07610
08B2R	4800	LH	R0,BINBLK+2		CHECK STATUS	AID07620
	08REK					
08B6R	4230	BNZ	IOERR			AID07630
	092AR					
08BAR	0302	BR	LINKS		RETURN	AID07640
08BCR	3805	BINBLK	DC	X'3805',0		AID07650
	0000					
08COR	0000	DC	0,0			AID07660
	0000					
*-----*						
* HIGH SPEED LIST OUTPUT						
08C4R	D0B0	LIST	STM	R11,SAVE2+18	SAVE REGS	AID07680
	0AD2R					AID07690
08C8R	4020	STH	LINKS,LT3		SAVE LINK	AID07700
	090AR					
08CCR	C8B0	LHI	R11,OUTEND-1		BUFFER END	AID07710
	08ADR					
08D0R	C800	LHI	R0,X'20'			AID07720
	0020					
08D4R	C5B0	LT1	CLHI	R11,OUTBUF	CHECK FOR BLANK LINE	AID07730
	0842R					
08D8R	2336	BES	LT2			AID07740
08DAR	D40B	CLB	R0,0(R11)		SPACE ?	AID07750
	0000					
08DER	2133	BNES	LT2		NO - SEARCH OVER	AID07760
08E0R	27B1	SIS	R11,1		DECREMENT POINTER	AID07770
08E2R	2207	BS	LT1		CONTINUE	AID07780
08E4R	40B0	LT2	STH	R11,LSTBLK+6	SAVE END ADDRESS	AID07790
	0912R					
08E8R	E110	SVC	1,LSTBLK		ISSUE I/O	AID07800
	090CR					
08ECR	4800	LH	R0,LSTBLK+2		CHECK STATUS	AID07810
	090ER					
08FOR	4230	UNZ	IOERR			AID07820
	092AR					
08F4R	C8D0	LHI	NUMBER,OUTEND-OUTBUF		BUFFER SIZE	AID07830
	006C					



08F8R	C8E0		LHI	FROM,X*2020*	CLEARING VALUE		AID07840
	2020						
08FCR	C8F0		LHI	TO,OUTBUF	BEGIN ADDR LSS		AID07850
	0842R						
0900R	4120		BAL	LINKS,CLEAR	CLEAR DATA AREA		AID07860
	07B6R						
0904R	D1B0		LM	R11,SAVE2+18	RESTORE REGS		AID07870
	0AD2R						
0908R	4300		DC	X*4300*	RETURN		AID07880
090AR	0000	LT3	DC	0			AID07890
090CR	2803	LSTBLK	DC	X*2803*,0			AID07900
	0000						
0910R	0842R		DC	OUTBUF			AID07910
0912R	08AER		DC	OUTEND			AID07920
*-----*							
* ERROR ROUTINES							
0914R	D1E0	ILLEG	LM	FROM,ILLMSG	MOVE ILLEGAL INSTR MESSAGE		AID07930
	0966R						AID07940
0918R	D0E0		STM	FROM,OUTBUF+2			AID07950
	0844R						
091CR	4120		BAL	LINKS,LIST	PRINT MESSAGE AND ADDRESS		AID07960
	08C4R						
0920R	4300		R	LDREG			AID07970
	0046R						
0924R	C8B0	CMDERR	LHI	R11,CMDMSG			AID07980
	0942R						
0928R	2305		RS	LOG			AID07990
092AR	E120	IOERR	SVC	2,UNPACK			AID08000
	0960R						AID08010
092ER	C8B0		LHI	R11,IOMSG			AID08020
	094ER						
0932R	E12B	LOG	SVC	2,0(R11)			AID08030
	0000						
0936R	4300		B	LDREG			AID08040
	0046R						
093AR	E120	PAWS	SVC	2,PAUSE	PAUSE OS AIDS		AID08050
	0964R						
093ER	4300		R	LDREG			AID08060
	0046R						
0942R	0007	CMDMSG	DC	7			AID08070
0944R	0008		DC	CMDND-CMDMS			AID08080
0946R	434D	CMDMS	DC	C*CMD-ERR*			AID08090
	442D						
	4552						
	5220						
094ER		CMDND	EQU	*			AID08100
094ER	0007	IOMSG	DC	7			AID08110
0950R	000E		DC	IOMEND-IOMS			AID08120
0952R	492F	IOMS	DC	C*I/O-ERR:			AID08130
	4F2D						
	4552						
	523A						
	2020						
	2020						
	2020						
0960R		IOMEND	EQU	*			AID08140

0960R	0006	UNPACK	DC	6		AID08150
0962R	095CR		DC	IOMEND-4		AID08160
0964R	0001	PAUSE	DC	1		AID08170
0966R	494C	ILLMSG	DC	C'ILL:'		AID08180
	4C3A					
*-----*						
		* TABLE SCANNER				AID08190
096AR	C880	TABC	LHI	R11,OPTBEN	END OF OPTION C	AID08200
	09B6R					AID08210
096ER	C8C0		LHI	R12,OPTCEN	BEGIN OF OPTION C	AID08220
	09C2R					
0972R	2308		BS	TBSCAN		AID08230
0974R	C8C0	TABB	LHI	R12,OPTBEN	BEGIN OF OPTION B	AID08240
	09B6R					
0978R	2303		BS	TAB		AID08250
097AR	C8C0	TABA	LHI	R12,OPTAEN	BEGIN OF OPTION A	AID08260
	09AFR					
097ER	C8B0	TAB	LHI	R11,OPTAB	END OF OPTIONS A & B	AID08270
	09A8R					
0982R	D40C	TBSCAN	CLB	R0,0(R12)	COMPARE CHARACTER	AID08280
	0000					
0986R	2335		BES	TBOUT	HIT, GET VECTOR	AID08290
0988R	27C1		SIS	R12,1	CHECK NEXT	AID08300
098AR	09CB		CHR	R12,R11	END OF TABLE	AID08310
098CR	0286		BLR	CMDER	YES = ERROR	AID08320
098ER	2206		BS	TBSCAN	CONTINUE LOOKING	AID08330
0990R	C8C0	TBOUT	SHI	R12,OPTAB	GET PURE INDEX	AID08340
	09A8R					
0994R	0ACC		AHR	R12,R12		AID08350
0996R	9380		LBR	MODE,R0	SET MODE	AID08360
0998R	481C		LH	LINK,OPN(R12)	PICK UP VECTOR ADDRESS	AID08370
	09C8R					
099CR	0301		BR	LINK	ENTER ROUTINE (RETURN VIA LINKS )	AID08380
099ER	C8B0	TABD	LHI	R11,OPTCEN	END OF OPTION D	AID08390
	09C2R					
09A2R	C8C0		LHI	R12,OPTDEN	BEGIN OF OPTION D	AID08400
	09C6R					
09A6R	220C		BS	TBOUT-2		AID08410
*-----*						
		* SECOND CHARACTER OPTIONS				AID08420
09A8R	48	OPTAB	DB	C'H'		AID08430
09A9R	44		DB	C'D'		AID08440
09AAR	41		DB	C'A'		AID08450
09ABR	42		DB	C'B'		AID08460
09ACR	43		DB	C'C'		AID08480
09ADR	46		DB	C'F'		AID08490
09AER	52		DB	C'R'		AID08500
09AFR	50		DB	C'P'		AID08510
09AFR		OPTAEN	EQU	*-1		AID08520
09B0R	46		DB	C'F'		AID08530
09B1R	41		DB	C'A'		AID08540
09B2R	42		DB	C'B'		AID08550
09B3R	48		DB	C'H'		AID08560
09B4R	44		DB	C'D'		AID08570
09B5R	43		DB	C'C'		AID08580
09B6R	45		DB	C'E'		AID08590

09B6R		OPTBEN	EQU	*-1	AID08600
09B7R	58		DB	C'X'	AID08610
09B8R	53		DB	C'S'	AID08620
09B9R	54		DB	C'T'	AID08630
09BAR	4C		DB	C'L'	AID08640
09BBR	48		DB	C'H'	AID08650
09BCR	43		DB	C'C'	AID08660
09BDR	41		DB	C'A'	AID08670
09BER	42		DB	C'B'	AID08680
09BFR	44		DB	C'D'	AID08690
09C0R	46		DB	C'F'	AID08700
09C1R	50		DB	C'P'	AID08710
09C2R	55		DB	C'U'	AID08720
09C2R		OPTCEN	EQU	*-1	AID08730
09C3R	58		DB	C'X'	AID08740
09C4R	53		DB	C'S'	AID08750
09C5R	54		DB	C'T'	AID08760
09C6R	50		DB	C'P'	AID08770
09C6R		OPTDEN	EQU	*-1	AID08780
09C7R	00		DB	*	AID08790
09C8R	040ER	OPN	DC	XLATEH	AID08800
09CAR	04E2R		DC	XLATED	AID08810
09CCR	167CR		DC	XLATEA	AID08820
09CER	167CR		DC	XLATEA	AID08830
09D0R	076CR		DC	XLATEC	AID08840
09D2R	0518R		DC	XLATEF	AID08850
09D4R	040ER		DC	XLATEH	AID08860
09D6R	040ER		DC	XLATEH	AID08870
09D8R	0672R		DC	XLATEL	AID08880
09DAR	158AR		DC	XLATEU	AID08890
09DCR	158AR		DC	XLATEU	AID08900
09DER	044CR		DC	XLATHB	AID08910
09E0R	04B2R		DC	XLATDB	AID08920
09E2R	079AR		DC	MOVE	AID08930
09E4R	044CR		DC	XLATHB	AID08940
09E6R	11B4R		DC	BREAK	AID08950
09E8R	1390R		DC	SNAP	AID08960
09EAR	1266R		DC	TRACE	AID08970
09ECR	08EER		DC	LOADER	AID08980
09EER	0CE0R		DC	HEXOMP	AID08990
09F0R	0CF8R		DC	CHRDMP	AID09000
09F2R	0CC6R		DC	ASMDMP	AID09010
09F4R	0CC6R		DC	ASMDMP	AID09020
09F6R	0884R		DC	DECDMP	AID09030
09F8R	0BCAR		DC	FLTDMP	AID09040
09FAR	149AR		DC	PROTEK	AID09050
09FCR	0D4AR		DC	UTIL	AID09060
09FER	1228R		DC	ZAPBRK	AID09070
0A00R	145CR		DC	ZAPSNP	AID09080
0A02R	1352R		DC	ZAPTRC	AID09090
0A04R	1506R		DC	ZAPPRO	AID09100
*-----*					
* COMMAND VECTOR TABLE					
0A06R	20	CMDCHR	DB	C' '	AID09120
0A07R	4F		DB	C'O'	AID09130
0A08R	0D		DB	X'0D'	AID09140
					AID09150

0A09R 20		DB	C'-'	AID09160
0A0AR 4A		DB	C'J'	AID09170
0A0BR 41		DB	C'A'	AID09180
0A0CR 53		DB	C'S'	AID09190
0A0DR 52		DB	C'R'	AID09200
0A0ER 4C		DB	C'L'	AID09210
0A0FR 42		DB	C'B'	AID09220
0A10R 45		DB	C'E'	AID09230
0A11R 49		DB	C'I'	AID09240
0A12R 5A		DB	C'Z'	AID09250
0A13R 44		DB	C'D'	AID09260
0A14R 50		DB	C'P'	AID09270
0A15R 47		DB	C'G'	AID09280
0A16R 4E		DB	C'N'	AID09290
0A17R	CMCREN	EQU	*	AID09300
0A17R 00		DB	*	AID09310
0A18R	CMDTBL	EQU	*	AID09320
0A18R 0284R		DC	NEXT	AID09330
0A1AR 0187R		DC	OPIN+1	AID09340
0A1CR 0284R		DC	NEXT	AID09350
0A1ER 0288R		DC	PRECED+1	AID09360
0A20R 0290R		DC	JUMP	AID09370
0A22R 02A5R		DC	ARITH+1	AID09380
0A24R 02A5R		DC	ARITH+1	AID09390
0A26R 0227R		DC	RPLACE+1	AID09400
0A28R 00DFR		DC	LU+1	AID09410
0A2AR 012DR		DC	BYAS+1	AID09420
0A2CR 0156R		DC	TERM	AID09430
0A2ER 0FEDR		DC	ENTER+1	AID09440
0A30R 1191R		DC	ZAP+1	AID09450
0A32R 0B2DR		DC	DUMP+1	AID09460
0A34R 093AR		DC	PAWS	AID09470
0A36R 0325P		DC	GO+1	AID09480
0A38R 0222R		DC	NEKST	AID09490
0A3AR	CMDEND	EQU	*	AID09500
*-----*				
* DELIMITER TABLE				
0A3AR 2F	DELIM	DB	C'/'	AID09520
0A3BR 0D		DB	X'0D'	AID09530
0A3CR 28	OPREN	DB	C'('	AID09540
0A3DR 29	CPREN	DB	C')'	AID09550
0A3ER 2C	COMMA	DB	C','	AID09560
0A3FR 2E		DB	C','	AID09570
0A40R 27	POP	DB	X'27'	AID09580
0A41R 00		DB	X'00'	AID09590
0A42R 20	SPACE	DB	X'20'	AID09600
0A42R	DELEND	EQU	*-1	AID09610
0A43R 00		DB	*	AID09620
*-----*				
* RESERVATIONS				
0A44R 0000	DECD2	DC	0	AID09630
0A46R 0000	PRORO	DC	0	AID09640
0A48R 0000	PROFLG	DC	0	AID09650
0A4AR 0000	PSWSAV	DC	0.0	AID09660
0000				AID09670
0A4ER	BUFFER	DS	40	AID09680
				AID09690
				AID09700

0A76R	2F2F	DC	C'///'	AID09710
0A76R		BUFEND EQU	*-2	AID09720
0A78R		OPRAND DS	20	AID09730
0A8CR		OPREND EQU	*	AID09740
0A8CR	534F	SNPMSG DC	C'SNP:'	AID09750
	503A			
0A90R	5052	PROMSG DC	C'PRO:'	AID09760
	4F3A			
0A94R	0000	TRCFLG DC	0	AID09770
0A96R	0000	OPSAVE DC	0	AID09780
0A98R		SAVREG EQU	*	AID09790
0A98R		REGSAV DS	32	AID09800
0AB8R	0000	DBLWK DC	E'0'	AID09810
	0000			
0ABCR	0000	DBLSAV DC	E'0'	AID09820
	0000			
0AC0R		SAVE2 DS	32	AID09830
0AE0R	0000	AIDSAV DC	0,0,0,0,0,0,CMDERR,0,C'HH'	AID09840
	0000			
	0000			
	0000			
	0000			
	0000			
	0924R			
	0000			
	4848			
0AF2R	0812R	DC	OUT,IN,0,0,0,0,0	AID09850
	004ER			
	0000			
	0000			
	0000			
	0000			
	0000			
0B00R		SAVE DS	32	AID09860
0R20R	0007	TITLE DC	7	AID09870
0R22R	0008	DC	TITLEN-TITLE	AID09880
0B24R	4F53	TITLE DC	C'OS AIDS'	AID09890
	2041			
	4944			
	5320			
0B2CR		TITLEN EQU	*	AID09900
		*-----*		AID09910
		* DUMP HANDLER		AID09920
0B2CR	4070	DUMP STH	LENGTH,D1 SAVE CURRENT CELL LENGTH	AID09930
	0B82R			
0B30R	D300	LR	R0,BUFFER+1 CHECK SECONDARY DIRECTIVE	AID09940
	0A4FR			
0B34R	C500	CLHI	R0,X'0055' IS IT A 'U' ?	AID09950
	0055			
0B38R	4330	BE	DUMPU YES , SKIP THE CALCULATIONS	AID09960
	0B7AR			
0B3CR	D3D0	LB	NUMBER,OPRAND GET OPERAND LENGTH	AID09970
	0A78R			
0B40R	C8E0	LHI	FROM,OPRAND+1 ADDRESS OF SENDER FIELD	AID09980
	0A79R			
0B44R	C8F0	LHI	TO,OPSAVE ADDRESS OR RECEIVER FIELD	AID09990

0B48R	0A96R 4000	STH	NUMBER,OPSAVE	CLEAR WORK AREA	AID10000
0B4CR	0A96R 4120	BAL	LINKS,XLATHB	TRANSLATE TO BINARY	AID10010
0B50R	044CR 4800	LH	R0,OPSAVE	LOAD START ADDRESS	AID10020
0B54R	0A96R C400	NHI	R0,X'FFFE'	ENSURE HALFWORD BOUNDARY	AID10030
0B58R	FFFE 0AED	AHR	FROM,NUMBER	POSITION TO NEXT OPERAND	AID10040
0B5AR	D3DE	LB	NUMBER,0(FROM)	OPERAND LENGTH	AID10050
0B5ER	0000 26E1	AIS	FROM,1	BUMP OPERAND POINTER	AID10060
0B60R	4120	BAL	LINKS,XLATHB	TRANSLATE TO BINARY	AID10070
0B64R	044CR 08E0	LHR	FROM,R0		AID10080
0B66R	4850	LH	VALUE,OPSAVE	GET HIGH LIMIT	AID10090
0B6AR	0A96R 05E5	CLHR	FROM,VALUE	COMPARE EXTENTS	AID10100
0B6CR	0386	BNLR	CMDER		AID10110
0B6ER	4870	LH	LENGTH,D1	RESTORE LENGTH	AID10120
0B72R	0B82R C8F0	LHI	TO,OUTBUF	SET UP OUTPUT BUFFER ADDRESS	AID10130
0B76R	0842R D300	LB	R0,BUFFER+1	DUMP TYPE	AID10140
0B7AR	0A4FR 4120	DUMP BAL	LINKS,TABC	GO TO HANDLER	AID10150
0B7ER	096AR 4300	B	NAME	ISSUE NAME & READ DIRECTIVE	AID10160
0B82R	004AR 0000	D1	DC 0		AID10170
*-----*					
* DECIMAL FORMAT DUMP					
0B84R	4020	DECOMP	STH	LINKS,DECD1	SAVE LINK
0B88R	0BC8R 40E0	DD1	STH	FROM,DECD2	MEMORY LOCATION
0B8CR	0A44R C8E0	LHI	FROM,DECD2	SET AS FROM ADDR	AID10220
0B90R	0A44R C8F0	LHI	TO,OUTBUF	RESET BUFFER POINTER	AID10230
0B94R	0842R 24D2	LIS	NUMBER,2		AID10240
0B96R	4120	BAL	LINKS,XLATEH	TRANSLATE ADDR TO ASCII	AID10250
0B9AR	040ER 26F6	AIS	TO,6	BUMP LINE POINTER	AID10260
0B9CR	24C4	LIS	R12,4	SET COUNTER	AID10270
0B9ER	48E0	LH	FROM,DECD2	RESTORE MEMORY LOCATION	AID10280
0BA2R	0A44R 4120	DD2	BAL	LINKS,XLATEH	TRANSLATE CORE TO ASCII
0BA6R	040ER C8B0	LHI	R11,C'='	EQUAL SIGN	AID10300
0BAAR	003D D2BF	STB	R11,4(TO)	STORE BETWEEN HEX AND DEC	AID10310
0BAER	0004 26F6	AIS	TO,6	BUMP LINE POINTER	AID10320
0BB0R	4120	BAL	LINKS,XLATED	TRANSLATE CORE TO DECIMAL ASCII	AID10330

04E2R					
0BB4R 26F7	AIS	TO,7	BUMP LINE POINTER		AID10340
0BB6R 26E2	AIS	FROM,2	INCREMENT MEMORY LOCATION		AID10350
0BB8R 27C1	SIS	R12,1	DECREMENT COUNTER		AID10360
0BBAR 203C	BNZS	DD2	CONTINUE LINE		AID10370
0BBCR 4120	BAL	LINKS,LIST	PRINT LINE		AID10380
0BC4R					
0BC0R 095E	CHR	VALUE,FROM	COMPARE TO HIGH LIMIT		AID10390
0BC2R 4380	HNL	DD1	CONTINUE		AID10400
0B88R					
0BC6R 4300	DC	X*4300'	RETURN		AID10410
0BC8R 0000	DECD1	DC	0		AID10420
					AID10430
					AID10440
					AID10450
0BCAR 4020	FLTDMP	STH	LINKS,FLT2	SAVE LINK REG	
0BE6R					
0BCER C8F0	FLT1	LHI	TO,OUTBUF	RESET LINE POINTER	AID10460
0842R					
0BD2R 4120	BAL	LINKS,XLATEF	TRANSLATE TO FLOATING POINT		AID10470
0518R					
0BD6R 0AE7	AHR	FROM,LENGTH	INCREMENT SENDER ADDRESS		AID10480
0BD8R C8F0	LHI	TO,OUTBUF+32	SET LINE POINTER		AID10490
0862R					
0BDCR 4120	BAL	LINKS,XLATEF	TRANSLATE		AID10500
0518R					
0BE0R 4120	BAL	LINKS,LIST	PRINT FLOATING POINT		AID10510
08C4R					
0BE4R 0AE7	AHR	FROM,LENGTH	INCREMENT SENDER ADDRESS		AID10520
0BE6R 095E	CHR	VALUE,FROM	DONE ?		AID10530
0BE8R 228D	BNLS	FLT1	NO		AID10540
0BEAR 4300	DC	X*4300'	RETURN		AID10550
0BE6R 0000	FLT2	DC	0		AID10560
					AID10570
					AID10580
					AID10590
0REER D000	LOADER	STM	R0,SAVE	SAVE REGISTERS	
0B00R					
0BF2R 4020	STH	LINKS,LDR4	SET RETURN		AID10600
0C70R					
0RF6R 07DD	XHR	NUMBER,NUMBER	CLEAR SEQUENCE NUMBER		AID10610
0BF8R 40F0	STH	TO,BINBLK+4	SET BUFFER BEGIN		AID10620
08C0R					
0BF6R C8B0	LHI	R11,OUTBUF+107	SET BUFFER END		AID10630
08ADR					
0C00R 40B0	STH	R11,BINBLK+6			AID10640
08C2R					
0C04R 4110	BAL	LINK,SETBUF	CLEAR AND SET UP BUFFER		AID10650
0CAAR					
0C08R C8A0	LHI	R10,X*35'	SET ABSOLUTE TOGGLE		AID10660
0035					
0C0CR D2A0	STB	R10,OUTBUF+4			AID10670
0846R					
0C10R D2E0	STB	FROM,OUTBUF+6	SET LOAD ADDRESS LSD		AID10680
0848R					
0C14R 08AE	LHR	R10,FROM			AID10690
0C16R 94AA	EXBR	R10,R10			AID10700
0C18R D2A0	STB	R10,OUTBUF+5	SET LOAD ADDRESS MSD		AID10710

0847R					
0C1CR	C840	LHI	OPEN,LDPACK		AID10720
	0C72R				
0C20R	248E	LIS	R11,14	SET BUFFER INDEX	AID10730
0C22R	055E	LDR1	CLHR	VALUE, FROM	AID10740
0C24R	4280		BL	LOREND	AID10750
	0C54R				
0C28R	C580	CLHI	R11,212	NO - END OF BUFFER ?	AID10760
	00D4				
0C2CR	2183	BLS	LDR2	NO - CONTINUE	AID10770
0C2ER	4110	BAL	LINK,LDROUT	YES - OUTPUT BUFFER	AID10780
	0C92R				
0C32R	24A8	LDR2	LIS	R10,8	SET LOADER ITEM TYPE
0C34R	0124		BALR	LINKS,OPEN	PACK A HALF BYTE
0C36R	487E		LH	LENGTH,0(FROM)	GET A HALFWORD
	0000				
0C3AR	94A7	EXBR	R10,LENGTH	MAKE MSB LOW	AID10820
0C3CR	90A4	SRLS	R10,4	SHIFT OUT LS BITS	AID10830
0C3ER	0124	BALR	LINKS,OPEN	PACK A HALF BYTE - MSD	AID10840
0C40R	94A7	EXBR	R10,LENGTH	MAKE MSB LOW	AID10850
0C42R	0124	BALR	LINKS,OPEN	PACK A HALF BYTE - LSD	AID10860
0C44R	08A7	LHR	R10,LENGTH	GET LSB	AID10870
0C46R	90A4	SRLS	R10,4	SHIFT OUT LS BITS	AID10880
0C48R	0124	BALR	LINKS,OPEN	PACK A HALF BYTE - MSD	AID10890
0C4AR	08A7	LHR	R10,LENGTH	GET LS BITS	AID10900
0C4CR	0124	BALR	LINKS,OPEN	PACK A HALF BYTE - LSD	AID10910
0C4ER	26E2	AIS	FROM,2	INCREMENT POINTER	AID10920
0C50R	4300	B	LDR1	CONTINUE	AID10930
	0C22R				
0C54R	C580	LDREND	CLHI	R11,215	INDEX = END ?
	00D7				
0C58R	2183	BLS	LDR5	USE SAME RECORD	AID10950
0C5AR	4110	BAL	LINK,LDROUT	OUTPUT BUFFER	AID10960
	0C92R				
0C5ER	24A3	LDR5	LIS	R10,3	SET ABSOLUTE TOGGLE
0C60R	0124		BALR	LINKS,OPEN	AID10980
0C62R	24A1		LIS	R10,1	SET END LOADER ITEM
0C64R	0124		BALR	LINKS,OPEN	AID11000
0C66R	4110		BAL	LINK,LDROUT	AID11010
	0C92R				
0C6AR	D100	LM	R0,SAVE	RESTORE REGISTERS	AID11020
	0B00R				
0C6ER	4300	DC	X'4300'	RETURN	AID11030
0C70R	0000	LDR4	DC	0	AID11040
0C72R	24CF	LDPACK	LIS	R12,15	MASK
0C74R	04AC		NHR	R10,R12	SINGLE OUT HALF BYTE
0C76R	91A4		SLLS	R10,4	AID11070
0C78R	081B		LHR	LINK,R11	SAVE INDEX
0C7AR	9011		SRLS	LINK,1	DIVIDE INDEX
0C7CR	2383		BNCS	LDPK1	CARRY ?
0C7ER	91C4		SLLS	R12,4	SHIFT MASK
0C80R	90A4		SRLS	R10,4	SHIFT HALF BYTE
0C82R	D301	LDPK1	LB	R0,OUTBUF(LINK)	PICK UP BYTE
	0842R				
0C86R	040C	NHR	R0,R12	ENSURE DESIRED HALF BYTE IS HIGH	AID11140
0C88R	060A	OHR	R0,R10	COMBINE WITH PACKING VALUE	AID11150



0C8AR	D201	STB	R0,OUTBUF(LINK)	STORE IN OUTPUT BUFFER	AID11160
	0842R				
0C8ER	26H1	AIS	R11,1	BUMP INDEX	AID11170
0C90R	0302	BR	LINKS		AID11180
0C92R	C8A0	LDR3	LHI R10,-1	*	AID11190
	FFFF				
0C96R	C8B0	LHI	R11,107	* CALCULATE	AID11200
	006B				
0C9AR	47AB	LDR3	XH R10,OUTBUF(R11)	* CHECKSVM	AID11210
	0842R				
0C9ER	27B2	SIS	R11,2	DECREMENT INDEX	AID11220
0CA0R	2213	BNMS	LDR3		AID11230
0CA2R	40A0	STH	R10,OUTBUF+2	STORE CHECKSUM IN RECORD	AID11240
	0844R				
0CA6R	4120	BAL	LINKS,BINOUT	OUTPUT RECORD	AID11250
	08AER				
0CAAR	08AE	SETBUF	LHR R10,FROM	SAVE FROM ADDRESS	AID11260
0CACR	080D	LHR	R0,NUMBER	SAVE RECORD NUMBER	AID11270
0CAER	C8D0	LHI	NUMBER,108		AID11280
	006C				
0CB2R	07EF	XHR	FROM,FROM	SET CLEARING VALUE	AID11290
0CB4R	4120	BAL	LINKS,CLEAR	CLEAR BUFFER	AID11300
	07B6R				
0CB8R	08EA	LHR	FROM,R10	RESTORE ADDRESS POINTER	AID11310
0CBAR	08D0	LHR	NUMBER,R0	RESTORE RECORD NUMBER	AID11320
0CBCR	27D1	SIS	NUMBER,1	DECREMENT RECORD NUMBER	AID11330
0CBER	40D0	STH	NUMBER,OUTBUF	STORE IN BUFFER	AID11340
	0842R				
0CC2R	24B8	LIS	R11,8	RESET INDEX	AID11350
0CC4R	0301	BR	LINK	RETURN	AID11360
					AID11370
*-----*					
* ASSEMBLY FORMAT DUMP					
0CC6R	4020	ASMDMP	STH LINKS,ASM1	SAVE LINK	AID11380
	0CDE R				AID11390
0CCAR	C8F0	LHI	TO,OUTBUF	RESET BUFFER POINTER	AID11400
	0842R				
0CCER	4120	BAL	LINKS,XLATEA	TRANSLATE TO ASSEMBLER	AID11410
	167CR				
0CD2R	4120	BAL	LINKS,LIST	PRINT LINE	AID11420
	08C4R				
0CD6R	0AE7	AHR	FROM,LENGTH	INCREMENT ADDRESS	AID11430
0CD8R	095E	CHR	VALUE,FROM	HIT HIGH LIMIT?	AID11440
0CDAR	2288	BNLS	ASMDMP+4	NO, CONTINUE	AID11450
0CDCR	4300	DC	X'4300'	RETURN	AID11460
0CDE R	0000	ASM1	DC 0		AID11470
					AID11480
*-----*					
* BINARY DUMP FORMAT					
0CE0R	C650	HEXDMP	OHI VALUE,1	ODD END ADDRESS	AID11490
	0001				AID11500
0CE4R	4020	STH	LINKS,HD1	SAVE RETURN LINK	AID11510
	0CF6R				
0CE8R	4050	STH	VALUE,BINBLK+6	PRIME PARAM BLOCK	AID11520
	08C2R				
0CECR	40E0	STH	FROM,BINBLK+4		AID11530
	08C0R				
0CF0R	4120	BAL	LINKS,BINOUT	WRITE BLOCK	AID11540

08AER						
0CF4R 4300		DC	X'4300'	RETURN		AID11550
0CF6R 0000	HD1	DC	0			AID11560
*-----*						
* CHARACTER FORMAT DUMP						
0CF8R 4020	CHRDMP	STH	LINKS,CHR1	SAVE LINK		AID11570
0D46R						AID11580
0CF6R 40E0		STH	FROM,CD1	SAVE FROM ADDRESS		AID11590
0D48R						
0D00R C8E0		LHI	FROM,CD1	SET AS SENDER FIELD		AID11600
0D48R						
0D04R 24D2		LIS	NUMBER,2	TWO BYTE ADDRESS		AID11610
0D06R 4120		BAL	LINKS,XLATEH	TRANSLATE TO ASCII		AID11620
040ER						AID11630
0D0AR 48E0		LH	FROM,CD1	RELOAD ADDRESS		AID11640
0D48R						
0D0ER 26F6		AIS	TO,6	BUMP POINTER		AID11650
0D10R 2408		LIS	R0,8	BLOCK COUNTER		AID11660
0D12R 4120	CD2	BAL	LINKS,XLATEH	TRANSLATE TO ASCII		AID11670
040ER						
0D16R 26E2		AIS	FROM,2	BUMP CORE ADDRESS		AID11680
0D18R 26F5		AIS	TO,5	BUMP LINE INDEX		AID11690
0D1AR 2701		SIS	R0,1	DECREMENT COUNTER		AID11700
0D1CR 2035		BNZS	CD2			AID11710
0D1ER C8C0		LHI	R12,X'2A'	STAR		AID11720
002A						
0D22R D2CF		STB	R12,0(T0)	INSERT STAR		AID11730
0000						
0D26R 26F2		AIS	TO,2	BUMP LINE POINTER		AID11740
0D28R 26DE		AIS	NUMBER,14	BUMP BYTE COUNT		AID11750
0D2AR 08E0		SHR	FROM,NUMBER	RESET CORE ADDRESS		AID11760
0D2CR 4120		BAL	LINKS,XLATEC	TRANSLATE TO CHARACTER		AID11770
076CR						
0D30R D2CF		STB	R12,17(T0)	INSERT STAR		AID11780
0011						
0D34R 0AED		AHR	FROM,NUMBER	BUMP ADDRESS BY SIXTEEN		AID11790
0D36R C8F0		LHI	TO,OUTBUF	RESET BUFFER		AID11800
0842R						
0D3AR 4120		BAL	LINKS,LIST	OUTPUT PRINT LINE		AID11810
08C4R						
0D3ER 095E		CHR	VALUE,FROM	HIT HIGH LIMIT?		AID11820
0D40R 4380		BNL	CHRDMP+4	NO, CONTINUE		AID11830
0CF6R						
0D44R 4300		DC	X'4300'	RETURN		AID11840
0D46R 0000	CHR1	DC	0			AID11850
0D48R 0000	CD1	DC	0			AID11860
*-----*						
* UTILITY (U) DUMP HANDLER						
0D4AR 24C6	UTIL	LIS	R12,6	TABLE COUNTER		AID11870
0D4CR 24D2		LIS	NUMBER,2	SET TRANSLATION LENGTH		AID11880
0D4ER C8E0		LHI	FROM,OPSAVE	SET WORK AREA		AID11890
0A96R						AID11900
0D52R 2408	UT1	LIS	R0,8	ITEM COUNTER		AID11910
0D54R C8F0		LHI	TO,OUTBUF+6	SET PRINT LINE		AID11920
0848R						AID11930
0D58R 48BC		LH	R11,MSGTBL(R12)	ADDRESS OF FIRST TABLE MSG		AID11940

OD5CR	ODDER 485B 0000	LH	VALUE,0(R11)	MESSAGE	AID11950
OD60R	4050 0842R	STH	VALUE,OUTBUF	STORE IN PRINT LINE	AID11960
OD64R	48BB 0002	LH	R11,2(R11)	MESSAGE	AID11970
OD68R	40B0 0844R	STH	R11,OUTBUF+2	STORE IN PRINT LINE	AID11980
OD6CR	488C 0DEER	LH	MODE,LENTBL(R12)	LENGTH OF TABLE ENTRY	AID11990
OD70R	485C 0DE6R	LH	VALUE,E1TBL(R12)	ADDRESS OF TABLE	AID12000
OD74R	48B5 0000	UT2 LH	R11,0(VALUE)	ADRESS OF SENTRY	AID12010
OD78R	C5B0 FFFF	CLHI	R11,-1	DELETED ?	AID12020
OD7CR	2336	BES	UT3	YES	AID12030
OD7ER	26F6	AIS	TO,6	BUMP LINE INDEX	AID12040
OD80R	40B0 0A96R	STH	R11,OPSAVE	STORE ADDRESS IN WORK AREA	AID12050
OD84R	4120 040ER	BAL	LINKS,XLATEH	TRANSLATE TO ASCII	AID12060
OD88R	0A58 0D8AR	UT3 AHR	VALUE,MODE	BUMP ENTRY BY INDEX	AID12070
OD8CR	203C	SIS	R0,1	DECREMENT ENTRY COUNTER	AID12080
OD8ER	4120 08C4R	BNZS	UT2		AID12090
OD92R	27C2	BAL	LINKS,LIST	PRINT LINE	AID12100
OD94R	4310 0D52R	SIS	R12,2	DECREMENT LIST COUNTER	AID12110
OD98R	24D1	BNM	UT1	CONTINUE	AID12120
OD9AR	C8F0 0851R	LIS	NUMBER,1	SET TRANSLATION LENGTH	AID12130
OD9ER	48B0 0A46R	LHI	TO,OUTBUF+15	PRINTLINE INDEX	AID12140
ODA2R	2336	LH	R11,PROR0	CHECK REG ZERO	AID12141
ODA4R	C8B0 0030	BZS	UT3A	NOT PROTECTED	AID12142
ODA8R	D2BF 0000	LHI	R11,C'0'	ZERO	AID12143
ODACR	26F6	STB	R11,0(T0)	STORE IN PRINTLINE	AID12144
ODAER	48B0 110AR	UT3A AIS	TO,6	BUMP INDEX	AID12145
ODB2R	07CC	LH	R11,PROREG-2	REG POSITIONAL COUNTER	AID12150
ODB4R	240F	XHR	R12,R12	CLEAR REG COUNTER	AID12160
ODB6R	90B1	LIS	R0,15	COUNTER	AID12170
ODB8R	238B	UT4 SRLS	R11,1	SHIFT OUT A POSITIONAL BIT	AID12180
ODBAR	08AC	BNCS	UT7	NOT SET	AID12190
ODBCR	C6A0 0030	LHR	R10,R12		AID12200
ODCOR	C5A0 003A	OHI	R10,C'0'	ASCII-IZE	AID12210
ODC4R	2182	CLHI	R10,X'3A'	TEN OR BETTER ?	AID12220
ODC6R	26A7	BLS	UT6	NO	AID12230
ODC8R	D2AF	UT6 AIS	R10,7	SET FOR ASCII TEN OR BETTER	AID12240
		STB	R10,0(T0)	STORE IN PRINT LINE	AID12250

0000						
0DCCR 26F6		AIS	T0,6	BUMP INDEX		AID12251
0DCER 26C1	UT7	AIS	R12,1	BUMP REG COUNTER		AID12260
0DD0R 2701		SIS	R0,1	DECREMENT REG COUNTER		AID12270
0DD2R 2112		BMS	UT5	DONE		AID12280
0DD4R 220F		BS	UT4	CONTINUE		AID12290
0DD6R 4120	UT5	BAL	LINKS,LIST			AID12300
08C4R						
0DDAR 4300		B	LDREG	RETURN		AID12310
0046R						
0DDER 0A90R	MSGTBL	DC	PROMSG			AID12320
0DE0R 0A8CR		DC	SNPMSG			AID12330
0DE2R 134CR		DC	TRCMSG			AID12340
0DE4R 1224R		DC	BRKMSG			AID12350
0DE6R 112ER	E1TBL	DC	PROCEL			AID12360
0DE8R 10AAR		DC	E1E4			AID12370
0DEAR 1056R		DC	E1E0			AID12380
0DECR 1088R		DC	E1E2			AID12390
0DEER 0004	LENTBL	DC	4			AID12400
0DF0R 000C		DC	12			AID12410
0DF2R 0006		DC	6			AID12420
0DF4R 0004		DC	4			AID12430
*-----*						
* EXECUTE - PROCESS INTERPRETIVELY						
0DF6R 4020	EX	STH	LINKS,EXIT+2	SAVE RETURN LINK		AID12450
0FB4R						AID12460
0DFAR 40C0		STH	R12,OPCELL	SAVE OPEN CELL		AID12470
0FC2R						
0DFER C800		LHI	R0,EX6A	SVC 13 RETURN ADDRESS		AID12480
0F9AR						
0E02R 4000		STH	R0,X'B6'			AID12490
00B6						
0E06R 240F		LIS	R0,15	SET FLAG		AID12500
0E08R 2474		LIS	LENGTH,4	PRESET INSTRUCTION LENGTH		AID12510
0E0AR D1E4		LM	FROM,0(OPEN)	GET INSTRUCTION		AID12520
0000						
0E0ER 94CE		EXBR	R12,FROM	OP CODE		AID12530
0E10R C3C0		THI	R12,X'40'	LONG INSTRUCTION?		AID12540
0040						
0E14R 2135		RNZS	X1	YES		AID12550
0E16R 2772		SIS	LENGTH,2	SET SIZE FOR SHORT INSTRUCTION		AID12560
0E18R C8F0		LHI	T0,X'200'	NO - NO-OP NEXT HALFWORD		AID12570
0200						
0E1CR 0700		XHR	R0,R0			AID12580
0E1ER D0E0	X1	STM	FROM,EXBLK	STORE INSTR IN EXECUTION BLOCK		AID12590
0FB6R						
0E22R C8B0		LHI	R11,OPSTOP-OPCODE	VOLATILE OP CODES		AID12600
000D						
0E26R D4CB	EX2	CLB	R12,OPCODE(R11)	CHECK LIST		AID12610
0FDER						
0E2AR 2335		BES	EX2B	FOUND		AID12620
0E2CR 27B1		SIS	R11,1	DECREMENT INDEX		AID12630
0E2ER 4210		BM	EX7	DONE - NO SIMULATION		AID12640
0EA0R						
0E32R 2206		BS	EX2	CONTINUE		AID12650
0E34R 0800	EX2B	LHR	R0,R0	CHECK FLAG		AID12660

0E36R	4230		BNZ	EX3	SET - LONG INSTRUCTION	AID12670
	0E80R					
0E3AR	C5B0		CLHI	R11,4	CHECK INDEX FOR SHORT BRANCHES	AID12680
	0004					
0E3ER	4380		BNL	EX4	NO - RR TYPE	AID12690
	0EC6R					
0E42R	C8B0		LHI	R11,X'4200'	BLANK BRANCH FORM	AID12700
	4200					
0E46R	C4C0		NHI	R12,X'0F'		AID12710
	000F					
0E4AR	27C2		SIS	R12,2	BRANCH TRUE OR FALSE ?	AID12720
0E4CR	2113		BMS	EX2A		AID12730
0E4ER	C6B0		OHI	R11,X'100'	FALSE BRANCH	AID12740
	0100					
0E52R	93CE	EX2A	LBR	R12,FROM	OP CODE	AID12750
0E54R	C4C0		NHI	R12,X'F0'	PICK UP CONDITION CODE MASK	AID12760
	00F0					
0E58R	06BC		OHR	R11,R12	SET BLANK'S MASK	AID12770
0E5AR	40B0		STH	R11,EXBLK	PLACE INTO EXECUTION BLOCK	AID12780
	0FB6R					
0E5ER	93BE		LBR	R11,FROM	FULL INSTRUCTION	AID12790
0E60R	C4B0		NHI	R11,X'0F'	MASK FOR HALFWORD BRANCH	AID12800
	000F					
0E64R	91B1		SLLS	R11,1	DOUBLE FOR BYTE	AID12810
0E66R	4810		LH	LINK,OPCELL	OPEN CELL ADDRESS	AID12820
	0FC2R					
0E6AR	08CE		LHR	R12,FROM	RESTORE OP CODE	AID12830
0E6CR	C3C0		THI	R12,X'100'	BRANCH FORWARD ?	AID12840
	0100					
0E70R	2333		BZS	EX2C	NOPE	AID12850
0E72R	0A1B		AHR	LINK,R11	INCREMETN P-COUNTER	AID12860
0E74R	2302		BS	EX2C+2		AID12870
0E76R	0B1B	EX2C	SHR	LINK,R11	DECREMENT P-COUNTER	AID12880
0E78R	4010		STH	LINK,EXBLK+2	STORE BRANCH ADDR IN EX BLOCK	AID12890
	0FB8R					
0E7CR	4300		B	EX5	EXECUTE BRANCH	AID12900
	0F2AR					
0E80R	C5B0	EX3	CLHI	R11,10	BTC,BFC OR BAL ?	AID12910
	000A					
0E84R	4280		BL	EX5	YES - EXECUTE	AID12920
	0F2AR					
0E88R	C5B0		CLHI	R11,12	BXLE OR BXH ?	AID12930
	000C					
0E8CR	238A		BNLS	EX7	N/- MUST BE LPSW OR SVC	AID12940
0E8ER	48B0		LH	R11,EXBLK+2	SAVE BRANCH ADDRESS	AID12950
	0FB8R					
0E92R	40B0		STH	R11,EXBLK+8		AID12960
	0FBER					
0E96R	C8B0		LHI	R11,EXBRC1	INSERT AIDS BRANCH ADDRESS	AID12970
	0FC4R					
0E9AR	40B0		STH	R11,EXBLK+2		AID12980
	0FB8R					
0E9ER	240F		LIS	R0,15	SET FLAG	AID12990
0EA0R	C810	EX7	LHI	LINK,EX6	RETURN ADDRESS AFTER EXECUTION	AID13000
	0F94R					
0EA4R	4010		STH	LINK,EXRTN	STORE IN EXECUTION BLOCK	AID13010

0EA8R	0FBCR 4840	LH	OPEN,OPCELL	OPEN CELL ADDRESS	AID13020
0EACR	0FC2R C5E0 E1E0	CLHI	FROM,X'E1E0'	CHECK FOR SENTRY	AID13030
0EB0R	2387	BNLS	XAB	YES	AID13040
0EB2R	0A47	AHR	OPEN,LENGTH	INCREMENT FOR INSTR LENGTH	AID13050
0EB4R	4040	STH	OPEN,OPCELL	SAVE NEW OPEN CELL	AID13060
0EB8R	0700	XHR	R0,R0	REGISTER RESTORE FLAG	AID13070
0EBAR	4300	B	EXCUTE	EXECUTE INSTRUCTION	AID13080
0EBER	0F7CR 4040	XAB	STH	OPEN,G3	ADDRESS FOR RETURN
0EC2R	040AR 4300	B	G10	RETURN	AID13100
0EC6R	03F8R C3C0	EX4	THI	R12,2	BRANCH ON REGISTER ?
0ECAR	0002 4230	BNZ	EX4B	YES	AID13120
0ECER	0F08R 93CE	LBR	R12,FROM	YES - GET REG1	AID13130
0ED0R	C4C0	NHI	R12,X'F0'		AID13140
0ED4R	00F0 90C3	SRLS	R12,3	INDEX-IZE	AID13150
0ED6R	48B0	LH	R11,OPCELL	OPEN CELL ADDRESS	AID13160
0EDAR	0F2CR 0AB7	AHR	R11,LENGTH	ADD INSTRUCTION LENGTH	AID13170
0EDCR	40BC	STH	R11,REGSAV(R12)	STORE IN REG SAVE AREA	AID13180
0EE0R	0A98R 93CE	LBR	R12,FROM	GET REG2	AID13190
0EE2R	D3B0	LB	R11,EXBLK	GET OP CODE	AID13191
0EE6R	0FB6R 27B1	SIS	R11,1	CHECK FOR BALR	AID13192
0EE8R	2133	BNZS	*+6	IF NOT SKIP AROUND	AID13193
0EEAR	40B0	STH	R11,EXBLK+2	ZERO EXECUTION BLOCK	AID13194
0EEER	0FB8R 07BB	XHR	R11,R11	CLEAR REG	AID13200
0EF0R	C4C0	NHI	R12,X'F0'	CHECK REG 2	AID13210
0EF4R	000F 2334	BZS	EX4A	NO INDEX	AID13220
0EF6R	91C1	SLLS	R12,1	INDEX-IZE	AID13230
0EF8R	48BC	LH	R11,REGSAV(R12)	LOAD INDEX	AID13240
0EFCR	0A98R 4AB0	EX4A	AH	R11,EXBLK+2	ADD BASE
0F00R	0FB8R 40B0	STH	R11,OPCELL		AID13260
0F04R	0FC2R 4300	B	EXRTRN	RETURN	AID13270
0F08R	0FA6R D3C0	EX4B	LB	R12,EXBLK+1	GET BRANCH REGISTER
0F0CR	0FB7R C4C0	NHI	R12,X'F0'		AID13272
0F10R	000F 91C1	SLLS	R12,1	INDEXIZE	AID13273
0F12R	48CC	LH	R12,SAVREG(R12)	GET REGISTER CONTENTS	AID13274
	0A98R				

0F16R	40C0		STH	R12,EXBLK+2	STORE AS ADDRESS	AID13275
	0FB8R					
0F1AR	48C0		LH	R12,EXBLK	PICK UP OP CODE ETC.	AID13276
	0FB6R					
0F1ER	C6C0		OHI	R12,X'4000'	CHANGE TO FULL BRANCH	AID13277
	4000					
0F22R	C4C0		NHI	R12,X'FFF0'	REMOVE INDEX REGISTER	AID13278
	FFF0					
0F26R	40C0		STH	R12,EXBLK	INSTRUCTION = FULL BRANCH	AID13279
	0FB6R					
0F2AR	D3C0	EX5	LB	R12,EXBLK	GET OP CODE	AID13280
	0FB6R					
0F2ER	C5C0		CLHI	R12,X'41'	BAL ?	AID13281
	0041					
0F32R	4330		BE	EX4	YES	AID13282
	0EC6R					
0F36R	D3B0		LB	R11,EXBLK+1	CONDITIONAL BRANCH HANDLER	AID13283
	0FB7R					
0F3AR	C4B0		NHI	R11,X'F'	GET REG2	AID13290
	000F					
0F3ER	2334		BZS	EX5A	SKIP REGISTER	AID13300
0F40R	91B1		SLLS	R11,1	INDEX-IZE	AID13310
0F42R	48BB		LH	R11,SAVREG(R11)	LOAD REGISTER CONTENTS	AID13320
	0A98R					
0F46R	4AB0	EX5A	AH	R11,EXBLK+2	ADD BASE ADDRESS	AID13330
	0FB8R					
0F4AR	40B0		STH	R11,EXBLK+8	SAVE USER BRANCH ADDR	AID13340
	0FBER					
0F4ER	C8B0		LHI	R11,EXBRCH	SET AIDS BRANCH INTERCEPT	AID13350
	0FC8R					
0F52R	07CC		XHR	R12,R12	CLEAR REG	AID13360
0F54R	40C0		STH	R12,EXBLK+2	CLEAR BASE	AID13370
	0FB8R					
0F58R	D3C0		LB	R12,EXBLK+1	GET R2	AID13380
	0FB7R					
0F5CR	C4C0		NHI	R12,X'F0'	KEEP R1 - ELIM R2	AID13390
	00F0					
0F60R	C6C0		OHI	R12,X'0B'	SET IN AIDS BRANCH REGISTER	AID13400
	000B					
0F64R	D2C0		STB	R12,EXBLK+1	STORE IN EXECUTION BLOCK	AID13410
	0FB7R					
0F68R	C810		LHI	LINK,EX6+4	SET RETURN	AID13420
	0F98R					
0F6CR	4010		STH	LINK,EXRTN		AID13430
	0FBCR					
0F70R	4840		LH	OPEN,OPCELL	NEW OPEN CELL CALC.	AID13440
	0FC2R					
0F74R	0A47		AHR	OPEN,LENGTH	SET LENGTH	AID13450
0F76R	4040		STH	OPEN,OPCELL	SAVE OPEN CELL (NEW)	AID13460
	0FC2R					
0F7AR	240F		LIS	R0,15	SET NO REG FLAG	AID13470
0F7CR	48C0	EXCUTE	LH	R12,CCSAVE	OLD CONDITION CODE	AID13480
	0FC0R					
0F80R	40C0		STH	R12,EXOUT	SET STATUS / CC	AID13490
	0F90R					
0F84R	0800		LHR	R0,R0	CHECK FOR USER REG RESTORATION	AID13500





0FE0R	D300	ENTER	LB	NUMBER,OPRAND	GET COMMAND	AID13870
	0A78R					
0FF0R	080D		LHR	NUMBER,NUMBER		AID13880
0FF2R	0336		BZR	CMDER	ZERO = COMMAND ERROR	AID13890
0FF4R	C800		LHI	R0,X'50'		AID13891
	0050					
0FF8R	D400		CLB	R0,BUFFER	PARAMETER CHECK	AID13892
	0A4ER					
0FFCR	0286		BLR	CMDER	LESS = COMMAND ERROR	AID13893
0FFER	C8E0		LHI	FROM,OPRAND+1	OPERAND STORAGE ADDR	AID13900
	0A79R					
1002R	C8F0		LHI	TO,OPSAVE		AID13910
	0A96R					
1006R	4000		STH	NUMBER,OPSAVE	CLEAR RECEIVER	AID13920
	0A96R					
100AR	4120		BAL	LINKS,XLATHB	TRANSLATE BEGIN ADDRESS	AID13930
	044CR					
100ER	4850		LH	VALUE,OPSAVE	KEEP IN REGISTER	AID13940
	0A96R					
1012R	0AED		AHR	FROM,NUMBER	BUMP TO NEXT OPERAND	AID13950
1014R	D3DE		LB	NUMBER,0(FROM)	PICK UP LENGTH	AID13960
	0000					
1018R	080D		LHR	NUMBER,NUMBER	NO OPERAND ?	AID13970
101AR	4330		BZ	E1	YES BYPASS	AID13980
	1040R					
101ER	26E1		AIS	FROM,1	OPERAND STORAGE ADDR	AID13990
1020R	40D0		STH	NUMBER,OPSAVE	CLEAR RECEIVER	AID14000
	0A96R					
1024R	4120		BAL	LINKS,XLATHB	TRANSLATE END ADDRESS	AID14010
	044CR					
1028R	4810		LH	LINK,OPSAVE		AID14020
	0A96R					
102CR	0AED		AHR	FROM,NUMBER	BUMP TO NEXT OPERAND	AID14030
102ER	D3DE		LB	NUMBER,0(FROM)	GET LENGTH	AID14040
	0000					
1032R	26E1		AIS	FROM,1		AID14050
1034R	40DF		STH	NUMBER,0(TO)	CLEAR CELL	AID14060
	0000					
1038R	080D		LHR	NUMBER,NUMBER	CHECK FOR DEFAULT	AID14070
103AR	2333		BZS	E1		AID14080
103CR	4120		BAL	LINKS,XLATDB	TRANSLATE NO OF EXECUTIONS	AID14090
	04B2R					
1040R	D300	E1	LB	R0,BUFFER+1	SECONDARY DIRECTIVE	AID14100
	0A4FR					
1044R	08F1		LHR	TO,LINK	END ADDRESS	AID14110
1046R	08E5		LHR	FROM,VALUE	DUMP TYPE	AID14120
1048R	D350		LB	VALUE,BUFFER+2	TYPE DUMP	AID14130
	0A50R					
104CR	4120		BAL	LINKS,TABC	GO TO HANDLER	AID14140
	096AR					
1050R	4300		B	LDREG		AID14150
	0046R					
*-----*						AID14160
* SENTRY POINT TABLES						AID14170
1054R	0000		DC	0		AID14180
1056R		E1E0	EQU	*		AID14190

1056R		DO	8		AID14200
1056R	FFFF	DC	-1,-1,-1		AID14210
	FFFF				
	FFFF				
105CR	FFFF	DC	-1,-1,-1		AID14210
	FFFF				
1062R	FFFF	DC	-1,-1,-1		AID14210
	FFFF				
1068R	FFFF	DC	-1,-1,-1		AID14210
	FFFF				
106ER	FFFF	DC	-1,-1,-1		AID14210
	FFFF				
1074R	FFFF	DC	-1,-1,-1		AID14210
	FFFF				
107AR	FFFF	DC	-1,-1,-1		AID14210
	FFFF				
1080R	FFFF	DC	-1,-1,-1		AID14210
	FFFF				
	FFFF				
1086R		E1E0N EQU	*		AID14220
1086R	0000	DC	0		AID14230
1088R		E1E2 EQU	*		AID14240
1088R		DO	8		AID14250
1088R	FFFF	DC	-1,-1		AID14260
	FFFF				
108CR	FFFF	DC	-1,-1		AID14260
	FFFF				
1090R	FFFF	DC	-1,-1		AID14260
	FFFF				
1094R	FFFF	DC	-1,-1		AID14260
	FFFF				
1098R	FFFF	DC	-1,-1		AID14260
	FFFF				
109CR	FFFF	DC	-1,-1		AID14260
	FFFF				
10A0R	FFFF	DC	-1,-1		AID14260
	FFFF				
10A4R	FFFF	DC	-1,-1		AID14260
	FFFF				
10A8R		E1E2N EQU	*		AID14270
10A8R	0000	DC	0		AID14280
10AAR		E1E4 EQU	*		AID14290
10AAR		DO	8		AID14300
10AAR	FFFF	DC	-1,-1,-1,-1,-1,-1		AID14310
	FFFF				
	FFFF				
	FFFF				
	FFFF				
10B6R	FFFF	DC	-1,-1,-1,-1,-1,-1		AID14310

	FFFF				
	FFFF				
	FFFF				
	FFFF				
10C2R	FFFF	DC	-1,-1,-1,-1,-1,-1		AID14310
	FFFF				
	FFFF				
	FFFF				
10CER	FFFF	DC	-1,-1,-1,-1,-1,-1		AID14310
	FFFF				
	FFFF				
	FFFF				
10DAR	FFFF	DC	-1,-1,-1,-1,-1,-1		AID14310
	FFFF				
	FFFF				
	FFFF				
10E6R	FFFF	DC	-1,-1,-1,-1,-1,-1		AID14310
	FFFF				
	FFFF				
	FFFF				
10F2R	FFFF	DC	-1,-1,-1,-1,-1,-1		AID14310
	FFFF				
	FFFF				
	FFFF				
10FER	FFFF	DC	-1,-1,-1,-1,-1,-1		AID14310
	FFFF				
	FFFF				
	FFFF				
	FFFF				
110AR		E1E4N	EQU	*	AID14320
110AR	0000		DC	0	AID14330
110CR		PROREG	EQU	*	AID14340
110CR			00	16	AID14350
110CR	0000		DC	0	AID14360
110ER	0000		DC	0	AID14360
1110R	0000		DC	0	AID14360
1112R	0000		DC	0	AID14360
1114R	0000		DC	0	AID14360
1116R	0000		DC	0	AID14360
1118R	0000		DC	0	AID14360
111AR	0000		DC	0	AID14360
111CR	0000		DC	0	AID14360
111ER	0000		DC	0	AID14360
1120R	0000		DC	0	AID14360

1122R	0000	DC	0		AID14360
1124R	0000	DC	0		AID14360
1126R	0000	DC	0		AID14360
1128R	0000	DC	0		AID14360
112AR	0000	DC	0		AID14360
112CR		PRORGN EQU	*		AID14370
112CR	0000	DC	0		AID14380
112ER		PROCEL EQU	*		AID14390
112ER		DO	8		AID14400
112ER	FFFF	DC	-1,-1		AID14410
	FFFF				
1132R	FFFF	DC	-1,-1		AID14410
	FFFF				
1136R	FFFF	DC	-1,-1		AID14410
	FFFF				
113AR	FFFF	DC	-1,-1		AID14410
	FFFF				
113ER	FFFF	DC	-1,-1		AID14410
	FFFF				
1142R	FFFF	DC	-1,-1		AID14410
	FFFF				
1146R	FFFF	DC	-1,-1		AID14410
	FFFF				
114AR	FFFF	DC	-1,-1		AID14410
	FFFF				
114ER		PROCLN EQU	*		AID14420
114ER	0000	BUFFLG DC	0		AID14430
*-----*					
* SUPERVISOR CALL (14) HANDLER					
1150R	D000	SVC14 STM	R0,SAVREG	SAVE USER REGS	AID14440
	0A98R				AID14460
1154R	D1E0	LM	FROM,X'96'		AID14470
	0096				
1158R	C4E0	NHI	FROM,X'7FFF'	RESET WAIT BIT	AID14480
	7FFF				
115CR	40E0	STH	FROM,CCSAVE	SAVE CONDITION CODE	AID14490
	0FC0R				
1160R	27F4	SIS	TO,4	BACK UP TO SVC	AID14500
1162R	084F	LHR	OPEN,TO	SVC ADDRESS	AID14510
1164R	08CF	LHR	R12,TO	SAVE SVC ADDR	AID14520
1166R	40C0	STH	R12,OPSAVE	SET UP FOR TRANSLATION	AID14530
	0A96R				
116AR	C8A0	LHI	R10,ILLEG		AID14540
	0914R				
116ER	48BF	LH	R11,0(TO)	PICK UP INSTR	AID14550
	0000				
1172R	C4B0	NHI	R11,15	MASK FOR X REGISTER	AID14560
	000F				
1176R	C8E0	SVC14A LHI	FROM,OPSAVE	OPEN CELL ADDRESS	AID14570
	0A96R				
117AR	C8F0	LHI	TO,OUTBUF+7	SET PRINT LINE	AID14580
	0849R				
117ER	24D2	LIS	NUMBER,2		AID14590
1180R	4120	BAL	LINKS,XLATEH	TRANSLATE ADDRESS TO ASCII	AID14600
	040ER				
1184R	482B	LH	LINKS,SVCTAB(R11)	PICK UP ADDRESS	AID14610

118AR	1188R 0302	BR	LINKS	BRANCH TO HANDLER	AID14620
	118AR 12CAR	SVCTAB DC	SVCE0		AID14630
	118CR 11EER	DC	SVCE2		AID14640
	118ER 13EER	DC	SVCE4		AID14650
*-----*					
		* ZAP (Z)			AID14660
	1190R D300	ZAP LB	R0,BUFFER+1	GET SECOND DIRECTIVE	AID14670
	0A4FR				AID14680
	1194R D3D0	LB	NUMBER,OPRAND	GET OPERAND LENGTH	AID14690
	0A78R				
	1198R 08DD	LHR	NUMBER,NUMBER	NO OPERAND ?	AID14700
	119AR 2339	BZS	Z1	YES - DELETE ALL	AID14710
	119CR C8E0	LHI	FROM,OPRAND+1	FROM ADDRESS	AID14720
	0A79R				
	11A0R C8F0	LHI	TO,OPSAVE	TO ADDRESS	AID14730
	0A96R				
	11A4R 4120	BAL	LINKS,XLATHB	TRANSLATE TO BINARY	AID14740
	044CR				
	11A8R 48E0	LH	FROM,OPSAVE		AID14750
	0A96R				
	11ACR 4120	Z1 BAL	LINKS,TABD	GO TO HANDLER	AID14760
	099ER				
	11B0R 4300	B	NAME		AID14770
	004AR				
*-----*					
		* BREAKPOINT (B) INSERTION HANDLER			AID14780
	11B4R 48B0	BREAK LH	R11,E1E2-2	ENTRY COUNTER	AID14790
	1086R				AID14800
	11B8R C5B0	CLHI	R11,0	LIST FULL ?	AID14810
	0008				
	11BCR 0386	BNLR	CMDR	YES - COMMAND ERROR	AID14820
	11BER 07BB	XHR	R11,R11	RESET INDEX	AID14830
	11C0R 48CB	BR1 LH	R12,E1E2(R11)	LOAD ADDRESS	AID14840
	1088R				
	11C4R C5C0	CLHI	R12,-1	ADDRESS DELETED ?	AID14850
	FFFF				
	11C8R 2333	BES	BR2	YES	AID14860
	11CAR 26B4	AIS	R11,4	NO - INCREMENT INDEX	AID14870
	11CCR 2206	BS	BR1	CONTINUE	AID14880
	11CER 40EB	BR2 STH	FROM,E1E2(R11)	STORE SENTRY ADDRESS	AID14890
	1088R				
	11D2R D3CE	LB	R12,0(FROM)	GET OP CODE	AID14900
	0000				
	11D6R 48CE	LH	R12,0(FROM)	GET INSTRUCTION	AID14910
	0000				
	11DAR 40CB	STH	R12,E1E2+2(R11)	STORE INSTRUCTION	AID14920
	108AR				
	11DER C8C0	LHI	R12,X'E1E2'	STORE SENTRY	AID14930
	E1E2				
	11E2R 40CE	STH	R12,0(FROM)		AID14940
	0000				
	11E6R 24C1	LIS	R12,1		AID14950
	11E8R 61C0	AHM	R12,E1E2-2	INCREMENT ENTRY COUNTER	AID14960
	1086R				
	11EER 0302	BR	LINKS	RETURN	AID14970

```

*-----*
*          BREAK EXECUTION HANDLER SVC 14.2
11EER C8B0 SVCE2 LHI R11,E1E2N-E1E2 LIST LENGTH FOR INDEX AID14980
    0020 AID14990
11F2R 45CB SVCE21 CLH R12,E1E2-4(R11) COMPARE ADDR AID15000
    1084R
11F6R 2334 BES SVCE22 HIT AID15020
11F8R 27B4 SIS R11,4 DECREMENT INDEX AID15030
11FAR 033A BZR R10 NOT FOUND = ILLEGAL INSTR AID15040
11FCR 2205 BS SVCE21 CONTINUE AID15050
11FER 48BB SVCE22 LH R11,E1E2-2(R11) PICK UP INSTRUCTION AID15060
    1086R
1202R 40B0 STH R11,DBLWK STORE IN WORK AREA AID15070
    0AB8R
1206R 48BC LH R11,2(R12) SECOND HALFWORD AID15080
    0002
120AR 40B0 STH R11,DBLWK+2 STORE IN WORK AREA AID15090
    0ABAR
120ER 08C4 LHR R12,OPEN SAVE OPEN CELL ADDR AID15100
1210R C840 LHI OPEN,DBLWK AID15110
    0AB8R
1214R 4120 BAL LINKS,EX EXECUTE INSTR AID15120
    0DF6R
1218R D1E0 LM FROM,BRKMSG AID15140
    1224R
121CR D0E0 STM FROM,OUTBUF+2 BREAK MESSAGE AID15150
    0844R
1220R 0129 BALR LINKS,OUTPUT PRINT MESSAGE AID15160
1222R 030A BR INPUT GET DIRECTIVE AID15170
1224R 4252 BRKMSG DC C:BRK: AID15180
    4B3A
*-----*
*          ZAP BREAKPOINT HANDLER
1228R C8B0 ZAPBRK LHI R11,E1E2N-E1E2 SET INDEX TO LIST SIZE AID15190
    0020 AID15200
122CR C8C0 LHI R12,-1 CLEARING VALUE AID15210
    FFFF
1230R 08DD LHR NUMBER,NUMBER SPECIFIC BRK OR ALL ? AID15230
1232R 4330 BZ ZB3 ALL AID15240
    1258R
1236R 45EB ZB1 CLH FROM,E1E2-4(R11) COMPARE ADDR AID15250
    1084R
123AR 2334 BES ZB2 HIT - DELETE AID15260
123CR 27B4 SIS R11,4 CONTINUE AID15270
123ER 0216 BMR CMDER NOT FOUND = ERROR AID15280
1240R 2205 BS ZB1 CONTINUE AID15290
1242R D1EB ZB2 LM FROM,E1E2-4(R11) AID15300
    1084R
1246R 05EC CLHR FROM,R12 DELETED ENTRY ? AID15310
1248R 0332 BER LINKS IGNORE AID15320
124AR 40FE STH TO,0(FROM) RESTORE INSTRUCTION AID15330
    0000
124ER 40CB STH R12,E1E2-4(R11) CLEAR ENTRY AID15340
    1084R
1252R 61C0 AHM R12,E1E2-2 DECREMENT ENTRY COUNTER AID15350
    1086R

```

1256R	0302	BR	LINKS	RETURN	AID15360
1258R	0812	ZB3	LHR	LINK, LINKS	AID15370
125AR	C820	LHI	LINKS, ZB4	SAVE RETURN	AID15380
	1260R				
125ER	220E	BS	ZB2	DELETE AN ENTRY	AID15390
1260R	27B4	SIS	R11,4	DECREMENT INDEX	AID15400
1262R	2032	BNZS	ZB4-2	CONTINUE DELETING	AID15410
1264R	0301	BR	LINK	RETURN	AID15420
					AID15430
					AID15440
					AID15450
1266R	48B0	TRACE	TRACE (T) INSERTION	HANDLER	
	1054R	LH	R11, E1E0-2	NUMBER OF ENTRIES	
126AR	C5B0	CLHI	R11,8	VACANCY IN TABLE ?	AID15460
	0008				
126ER	0386	BNLR	CMDER	NO - COMMAND ERROR	AID15470
1270R	0755	XHR	VALUE, VALUE		AID15480
1272R	48C5	TR1	LH	R12, E1E0 (VALUE) GET BEGIN ADDR	AID15490
	1056R				
1276R	C5C0	CLHI	R12, -1	CLEARED ?	AID15500
	FFFF				
127AR	2336	BES	TR2	YES	AID15510
127CR	05EC	CLHR	FROM, R12	FROM LESS THAN BEGIN ?	AID15520
127ER	2188	BLS	TR3	YES	AID15530
1280R	45E5	CLH	FROM, E1E0+2 (VALUE)	NO - TO GREATER THAN END ?	AID15540
	1058R				
1284R	0286	BLR	CMDER	NO - COMMAND ERROR - NESTING	AID15550
1286R	2656	TR2	AIS	VALUE, 6	AID15560
1288R	27B1	SIS	R11, 1	BUMP INDEX	AID15570
128AR	2115	BMS	TR4	DECREMENT COUNTER	AID15580
128CR	2200	BS	TR1	CHECKING COMPLETE	AID15590
128ER	05FC	TR3	CLHR	TO, R12	AID15600
1290R	0386	BNLR	CMDER	'TO' LESS THAN END ?	AID15610
1292R	2206	BS	TR2	NO - COMMAND ERROR - NESTING	AID15620
1294R	05EF	TR4	CLHR	FROM, TO	AID15630
1296R	0386	BNLR	CMDER	CONTINUE CHECK	AID15640
1298R	2401	LIS	R0, 1	CHECK EXTENTS	AID15650
129AR	6100	AHM	R0, E1E0-2	INCREMENT COUNT	AID15660
	1054R				
129ER	C8B0	LHI	R11, E1E0	ADDRESS OF SVC 14,0 LIST	AID15670
	1056R				
12A2R	C8C0	LHI	R12, -1	EMPTY ENTRY MASK	AID15680
	FFFF				
12A6R	45CB	TR5	CLH	R12, 0 (R11)	EMPTY ?
	0000				AID15690
12AAR	2333	BES	TR6	YES	AID15700
12ACR	26B6	AIS	R11, 6	NO - BUMP INDEX	AID15710
12AER	2204	BS	TR5	CONTINUE	AID15720
12B0R	D3CE	TR6	LB	R12, 0 (FROM)	GET OP CODE
	0000				AID15730
12B4R	48CE	LH	R12, 0 (FROM)	PULL INSTRUCTION HALFWORD OUT	AID15740
	0000				
12B8R	D0EB	STM	FROM, 0 (R11)	INSERT EXTENTS	AID15750
	0000				
12BCR	40CB	STH	R12, 4 (R11)	SAVE IN TABLE ENTRY	AID15760
	0004				
12C0R	C8B0	LHI	R11, X*E1E0'	SVC 14, 0	AID15770

12C4R	E1E0 40BE 0000	STH	R11,0(FROM)	STORE IN CORE	AID15780
12C8R	0302	BR	LINKS	RETURN	AID15790
*-----*					
* TRACE EXECUTION HANDLER SVC 14,0					
12CAR	C850 0030	SVCE0	LHI	VALUE,E1E0N-E1E0 LIST LENGTH FOR INDEX	AID15800 AID15810 AID15820
12CER	45C5 1050R	SVCE01	CLH	R12,E1E0-6(VALUE) COMPARE ADDRESSES	AID15830
12D2R	2334	BES	SVCE02	FOUND	AID15840
12D4R	2756	SIS	VALUE,6	DECREMENT INDEX	AID15850
12D6R	033A	BZR	R10	NOT FOUND = ILLEGAL	AID15860
12D8R	2205	BS	SVCE01	CONTINUE	AID15870
12DAR	D1D5 1050R	SVCE02	LM	NUMBER,E1E0-6(VALUE)	AID15880
12DER	40F0 0AB8R	STH	TO,DBLWK	SET INSTR IN WORK AREA	AID15890
12E2R	48F4 0002	LH	TO,2(OPEN)	GET SECOND HALFWORD	AID15900
12E6R	40F0 0ABAR	STH	TO,DBLWK+2	STORE IN WORK AREA	AID15910
12EAR	08C4	LHR	R12,OPEN	SAVE OPEN CELL ADDR	AID15920
12ECR	40C0 0A94R	STH	R12,TRCFLG	SET TRACE FLAG	AID15930
12FOR	C840 0AB8R	LHI	OPEN,DBLWK		AID15940
12F4R	4050 1350R	STH	VALUE,TRCSAV	SAVE INDEX	AID15950
12F8R	4120 0DF6R	BAL	LINKS,EX	EXECUTE INSTR	AID15960
12FCR	D1E0 134CR	LM	FROM,TRCMG	TRACE MESSAGE	AID15970
1300R	D0E0 0844R	STM	FROM,OUTBUF+2		AID15980
1304R	4120 08C4R	BAL	LINKS,LIST	PRINT TRACE AND ADDRESS	AID15990
1308R	4850 1350R	SVCE03	LH	VALUE,TRCSAV RESTORE INDEX	AID16000
130CR	4545 1052R	CLH	OPEN,E1E0-4(VALUE) END OF TRACING ?		AID16010
1310R	2337	BES	TRC3	IF EQUAL CONTINUE	AID16020
1312R	4380 133ER	BNL	SVCE05	YES	AID16030
1316R	4545 1050R	CLH	OPEN,E1E0-6(VALUE) OUT OF BOUNDS ?		AID16040
131AR	4280 133ER	BL	SVCE05	YES	AID16050
131ER	4040 0A96R	TRC3	STH	OPEN,OPSAVE	AID16060
1322R	C8E0 0A96R	LHI	FROM,OPSAVE	WORK ADDRESS	AID16070
1326R	C8F0 0849R	LHI	TO,OUTBUF+7	PRINT AREA	AID16080
132AR	24D2	LIS	NUMBER,2	BYTE COUNT	AID16090
132CR	4120	BAL	LINKS,XLATEH	TRANSLATE TO ASCII HEX	AID16100



040ER					
1330R	4120	BAL	LINKS,LIST	PRINT ADDR	AID16110
	08C4R				
1334R	C820	SVCE04	LHI	LINKS,SVCE03	SET RETURN ADDRESS
	1308R				AID16130
1338R	08C4	LHR	R12,OPEN	SAVE OPEN CELL ADDRESS	AID16140
133AR	4300	B	EX	EXECUTE INSTR	AID16150
	00F6R				
133ER	0700	SVCE05	XHR	R0,R0	
1340R	4000	STH	R0,TRCFLG	CLEAR TRACE FLAG	AID16160
	0A94R				AID16170
1344R	4040	SVCE06	STH	OPEN,G3	SET RETURN ADDR
	040AR				AID16180
1348R	4300	B	G1B	EXIT	AID16190
	0350R				
134CR	5452	TRCMSG	DC	C*TRC:'	AID16200
	433A				
1350R	0000	TRCSAV	DC	0	AID16210
*-----*					
* ZAP TRACE HANDLER					
1352R	C8B0	ZAPTRC	LHI	R11,E1E0N-E1E0	LIST LENGTH FOR INDEX
	0030				AID16220
1356R	C8C0	LHI	R12,-1		AID16230
	FFFF				AID16240
135AR	0800	LHR	NUMBER,NUMBER	CLEAR TRACE OR ALL ?	AID16260
135CR	4330	BZ	ZT3	ALL	AID16270
	1382R				
1360R	45EB	ZT1	CLH	FROM,E1E0-6(R11)	COMPARE ADDR
	1050R				AID16280
1364R	2334	BES	ZT2	HIT - DELETE	AID16290
1366R	27B6	SIS	R11,6	DECREMENT INDEX	AID16300
1368R	0216	BMR	CMDER	NOT FOUND = ERROR	AID16310
136AR	2205	BS	ZT1	CONTINUE	AID16320
136CR	D1DB	ZT2	LM	NUMBER,E1E0-6(R11)	ADDR AND INSTRUCTION
	1050R				AID16330
1370R	05DC	CLHR	NUMBER,R12	DELETED ENTRY ?	AID16340
1372R	0332	BER	LINKS	YES - IGNORE	AID16350
1374R	40FD	STH	TO,0(NUMBER)	REPLACE INSTR	AID16360
	0000				
1378R	40CB	STH	R12,E1E0-6(R11)	DELETE ADDR	AID16370
	1050R				
137CR	61C0	AHM	R12,E1E0-2	DECREMENT ENTRY COUNTER	AID16380
	1054R				
1380R	0302	BR	LINKS	RETURN	AID16390
1382R	0812	ZT3	LHR	LINK,LINKS	SAVE RETURN
1384R	C820	LHI	LINKS,ZT4	SET RETURN	AID16400
	138AR				AID16410
1388R	220E	BS	ZT2		AID16420
138AR	27B6	SIS	R11,6	DECREMENT INDEX	AID16430
138CR	2032	BNZS	ZT4-2	CONTINUE	AID16440
138ER	0301	BR	LINK	RETURN	AID16450
*-----*					
* SNAPSHOT (S) INSERTION HANDLER					
1390R	48B0	SNAP	LH	R11,E1E4-2	ENTRY COUNTER
	10A8R				AID16460
1394R	C5B0	CLHI	R11,8	CHECK FOR VACANCY	AID16470
					AID16480
					AID16490

0008					
1398R 0386		BNLR	CMDR	FULL = ERROR	AID16500
139AR 078B		XHR	R11,R11	CLEAR INDEX	AID16510
139CR 48CB	SN1	LH	R12,E1E4(R11)	GET ENTRY	AID16520
10AAR					
13A0R C5C0		CLHI	R12,-1	DELETED ?	AID16530
FFFF					
13A4R 2333		BES	SN2	YES	AID16540
13A6R 26BC		AIS	R11,12	BUMP INDEX	AID16550
13A8R 2206		BS	SN1	CONTINUE	AID16560
13AAR 0814	SN2	LHR	LINK,OPEN	RELATIVE ADDRESS	AID16570
13ACR 0A13		AHR	LINK,BIAS	ABSOLUTE ADDRESS	AID16580
13AER 05EF		CLHR	FROM,TO	EXTENT CHECK	AID16590
13B0R 0386		BNLR	CMDR		AID16600
13B2R C550		CLHI	VALUE,X'20'	CHECK DUMP TYPE	AID16601
0020					
13B6R 0336		BER	CMDR	ERROR	AID16602
13B8R D3C1		LB	R12,0(LINK)	GET OP CODE	AID16610
0000					
13BCR 401B		STH	LINK,E1E4(R11)	STORE IN TABLE	AID16620
10AAR					
13C0R 48C1		LH	R12,0(LINK)	GET INSTRUCTION	AID16630
0000					
13C4R 40CB		STH	R12,E1E4+2(R11)	STORE IN TABLE	AID16640
10ACR					
13C8R D25B		STB	VALUE,E1E4+5(R11)	STORE DUMP TYPE IN TABLE	AID16650
10AFR					
13CCR D3C0		LB	R12,OPSAVE+1	NO. OF EXECUTIONS BETWEEN DUMPS	AID16660
0A97R					
13D0R 08CC		LHR	R12,R12	CHECK FOR ZERO	AID16670
13D2R 2332		BZS	SN3	SKIP IF ZERO	AID16680
13D4R 27C1		SIS	R12,1	ZERO RELATIV-IZE	AID16690
13D6R 40CB	SN3	STH	R12,E1E4+6(R11)	STORE IN TABLE	AID16700
10B0R					
13DAR D0EB		STM	FROM,E1E4+8(R11)	STORE DUMP EXTENTS	AID16710
10B2R					
13DER 24B1		LIS	R11,1		AID16720
13E0R 61B0		AHM	R11,E1E4-2	INCREMENT TABLE COUNTER	AID16730
10A8R					
13E4R C8B0		LHI	R11,X'E1E4'		AID16740
E1E4					
13E8R 40B1		STH	R11,0(LINK)	STORE SENTINEL IN MEMORY	AID16750
0000					
13ECR 0302		BR	LINKS	RETURN	AID16760
*-----*					
* SNAPSHOT EXECUTION HANDLER SVC 14,4					
13EER C850	SVCE4	LHI	VALUE,E1E4N-E1E4	LIST LENGTH	AID16770
0060					
13F2R 45C5	SVCE41	CLH	R12,E1E4-12(VALUE)	COMPARE ADDR WITH LIST ENTRY	AID16800
109ER					
13F6R 2334		BES	SVCE42	FOUND	AID16810
13F8R 275C		SIS	VALUE,12	DECREMENT INDEX	AID16820
13FAR 033A		BZR	R10	NOT FOUND = ILLEGAL INSTR	AID16830
13FCR 2205		BS	SVCE41		AID16840
13FER 48C5	SVCE42	LH	R12,E1E4-10(VALUE)	PICK UP INSTRUCTION	AID16850
10A0R					

1402R	40C0		STH	R12,DBLWK	STORE IN WORK AREA	AID16860
	0AB8R					
1406R	48C4		LH	R12,2(OPEN)	PICK UP SECOND HALFWORD	AID16870
	0002					
140AR	40C0		STH	R12,DBLWK+2	STORE IN WORK AREA	AID16880
	0ABAR					
140ER	08C4		LHR	R12,OPEN	SAVE OPEN CELL ADDR	AID16890
1410R	C840		LHI	OPEN,DBLWK	SET OPEN CELL TO WORK AREA	AID16900
	0AB8R					
1414R	4050		STH	VALUE,SAVE	SAVE INDEX	AID16910
	0B00R					
1418R	4120		BAL	LINKS,EX	EXECUTE INSTRUCTION	AID16920
	0DF6R					
141CR	4850		LH	VALUE,SAVE	RESTORE INDEX	AID16930
	0B00R					
1420R	D1C5		LM	R12,E1E4-8(VALUE)	PICK UP ENTRY INFO	AID16940
	10A2R					
1424R	930D		LBR	R0,NUMBER	NO	AID16950
1426R	90D8		SRLS	NUMBER,8	POSITION CURRENT EXECUTIONS	AID16960
1428R	090D		CHR	R0,NUMBER	COMPARE TO EXECUTIONS BETWEEN	AID16970
142AR	2336		BES	SVCE43	EQUAL - SNAP DUMP	AID16980
142CR	26D1		AIS	NUMBER,1	INCREMENT EXECUTION COUNT	AID16990
142ER	D2D5		STB	NUMBER,E1E4-6(VALUE)	STORE IN ENTRY	AID17000
	10A4R					
1432R	4300		B	SVCE06	RETURN	AID17010
	1344R					
1436R	0700	SVCE43	XHR	R0,R0		AID17020
1438R	D2D5		STB	R0,E1E4-6(VALUE)	CLEAR CURRENT EXECUTIONS	AID17030
	10A4R					
143CR	080E		LHR	R0,FROM	SAVE REG	AID17040
143ER	085F		LHR	VALUE,TO	END ADDRESS	AID17050
1440R	D1E0		LM	FROM,SNPMSG	SNAP MESSAGE	AID17060
	0A8CR					
1444R	D0E0		STM	FROM,OUTRUF+2		AID17070
	0844R					
1448R	08E0		LHR	FROM,R0	RESTORE REG	AID17080
144AR	C8F0		LHI	TO,OUTBUF	SET PRINT INDEX	AID17090
	0842R					
144ER	4120		BAL	LINKS,LIST	PRINT SNAP MESSAGE	AID17100
	08C4R					
1452R	930C		LBR	R0,R12	GET DUMP MODE	AID17110
1454R	4120		BAL	LINKS,TABC	EXECUTE DUMP	AID17130
	096AR					
1458R	4300		B	SVCE06	EXIT	AID17140
	1344R					
*-----*						AID17150
* ZAP SNABPSHOT HANDLER						AID17160
145CR	C8B0	ZAPSNP	LHI	R11,E1E4N-E1E4	SET INDEX TO LIST SIZE	AID17170
	0060					
1460R	C8C0		LHI	R12,-1	CLEARING VALUE	AID17180
	FFFF					
1464R	08DD		LHR	NUMBER,NUMBER	SPECIFIC SNAP OR ALL?	AID17190
1466R	4330		BZ	ZS3	ALL	AID17200
	148CR					
146AR	45EB	ZS1	CLH	FROM,E1E4-12(R11)	COMPARE ADDR	AID17210
	109ER					

146ER	2334	BES	ZS2	HIT - DELETE	AID17220	
1470R	27BC	SIS	R11,12	DECREMENT	AID17230	
1472R	0216	BMR	CMDER	NOT FOUND = ERROR	AID17240	
1474R	2205	BS	ZS1	CONTINUE	AID17250	
1476R	D1EB	ZS2	LM	FROM,E1E4-12(R11) ADDR AND INSTRUCTION	AID17260	
	109ER					
147AR	05EC	CLHR	FROM,R12	DELETED ENTRY ?	AID17270	
147CR	0332	BER	LINKS	YES - IGNORE	AID17280	
147ER	40FE	STH	TO,0(FROM)	REPLACE INSTRUCTION	AID17290	
	0000					
1482R	40CB	STH	R12,E1E4-12(R11)	DELETE ADDRESS	AID17300	
	109ER					
1486R	61C0	AHM	R12,E1E4-2	DECREMENT ENTRY COUNTER	AID17310	
	10A8R					
148AR	0302	BR	LINKS	RETURN	AID17320	
148CR	0812	ZS3	LHR	LINK,LINKS	SAVE RETURN	AID17330
148ER	C820	LHI	LINKS,ZS4	SET RETURN	AID17340	
	1494R					
1492R	220E	ZS4	BS	ZS2	DELETE ENTRY	AID17350
1494R	27BC	SIS	R11,12	DECREMENT INDEX	AID17360	
1496R	2032	BNZS	ZS4-2	CONTINUE	AID17370	
1498R	0301	BR	LINK		AID17380	
					AID17390	
					AID17400	
					AID17410	
149AR	C550	PROTEK	CLHI	PROTECTION (P) INSERTION HANDLER VALUE,C'R'	REGISTER PROTECTION ?	
	0052					
149ER	4230	BNE	PR1	NO = CELL PROTECTION	AID17420	
	14D2R					
14A2R	C5E0	CLHI	FROM,X*10'	RANGE CHECK	AID17430	
	0010					
14A6R	0386	BNLR	CMDER		AID17440	
14A8R	085E	LHR	VALUE,FROM		AID17450	
14AAR	0AEE	AHR	FROM,FROM	INDEXIZE REG	AID17460	
14ACR	2135	BNZS	PRO	SKIP	AID17470	
14AER	24CF	LIS	R12,15	LOAD REG 12	AID17480	
14BOR	40C0	STH	R12,PROR0	SET REG ZERO FLAG	AID17490	
	0A46R					
14B4R	2309	BS	PROA		AID17500	
14B6R	48B0	PRO	LH	R11,PROREG-2	BIT POSITIONAL HALFWORD	AID17510
	110AR					
14BAR	24C1	LIS	R12,1		AID17520	
14BCR	CDC5	SLHL	R12,0(VALUE)	SHIFT BIT BY REG INDEX	AID17530	
	0000					
14COR	06BC	OHR	R11,R12	OR BIT INTO POSITIONAL HALFWORD	AID17540	
14C2R	40B0	STH	R11,PROREG-2	STORE REG POSITION COUNTER	AID17550	
	110AR					
14C6R	485E	PROA	LH	VALUE,SAVREG(FROM)	LOAD USER REG	AID17560
	0A98R					
14CAR	405E	STH	VALUE,PROREG(FROM)	STORE USER REG IN LIST	AID17570	
	110CR					
14CEK	4300	B	PR4		AID17580	
	1500R					
14D2R	48B0	PR1	LH	R11,PROCEL-2	ENTRY COUNT	AID17590
	112CR					
14D6R	2114	BMS	PR2-2	EMPTY LIST	AID17600	
14D8R	C5B0	CLHI	R11,8	COMPARE TO EIGHT	AID17610	

0008					
14DCR 0386		BNLR	CMDR	NOT LESS = NO VACANCY	AID17620
14DFR 070H		XHR	R11,R11	CLEAR INDEX	AID17630
14EOR 48CR	PR2	LH	R12,PROCEL(R11)	GET AN ADDRESS	AID17640
112ER					
14E4R C5C0		CLHI	R12,-1	BLANKED ENTRY ?	AID17650
FFFF					
14E8R 2333		BES	PR3	YES	AID17660
14EAR 26B4		AIS	R11,4	NO - INCREMENT INDEX	AID17670
14ECR 2206		BS	PR2	CONTINUE	AID17680
14EER 40EB	PR3	STH	FROM,PROCEL(R11)	STORE PROTECTED ADDR	AID17690
112ER					
14F2R 48EE		LH	FROM,0(FROM)		AID17700
0000					
14F6R 40EB		STH	FROM,PROCEL+2(R11)	STORE PROTECTED VALUE	AID17710
113OR					
14FAR 24C1		LIS	R12,1		AID17720
14FCR 61C0		AHM	R12,PROCEL-2	BUMP ENTRY COUNTER	AID17730
112CR					
1500R 40C0	PR4	STH	R12,PROFLG	SET PROTECT FLAG FOR GO ROUTINE	AID17740
0A48R					
1504R 0302		BR	LINKS	RETURN	AID17750
*-----*					
			ZAP	PROTECTION HANDLER	AID17760
1506R C8B0	ZAPPRO	LHI	R11,PROCLN-PROCEL	LIST LENGTH FOR INDEX	AID17770
0020					AID17780
150AR C8C0		LHI	R12,-1		AID17790
FFFF					
150ER 08DD		LHR	NUMBER,NUMBER	SPECIFIC CELL/REG OR ALL ?	AID17800
1510R 4330		BZ	ZP3	ALL	AID17810
156CR					
1514R C800		LHI	R0,C' R'	JUST REGISTER ?	AID17820
2052					
1518R D400		CLB	R0;BUFFER+2		AID17830
0A50R					
151CR 4230		BNE	ZP1	NO	AID17840
1556R					
1520R 08EE		LHR	FROM,FROM	CHECK FOR ZERO	AID17850
1522R 2134		BNZS	ZP0		AID17860
1524R 40E0		STH	FROM,PROR0	CLEAR REG ZERO FLAG	AID17870
0A46R					
1528R 230B	ZP0	BS	ZP5		AID17880
152AR 2401		LIS	R0,1		AID17890
152CR CD0E		SLHL	R0,0(FROM)	SHIFT BIT POSITIONAL COUNTER	AID17900
0000					
1530R 4400		NH	R0,PROREG-2	ISOLATE BIT	AID17910
110AR					
1534R 0336		BZR	CMDR	IF NOT SET = ERROR	AID17920
1536R 4700		XH	R0,PROREG-2	ELIMINATE BIT THAT EQUALS REG	AID17930
110AR					
153AR 4000		STH	R0,PROREG-2	RESTORE COUNTER	AID17940
110AR					
153ER 4800	ZP5	LH	R0,PROCEL-2	GET PROTECTED CELL COUNT	AID17950
112CR					
1542R 0232		BNZR	LINKS		AID17960
1544R 4800		LH	R0,PROR0	CHECK REG ZERO	AID17970

1548R	0A46R 0232	BNZR	LINKS		AID17980
154AR	4800	LH	R0,PROREG-2	CHECK REG COUNT	AID17990
	110AR				
154ER	0232	BNZR	LINKS		AID18000
1550R	4000	STH	R0,PROFLG		AID18010
	0A48R				
1554R	0302	BR	LINKS		AID18020
1556R	45ER	ZP1 CLH	FROM,PROCEL-4(R11)	COMPARE CELL ADDR	AID18030
	112AR				
155AR	2334	BES	ZP2	HIT - DELETE	AID18040
155CR	27B4	SIS	R11,4	DECREMENT INDEX	AID18050
155ER	0336	BZR	CMDER	NOT FOUND = ERROR	AID18060
1560R	2205	BS	ZP1	CONTINUE	AID18070
1562R	40CR	ZP2 STH	R12,PROCEL-4(R11)	DELETE ADDRESS	AID18080
	112AR				
1566R	61C0	AHM	R12,PROCEL-2	DECREMENT ENTRY COUNTER	AID18090
	112CR				
156AR	0302	BR	LINKS	RETURN	AID18100
156CR	40D0	ZP3 STH	NUMBER,PROREG-2	CLEAR BITS	AID18110
	110AR				
1570R	0812	LHR	LINK,LINKS	SAVE RETURN	AID18120
1572R	C820	LHI	LINKS,ZP4	SET RETURN	AID18130
	1578R				
1576R	220A	BS	ZP2	DELETE AN ENTRY	AID18140
1578R	27B4	ZP4 SIS	R11,4	DECREMENT INDEX	AID18150
157AR	203C	BNZS	ZP2	CONTINUE	AID18160
157CR	40B0	STH	R11,PROCEL-2	CLEAR CELL COUNTER	AID18170
	112CR				
1580R	40B0	STH	R11,PROR0	CLEAR REG ZERO	AID18180
	0A46R				
1584R	40B0	STH	R11,PROFLG	CLEAR FLAG	AID18190
	0A48R				
1588R	0301	BR	LINK	RETURN	AID18200
					AID18210
					AID18220
					AID18230
158AR	D0A0	*XLATEU	STM	R10,SAVE2	SAVE REGS
	0AC0R				
158ER	4020	STH	LINKS,XU8	SAVE LINK	AID18240
	1648R				
1592R	C8A0	LHI	R10,X'2020'		AID18250
	2020				
1596R	40A0	STH	R10,DBLWK+2	CLEAR WORK AREA	AID18260
	0ABAR				
159AR	08CF	LHR	R12,TO	SAVE TO ADDRESS	AID18270
159CR	C4C0	NHI	R12,X'FFFE'	INSURE HALF WORD BOUNDARY	AID18280
	FFFE				
15A0R	C8E0	LHI	FROM,OPRAND+1	SET FROM ADDRESS	AID18290
	0A79R				
15A4R	C8F0	LHI	TO,DBLWK	SET WORK AREA AS TO ADDRESS	AID18300
	0AB8R				
15A8R	D3D0	LB	NUMBER,OPRAND	BYTE COUNT	AID18310
	0A78R				
15ACR	4120	BAL	LINKS,MOVE	MOVE DATA	AID18320
	079AR				
15B0R	08FC	LHR	TO,R12	RESTORE TO ADDRESS	AID18330

15B2R	C8B0 01F0		LHI	R11,MSDEND-MSD0	OP CODE MNEMONIC TABLE LENGTH	AID18340
15B6R	48CB 17E0R	XU1	LH	R12,MSD0-4(R11)		AID18350
15BAR	48DB 17E2R		LH	NUMBER,MSD0-2(R11)		AID18360
15BER	4BC0 0AB8R		SH	R12,DBLWK	COMPARE OP CODE MNEMONIC	AID18370
15C2R	4FD0 0ABAR		SCH	NUMBER,DBLWK+2	WITH INPUT	AID18380
15C6R	2334		BZS	XU2	FOUND	AID18390
15C8R	27B4		SIS	R11,4	DECREMENT POINTER	AID18400
15CAR	0216		BMR	CMDER	NOT FOUND	AID18410
15CCR	220B		BS	XU1	CONTINUE SEARCH	AID18420
15CER	C8DB 17E1R	XU2	LHI	NUMBER,MSD0-3(R11)	ASSEMBLY TABLE POINTER	AID18430
15D2R	24CF		LIS	R12,15	MSD COUNTER	AID18440
15D4R	C8B0 17E2R		LHI	R11,ASMEND-2	MNEMONIC POINTER TABLE	AID18450
15D8R	48AB 0000	XU3	LH	R10,0(R11)	POINTER TABLE ENTRY	AID18460
15DCR	2333		BZS	XU4	IGNORE ENTRY	AID18470
15DER	05DA		CLHR	NUMBER,R10	CHECK FOR VECTOR	AID18480
15E0R	2384		BNLS	XU5	FOUND	AID18490
15E2R	27C1	XU4	SIS	R12,1	DECREMENT COUNTER	AID18500
15E4R	27B2		SIS	R11,2	DECREMENT INDEX	AID18510
15E6R	2207		BS	XU3	CONTINUE SCAN	AID18520
15E8R	91C4	XU5	SLLS	R12,4	COUNTER = MSD	AID18530
15EAR	0BDA		SHR	NUMBER,R10	GET POINTER INDEX	AID18540
15ECR	26D1		AIS	NUMBER,1	ROUND UP	AID18550
15EER	90D2		SRLS	NUMBER,2	QUARTER REMAINDER (ENTRY = 4 BYTES)	AID18560
15FOR	0700		XHR	R0,R0		AID18570
15F2R	400F 0000		STH	R0,0(TO)	CLEAR INSTRUCTION LOCATION	AID18580
15F6R	06CD		OHR	R12,NUMBER	COMBINE LSD AND MSD	AID18590
15F8R	D2CF 0000		STB	R12,0(TO)	PLACE OP CODE IN MEMORY	AID18600
15FCR	D3C0 0A78R		LB	R12,OPRAND		AID18610
1600R	C8CC 0A79R		LHI	R12,OPRAND+1(R12)	ADDRESS OF NEXT OPERAND	AID18620
1604R	D3DC 0000		LB	NUMBER,0(R12)	GET OPERAND SIZE	AID18630
1608R	C5D0 0001		CLHI	NUMBER,1	MUST BE ONE BYTE (FOR REG)	AID18640
160CR	0236		BNER	CMDER	IF NOT, ERROR	AID18650
160ER	C8EC 0001		LHI	FROM,1(R12)	SET SENDER ADDRESS	AID18660
1612R	4120 044CR		BAL	LINKS,XLATHB	TRANSLATE TO BINARY	AID18670
1616R	D30F 0001		LB	R0,1(TO)	SAVE R2 VALUE	AID18680
161AR	9104		SLLS	R0,4	ADJUST R1	AID18690
161CR	2472		LIS	LENGTH,2	SET LENGTH	AID18700
161ER	C3A0 0001		THI	R10,1	FLAG (RX INSTR) SET ?	AID18710

1622R	4230		BNZ	XU7	YES	AID18720
	164AR					
1626R	CAEC	XU6	LHI	FROM,3(R12)	ADDRESS OF SENDER FIELD	AID18730
	0003					
162AR	D3DC		LB	NUMBER,2(R12)	BYTE COUNT	AID18740
	0002					
162ER	C5D0		CLHI	NUMBER,1	MUST BE ONE OR LESS	AID18750
	0001					
1632R	2186		BLS	XU6A	IF ZERO SKIP	AID18760
1634R	0236		BNER	CMDER		AID18770
1636R	4120		BAL	LINKS,XLATHB	TRANSLATE TO BINARY	AID18780
	044CR					
163AR	460F		OH	R0,0(T0)	COMBINE WITH R1	AID18790
	0000					
163ER	D20F	XU6A	STB	R0,1(T0)	STORE AFTER OP CODE	AID18800
	0001					
1642R	D1A0		LM	R10,SAVE2	RESTORE REGS	AID18810
	0AC0R					
1646R	4300		DC	X'4300'	RETURN	AID18820
1648R	0000	XU8	DC	0		AID18830
164AR	D3BC	XU7	LB	R11,2(R12)		AID18840
	0002					
164ER	CABC		AHI	R11,1(R12)	SAVE POINTER TO R2 OPERAND	AID18850
	0001					
1652R	D3DC		LB	NUMBER,2(R12)	OPERAND LENGTH	AID18860
	0002					
1656R	C8EC		LHI	FROM,3(R12)		AID18870
	0003					
165AR	26F2		AIS	T0,2	BUMP RECEIVER	AID18880
165CR	40DF		STH	NUMBER,0(T0)	CLEAR ADDRESS FIELD	AID18890
	0000					
1660R	4120		BAL	LINKS,XLATHB	TRANSLATE TO BINARY	AID18900
	044CR					
1664R	D3A0		LB	R10,BUFFER+1	GET DIRECTIVE TYPE	AID18910
	0A4FR					
1668R	C5A0		CLHI	R10,X'42'	BIAS ?	AID18920
	0042					
166CR	2133		BNES	XU7A	NO	AID18930
166ER	613F		AHM	BIAS,0(T0)	ADD BIAS	AID18940
	0000					
1672R	27F2	XU7A	SIS	T0,2	DECREMENT RECEIVER	AID18950
1674R	2474		LIS	LENGTH,4	SET LENGTH ATTRIBUTE	AID18960
1676R	08CB		LHR	R12,R11	RESTORE OPERAND POINTER	AID18970
1678R	4300		B	XU6	PRIME REGISTER	AID18980
	1626R					
						AID18990
						AID19000
						AID19010
167CR	D0A0	XLATEA	STM	R10,SAVE2	SAVE REGS	AID19020
	0AC0R					
1680R	4020		STH	LINKS,XA2	SAVE LINK	AID19020
	171AR					
1684R	C580		CLHI	MODE,X'42'	BIAS ?	AID19030
	0042					
1688R	2138		BNES	XA0	NO	AID19040
168AR	C800		LHI	R0,X'0052'	'R' FOR RELOCATABLE	AID19050
	0052					



168ER	D20F		STB	R0,4(TO)	STORE AFTER ADDRESS	AID19060
	0004					
1692R	48E0		LH	FROM,SAVE2+8	GET ADDRESS	AID19070
	0AC8R					
1696R	08E3		SHR	FROM,BIAS	UNBIAS	AID19080
1698R	40E0	XAO	STH	FROM,OPSAVE	REPLACE IN WORK AREA	AID19090
	0A96R					
169CR	C8E0		LHI	FROM,OPSAVE	SET SENDER ADDRESS	AID19100
	0A96R					
16A0R	24D2		LIS	NUMBER,2		AID19110
16A2R	4120		BAL	LINKS,XLATEH	TRANSLATE TO ASCII	AID19120
	040ER					
16A6R	48E0		LH	FROM,SAVE2+8	RESTORE ABSOLUTE ADDRESS	AID19130
	0AC8R					
16AAR	26F6		AIS	TO,6	BUMP RECEIVER	AID19140
16ACR	2404		LIS	NUMBER,4	SET INSTRUCTION LENGTH	AID19150
16AER	4120		BAL	LINKS,XLATEH	TRANSLATE TO ASCII	AID19160
	040ER					
16B2R	26FC		AIS	TO,12	BUMP PRINT LINE	AID19170
16B4R	D3AE		LB	R10,0(FROM)	GET OP CODE OF OPEN CELL	AID19180
	0000					
16B8R	08BA		LHR	R11,R10	SAVE OP CODE	AID19190
16BAR	90A4		SRLS	R10,4	SHIFT OFF LSD	AID19200
16BCH	0AAA		AHR	R10,R10	INDEX-IZE BY TWO	AID19210
16BER	48AA		LH	R10,ASMPTR(R10)	CHECK FOR ZERO	AID19220
	17C4R					
16C2R	4330		BZ	DEFCON	SET UP DEFINE CONSTANT	AID19230
	178ER					
16C6R	C4B0		NHI	R11,X*F	GET LSD	AID19240
	000F					
16CAR	91B2		SLLS	R11,2	INDEXIZE VALUE	AID19250
16CCR	0ABA		AHR	R11,R10	FORM ADDRESS OF MNEMONIC	AID19260
16CER	48CB		LH	R12,0(R11)	VALID OP CODE	AID19270
	0000					
16D2R	4210		BM	DEFCON	NO, SET UP DEFINE CONSTANT	AID19280
	178ER					
16D6R	40CF		STH	R12,0(TO)	SET OP CODE MSD	AID19290
	0000					
16DAR	48CB		LH	R12,2(R11)		AID19300
	0002					
16DER	40CF		STH	R12,2(TO)	SET OP CODE	AID19310
	0002					
16E2R	24D1		LIS	NUMBER,1	R1/M1 FIELD	AID19320
16E4R	26E1		AIS	FROM,1	FROM OPEN CELL + 1	AID19330
16E6R	26F6		AIS	TO,6	TO PRINT BUFFER	AID19340
16E8R	4120		BAL	LINKS,XLATEH	TRANSLATE TO ASCII	AID19350
	040ER					
16ECR	26F1		AIS	TO,1	BUMP BUFFER POINTER	AID19360
16EER	D3C0	XA1	LB	R12,COMMA	GET COMMA	AID19370
	0A3ER					
16F2R	D30F		LB	R0,0(TO)	SAVE R2	AID19380
	0000					
16F6R	D2CF		STB	R12,0(TO)	PLACE COMMA IN INSTRUCTION	AID19390
	0000					
16FAR	26F1		AIS	TO,1	BUMP POINTER	AID19400
16FCR	C3A0		THI	R10,1	RX INSTRUCTION ?	AID19410

0001						
1700R	213E		BNZS	IDXASM	YES	AID19420
1702R	020F		STB	R0,0(TO)	ENTER R2	AID19430
	0000					
1706R	2472	XA4	LIS	LENGTH,2	SET LENGTH	AID19440
1708R	C800		LHI	R0,X'2020'	SPACES	AID19450
	2020					
170CR	400F		STH	R0,-16(TO)	CLEAR SECOND HALFWORD	AID19460
	FFF0					
1710R	400F		STH	R0,-14(TO)	CLEAR SECOND HALF WORD	AID19470
	FFF2					
1714R	01A0	XA3	LM	R10,SAVE2	RESTORE REGS	AID19480
	0AC0R					
1718R	4300		DC	X'4300'	RETURN	AID19490
171AR	0000	XA2	DC	0		AID19500
171CR	2474	IDXASM	LIS	LENGTH,4	SET LENGTH	AID19510
171ER	D3AE		LB	R10,0(FROM)	SAVE INDEX REGS	AID19520
	0000					
1722R	26E1		AIS	FROM,1	BUMP SENDER	AID19530
1724R	24D2		LIS	NUMBER,2	SET BYTE COUNT	AID19540
1726R	08BE		LHR	R11,FROM	SAVE POINTER	AID19541
1728R	C580		CLHI	MODE,X'42'	BIASED INSTRUCTION	AID19550
	0042					
172CR	4330		BE	XA6	YES, UNBIAS ADDRESS REFERENCES	AID19560
	1760R					
1730R	4120	XA5	BAL	LINKS,XLATEH	TRANSLATE TO ASCII	AID19570
	040ER					
1734R	08EB		LHR	FROM,R11	RESTORE POINTER	AID19571
1736R	27E1		SIS	FROM,1	DECREMENT TO FIRST H-WORD	AID19580
1738R	D30E		LB	R0,0(FROM)	GET REGISTERS	AID19590
	0000					
173CR	C300		THI	R0,X'0F'	R2 = ZERO ?	AID19600
	000F					
1740R	233E		BZS	XA5A	YES - SKIP IT	AID19610
1742R	26F4		AIS	TO,4	INCREMENT PRINT BUFFER	AID19620
1744R	24D1		LIS	NUMBER,1	SET BYTE COUNT	AID19630
1746R	4120		BAL	LINKS,XLATEH	TRANSLATE TO ASCII	AID19640
	040ER					
174AR	D3C0		LB	R12,OPREN		AID19650
	0A3CR					
174ER	D2CF		STB	R12,0(TO)	SET OPEN PAREN	AID19660
	0000					
1752R	D3C0		LB	R12,CPREN		AID19670
	0A3DR					
1756R	D2CF		STB	R12,2(TO)	SET CLOSE PAREN	AID19680
	0002					
175AR	27F1		SIS	TO,1	DECREMENT POINTER	AID19690
175CR	4300	XA5A	B	XA3	FINISH UP	AID19700
	1714R					
1760R	480E	XA6	LH	R0,0(FROM)	GET ABSOLUTE ADDRESS REFERENCE	AID19710
	0000					
1764R	0503		CLHR	R0,BIAS	CHECK AGAINST BIAS	AID19720
1766R	4280		BL	XA5	DON'T BIAS	AID19730
	1730R					
176AR	0B03		SHR	R0,BIAS	REMOVE BIAS	AID19740
176CR	4000		STH	R0,OPSAVE	SAVE IN WORK AREA	AID19750

0A96R					
1770R	08BE	LHR	R11, FROM	SAVE POINTER	AID19751
1772R	C8E0	LHI	FROM, OPSAVE	SET SENDER ADDRESS	AID19760
	0A96R				
1776R	CBF0	SHI	TO, 16	ADJUST LINE POINTER	AID19770
	0010				
177AR	4120	BAL	LINKS, XLATEH	TRANSLATE TO ASCII	AID19780
	040ER				
177ER	26F4	AIS	TO, 4	INCREMENT LINE POINTER	AID19790
1780R	C800	LHI	R0, X'0052'	'R' FOR RELOCATABLE	AID19800
	0052				
1784R	D20F	STB	R0, 0(T0)	STORE IN PRINTLINE	AID19810
	0000				
1788R	26FC	AIS	TO, 12	RESTORE PRINTLINE POINTER	AID19820
178AR	4300	B	XA5	RETURN	AID19830
	1730R				
178ER	C8C0	DEFCON LHI	R12, C'DC'	SET UP DEFINE CONSTANT	AID19840
	4443				
1792R	40CF	STH	R12, 0(T0)	STORE IN PRINT LINE	AID19850
	0000				
1796R	26F6	AIS	TO, 6	INCREMENT INDEX	AID19860
1798R	C8C0	LHI	R12, X'5827'	LOAD A 'X'	AID19870
	5827				
179CR	40CF	STH	R12, 0(T0)	PUT IN PRINT LINE	AID19880
	0000				
17A0R	24D2	LIS	NUMBER, 2	SET BYTE COUNT	AID19890
17A2R	26F2	AIS	TO, 2	BUMP RECEIVER	AID19900
17A4R	4120	BAL	LINKS, XLATEH	TRANSLATE TO ASCII	AID19910
	040ER				
17A8R	26F4	AIS	TO, 4	BUMP POINTER	AID19920
17AAR	D2CF	STB	R12, 0(T0)	STORE CLOSE PAREN	AID19930
	0000				
17AER	26F4	AIS	TO, 4	BUMP INDEX	AID19940
17B0R	C8C0	LHI	R12, X'433D'	CHARACTER TRANSLATION	AID19950
	433D				
17B4R	40CF	STH	R12, 0(T0)	STORE 'C:'	AID19960
	0000				
17B8R	26F3	AIS	TO, 3	BUMP INDEX	AID19970
17BAR	4120	BAL	LINKS, XLATEC	TRANSLATE TO CHARACTER	AID19980
	076CR				
17BER	27FA	SIS	TO, 10	RESET LINE POINTER	AID19990
17C0R	4300	B	XA4		AID20000
	1706R				
*-----*					AID20010
* ASSEMBLY TABLE POINTERS					AID20020
17C4R	17E4R	ASMPTR DC	MSD0, 0		AID20030
	0000				
17C8R	1824R	DC	MSD2, 0		AID20040
	0000				
17CCR	1865R	DC	MSD4+1, 0		AID20050
	0000				
17D0R	18A5R	DC	MSD6+1, 0, 0		AID20060
	0000				
	0000				
17D6R	18E4R	DC	MSD9, 0, 0		AID20070
	0000				

0000				
17DCR	1925R	DC	MSDC+1,MSDD+1	AID20080
	1965R			
17E0R	19A5R	DC	MSDE+1,0	AID20090
	0000			
17E4R		ASMEND	EQU *	AID20100
		*	-----*	AID20110
		*		AID20120
		*	ASSEMBLER MNEMONICS	AID20130
17E4R	FFFF	MSD0	DC -1,-1	AID20140
	FFFF			
17E8R	4241	DC	C*BALR'	AID20150
	4C52			
17ECR	4254	DC	C*BTCR'	AID20160
	4352			
17F0R	4246	DC	C*BFCR'	AID20170
	4352			
17F4R	4E48	DC	C*NHR'	AID20180
	5220			
17F8R	434C	DC	C*CLHR'	AID20190
	4852			
17FCR	4F48	DC	C*OHR'	AID20200
	5220			
1800R	5848	DC	C*XHR'	AID20210
	5220			
1804R	4C48	DC	C*LHR'	AID20220
	5220			
1808R	4348	DC	C*CHR'	AID20230
	5220			
180CR	4148	DC	C*CHR'	AID20240
	5220			
1810R	5348	DC	C*SHR'	AID20250
	5220			
1814R	4D48	DC	C*MHR'	AID20260
	5220			
1818R	4448	DC	C*DHR'	AID20270
	5220			
181CR	4143	DC	C*ACHR'	AID20280
	4852			
1820R	5343	DC	C*SCHR'	AID20290
	4852			
1824R	4254	MSD2	DC C*BTBS'	AID20300
	4253			
1828R	4254	DC	C*BTFS'	AID20310
	4653			
182CR	4246	DC	C*BFBS'	AID20320
	4253			
1830R	4246	DC	C*BFFS'	AID20330
	4653			
1834R	4C49	DC	C* LIS'	AID20340
	5320			
1838R	4C43	DC	C* LCS'	AID20350
	5320			
183CR	4149	DC	C* AIS'	AID20360
	5320			
1840R	5349	DC	C* SIS'	AID20370

5320				
1844R	4C45		DC C'LER *	AID20380
	5220			
1848R	4345		DC C' CER *	AID20390
	5220			
184CR	4145		DC C' AER *	AID20400
	5220			
1850R	5345		DC C' SER *	AID20410
	5220			
1854R	4D45		DC C' MER *	AID20420
	5220			
1858R	4445		DC C' DER *	AID20430
	5220			
185CR	FFFF		DC -1,-1,-1,-1	AID20440
	FFFF			
	FFFF			
	FFFF			
1864R	5354	MSD4	DC C' STH *	AID20450
	4820			
1868R	4241		DC C' BAL *	AID20460
	4C20			
186CR	4254		DC C' BTC *	AID20470
	4320			
1870R	4246		DC C' BFC *	AID20480
	4320			
1874R	4E48		DC C' NH *	AID20490
	2020			
1878R	434C		DC C' CLH *	AID20500
	4820			
187CR	4F48		DC C' OH *	AID20510
	2020			
1880R	5848		DC C' XH *	AID20520
	2020			
1884R	4C48		DC C' LH *	AID20530
	2020			
1888R	4348		DC C' CH *	AID20540
	2020			
188CR	4148		DC C' AH *	AID20550
	2020			
1890R	5348		DC C' SH *	AID20560
	2020			
1894R	4D48		DC C' MH *	AID20570
	2020			
1898R	4448		DC C' DH *	AID20580
	2020			
189CR	4143		DC C' ACH *	AID20590
	4820			
18A0R	5343		DC C' SCH *	AID20600
	4820			
18A4R	5354	MSD6	DC C' STE *	AID20610
	4520			
18A8R	4148		DC C' AHM *	AID20620
	4D20			
18ACR	FFFF		DC -1,-1,-1,-1	AID20630
	FFFF			
	FFFF			

18B4R	FFFF 4154 4C20	DC	C*ATL *	AID20640
18B8R	4142 4C20	DC	C*ABL *	AID20650
18BCR	5254 4C20	DC	C*RTL *	AID20660
18C0R	5242 4C20	DC	C*RBL *	AID20670
18C4R	4C45 2020	DC	C*LE *	AID20680
18C8R	4345 2020	DC	C*CE *	AID20690
18CCR	4145 2020	DC	C*AE *	AID20700
18D0R	5345 2020	DC	C*SE *	AID20710
18D4R	4D45 2020	DC	C*ME *	AID20720
18D8R	4445 2020	DC	C*DE *	AID20730
18DCR	FFFF FFFF FFFF	DC	-1,-1,-1,-1	AID20740
18E4R	5352 4C53	MSD9 DC	C*SRLS*	AID20750
18E8R	534C 4C53	DC	C*SLLS*	AID20760
18ECR	5354 4252	DC	C*STBR*	AID20770
18F0R	4C42 5220	DC	C*LBR *	AID20780
18F4R	4558 4252	DC	C*EXBR*	AID20790
18F8R	4550 5352	DC	C*EPSR*	AID20800
18FCR	5742 5220	DC	C*WBR *	AID20810
1900R	5242 5220	DC	C*RBR *	AID20820
1904R	5748 5220	DC	C*WHR *	AID20830
1908R	5248 5220	DC	C*RHR *	AID20840
190CR	5744 5220	DC	C*WDR *	AID20850
1910R	5244 5220	DC	C*RDR *	AID20860
1914R	4D48 5552	DC	C*MHR *	AID20870
1918R	5353 5220	DC	C*SSR *	AID20880
191CR	4F43 5220	DC	C*OCR *	AID20890
1920R	4149	DC	C*AIR *	AID20900

1924R	5220 4258 4820	MSDC	DC	C'BXH °	AID20910
1928R	4258 4C45		DC	C'BXLE°	AID20920
192CR	4C50 5357		DC	C'LPSW°	AID20930
1930R	5448 4920		DC	C'THI °	AID20940
1934R	4E48 4920		DC	C'NHI °	AID20950
1938R	434C 4849		DC	C'CLHI°	AID20960
193CR	4F48 4920		DC	C'OHI °	AID20970
1940R	5848 4920		DC	C'XHI °	AID20980
1944R	4C48 4920		DC	C'LHI °	AID20990
1948R	4348 4920		DC	C'CHI °	AID21000
194CR	4148 4920		DC	C'AH I °	AID21010
1950R	5348 4920		DC	C'SHI °	AID21020
1954R	5352 484C		DC	C'SRHL°	AID21030
1958R	534C 484C		DC	C'SLHL°	AID21040
195CR	5352 4841		DC	C'SRHA°	AID21050
1960R	534C 4841		DC	C'SLHA°	AID21060
1964R	5354 4D20	MSDD	DC	C'STM °	AID21070
1968R	4C4D 2020		DC	C'LM °	AID21080
196CR	5354 4220		DC	C'STB °	AID21090
1970R	4C42 2020		DC	C'LB °	AID21100
1974R	434C 4220		DC	C'CLB °	AID21110
1978R	414C 2020		DC	C'AL °	AID21120
197CR	5742 2020		DC	C'WB °	AID21130
1980R	5242 2020		DC	C'RB °	AID21140
1984R	5748 2020		DC	C'WH °	AID21150
1988R	5248 2020		DC	C'RH °	AID21160
198CR	5744 2020		DC	C'WD °	AID21170
1990R	5244		DC	C'RD °	AID21180

2020				
1994R 4048		DC	C'MHU *	AID21190
5520				
1998R 5353		DC	C'SS *	AID21200
2020				
199CR 4F43		DC	C'OC *	AID21210
2020				
19A0R 4149		DC	C'AI *	AID21220
2020				
19A4R FFFF	MSDE	DC	-1,-1	AID21230
FFFF				
19A8R 5356		DC	C'SVC *	AID21240
4320				
19ACR 5349		DC	C'SINT*	AID21250
4E54				
19B0R		DO	7	AID21260
19B0R FFFF		DC	-1,-1	AID21270
FFFF				
19B4R FFFF		DC	-1,-1	AID21270
FFFF				
19B8R FFFF		DC	-1,-1	AID21270
FFFF				
19BCR FFFF		DC	-1,-1	AID21270
FFFF				
19C0R FFFF		DC	-1,-1	AID21270
FFFF				
19C4R FFFF		DC	-1,-1	AID21270
FFFF				
19C8R FFFF		DC	-1,-1	AID21270
FFFF				
19CCR 5252		DC	C'RRL *	AID21280
4C20				
19D0R 524C		DC	C'RLL *	AID21290
4C20				
19D4R	MSDFEND	EQJ	*	AID21300
19D4R		END	OSAIDS	AID21310



## NO ERRORS

ADD	02E8R
ADFL0T	0316R
AIDSAV	0AE0R
ARITH	02A4R
ASM1	0CDEr
ASMDMP	0CC6R
ASMEND	17E4R
ASMPTR	17C4R
BIAS	0003
BINBLK	08BCR
BINOUT	08AER
BR1	11C0R
BR2	11CER
BREAK	11B4R
BRKMSG	1224R
BUFEND	0A76R
BUFFER	0A4ER
BUFFLG	114ER
BYAS	012CR
CCSAVE	0FC0R
CD1	0D48R
CD2	0D12R
CHK1	005ER
CHK2	006AR
CHK2A	0076R
CHK3	007CR
CHK3A	0094R
CHK4	009ER
CHK5	00B0R
CHK6	00C6R
CHK7	00D4R
CHR1	0D46R
CHRDMP	0CF8R
CLEAR	07B6R
CMCREN	0A17R
CMDCHR	0A06R
CMDEND	0A3AR
CMDER	0006
CMDERR	0924R
CMDINP	080AR
CMDMS	0946R
CMDMSG	0942R
CMDND	094ER
CMDTBL	0A18R
COMMA	0A3ER
CPREN	0A30R
D1	0B82R
D10	066ER
D3C	0666R
DBLSAV	0ABCR
DBLWK	0AB8R
DD1	0B88R
DD2	0BA2R
DECD1	0BC8R
DECD2	0A44R

DECDMP	0B84R
DEFCON	178ER
DELEND	0A42R
DELIM	0A3AR
DUMP	0B2CR
DUMPU	0B7AR
E1	1040R
E1E0	1056R
E1E0N	1086R
E1E2	1088R
E1E2N	10A8R
E1E4	10AAR
E1E4N	110AR
E1TRL	0DE6R
ENTER	0FECH
EX	0DF6R
EX2	0E26R
EX2A	0E52R
EX2B	0E34R
EX2C	0E76R
EX3	0E80R
EX4	0EC6R
EX4A	0EFCR
EX4B	0F08R
EX5	0F2AR
EX5A	0F46R
EX6	0F94R
EX6A	0F9AR
EX7	0EA0R
EXBLK	0FB6R
EXBRC1	0FC4R
EXBRCH	0FC8R
EXCUTE	0F7CR
EXIT	0FB2R
EXLPSW	0F8CR
EXOUT	0F90R
EXRTN	0FBCR
EXRTRN	0FA6R
FCON1	066AR
FLOT	02F2R
FLT1	0BCER
FLT2	0BECR
FLTDMP	0BCAR
FROM	000E
G1	0340R
G10	03F8R
G11	0408R
G12	040CR
G1A	0364R
G1B	0350R
G1C	0376R
G3	040AR
G4	0400R
G5	0380R
G5B	0384R
G6	0392R

G6A	03A2R
G6B	03CAR
G7	03AAR
G8	03C2R
G8A	03CER
G9	03D2R
G0	0324R
HD1	0CF6R
HEXDMP	0CE0R
IDXASM	171CR
ILLEG	0914R
ILLMSG	0966R
IN	004ER
INPUT	000A
IOERR	092AR
IOMEND	0960R
IOMS	0952R
IOMSG	094ER
J1	02A0R
JUMP	0290R
LDPACK	0C72R
LDPK1	0C82R
LDR1	0C22R
LDR2	0C32R
LDR3	0C9AR
LDR4	0C70R
LDR5	0C5ER
LDREG	0046R
LDREND	0C54R
LDROUT	0C92R
LENGTH	0007
LENTBL	0DEER
LINK	0001
LINKS	0002
LIST	08C4R
LOADER	08EER
LOG	0932R
LSTBLK	090CR
LT1	08D4R
LT2	08E4R
LT3	090AR
LU	00DER
M1	079ER
M2	07B0R
MODE	0008
MOVE	079AR
MSD0	17E4R
MSD2	1824R
MSD4	1864R
MSD6	18A4R
MSD9	18E4R
MSDC	1924R
MSDD	1964R
MSDE	19A4R
MSDEND	19D4R
MSGTBL	00DER

NAME	004AR
NEKST	0222R
NEXT	0284R
NUMBER	0000
OP1	01C6R
OP1A	01CAR
OP2	01E8R
OP3	01FAR
OP4	01FCR
OPCELL	0FC2R
OPCODE	0FDER
OPCONT	0220R
OPEN	0004
OPIN	01B6R
OPN	09C8R
OPRAND	0A78R
OPREN	0A3CR
OPREND	0A8CR
OPSAVE	0A96R
OPSTOP	0FE8R
OPTAB	09A8R
OPTAEN	09AFR
OPTBEN	09B6R
OPTCEN	09C2R
OPTDEN	09C6R
OS	001CR
OS1	0026R
OS2	0042R
OS AIDS	0000R
OT1	081AR
OT2	0826R
OUT	0812R
OUTBLK	083ER
OUTBUF	0842R
OUTEND	08AER
OUTPUT	0009
PARSER	009AR
PAUSE	0964R
PAWS	093AR
POP	0A40R
PRO	14B6R
PROA	14C6R
PR1	14D2R
PR2	14E0R
PR3	14EER
PR4	1500R
PRECED	028AR
PROCEL	112ER
PROCHK	0370R
PROCLN	114ER
PROFLG	0A48R
PROMSG	0A90R
PROK0	0A46R
PROREG	110CR
PRORGN	112CR
PROTEK	149AR

PSWSAV	0A4AR
R0	0000
R1	0001
R10	000A
R11	000B
R12	000C
R13	000D
R14	000E
R15	000F
R2	0002
R3	0003
R4	0004
R5	0005
R6	0006
R7	0007
R8	0008
R9	0009
RDIN	07CAR
READ	07F8R
REGSAV	0A98R
RP1	023ER
RP2	0244R
RP3	024ER
RP3A	0258R
RP4	0264R
RP5	0276R
RPLACE	0226R
SAVE	0B00R
SAVE2	0AC0R
SAVREG	0A98R
SBFLOT	0308R
SETBUF	0CAAR
SN1	139CR
SN2	13AAR
SN3	13D6R
SNAP	1390R
SNPMSG	0A8CR
SPACE	0A42R
SVC14	1150R
SVC14A	1176R
SVCE0	12CAR
SVCE01	12CER
SVCE02	12DAR
SVCE03	1308R
SVCE04	1334R
SVCE05	133ER
SVCE06	1344R
SVCE2	11EER
SVCE21	11F2R
SVCE22	11FER
SVCE4	13EER
SVCE41	13F2R
SVCE42	13FER
SVCE43	1436R
SVCTAB	118AR
TAB	097ER

ABA	097AK
ABB	0974K
ABC	096AK
ABD	099ER
BOUT	0990R
BSCAN	0982R
ERM	0156R
ITLE	0B20R
ITLEN	0B2CR
O	000F
R1	1272R
R2	1286R
R3	128ER
R4	1294R
R5	12A6R
R6	12B0R
RACE	1266R
RC3	131ER
RCFLG	0A94R
RCMSG	134CR
RCSAV	1350R
TLE	0B24R
NPACK	0960R
T1	0D52R
T2	0D74R
T3	0D88R
T3A	0DAER
T4	0DR6R
T5	0DD6R
T6	0DC8R
T7	0DCER
ITIL	0D4AR
VALUE	0005
I1	0E1ER
IA0	169BR
IA1	16EER
IA2	171AR
IA3	1714R
IA4	1706R
IA5	1730R
IA5A	175CR
IA6	1760R
IA8	0EBER
IB1	04COR
IB2	04CCR
IC1	0770R
IC2	077AR
IC3	077ER
IC4	0786R
ID1	04EER
ID2	04F4R
ID3	050ER
IF1	0568R
IF2	0578R
IF3	0584R
IF4	0598R

XF5	059ER
XF6	05ACR
XF7	05EAR
XF8	060CR
XF9	0632R
XF9A	0644R
XFR0	05C0R
XFR2	05CER
XFSAVE	065ER
XH1	0416R
XH2	042ER
XH3	0434R
XHB1	0458R
XHB2	045AR
XHB2A	045ER
XHB3	0466R
XHB4	0494R
XHB5	04ACR
XL2	06CER
XL3	06D4R
XLATDB	0482R
XLATEA	167CR
XLATEC	076CR
XLATED	04E2R
XLATEF	0518R
XLATEH	040ER
XLATEL	0672R
XLATEU	158AR
XLATHB	044CR
XLCMEX	0730R
XLDCNT	0720R
XLDEC	06BAR
XLDEC2	06C8R
XLDONE	0716R
XLEXDG	0738R
XLEXPT	06FOR
XLFT1	06A0R
XLFT2	0702R
XLFT3	0704R
XLOUT	0746R
XLSAVE	0768R
XLTF	06EAR
XREDUZ	0654R
XU1	15B6R
XU2	15CER
XU3	15D8R
XU4	15E2R
XU5	15E8R
XU6	1626R
XU6A	163ER
XU7	164AR
XU7A	1672R
XU8	1648R
XZERO	063CR
Z1	11ACR
ZAP	1190R

ZAPBRK	122BR
ZAPPRO	150BR
ZAPSNP	145CR
ZAPTRC	1352R
ZB1	1236R
ZB2	1242R
ZB3	1258R
ZB4	1260R
ZP0	152AR
ZP1	1556R
ZP2	1562R
ZP3	156CR
ZP4	1578R
ZP5	153ER
ZS1	146AR
ZS2	1476R
ZS3	148CR
ZS4	1494R
ZT1	1360R
ZT2	136CR
ZT3	1382R
ZT4	138AR



```

*
* BASIC REL-LOADER
*
* COPYRIGHT INTERDATA, INC. JUNE, 1973
*
* REVISED BY: J. PRATT
*
* THIS LOADER OPERATES STAND-ALONE. IT ACCEPTS STANDARD
* LOADER-FORMAT, INTERDATA TYPE M08/09 OR M16/17 FORMATS
* PROGRAM-RELOCATION AND FORWARD-REFERENCE CHAINING IS
* ALLOWED FOR.
* LABEL AND ENTRY ITEMS ARE IGNORED
* EXTRN ITEMS HALT, BUT ARE IGNORED ON RESTART
* COMMON ITEMS HALT, WITH NO RESTART AVAILABLE
*
*
* FOR EXECUTION WITH CASSETTE OR MAG TAPE LIBRARIES - TO POSITION
* TAPE FOR LOADING - ENTER THE NUMBER OF FILEMARKS TO BE SKIPPED ON
* THE MAINTENANCE PANEL. WHEN NO SKIPPING IS DESIRED, THE MSB SHOULD
* BE X'00' TO AVOID INADVERTANT SKIPPING.
*
* ERROR-STOP DISPLAYS:
* 01 CHECKSUM ERROR
* 02 SEQUENCE-ERROR
* 03 ATTEMPT TO LOAD OVER LOADER
* 04 REF-CHAIN LOOP
* FX X-CONTROL-ITEM ENCOUNTERED
*
*
* START THE LOADER AT ORG TO LOAD AT 80
* TO REDEFINE BIAS, INSERT NEW BIAS AT ORG+A AND
* EXECUTE AT ORG+8
* CONTINUE IS AT ORG+26

```

```

0000 R0 EQU 0
0001 R1 EQU 1
0002 R2 EQU 2
0003 R3 EQU 3
0004 BYTE EQU 4
0005 PICK EQU 5
0006 SEGNUM EQU 6
0007 ONE EQU 7
0008 TWO EQU 8
0009 FOUR EQU 9
000A A EQU 10
000B B EQU 11
000C C EQU 12
000D D EQU 13
000E E EQU 14
000F ABSF EQU 15
0078 BINDV EQU X'78'
* REDEFINE SOME REGISTERS
0002 RTN EQU R2
000D DEV EQU D
000A AC1 EQU A

```

```

LINK REGISTER
DEVICE ADDRESS

```

```

RFL0003
REL0004
RFL0004
RFL0004
RFL0004
RFL0004
REL0005
RFL0006
REL0007
REL0008
RFL0009
REL0010
RFL0011
RFL0012
RFL0013
RFL0014
REL0015
REL0016
RFL0017
REL0018
REL0019
RFL0020
REL0021
RFL0022
REL0023
RFL0024
RFL0025
RFL0026
REL0027
REL0028
RFL0029
REL0030
RFL0031
RFL0032
REL0033
REL0034
REL0035
RFL0036
REL0037
REL0038
RFL0039
REL0040
RFL0041
RFL0042
REL0043
RFL0044
RFL0045
RFL0046
REL0047
RFL0048
RFL0049
RFL0050
RFL0051
RFL0052
RFL0053

```

Address	Label	Op	Op2	Op3	Description	Reloc
0001	DAT	EQU	R1		INPUT DATA & STATUS	REL00540
000E	CBA	EQU	E		CUR BYTE ADRS POINTER	REL00550
000B	CRB	EQU	B		TEMP STOR FOR BYTE ASMB	REL00560
000C	ZERO	EQU	C			REL00570
	*					REL00580
0000R	C8A0	START	LHI	A,X'80'	INITIALIZE LOC,BIAS	REL00590
	0080					
0004R	4300		B	**+B		REL00600
	000CR					
0008R	C8A0	REDEF	LHI	A,X'80'	BIAS REDEFINITION	RFL00610
	0080					
0000CR	40A0		STH	A,LOC		RFL00620
	0384R					
0010R	40A0		STH	A,BIAS		REL00630
	0388R					
0014R	40A0		STH	A,PTOP	RESET PTOP TO BIAS	REL00640
	038AR					
0018R	0BAA		SHR	A,A	CLEAR EXECUTE ADRS	RFL00650
001AR	40A0	CLEAR	STH	A,LOCX		RFL00660
	0382R					
001ER	4200		NOP			REL00661
	0000					
0022R	4200		NOP			RFL00662
	0000					
0026R	0B66	CONT	SHR	SEQNUM,SEQNUM	CLEAR SEQNUM	REL00670
0028R	0BFF		SHR	ABSF,ABSF	SET REL MODE	RFL00680
002AR	C870		LHI	ONE,1	SET CONSTANTS 1,2,4	REL00690
	0001					
002ER	C880		LHI	TWO,2		REL00700
	0002					
0032R	C890		LHI	FOUR,4		RFL00710
	0004					
	*	SKIP	NO. FILEMARKS	ENTERED ON SWITCHES (MSB)		RFL00720
0036R	D3D0	LB	DEV,RINDV	GET DEVICE ADDRESS		RFL00730
	0078					
003AR	080D	LHR	R0,DEV	DEVICE ADDRESS		RFL00740
003CR	0409	NHR	R0,FOUR	MAGNETIC DEVICE ?		REL00745
003ER	4330	BZ	NEXT	NO		REL00750
	005ER					
0042R	4130	BAL	R3,MTCHK	CHECK MOTION - ETC.		RFL00760
	035AR					
0046R	997B	RHR	ONE,CRB	GET NUMBER OF FILE MARKS		REL00770
0048R	C4B0	NHI	CRB,X'00FF'	REMOVE LSB		REL00780
	00FF					
004CR	4330	BZ	NEXT	IF ZERO - NO SKIP		RFL00790
	005ER					
0050R	DED0	SKPFM	OC	DEV,SKIP	SKIP COMMAND	REL00800
	0390R					
0054R	4130	BAL	R3,MTCHK	CHECK MOTION - ETC.		REL00810
	035AR					
0058R	0BB7	SHR	CRB,ONE	DECREMENT FILE MARK COUNT		REL00820
005AR	4230	BNZ	SKPFM	CONTINUE		REL00830
	0050R					
005ER	0B67	NEXT	SHR	SEQNUM,ONE	DECR SEQ COUNT	REL00840
0060R	4120	BAL	R2,INPUT	INPUT ONE RECORD		REL00850
	024CR					

0064R	07CC		XHR	C,C	CLEAR FOR CHECKSUM	REL00860
0066R	C8A0		LHI	A,106	COMPUTE CHECKSUM	REL00870
	006A					
006AR	47CA	CKIT	XH	C,BUFF(A)	BY XH OF EVERY HW OF BUFR	REL00880
	0392R					
006ER	08A8		SHR	A,TWO	AND WHEN DONE,	REL00890
0070R	4310		BNM	CKIT		REL00900
	006AR					
0074R	C7C0		XHI	C,-1	WITH FFFF. RESULT SHOULD	REL00910
	FFFF					
0078R	4230		BNZ	CERR	BE ZERO.	REL00920
	00E8R					
007CR	4560		CLH	SEQNUM,BUFF	COMPARE TO SEQ NUM	REL00930
	0392R					
0080R	4230		BNE	SERR		REL00940
	00FOR					
0084R	C850		LHI	PICK,BUFF+4	ADJUST PICK,BYTE	REL00950
	0396R					
0088R	C840		LHI	BYTE,12		REL00960
	000C					
		*				
008CR	C550	LOOP	CLHI	PICK,BUFF+108	TEST IF RECORD DONE	REL00970
	03FER					REL00980
0090R	4380		BNL	NEXT		REL00990
	005ER					
0094R	48A5		LH	A,0(PICK)	EXTRACT NEXT COMMAND	REL01000
	0000					
0098R	4110		BAL	R1,EXTR		REL01010
	0236R					
009CR	08EA		LHR	E,A	SAVE ITEM FOR ERR STOP	REL01020
009ER	0AAA		AHR	A,A		REL01030
00A0R	48BA		LH	B,JUMP(A)	GO TO COMMAND ROUTINE	REL01040
	00A6R					
00A4R	030B		BR	B		REL01050
		*				
00A6R	005ER	JUMP	DC	NEXT,END,CHAIN,FLIP		REL01060
	010ER					REL01070
	01EER					
	012CR					
00AER	0144R		DC	LDX,LDL,RFIN,DFIN		REL01080
	0150R					
	01B8R					
	01C4R					
00B6R	0178R		DC	UNAB,UNRL,DUAB,DURL		REL01090
	0180R					
	0198R					
	01B0R					
00BER	01D8R		DC	RBCD,DBCD,EITM,LABEL		REL01100
	01D0R					
	00C6R					
	01D0R					
		*				
00C6R	48A5	EITM	LH	A,0(PICK)	FETCH SECOND DIGIT	REL01110
	0000					REL01120
00CAR	4110		BAL	R1,EXTR	OF EX CONTROL ITEMS	REL01130
	0236R					

00CER 05A9		CLHR	A,FOUR	IF IT'S E4, ZERO SEQ	REL01140
00DOR 4230		BNE	LERR	OTHERWISE, LOAD-ERR	REL01150
00DAR 00DAR					
00D4R 0766		XHR	SEQNUM,SEQNUM		REL01160
00D6R 4300		B	LOOP		REL01170
008CR					
00DAR C8BE	LERR	LHI	B,X'F0'(E)	FETCH ERROR DISPLAY	REL01180
00F0					
00DER 9A7B		WDR	ONE,B	SHOW ERROR AND STP	REL01190
00EOR C200		LPSW	**4		REL01200
00E4R 00E4R					
8000		DC	X'8000',LOOP		REL01210
008CR					
	*				
00E8R C8A0	CERR	LHI	A,1		REL01220
0001					REL01230
00ECR 4300		B	ERROR		REL01240
0104R					
00FOR C8A0	SERR	LHI	A,2		REL01250
0002					
00F4R 4300		B	ERROR		REL01260
0104R					
00F8R C8A0	ADER	LHI	A,3		REL01270
0003					
00FCR 4300		B	ERROR		REL01280
0104R					
0100R C8A0	REL P	LHI	A,4		REL01290
0004					
0104R 9A7A	ERROR	WDR	ONE,A	SHOW ERROR STOP	REL01300
0106R C200		LPSW	**4		REL01310
010AR 010AR					
8000		DC	X'8000',A(NEXT+2)		REL01320
0060R					
010ER 4800	END	LH	R0,PTOP	FETCH TOP OF LOADED CODE	REL01330
038AR					
0112R 4000		STH	R0,BIAS	BUMP BIAS	REL01340
0388R					
0116R 4000		STH	R0,LOC	AND LOC COUNTER	REL01350
0384R					
011AR 07AA		XHR	A,A	GET A ZERO	REL01360
011CR 9A7A		WDR	ONE,A	DISPLAY NORM END CO	REL01370
011ER 48A0		LH	A,LOCX	FETCH XFER ADRS	REL01380
0382R					
0122R 023A		BNZR	A	XFER IF NON-ZERO	REL01390
0124R C200		LPSW	**4		REL01400
0128R 0128R					
8000		DC	X'8000',CONT		REL01410
0026R					
	*				
012CR C7F0	FLIP	XHI	ABSF,X'FFFF'	FLIP THE ABS FLAG	REL01420
FFFF					REL01430
0130R 48A0		LH	A,LOC	FLIP LOC COUNTERS	REL01440
0384R					
0134R 48B0		LH	B,LOC+2		REL01450
0386R					
0138R 40A0		STH	A,LOC+2		REL01460

013CR	0386R 40B0		STH	B,LOC		REL01470
0140R	0384R 4300 008CR		B	LOOP		REL01480
0144R	4130	* LDX	BAL	R3,GETT	SET EXECUTION ADRS	REL01490 REL01500
0148R	020ER 40D0		STH	D,LOCX		REL01510
014CR	0382R 4300 008CR		B	LOOP		REL01520
0150R	4130	* LDL	BAL	R3,GETT	SET LOAD LOCATION	REL01530 REL01540
0154R	020ER C5D0	LDLO	CLHI	D,START	CHECK FOR LOAD-LOC	REL01550
0158R	0000R 4280		BL	LDLX	OVERLAPPING LOADER	REL01560
015CR	0164R C5D0		CLHI	D,LTOP		REL01570
0160R	0400R 4280		BL	ADER		REL01580
0164R	00F8R 40D0	LDLX	STH	D,LOC		REL01590
0168R	0384R 45D0	LDL1	CLH	D,PTOP		REL01600
016CR	038AR 4280		BL	LOOP		REL01610
0170R	008CR 40D0		STH	D,PTOP	UPDATE PTOP IF NEC.	REL01620
0174R	038AR 4300 008CR		B	LOOP		REL01630
0178R	4120	* UNAB	BAL	R2,WORD	LOAD 2 BYTES ABS	REL01640 REL01650
017CR	021ER 4300		B	UNRX		REL01660
0180R	0188R 4120	UNRL	BAL	R2,WORD	LOAD 2 BYTES REL	REL01670
0184R	021ER 4AD0		AH	D,BIAS		REL01680
0188R	0388R 48C0	UNRX	LH	C,LOC		REL01690
018CR	0384R 40DC		STH	D,0(C)		REL01700
0190R	0000 0AC8		AHR	C,TWO	BUMP LOAD LOCATION	REL01710
0192R	08DC		LHR	D,C		REL01720
0194R	4300 0154R		B	LDL0		REL01730
0198R	C8E0	* DUAB	LHI	E,UNAB	LOAD 4 BYTES ABS	REL01740 REL01750
019CR	0178R 4120	DU	BAL	R2,WORD		REL01760
01A0R	021ER 48C0		LH	C,LOC		REL01770

01A4R	0384R 40DC	STH	D,0(C)		REL01780
01A8R	0000 0AC8	AHR	C,TWO		REL01790
01AAR	40C0	STH	C,LOC		REL01800
01AER	0384R 030E	BR	E		REL01810
01BOR	C8E0	* DURL	LHI	E,UNRL	LOAD 4 BYTES REL
01B4R	0180R 4300 019CR	B	DU		REL01840
01B8R	4130	* RFIN	BAL	R3,GETT	GET REF VALUE
01BCR	020ER 40D0	STH	D,REF		REL01870
01COR	037ER 4300 008CR	B	LOOP		REL01880
01C4R	4130	* DFIN	BAL	R3,GETT	GET DEF VALUE
01C8R	020ER 40D0	STH	D,DEF		REL01910
01CCR	0380R 4300 008CR	B	LOOP		REL01920
01D0R	C8B0	* DRCD	LHI	B,LOOP	REL01930
01D4R	008CR 4300	B	SKIPD		REL01940
01D8R	01E0R C8B0	RBCD	LHI	B,LERR	REL01950
01DCR	00DAR 4300 01E0R	B	SKIPD		REL01960
01D0R	01E0R	LABEL	EQU	D8CD	REL01970
01E0R	4120	SKIPD	BAL	R2,WORD	REL01980
01E4R	021ER 4120	BAL	R2,WORD	SKIP DATA-ITEMS	REL01990
01E8R	021ER 4120	BAL	R2,WORD		REL02000
01ECR	021ER 030B	BR	B		REL02010
01EER	48D0	* CHAIN	LH	D,DEF	REL02020
01F2R	0380R 48E0	LH	E,REF	DEFINE CHAIN:	REL02030
01F6R	037ER 48CE	CH1	LH	C,0(E)	REL02040
01FAR	0000 40DE	STH	D,0(E)	FOLLOW THE THREAD	REL02050
01FER	0000	LHR	E,C	AND DEF EACH REF	REL02060
0200R	4330	BZ	LOOP	AND THIS IS A REF-LOOP.	REL02070
0204R	008CR 05CD	CLHR	C,D	BUT IF A REF=A DEF,	REL02080
					REL02090
					REL02100

0206R	4330		BE	RELP	THEN WE'VE BEEN HERE BEFORE	REL02110
	0100R					
020AR	4300		B	CH1		REL02120
	01F6R					
		*				REL02130
020ER	4120	GETT	BAL	R2,WORD	GET 2 BYTES OF DATA	REL02140
	021ER					
0212R	08FF		LHR	ABSF,ABSF	AND ADD BIAS TO IT	REL02150
0214R	4233		HNZ	0(R3)	IF IN REL MODE	REL02160
	0000					
0218R	4AD0		AH	D,BIAS		REL02170
	0388R					
021CR	0303		BR	R3		REL02180
		*				REL02190
021ER	08C9	WORD	LHR	C,FOUR	ASSEMBLE 1 WORD OR	REL02200
0220R	48A5	WORD1	LH	A,0(PICK)	TWO BYTES OF DATA	REL02210
	0000					
0224R	4110		BAL	R1,EXTR	INTO REG D.	REL02220
	0236R					
0228R	CDD0		SLHL	D,4		REL02230
	0004					
022CR	06DA		OHR	D,A		REL02240
022ER	08C7		SHR	C,ONE		REL02250
0230R	4230		BNZ	WORD1		REL02260
	0220R					
0234R	0302		BR	R2		REL02270
		*				REL02280
0236R	CCA4	EXTR	SRHL	A,0(BYTE)	EXTRACT ONE FOUR BIT	REL02290
	0000					
023AR	C4A0		NHI	A,X'F'	BYTE FROM THE DATA	REL02300
	000F					
023ER	0B49		SHR	BYTE,FOUR	IN REG A.	REL02310
0240R	4311		BNM	0(R1)		REL02320
	0000					
0244R	C840		LHI	BYTE,12	UPDATE PICK AND BYTE	REL02330
	000C					
0248R	0A58		AHR	PICK,TWO		REL02340
024AR	0301		BR	R1		REL02350
		*				REL02360
		* INPUT ROUTINE				REL02370
		*				REL02380
		*				REL02390
		* CALL	BAL	RTN,INPUT		REL02400
024CR	D3D0	INPUT	LB	DEV,BINDV	FETCH INPUT DEV ADRS	REL02410
	0078					
0250R	0830		LHR	R3,DEV	DEVICE ADDRESS	REL02420
0252R	0437		NHR	R3,ONE	IF TTY, SELECT WRITE	REL02425
0254R	4230		BNZ	IN1	HSPTRP, MT OR CAS	REL02430
	026CR					
0258R	DED0		OC	DEV,TWRT	AND OUTPUT X-ON	REL02440
	037AR					
025CR	4240		RTC	4,*-4	REPEAT IF FSYN	REL02450
	0258R					
0260R	4130		BAL	R3,DEVCHK	CHECK	REL02460
	034ER					
0264R	DAD0		WD	DEV,XON		REL02470

0268R	037CR 4240		BTC	4,*-4	REPEAT WRT IF FSYN.	REL02480
026CR	0264R DED0	IN1	OC	DEV,BINDV+1	START DEVICE IN READ-	REL02490
0270R	0079 4240		BTC	4,*-4	MODE (LOOP IF FSYN)	REL02500
0274R	026CR 083D		LHR	R3,DEV	DEVICE ADDRESS	REL02510
0276R	0439		NHR	R3,FOUR	MT / CAS ?	REL02515
0278R	4230		BNZ	MTCAS		REL02520
027CR	0316R 07EE	IN2	XHR	CBA,CBA		REL02530
027ER	40E0		STH	CBA,CFLG	AND BUFFER-INDEX	REL02540
0282R	0376R 40E0		STH	CBA,BFLG		REL02550
0286R	0378R 08CE		LHR	ZERO,CBA		REL02560
0288R	4130	* READIT	BAL	R3,DEVCHK	CHECK FOR NOT BUSY	REL02570 REL02580
028CR	034ER 9B01		RDR	DEV,DAT	READ A BYTE	REL02590
028ER	4240		BTC	4,*-2	REPEAT IF FSYN	REL02600
0292R	028CR 9A71		WDR	ONE,DAT	SHOW DATA READ	REL02610
0294R	48A0		LH	AC1,CFLG	TEST C FLAG	REL02620
0298R	0376R 4220		BP	STORE	IF + ITS M16/17 FORMAT	REL02630
029CR	02E4R 4210		BM	TESTB	IF - ITS M08/09 FORMAT	REL02640
02A0R	02B0R C510		CLHI	DAT,X'F0'	ELSE, WAIT & SEE	REL02650
02A4R	00F0 4230		BNE	TESTB	IF X'F0' SET C +	REL02660
02A8R	02B0R 4070		STH	ONE,CFLG		REL02670
02ACR	0376R 4300		B	READIT	AND READ IN BUFFER	REL02680
02B0R	0288R 08A1	TESTB	LHR	AC1,DAT	OTHERWISE SEE IF	REL02690
02B2R	C4A0		NHI	AC1,15	LEGAL M08/09 CHAR	REL02700
02B6R	000F D3AA		LB	AC1,ZTAB(AC1)	FETCH TABLED-ENTRY	REL02710
02BAR	0366R 051A		CLHR	DAT,AC1	AND IS IT = DATA?	REL02720
02BCR	4230		BNE	READIT	IF NOT, IGNORE	REL02730
02C0R	0288R 48A0		LH	AC1,BFLG	IF SO, TEST B FLAG	REL02740
02C4R	0378R 4230		BNZ	ASMB	IF SET, THIS IS AN ODD	REL02750
02C8R	02DAR 4070		STH	ONE,BFLG	HALF-BYTE SO ASSEMBLE	REL02760
02CCR	0378R 08B1		LHR	CRB,DAT	IF RESET, SAVE EVEN-	REL02770
02CER	CDB0		SLHL	CRB,4	HALF-BYTE AND SET B	REL02780
02D2R	0004 4060		STH	SEQNUM,CFLG	IF LEGAL ZONES, SET C -	REL02790



02D6R	0376R 4300		B	READIT		REL0280
02DAR	0288R 40C0	ASMB	STH	ZERO,BFLG	ON ODD-ITEMS, RESET B	REL0281
02DER	0378R C410		NHI	DAT,15	AND ASSEMBLE A WHOLE	REL0282
02E2R	000F 061B		OHR	DAT,CRB	BYTE	REL0283
02E4R	021E	STORE	STB	DAT,BUFF(CBA)	PUT IT IN BUFFER	REL0284
02E8R	0392R 0AE7		AHR	CBA,ONE	BUMP THE POINTER	REL0285
02EAR	C5E0		CLHI	CBA,108	THRU YET???	REL0286
02EER	006C 4280		BL	READIT	NO, GET SOME MORE	REL0287
02F2R	0288R 083D		LHR	R3,DEV	DEVICE ADDRESS	REL0288
02F4R	0437		NHR	R3,ONE	HSPTRP ?	REL0288
02F6R	4230		BNZ	DVROUT		REL0289
02FAR	0310R DED0	TTYOFF	OC	DEV,TWRT	WRITE COMMAND	REL0290
02FER	037AR 4240		BTC	4,*-4	WAIT FOR A CHANCE	REL0291
0302R	02FAR 9DD1		SSR	DEV,DAT	AND SEND X-OFF	REL0292
0304R	42F0		BTC	15,*-2		REL0293
0308R	0302R DAD0		WD	DEV,XOFF		REL0294
030CR	037DR 4240		BTC	4,*-4		REL0295
0310R	0308R C880	DVROUT	LHI	TWO,2	RESTORE RETRY COUNTER	REL0296
0314R	0002 0302		BR	R2	RETURN	REL0297
0316R	9DD1	MTCAS	SSR	DEV,DAT	COMMAND ACCEPTED ?	REL0298
0318R	4320		BNP	MTCAS1	YES ,EOM DROPPED	REL0299
031CR	0324R 4130		BAL	R3,MTCHK	NO ,WAIT FOR NMTN	REL0300
0320R	035AR 4300		B	IN1	RETRY	REL0301
0324R	026CR D7D0	MTCAS1	RB	DEV,BUFBK	READ A BLOCK	REL0302
0328R	038CR 4130		BAL	R3,DEVCHK	CHECK FOR ERRORS ON MT / CASSETTE	REL0303
032CR	034ER 4320		BNP	DEVCHK	WAIT FOR EOM TO SET	REL0304
0330R	034ER C410		NHI	DAT,X'A0'	CHECK ERR AND ET BITS	REL0305
0334R	00A0 4330		BZ	DVROUT	NOT SET - RETURN	REL0306
0338R	0310R 4130		BAL	R3,MTCHK	WAIT FOR NO MOTION	REL0307
033CR	035AR DED0		OC	DEV,BAKREC	BACKSPACE A RECORD	REL0308
0340R	037CR 4130		BAL	R3,MTCHK	WAIT FOR NO MOTION	REL0309
	035AR					



NO ERRORS

A	000A
ABSF	000F
AC1	000A
ADER	00F8R
ASMB	02DAR
B	000B
BAKREC	037CR
BFLG	0378R
BIAS	0388R
BINDV	0078
BUFFLK	038CR
BUFF	0392R
BYTE	0004
C	000C
CBA	000E
CERR	00E8R
CFLG	0376R
CH1	01F6R
CHAIN	01EER
CKIT	006AR
CLEAR	001AR
CONT	0026R
CRB	000B
D	000D
DAT	0001
DRCU	0100R
DEF	038UR
DEV	000D
DEVCHK	034ER
DFIN	01C4R
DU	019CR
DUAB	0198R
DURL	0180R
DVROUT	0310R
E	000E
EITM	00C6R
END	010ER
ERROR	0104R
EXTR	0236R
FLIP	012CR
FOUR	0009
GETT	020ER
IN1	026CR
IN2	027CR
INPUT	024CR
JUMP	00A6R
LABEL	01D0R
LDL	0150R
LDL0	0154R
LDL1	0168R
LDLX	0164R
LDX	0144R
LERR	00DAR
LOC	0384R
LOCX	0382R

LOOP	008CR
LTOP	0400R
MTCAS	0316R
MTCAS1	0324R
MTCHK	035AR
MTCLR	0391R
MTERR	034AR
NEXT	005ER
ONE	0007
PICK	0005
PTOP	038AR
R0	0000
R1	0001
R2	0002
R3	0003
RBCD	0108R
READIT	0288R
REDEF	0008R
REF	037ER
RELP	0100R
RFIN	0188R
RTN	0002
SEQNUM	0006
SERR	00F0R
SKIP	0390R
SKIPD	01E0R
SKPFM	0050R
START	0000R
STORE	02F4R
TESTB	0280R
TTYOFF	02FAR
TWO	0008
TWRT	037AR
UNAB	0178R
UNRL	0180R
UNRX	0188R
WORD	021ER
WORD1	0220R
XOFF	037DR
XON	037CR
ZERO	000C
ZTAB	0366R

LGC 184ER must be  
 Changed to 0A00 as 306  
 printer has no form feed  
 change 184E to 0A0A  
 & next 3 to same  
 so that FEED gets 10h

0000R	* COPYRIGHT, INTERDATA, INC. MARCH 1972	ASM00025
	ENTRY PASS1,PASS22,PASS23,PASS33	ASM00030
	* REGISTER DEFINITIONS	ASM00040
	* CONSTANTS	ASM00050
0000	ZERO EQU 0	ASM00060
0001	ONE EQU 1	ASM00070
0002	TWO EQU 2	ASM00080
	* ADDRESS	ASM00090
0003	GET EQU 3 GETCHR	ASM00100
	* POINTER INTO SOURCE STATEMENT	ASM00110
0004	SRX EQU 4	ASM00120
	* TEMPORARY USAGE	ASM00130
0005	R EQU 5	ASM00140
0006	REG EQU 6	ASM00150
0007	HOLD EQU 7	ASM00160
0008	TEMP EQU 8	ASM00170
	* GENERAL COUNTER	ASM00180
0009	COUNT EQU 9	ASM00190
	* GENERAL POINTER	ASM00200
000A	P EQU 10	ASM00210
	* CHARACTER FROM SOURCE STATEMENT	ASM00220
000B	CHAR EQU 11	ASM00230
000C	VALUE EQU 12	ASM00240
	* LINKAGE REGISTERS	ASM00250
000D	RETURN EQU 13	ASM00260
000E	LINK EQU 14	ASM00270
000F	EXIT EQU 15	ASM00280
	* REGISTER REDEFINITION	ASM00290
0004	STX EQU SRX	ASM00300
000B	NXTSYM EQU CHAR	ASM00310
	SYMBOL TABLE DUMP INDEX AND POINTER TO NEXT SYMROL	ASM00320
	* SYMBOL'S "SYMBOL TABLE ENTRY" CODE BYTE BIT DEFINITIONS	ASM00330
0200	DEFBIT EQU X'0200'	ASM00340
1000	USEDDBT EQU X'1000'	ASM00350
0100	RELBIT EQU X'0100'	ASM00360
0800	ENRBIT EQU X'0800'	ASM00370
0400	EXTBIT EQU X'0400'	ASM00380
0600	UNDEFX EQU X'0600'	ASM00390
0E00	XENTDF EQU X'0E00'	ASM00400
0C00	EXTENT EQU X'0C00'	ASM00410
	*****	ASM00420
	* ASSEMBLER MAIN FLOW - PASS & OPTION CONTROL	ASM00430
	*****	ASM00440
	* "PASS1" IS THE EXECUTION ENTRY POINT FOR	ASM00450
	* PASS ONE ASSEMBLY OPERATIONS OF EITHER	ASM00460
	* SINGLE OR MULTIPLE PASS ASSEMBLIES	ASM00470
0000R E120	PASS1 SVC 2,SIZES	ASM00480
18A0R		
0004R 4855	LH R,0(R)	ASM00490
0000		
0008R 4050	STH R,SIZE	ASM00500
162ER		
000CR D150	LM 5,OPTS	ASM00510
1638R		
	* SET ASSUMED OPTIONS=PASS 1 OF A PASS2 ASSEMBLY	ASM00520
	* STOP,PRINT,PUNCH,NO SCRATCH,NO SEQUENC CHECK,FLOATING POINT	ASM00530

		* SYMBOL TABLE POINTERS SET =SYMTAB,AND FIRST NULL	ASM00540
		* ENTRY SET = X'8000'	ASM00550
0010R	D070	STM 7,OPTION	ASM00560
	161CR		
0014R	406F	STH 6,0(15) STORE X'8000' BEGIN SYMTAB	ASM00570
	0000		
0018R	D250	STB 5,SRBLK+1 ON PASS1 SET SR LU =1	ASM00580
	18A5R		
001CR	4300	B BEGIN1	ASM00590
	003FR		
		* "PASS22" IS THE EXECUTION ENTRY POINT FOR	ASM00600
		* PASS TWO OF A TWO-PASS ASSEMBLY	ASM00610
0020R	C860	PASS22 LHI REG,2 2-PASS ASSEMBLY	ASM00620
	0002		
0024R	4300	B PASS33+4 SETUP CURRENT PASS=2	ASM00630
	0038R		
		* "PASS23" IS THE EXECUTION ENTRY POINT FOR	ASM00640
		* PASS TWO OF A THREE-PASS ASSEMBLY	ASM00650
0028R	C860	*PASS23 LHI REG,3 3-PASS ASSEMBLY	ASM00660
	0003		
002CR	C850	LHI R,2 SETUP CURRENT PASS=2	ASM00670
	0002		
0030R	4300	B BEGIN	ASM00680
	003AR		
		* "PASS33" IS THE EXECUTION ENTRY POINT FOR	ASM00690
		* PASS THREE OF A THREE-PASS ASSEMBLY	ASM00700
0034R	C860	*PASS33 LHI REG,3 3-PASS ASSEMBLY	ASM00710
	0003		
0038R	0856	LHR R,REG SETUP CURRENT PASS=2	ASM00720
003AR	4060	BEGIN STH REG,OPASS SET TOTAL NO OF PASSES	ASM00730
	161CR		
003ER	41E0	BEGIN1 BAL LINK,PASMSG WRITE PASS MESSAGE	ASM00740
	0046R		
0042R	4300	B GO SKIP PAUSE ON PASS1/OR RESTART	ASM00750
	0068R		
		* INITIALIZATION REQUIRED PRIOR TO EACH ASSEMBLY PASS:	ASM00760
0046R	4050	PASMSG STH R,PASS SET CURRENT PASS NUMBER	ASM00770
	161AR		
004AR	C855	LHI R,X'30'(R) SET IN PASS NUMBER MESSAGE	ASM00780
	0030		
004ER	D250	STB R,PASNUM	ASM00790
	18F5R		
0052R	E120	SVC 2,PASBLK REQUEST "PASS #" MESSAGE	ASM00800
	18DCR		
0056R	030E	BR LINK RETURN TO MAIN FLOW	ASM00810
		* RE-ENTRY POINT BETWEEN PASSES OF MULTIPLE ASSEMBLIES.	ASM00820
0058R	41E0	BEGINP BAL LINK,PASMSG WRITE PASS MESSAGE	ASM00830
	0046R		
005CR	4850	LH R,OPAUSE IS "STOP" OPTION IN EFFECT?	ASM00840
	161ER		
0060R	4330	BZ GO NO, GO ON.	ASM00850
	0068R		
0064R	F120	SVC 2,SUSPND PAUSE BETWEEN PASSES	ASM00860
	18D6R		
0068R	D100	GO LM 0,PSETUP	ASM00870
	1662R		

				* INITIALIZE GENERAL REGISTERS ZERO, ONE AND TWO TO 0,1,2	ASM00880	
				* INITIALIZE MODE=1, ASSEM=2, LCOUNT=0, DOC=0, LOC=0, MAXLOC=0	ASM00890	
				* FLIPS=0, SEQNUM=SPACES, DOUBLE PUNCH BUFFER SWITCH AT	ASM00900	
				* PBSWCH=PB1/PB2, AND FIRST OBJECT RECORD SEQNUM=-1	ASM00910	
				* PAGE NO.= 001, PSTART AND LSTART =0	ASM00920	
				* ASSEMBLY LISTING ERRORS=0 AND BLANK OUT HEADER LINE.	ASM00925	
006CR	D000		STM	ZERO, PSTART	SET 16 HALFWORDS=PSETUP	ASM00930
	1684R					
0070R	4000		STH	ZERO, LSTART	NO LISTING I/O YET	ASM00940
	1682R					
0074R	4000		STH	ZERO, ERRCNT	NO LISTING ERRORS YET	ASM00950
	1912R					
0078R	40F0		*STH	15, PAGE	INITIALIZE PAGE NO.=1	ASM00960
	189CR					
007CR	D280		*STB	8, PAGE-1	INITIALIZE HUNDREDTHS POSITION	ASM00970
	189BR					
0080R	41F0		*BAL	EXIT, NTITLE	BLANK OUT TITLE RECORD	ASM00980
	051CR					
				* CLEAR RELOCATABLE BIT, AND VALUE OF ALL UNDEFINED SYMBOLS	ASM00990	
				* AT BEGINNING OF EACH PASS.	ASM01000	
0084R	48A0		LH	P, STO	SYMBOL TABLE ORIGIN	ASM01010
	162AR					
0088R	0BA2		SHR	P, TWO	TEMPORARY DECREMENT	ASM01020
008AR	0AA2	LOOPSM	AHR	P, TWO	BUMP INDEX INTO SYMBOLS	ASM01030
008CR	45A0		CLH	P, STE	EQUAL TO SYMBOL TABLE END?	ASM01040
	162CR					
0090R	4330		BE	STDONE	YES, FINISHED.	ASM01050
	00B6R					
0094R	485A		LH	R, 0(P)	FETCH CODE BYTE	ASM01060
	0000					
0098R	4310		BNM	LOOPSM	IF NOT MINUS, IN MIDDLE	ASM01070
	00BAR					
009CR	0865		LHR	REG, R	DEF BIT=0, DEFINED	ASM01080
009ER	C460		NHI	REG, DEFBIT	DEFBIT=1, UNDEFINED	ASM01090
	0200					
00A2R	4330		BZ	LOOPSM	LOOP ON DEFINEDS	ASM01100
	00BAR					
00A6R	C450		NHI	R, X'FE00'	STRIP REL BIT AND	ASM01110
	FE00					
00AAR	405A		STH	R, 0(P)	VALUE	ASM01120
	0000					
00AER	D25A		STB	R, 3(P)	CLEAR VALUE (ABS ZERO)	ASM01130
	0003					
00B2R	4300		B	LOOPSM	CONTINUE LOOPING THRU TABLE	ASM01140
	00BAR					
00B6R	C8D0	STDONE	LHI	RETURN, DOLINE	GO DOLINE AFTER ZEROING PB	ASM01150
	011CR					
00BAR	C850	ZEROPB	LHI	R, 104	ZERO THE CURRENT PUNCHBUFF	ASM01160
	0068					
00BER	4A50		AH	R, PBSWCH	PB1 FIRST PUNCH BUFFER	ASM01170
	169CR					
00C2R	4005	CLRPB	STH	ZERO, 2(R)	CLEAR 52 HALFWORDS	ASM01180
	0002					
00C6R	0852		SHR	R, TWO		ASM01190
00C8R	4550		CLH	R, PBSWCH	DON'T CLEAR RECORD SEQNUM	ASM01200
	169CR					

00CCR 4230	RNE	CLRPH		ASM01210
00C2R 0000				
00D0R 4000	STH	ZERO,PBX	RESET PUNCH BUFFER INDEX	ASM01220
1788R				
00D4R 0300	BR	RETURN		ASM01230
	*****			ASM01240
	* "IOWAIT" ISSUES I/O WAIT WITH IMMEDIATE STATUS CHECK			ASM01250
	* OR ISSUES "WAIT" WITH STATUS CHECK ON PREVIOUS			ASM01260
	* I/O AND PROCEED DATA TRANSFER MADE.			ASM01270
	* ON GOOD DATA TRANSFER RETURN TO PROCESSING			ASM01280
	*			ASM01290
	* FOR TROUBLED I/O DUE TO ILLEGAL,OR DEVICE UNAVAILABLE*			ASM01300
	* "IOWAIT" LISTS "XXNN I/O ERROR" MESSAGE ON SYSTEM			ASM01310
	* CONSOLE AND PAUSES FOR OPERATOR INTERVENTION			ASM01320
	* NN = PHYSICAL DEVICE ADDRESS OF TROUBLED I/O			ASM01330
	* XX = ERROR STATUS SENT BACK TO ASSEMBLER BY OS			ASM01340
	* AFTER OPERATOR CONTINUES, "IOWAIT" REISSUES THE			ASM01350
	* SVC I/O CALL IN TROUBLE UNTIL GOOD TRANSFER OCCURS			ASM01360
	*			ASM01370
	* FOR TROUBLED I/O DUE TO EOM,EOF,UNRECOVERABLE ERROR,			ASM01380
	* OR OVERFLOW "IOWAIT" LISTS I/O ERROR MESSAGE AND			ASM01390
	* RETURNS TO CONTINUE PROCESSING BEST GUESS DATA			ASM01400
	*			ASM01410
	* CALLING SEQUENCE FOR STATUS CHECK ON LAST I/O & PROCEED:			ASM01420
	* BAL RETURN,IOWAIT			ASM01430
	DC	A(WAIT PARAMETER BLOCK)		ASM01440
	DC	A(I/O & PROCEED PARAMETER BLOCK)		ASM01450
	*			ASM01460
	* CALLING SEQUENCE FOR I/O WAIT AND CHECK ON STATUS			ASM01470
	* BAL RETURN,IOWAIT			ASM01480
	DC	A(I/O & WAIT PARAMETER BLOCK)		ASM01490
	DC	A(SAME I/O & WAIT PARAMETER BLOCK)		ASM01500
	*			ASM01510
	* (INTEGRITY OF REGISTER ZERO=0000 MAINTAINED FOR			ASM01520
	* ASSEMBLER CONSTANT)			ASM01530
	*****			ASM01540
00D6R 485D	IOWAIT	LH	R,0(RETURN)      FETCH LU/WAIT PARAMETER ADRS	ASM01550
0000				
00DAR 486D		LH	REG,2(RETURN)      FETCH ADRS LAST BLOCK	ASM01560
0002				
00DER E115	SVCALL	SVC	1,0(R)             ISSUE WAIT ON LU	ASM01570
0000				
00E2R 4806		LH	0,2(REG)           GET STATUS TO CHECK	ASM01580
0002				
00E6R 433D		BZ	4(RETURN)          DATA TRANSFER GOOD,GO ON	ASM01590
0004				
00EAR C400		NHI	0,X'7700'          REISSUE SVC ON EOF ONLY	ASM01600
7700				
00EER 4330		BZ	SVCALL	ASM01610
000R				
00F2R 4806		LH	0,2(REG)           GET STATUS AGAIN	ASM01620
0002				
00F6R E120		SVC	2,ERBLK1          REQUEST STATUS PACKING	ASM01630
18E6R				
00FAR E120		SVC	2,ERBLK2          REQUEST I/O ERROR MESSAGE	ASM01640
18EAR				



00FER	E120 18D6R	SVC	2,SUSPND	PAUSE FOR OPERATOR INTERVENTION	ASM01650
0102R	C400 6000	NHI	0,X*6000	ILLEG,DU = BAD TRANSFER	ASM01660
0106R	433D 0004	BZ	4(RETURN)	BAD DATA BUT CONTINUE	ASM01670
010AR	E116 0000	SVC	1,0(REG)	REISSUE TROUBLED I/O	ASM01680
				* DON'T REISSUE WAIT BIT OF TROUBLED I/O TWICE	ASM01690
010ER	0556	CLHR	R,REG	IF EQUAL, ARGUMENT HAS I/O	ASM01700
0110R	4330 00E2R	BE	SVCALL*4	WAIT BIT IMBEDDED	ASM01710
0114R	4300 00DER	B	SVCALL	GO BACK TO CHECK AGAIN	ASM01720
				*****	ASM01730
0118R	E120 18D6R			* ASSEMBLER DIRECTIVES (PSEUDO-OP) PAUSE	ASM01740
		PAUSE	SVC 2,SUSPND	CONTINUE AT DOLINE	ASM01750
				*****	ASM01760
				* ASSEMBLER MAIN FLOW - SOURCE LINE EVALUATION	ASM01770
				*****	ASM01780
				* MAIN FLOW RE-ENTRY POINT TO PROCESS NEXT SOURCE STATEMENT	ASM01790
				* INITIALIZATION REQUIRED PRIOR TO PROCESSING ONE LINE:	ASM01800
				* CLEAR LINE ERROR BYTE FLAGS :S,F,O,R,U,T,M	ASM01810
				* FILL PRINT AND SOURCE BASIC BUFFERS WITH SPACES	ASM01820
011CR	C890 000E	DOLINE	LHI COUNT,14	CLEAR ERROR INDICATORS	ASM01830
0120R	D209 1560R	CLRERR	STB ZERO,ERRORE(COUNT)		ASM01840
0124R	0892	SHR	COUNT,TWO		ASM01850
0126R	4310 0120R	BNM	CLRERR		ASM01860
012AR	4000 15D4R	STH	ZERO,EOL	CLEAR END OF LINE FLAG	ASM01870
				* SAVE CURRENT LOCATION COUNTER	ASM01880
012ER	4850 168ER	LH	R,LOC	LOC= LOC AT BEGINNING OF LINE	ASM01890
0132R	4050 15C8R	STH	R,SAVLOC		ASM01900
				* BETWEEN SOURCE READS WAIT HERE ON SCRATCH	ASM01910
0136R	4850 1626R	LH	R,OPSCRT	IF NOT IN SCRATCH MODE	ASM01920
013AR	4330 014ER	BZ	NOSCRT	SKIP WAIT ON SCRATCH	ASM01930
013ER	4510 161AR	CLH	ONE,PASS	OR IF NOT ON FIRST PASS	ASM01940
0142R	4230 014ER	BNE	NOSCRT	ALSO SKIP WAIT ON SCRATCH	ASM01950
0146R	41D0 00D6R	BAL	RETURN,IOWAIT	CHECK STATUS ON LAST I/O	ASM01960
014AR	18BCR	DC	SCWAIT	WAIT & CHECK LAST I/O PROCEED	ASM01970
014CR	1884R	DC	SCBLK	REISSUE SCRATCH ON BAD STATUS	ASM01980
014ER	4850 168CR	NOSCRT	LH R,DOC	IN MIDST OF DOING "DO"?	ASM01990
0152R	4220 0256R	BP	DECDOC	SKIP READING IN ANOTHER LINE	ASM02000

0156R	C8F0	READIN	LHI	EXIT,READ		ASM02010
	021ER					
015AR	C890		LHI	COUNT,SREND-PR		ASM02020
	005E					
015ER	C850	BLKOUT	LHI	R,X'2020'	BLANK OUT PR AND SR	ASM02030
	2020					
0162R	4059		STH	R,PR(COUNT)		ASM02040
	178AR					
0166R	0B92		SHR	COUNT,TWO		ASM02050
0168R	4310		BNM	BLKOUT+4		ASM02060
	0162R					
016CR	030F		BR	EXIT		ASM02070
						ASM02080
016ER	C5B0	FINDSP	CLHI	CHAR,X'20'	FIND A SPACE PRIOR TO END	ASM02090
	0020					
0172R	4330		BE	SPACER		ASM02100
	0186R					
0176R	C5B0		CLHI	CHAR,X'0D'	IF AT END	ASM02110
	000D					
017AR	033E		BER	LINK	HIT END BEFORE A SPACE	ASM02120
017CR	D210		STB	ONE,ERRORF	NOSPACE AFTER LABEL OR OP	ASM02130
	156CR					
0180R	01F3		BALR	EXIT,GET	BYPASS ILLEGAL CHARACTER	ASM02140
0182R	4300		B	FINDSP	SEEK A SPACE TO CONTINUE	ASM02150
	016ER					
0186R	0799	SPACER	XHR	COUNT,COUNT	CHAR HOLDS A SPACE	ASM02160
0188R	0A91	LOOPB	AHR	COUNT,ONE	SPACEOVER TO 1ST NON-BLANK	ASM02170
018AR	C590		CLHI	COUNT,16	PRIOR TO END	ASM02180
	0010					
018ER	033F		BER	LINK	TREAT SPAN OF 15 SPACES=CR	ASM02190
0190R	01F3		BALR	EXIT,GET		ASM02200
0192R	C5B0		CLHI	CHAR,X'20'		ASM02210
	0020					
0196R	4330		BE	LOOPB		ASM02220
	0188R					
019AR	C5B0		CLHI	CHAR,X'0D'	END OF LINE	ASM02230
	000D					
019ER	033E		BER	LINK		ASM02240
01A0R	430E		B	4(LINK)	FOUND 1ST NON-BLANK CHAR	ASM02250
	0004					
						ASM02260
01A4R	C5B0	TRYSPC	CLHI	CHAR,X'20'	FIRST CHARACTER = SPACE?	ASM02270
	0020					
01A8R	4230		BNE	INERR	NO	ASM02280
	0188R					
01ACP	41E0		BAL	LINK,SPACER		ASM02290
	0186R					
01B0R	4300		B	PRINTR	PRINT BLANK LINE	ASM02300
	0CC6R					
01B4R	4300		B	COMAND	NO LABEL TRY FOR COMMAND	ASM02310
	0292R					
						ASM02320
01B8R	41E0	INERR	BAL	LINK,FINDSP		ASM02330
	016ER					
01RCR	4300		B	RESERR	BAD LABEL & NO COMMAND	ASM02340
	01D0R					

01C0R	4300 0292R	B	COMAND	BAD LABEL,BUT TRY FOR OP	ASM02350
01C4R	41F0 09E8R	LONLY	BAL	EXIT,EDITC	EDIT SOLITARY LABEL ON LINE
01C8R	4300 0CC6R	B	PRINTR	PRINT, THEN GOTO DOLINE	ASM02380
01CCR	D210 1568R	OPERR	STB	ONE,ERRORO	ASM02390 ASM02400
01D0R	41F0 09E8R	RESERR	BAL	EXIT,EDITC	EDIT LABELW/VALUE=LOCOUNTR
01D4R	C8C0 0004	LHI	VALUE,4		ASM02420
01D8R	4300 06F2R	B	RESERV		ASM02430
01DCR	4850 1688R	XSQCHK	LH	R,ASSEM	DON'T CHECK SEQNUM IF IN
01E0R	4330 0242R	* BZ	CHKDOC		ASM02440 ASM02450
01E4R	0766	* XHR	REG,REG	SEQNUM = 4 HALFWORDS	ASM02470
01E6R	4856 17E2R	* LOOPSQ	LH	R,SR+72(REG)	FETCH NEW SEQNUM
01EAR	4556 1694R	* CLH	R,SEQNUM(REG)	COMPARE TO OLD SEQNUM	ASM02480 ASM02490
01EER	4280 0200R	* BL	SQERR	IF NEW LESS THAN OLD=ERROR	ASM02500
01F2R	4230 0212R	* BNE	UPDATE	IF NEW GT THEN OLD GO UPDATE	ASM02510 ASM02520
01F6R	0A62	* AHR	REG,TWO	BUMP INDEX	ASM02530
01F8R	C560 0008	* CLHI	REG,8	WERE THESE THE LAST 2 DIGITS	ASM02540
01FCR	4280 01E6R	* BL	LOOPSQ	CONTINUE TO NEXT 2DIGITS	ASM02550
0200R	C850 0023	* SQERR	LHI	R,X'23'	SEQUENCE FLAG IS NUMBER SIGN
0204R	D250 178BR	* STB	R,PR+1	SET ERROR FLAG INTO PR	ASM02560 ASM02570
0208R	4850 1912R	* LH	R,ERRCNT	ADD ONE TO FLAGS	ASM02580 ASM02581
020CR	0A51	* AHR	R,ONE	COUNT, FOR EACH SEQNUM	ASM02582
020ER	4050 1912R	* STH	R,ERRCNT	(#) ERROR FLAG THAT'S SET.	ASM02583
0212R	D1C0 17E2R	UPDATE	LM	12,SR+72	ASM02590
0216R	D0C0 1694R	* STM	12,SEQNUM	UPDATE NEW SEQNUM	ASM02600
021AR	4300 0242R	* B	CHKDOC	RETURN TO MAIN ASSEMBLY FLOW	ASM02610
021ER	41D0 00D6R	READ	BAL	RETURN,IOWAIT	READ/WAIT/CHECK STATUS
0222R	18A4R	DC	SRBLK	REQUEST SOURCE READ	ASM02620 ASM02630 ASM02640

0224R	18A4R	DC	SRBLK	REISSUE REQUEST ON BAD STATUS	ASM02650
0226R	4850	LH	R,OPSCRT	TO SCRATCH OR NOT TO SCRATCH	ASM02660
	1626R				
022AR	4330	BZ	QUESTS	NO, GO CHECK SEQ	ASM02670
	023AR				
022ER	4510	CLH	ONE,PASS	ON PASS 1 ONLY SCRATCH	ASM02680
	161AR				
0232R	4230	BNE	QUESTS		ASM02690
	023AR				
0236R	E110	SVC	1,SCBLK	WRITE SOURCE LINE TO SCRATCH	ASM02700
	18B4R				
023AR	4850	QUESTS LH	R,OPSQNM	WAS A SEQNUM CHK REQUESTED?	ASM02710
	1628R				
023ER	4230	BNZ	SQCHK	YES, GO COMPARE SEQNUMS	ASM02720
	01DCR				
0242R	4850	CHKDOC LH	R,DOC	GET DO COUNTER	ASM02730
	168CR				
0246R	4330	BZ	PROCES	NOT IN MIDST OF DOING A "DO"	ASM02740
	0264R				
024AR	C450	NHI	R,X'7FFF'	DELETE 1ST TIME NEG FLAG	ASM02750
	7FFF				
024ER	4050	STH	R,DOC	STORE ZERO/OR POS PORTION	ASM02760
	168CR				
0252R	4330	BZ	PRINTR	IF DOC=0,DON'T ASSEMBLE	ASM02770
	0CC6R				
0256R	0B51	DECDOC SHR	R,ONE	DECREMENT DO COUNTER	ASM02780
0258R	4050	STH	R,DOC	UPDATE DO COUNTER	ASM02790
	168CR				
025CR	C860	LHI	REG,X'20'	REPLACE CR WITH SPACE	ASM02800
	0020				
0260R	D260	STB	REG,SR-1	BETWEEN DO ITERATIONS	ASM02810
	1799R				
0264R	C840	PROCES LHI	SRX,SR	INITIALIZE SRX	ASM02820
	179AR				
0268R	C830	LHI	GET,GETCHR	INITIALISE FOR BALR TOGET	ASM02830
	0942R				
026CR	01F3	BALR	EXIT,GET	FETCH 1ST CHAR	ASM02840
026ER	C5B0	CLHI	CHAR,C'*	IF ITS A COMMENT STATEMENT	ASM02850
	002A				
0272R	4330	BE	PRINTR	PRINT IT AND GET NEXT LINE	ASM02860
	0CC6R				
0276R	C5B0	CLHI	CHAR,X'0D'	IF CARRIAGE RETURN,	ASM02870
	000D				
027AR	4330	BE	PRINTR	PRINT A BLANK LINE	ASM02880
	0CC6R				
027ER	C850	LABEL LHI	R,10	FLAG FOR SYMBOL TOGOIN LS	ASM02890
	000A				
0282R	41D0	BAL	RETURN,SYMBOL*2	FETCH & PACK LABEL	ASM02900
	094CR				
0286R	4300	B	TRYSPC	1ST CHAR WAS NOT LETTER	ASM02910
	01A4R				
028AR	41E0	BAL	LINK,FINDSP		ASM02920
	016FR				
028ER	4300	B	LBONLY		ASM02930
	01C4R				
0292R	41D0	COMAND BAL	RETURN,SYMBOL		ASM02940

0296R	094AR 4300 01CCR	B	OPERR	1ST CHAR NOT A LETTER	ASM02950
029AR	41E0 016ER	BAL	LINK,FINDSP	MATCH FOUND, SPACEOVER EXP	ASM02960
029ER	0210 15D4R	STB	ONE,EOL		ASM02970
02A2R	41D0 0F34R	BAL	RETURN,SEARCH	LOOK FOR MATCH IN OP TABLE	ASM02980
02A6R	1630R	DC	OPORG		ASM02990
02A8R	15E6R	DC	SS		ASM03000
02AAR	4300	B	SPECOP	NOT FOUND,CHECK SPECIAL OPS	ASM03010
02AER	03AAR 41E0 0BC8R	BAL	LINK,UNPACK	UNPACK SYMBOL'S CODE&VALUE	ASM03020
02B2R	4070 15D8R	STH	HOLD,SAVE	SAVE RIGHT HALF OF CODE	ASM03030
02B6R	C560 000D	CLHI	REG,X'D'	X'D' FOR "IF" & "END"	ASM03040
02BAR	4330 04B6R	BE	IFEND		ASM03050
02BER	4850 1688R	LH	R,ASSEM	IGNORE ALL OTHER LINES	ASM03060
02C2R	4330 011CR	BZ	DOLINE	IF IN CONDITIONAL ASSEM	ASM03070
02C6R	C560 000C	CLHI	REG,X'C'		ASM03080
02CAR	4330 04E4R	BE	SWITCH	MUST BE AN EQU,ORG, OR OPT	ASM03090
02CER	C560 000E	CLHI	REG,X'E'	CODE=E FOR ALL OTHER	ASM03100
02D2R	4330 04DCR	BE	SWTCH1	EDIT LABEL PRIOR SWITCHING	ASM03110
02D6R	C560 0008	CLHI	REG,8	IS IT A LEGAL INSTRUCTION	ASM03120
02DAR	4230 01CCR	BNE	OPERR	NO,GO RESERVE 4BYTES	ASM03130
				*****	ASM03140
				* SOURCE LINE EVALUATION - MACHINE INSTRUCTIONS	ASM03150
				*****	ASM03160
				* ASSEMBLE SOURCE STATEMENTS OF THE FORM:	ASM03170
				* (LABEL) OP R1,AFIELD(X2) TWO WORD INSTRUCTION	ASM03180
				* (LABEL) OPX AFIELD(X2) TWO WORD EXTENDED INSTR	ASM03190
				* (LABEL) OP R1,R2 ONE WORD INSTRUCTION	ASM03200
				* (LABEL) OPX R2 ONE WORD EXTENDED INSTR	ASM03210
				* (LABEL) OPX A	ASM03220
				*****	ASM03230
02DER	40C0 15E0R	INSR	STH	VALUE,W3	ASM03240
02E2R	41F0 09E8R	BAL	EXIT,EDITC	INSERT LABEL & PACK PR	ASM03250
02E6R	48C0 15E0R	LH	VALUE,W3		ASM03260
02EAR	40C0 15DCR	STH	VALUE,W1		ASM03270
02EER	C850	LHI	R,8		ASM03280

0008					
02F2R 4050		STH	R,PC		ASM03290
158ER					
02F6R 4870		LH	HOLD,SAVE	GET RIGHT HALF OP'S CODE	ASM03300
15D8R					
02FAR C570		CLHI	HOLD,3	3=TWO WORD EXTENDED	ASM03310
0003					
02FER 4330		BE	TWOWRX		ASM03320
0336R					
0302R C570		CLHI	HOLD,X'C'	C=ONE WORD EXTENDED	ASM03330
000C					
0306R 4330		BE	ONEWRX		ASM03340
0390R					
030AR C570		CLHI	HOLD,X'D'	CODE= D FOR SHORT EXTENDED	ASM03350
000D					
030ER 4330		BF	SHORTX	BRANCHES	ASM03360
03E0R					
0312R 935C		LBR	R,VALUE	ONEWRD/TWOWRD OP VALUES=XX00	ASM03370
0314R D250		STB	R,SAVE	UNLESS EVEN CHECK REQ'D	ASM03380
15D8R					
0318R C4C0		NHI	VALUE,X'FF00'	STRIP EVENNESS CHECK BITS	ASM03390
FF00					
031CR 40C0		STH	VALUE,W1	UPDATE W1 PRIOR"ORING"FIELDS	ASM03400
15DCR					
0320R C570		CLHI	HOLD,8	IF 8,ONE WORD INSTRUCTION	ASM03410
0008					
0324R 4330		BE	ONEWRD		ASM03420
037ER					
0328R 0527		CLHR	TWO,HOLD		ASM03430
032AR 4230		BNE	OPERR		ASM03440
01CCR					
032ER 41E0	TWOWRD	BAL	LINK,R1	GET R1 FIELD	ASM03450
0468R					
0332R 41E0		BAL	LINK,EVENR1	SEE IF R1 NEEDS EVEN CHECK	ASM03460
04A2R					
0336R 4020	TWOWRX	STH	TWO,RFCF		ASM03470
15CCR					
033AR 4020		STH	TWO,AFIELD		ASM03480
15CER					
033ER 41E0		BAL	LINK,AEXP	GET ADDRESS FIELD	ASM03490
117CR					
0342R 40C0		STH	VALUE,W2	PACK IT	ASM03500
15DER					
0346R 4000		STH	ZERO,RFCF		ASM03510
15CCR					
034AR 4000		STH	ZERO,AFIELD		ASM03520
15CER					
034ER CA60		AHI	REG,X'A'	MAKE PC=B IF EXP RELOCATABLE	ASM03540
000A					
0352R 4060		STH	REG,PC	OR PC=A IF EXP ABSOLUTE	ASM03550
158ER					
0356R C5B0		CLHI	CHAR,C'('	IS THIS AN INDEXED INSTR	ASM03560
0028					
035AR 4230		BNE	FINEND	NO,GO CHECK FOR PROPER END	ASM03570
039AR					
035ER 01F3		BALR	EXIT,GET	BYPASS LEFT PAREN	ASM03580

0360R	4050 15CAR	STH	R,SAVFRF	OF ADDRESS FIELD	ASM03600
0364R	41E0 048AR	BAL	LINK,R2	EVALUATE&PACK XZ INDEX	ASM03610
0368R	4850 15CAR	LH	R,SAVFRF	RESTORE FORWARD REF FLAG	ASM03620
036CR	4050 15B4R	STH	R,FRF	OF ADDRESS FIELD	ASM03630
0370R	C5B0 0029	CLHI	CHAR,C'11	IF NO RIGHT PAREN.	ASM03640
0374R	4230 039ER	BNE	FINEND+4	GO TO FLAG FORMAT ERROR	ASM03650
0378R	01F3	BALR	EXIT,GET	BYPASS RIGHT PAREN	ASM03660
037AR	4300 039AR	B	FINEND		ASM03670
037ER	41E0 0468R	ONEWRD	BAL LINK,R1	***** GET R1 FIELD	ASM03680 ASM03690
0382R	41E0 04A2R	BAL	LINK,EVENR1	SEE IF R1 NEEDS EVEN CHECK	ASM03700
0386R	0350 15D8R	LB	R,SAVE	FETCH KEY FOR EVEN CHECKS	ASM03710
038AR	0A55	AHR	R,R	AFTER ONEWRD R1 CHECKED,	ASM03720
038CR	D250 15D8R	STB	R,SAVE	MOVE R2 CHECK KEY TO SIGN BIT	ASM03730
0390R	41E0 048AR	ONEWRX	BAL LINK,R2	GET R2 FIELD	ASM03740
0394R	0861	LHR	REG,ONE	CHECK BIT 15 FOR EVENESS OF	ASM03750
0396R	41E0 04A6R	BAL	LINK,EVENR2	R2 FIELD OF FLTG-PT RR INSTR	ASM03760
039AR	41E0 0C2CR	FINEND	BAL LINK,CHKEND	CHECK IF END PROPER	ASM03770
039ER	D210 156CR	STB	ONE,ERRORF	IF NOT, SET FORMAT ERROR	ASM03780
03A2R	41F0 0A30R	BAL	EXIT,EDIT		ASM03790
03A6R	4300 011CR	B	DOLINE		ASM03800
03AAR	4850 1688R	SPECOP	LH R,ASSEM	IGNORE IF IN CONDITIONAL ASSEM	ASM03810
03AER	4330 011CR	BZ	DOLINE		ASM03820
03B2R	41D0 0F34R	BAL	RETURN,SEARCH		ASM03830
03B6R	162AR	DC	STO		ASM03840
03B8R	15E6R	DC	SS		ASM03850
03BAR	4300 01CCR	B	OPERR	NOT INTABLE, ERROR	ASM03860
03BER	485A 0000	LH	R,0(P)	FOUND MATCH AT "P"	ASM03870
03C2R	C450 0F00	NHI	R,X'0F00'	VALUE MUST BE ABS/DEFINED	ASM03880
03C6R	4230 01CCR	BNZ	OPERR	AND NOT AN ENTRY/EXTRN	ASM03890
03CAR	41E0 0BC8R	BAL	LINK,UNPACK	V1=OPCODE	ASM03900

03CER	087C	LHR	HOLD,VALUE	V2=EXTENSION +2,3,,8 OR C	ASM03910
03D0R	C470	NHI	HOLD,X'F'	HOLD HOLDS 2,3,8,OR'C'	ASM03920
	000F				
03D4R	4070	STH	HOLD,SAVE	SAVE RIGHT HALF OF CODE	ASM03930
	15D8R				
03D8R	C4C0	NHI	VALUE,X'FFF0'		ASM03940
	FFF0				
03DCR	4300	B	INSTR		ASM03950
	02DER				
		*****			ASM03960
03E0R	41E0	SHORTX	BAL	LINK,AEXP	GET VALUE OF OPERAND
	117CR				ASM03970
03E4R	4330	BZ	**16	AEXP RETURNS W/LOAD FRF	ASM03980
	03F4R				
03E8R	D210	STB	ONE,ERRORU	SET UERROR FOR UNDEFINEDS	ASM03990
	1564R				
03ECP	4860	LH	REG,MODE	DON'T COVER UERROR,	ASM04000
	1686R				
03F0R	48C0	LH	VALUE,LOC	ASSURE NO F ERROR GETS SET.	ASM04010
	168ER				
03F4R	4560	CLH	REG,MODE	VALUE'S MODE= LOC'S MODE	ASM04020
	1686R				
03F8R	4330	BE	**8		ASM04030
	0400R				
03FCR	D210	STB	ONE,ERRORF	ELSE ERROR	ASM04040
	156CR				
0400R	4860	LH	REG,LOC	CURRENT LOCATION COUNTER	ASM04050
	168ER				
0404R	0788	XHR	TEMP,TEMP	FLAG=ZERO FOR BACKWARD	ASM04060
0406R	056C	CLHR	REG,VALUE	COMPARE LOC TO VALUE	ASM04070
0408R	4280	BL	FORWRD	CARRY MEANS FORWARD BRANCH	ASM04080
	042ER				
040CR	086C	BOTHBF	SHR	REG,VALUE	ASM04090
040ER	CF60	SRHA	REG,1	GET HALFWORD DISPLACEMENT	ASM04100
	0001				
0412R	4380	BNC	**8	EVEN DIFFERENCE OK	ASM04110
	041AR				
0416R	D210	STB	ONE,ERRORF	ODD BYTE DISPLACEMENT ERROR	ASM04120
	156CR				
041AR	08C6	LHR	VALUE,REG		ASM04130
041CR	41E0	BAL	LINK,TEST	CATCH OUT-OF-RANGE ERRORS	ASM04140
	043CR				
0420R	06C8	OHR	VALUE,TEMP	CHANGE BACKWARD TO FORWARD	ASM04150
0422R	46C0	OH	VALUE,W1	IF TEMP=X'0100',ELSE NOT	ASM04160
	15DCR				
0426R	40C0	STH	VALUE,W1	UPDATE OBJECT DATA	ASM04170
	15DCR				
042AR	4300	B	FINEND	CHECK END AND EDIT	ASM04180
	039AR				
042ER	0856	FORWRD	LHR	R,REG	SWITCH UPPER/LOWER LIMITS
0430R	086C	LHR	REG,VALUE		ASM04200
0432R	08C5	LHR	VALUE,R		ASM04210
0434R	C880	LHI	TEMP,X'0100'	FLAG FOR FORWARD OP CODE	ASM04220
	0100				
0438R	4300	B	BOTHRF		ASM04230
	040CR				



				*****	ASM04240	
				* TEST TO SEE IF EXPRESSION HAS A VALUE BETWEEN 0 & 15.	ASM04250	
				* IF OUT-OF-RANGE, CHANGE VALUE TO ZERO	ASM04260	
043CR	C5C0	TEST	CLHI	VALUE,16	IF VALUE WITHIN THAT RANGE	ASM04270
	0010					
0440R	028E		BLR	LINK	RETURN.	ASM04280
0442R	D210	BADVAL	STB	ONE,ERRORF	OTHERWISE FLAG FORMAT ERR	ASM04290
	156CR					
0446R	07CC		XHR	VALUE,VALUE	AND RESET VALUE TO ZERO	ASM04300
0448R	030E		BR	LINK	BEFORE RETURNING	ASM04310
						ASM04320
044AR	40E0	SCAN	STH	LINK,ENDSCN+2	SAVE LINKAGE ADDRESS	ASM04330
	0466R					
044ER	41E0		BAL	LINK,AEXP	EVALUATE EXPRESSION	ASM04340
	117CR					
0452R	4330		BZ	**8	FRF GET SET? NO, IT'S OK	ASM04360
	045AR					
0456R	D210		STB	ONE,ERRORU	SET "U" ERROR FOR UNDEFINED	ASM04370
	1564R					
045AR	0866		LHR	REG,REG	WAS VALUE RELOCATABLE?	ASM04380
045CR	4330		BZ	**8	NO, SKIP OVER ERROR	ASM04390
	0464R					
0460R	D210		STB	ONE,ERRORF	YES, SET FORMAT ERROR	ASM04400
	156CR					
0464R	4300	ENDSCN	B	0000	RETURN	ASM04410
	0000					
						ASM04420
0468R	40E0	R1	STH	LINK,R2END+2	RU EXPRESSION MUST NOT BE	ASM04430
	04A0R					
046CR	41E0		BAL	LINK,SCAN	FRF/REL OR OUTSIDE 0THRU15	ASM04440
	044AR					
0470R	41E0		BAL	LINK,TEST		ASM04450
	043CR					
0474R	CDC0		SLHL	VALUE,4	POSITION VALUE TO R1 FIELD	ASM04460
	0004					
0478R	C5B0		CLHI	CHAR,C1,1	COMMA MUST FOLLOW 1ST	ASM04490
	002C					
047CR	4330		BE	**8	OPERAND	ASM04500
	0484R					
0480R	D210		STB	ONE,ERRORF		ASM04510
	156CR					
0484R	01F3		BALR	EXIT,GET		ASM04520
0486R	4300		B	R1R2		ASM04530
	0496R					
						ASM04540
048AR	40E0	R2	STH	LINK,R2END+2	R2/X2 EXPRESSION CANNOT BE	ASM04550
	04A0R					
048ER	41E0		BAL	LINK,SCAN	FRF/REL OR OUTSIDE 0THRU15	ASM04560
	044AR					
0492R	41E0		BAL	LINK,TEST		ASM04570
	043CR					
0496R	46C0	R1R2	OH	VALUE,W1	"OR" INTO R2/X2 FIELD OF	ASM04580
	15DCR					
049AR	40C0		STH	VALUE,W1	MACHINE INSTRUCTION	ASM04590
	15DCR					
049ER	4300	R2END	B	0000		ASM04600

0000

```

*****
04A2R C860  EVENR1 LHI  REG,X'0010'  CHECK BIT 11 FOR R1 EVENESS  ASM04610
      0010  ASM04620
04A6R 4850  EVENR2 LH    R,SAVE      LEFT BYTE=EVENCHK KEY,RT=TYPE  ASM04630
      15D8R
04AAR 031E  BNMR  LINK      RETURN WHEN NO CHECK REQ'D  ASM04640
04ACR 04C6  NHR   VALUE,REG  CHECK EVENESS OF R1 OR R2  ASM04650
04AER 033E  BZR  LINK      R1/R2 VALUE IS EVEN,SO RETURN  ASM04660
04B0R 0210  STB  ONE,ERROR  ELSE,SET E ERROR FLAG  ASM04670
      1560R
04B4R 030E  BR   LINK      BEFOR RETURNING  ASM04680
*****
* ASSEMBLER DIRECTIVES (PSEUDO-OP)  IF  ASM04690
* ASSEMBLE SOURCE STATEMENT OF THE FORM:  ASM04700
* IF (AN EXPRESSION) (COMMENT)  ASM04710
*****
* IF OPERAND OF "IF" LINE EQUAL ZERO  ASM04720
* NO LINES WILL BE ASSEMBLED UNTIL THE NEXT  ASM04730
* "IF" OPERATION OR "END" STATEMENT.  ASM04740
*  ASM04750
* IF OPERAND NOT EQUAL ZERO, THE ASSEMBLY PROCESS  ASM04760
* CONTINUES AS PER OPTIONS SELECTED.  ASM04770
IFEND CLHI HOLD,3 3= END, 0= IF STATEMENT  ASM04780
      ASM04790
04B6R C570  BNE  IF  ASM04800
      0003  ASM04810
04BAR 4230  STH  TWO,ASSEM  FORCE UNCONDITIONAL ASSEMBLY  ASM04820
      04C6R  ASM04830
04BER 4020  B    SWTCH1  OF THE END CARD
      1688R  ASM04840
04C2R 4300  *"IF" STATEMENT TO PROCESS  ASM04850
      04DCR  IF  BAL  LINK,AEXP  GET VALUE OF "IF" OPERAND
04C6R 41E0  BZ  ++10  OPERAND'S VALUE SETS ASSEM  ASM04870
      117CR  ASM04880
04CAR 4330  LHR  VALUE,ONE  UNDEFINED?FORCE ASSEMBLY.  ASM04890
      04D4R  STB  ONE,ERRORU
04CER 08C1  STH  VALUE,ASSEM  ASM04900
04D0R 0210  B    ENDCHK  ASM04910
      1564R
04D4R 40C0  *****
04D8R 4300  * SWITCHING PROCESS THRU "DTABLE" TO PSEUDO-OP ROUTINE  ASM04920
      0CBER  SWTCH1 BAL  EXIT,EDITC  EDIT LABEL  ASM04930
04DCR 41F0  LH  HOLD,SAVE  ASM04940
      09E8R  ASM04950
04E0R 4870  SWITCH AHR  HOLD,HOLD  YES, DOUBLE FOR HW ADRS  ASM04960
      15D8R  LH  HOLD,DTABLE(HOLD)  ASM04970
04E4R 0A77  BR  HOLD  GOTO PSEUDO-OP ROUTINE  ASM04980
04E6R 4877  *****
      1548R  * ASSEMBLER DIRECTIVES (PSEUDO-OP)  TITLE  ASM04990
04EAR 0307  * ASSEMBLE SOURCE STATEMENT OF THE FORM:  ASM05000
      ASM05010

```

		* TITLE (UP TO 56 CHARACTERS IN OPERAND)					
		*****					
✓	04ECR 41F0	✓	TITLE	BAL	EXIT,NTITLE	BLANK OUT TITLE RECORD	ASM05020 ASM05030 ASM05040
	051CR						
	04F0R 4850	✓	LH	R,EOL		IF END OF LINE GOT SET	ASM05050
	15D4R						
	04F4R 4230	✗	BNZ	EJECT		ASSUME NO TITLE	ASM05060
	0514R						
	04F8R C5B0	✗	LOOPTR	CLHI	CHAR,X'0D'	END AT CR	ASM05070
	000D						
	04FCR 4330	✗	BE	EJECT			ASM05080
	0514R						
	0500R C550	✗	CLHI	R,56		OR AT MAXIMUM OF 56 CHARS.	ASM05090
	0038						
	0504R 4380	✗	BNL	EJECT			ASM05100
	0514R						
	0508R D2B5	✗	STB	CHAR,TR(R)		STORE CHARACTER	ASM05110
	185CR						
	050CR 0A51	✗	AHR	R,ONE			ASM05120
	050ER 01F3	✗	BALR	EXIT,GET		TITLE RECORD FOR FUTURE	ASM05130
	0510R 4300	✗	B	LOOPTR		PRINTING.	ASM05140
	04F8R						
	0514R 41E0	✓	EJECT	BAL	LINK,PRINT		ASM05160
	0CCAR						
	0518R 4300	✗	B	DOLINE			ASM05170
	011CR						
	051CR 0755	✗	NTITLE	XHR	R,R	BLANK OUT TITLE RECORD	ASM05180
	051ER C860	✓	LHI	REG,X'2020'		WITH SPACES	ASM05190
	2020						
	0522R 4065	✗	LOOPTR	SIH	REG,TR(R)		ASM05200
	185CR						
	0526R 0A52	✗	AHR	R,TWO			ASM05210
	0528R C550	✗	CLHI	R,56			ASM05220
	0038						
	052CR 4280	✗	BL	LOOPTR			ASM05230
	0522R						
	0530R 030F	✗	BR	EXIT			ASM05240
							ASM05250
							ASM05260
							ASM05270
							ASM05280
							ASM05290
							ASM05300
							ASM05310
							ASM05320
	0532R D210		OPTERR	STB	ONE,ERRORF	SET FORMAT ERROR	
	156CR						
	0536R C5B0		OPTEND	CLHI	CHAR,C','	COMMA?	ASM05330
	002C						
	053AR 4230		BNE	OPTPRT		END OF OPTION STATEMENT	ASM05340
	05BAR						
	053ER 01F3		BALR	EXIT,GET		BYPASS COMMA	ASM05350
	0540R 41D0	OPT	BAL	RETURN,SYMBOL			ASM05360
	094AR						
	0544R 4300		B	OPTERR		SYMBOL'S 1ST CHAR NOT LETTER	ASM05370
	0532R						
	0548R 41D0		BAL	RETURN,SEARCH		FIND MATCH IN OP TABLE	ASM05380

0F34R				
054CR 1630R		DC	OPORG	ASM05390
054ER 15E6R		DC	SS	ASM05400
0550R 4300		B	OPTERR	NO MATCH,BYPASS BAD SYMBOL
0532R				ASM05410
0554R 41E0		BAL	LINK,UNPACK	UNPACK CODES FOR OPTIONS
08C8R				ASM05420
0558R C560		CLHI	REG,9	FIRST HALF OF CODE=9
0009				ASM05430
055CR 4230		BNE	OPTERR	MATCHED BUT NOT AN OPT SYMBOL
0532R				ASM05440
0560R C570		CLHI	HOLD,X'F'	CODE=X'F'FOR"LAB="OPTION
000F				ASM05450
0564R 4230		BNE	OPT2	
0596R				ASM05460
0568R C580	LAB	CLHI	CHAR,C'='	CORRECT FORMAT SHOULD BE
003D				ASM05470
056CR 4330		BE	**8	LAB=SYMBOL
0574R				ASM05480
0570R D210		STB	ONE,ERRORF	SET FORMAT ERROR
156CR				ASM05490
0574R 4070		STH	HOLD,PC	LOADER CONTROL ITEM FOR LAB=
158ER				ASM05500
0578R 01F3		BALR	EXIT,GET	BYPASS EQUAL OR ERR CHAR
057AR 41D0		BAL	RETURN,SYMBOL	GET LABEL OF ENTIRE PROGRAM
094AR				ASM05510
057ER 4300		B	OPTERR	ERRONEOUS FIRST CHARACTER
0532R				ASM05530
0582R C850		LHI	R,W1	PACK TEMP STORAGE W1,W2,W3
15D0R				ASM05540
0586R C8A0		LHI	P,SS	WITH ASCII CODED SYMBOL
15E6R				ASM05550
058AR 41E0		BAL	LINK,PACSYM	IN TEMP SYMBOL STORAGE
08EAR				ASM05560
058ER 41F0		BAL	EXIT,PACKPB	ON PUNCHING PASS,PACK PB1
0AAAAR				ASM05570
0592R 4300		B	OPTEND	CHECK IF AT END
0536R				ASM05580
0596R D35A	OPT2	LB	R,1(P)	GET FLAG VALUE OF OPTEDSYMBOL
0001				ASM05590
059AR C570		CLHI	HOLD,X'A'	IS THIS THE "SCRT" OPTION
000A				ASM05600
059ER 4230		BNE	SETOPT	NO,CONTINUE
0582R				ASM05610
05A2R 4510		CLH	ONE,PASS	IS THIS PASS 1
161AR				ASM05620
05A6R 4230		BNE	SETOPT	NO,CONTINUE
0582R				ASM05630
05AAR E110		SVC	1,SCRWD	REWIND SCRATCH DEVICE
18BER				ASM05640
05AER E110		SVC	1,SCBLK	WRITE OPT LINE TO SCRATCH
1884R				ASM05650
05B2R 4057	SETOPT	STH	R,OPTION(HOLD)	SET THIS OPTION
161CR				ASM05660
05B6R 4300		B	OPTEND	CHECK IF AT END
0536R				ASM05670

05BAR	41E0 0C2CR	OPTPRT	BAL	LINK,CHKEND	LINE SHOULD END WITH CR/SPACE	ASM05680
05BER	4300 0CC2R		B	PRINTF	IF IMPROPER END ERROR	ASM05690
05C2R	D350 156CR		LB	R,ERRORF	IF OPT STATEMENT HAD F ERR	ASM05700
05C6R	0855		LHR	R,R	PRINT IT / OTHERWISE	ASM05710
05C8R	4330 011CR		BZ	DOLINE	DON'T PRINT CORRECT OPTION	ASM05720
05CCR	4300 0CC2R		B	PRINTF	PRINT IF ERROR,AS WARNING.	ASM05730
					*****	ASM05740
					* ASSEMBLER DIRECTIVES (PSEUDO-OP)      EXTRN	ASM05750
					* ASSEMBLE SOURCE STATEMENT OF THE FORM:	ASM05760
					* (LABEL) EXTRN SYMBOL1,SYMBOL2,....,SYMBOL (COMMENT)	ASM05770
					*****	ASM05780
05D0R	C850 0600		EXTRN	LHI R,X'0600'	SET EXTRN/UNDEFINED BITS	ASM05790
05D4R	4300 05DCR		B	**8		ASM05800
					*****	ASM05810
					* ASSEMBLER DIRECTIVES (PSEUDO-OP)      ENTRY	ASM05820
					* ASSEMBLE SOURCE STATEMENT OF THE FORM:	ASM05830
					* (LABEL) ENTRY SYMBOL1,SYMBOL2,....,SYMBOL (COMMENT)	ASM05840
					*****	ASM05850
05D8R	C850 0A00		ENTRY	LHI R,X'0A00'	SET ENTRY/UNDEFINED BITS	ASM05860
05DCR	4050 15C6R		STH	R,XNBITS		ASM05870
05E0R	4300 05E6R		B	**6		ASM05880
05E4R	01F3	COMA	BALR	EXIT,GET	BYPASS COMMA	ASM05890
05E6R	41D0 094AR		BAL	RETURN,SYMBOL		ASM05900
05EAR	4300 0CC2R		B	PRINTF	PRINT LINE W/ F ERROR	ASM05910
05EER	41D0 0F34R		BAL	RETURN,SEARCH		ASM05920
05F2R	162AR		DC	STO		ASM05930
05F4R	15E6R		DC	SS		ASM05940
05F6R	4300 064ER		B	ENTER	SYMBOL NOT IN TABLE PASS 1	ASM05950
05FAR	485A 0000		LH	R,0(P)		ASM05960
05FER	0875		LHR	HOLD,R		ASM05970
0600R	C450 0C00		NHI	R,EXTENT		ASM05980
0604R	4330 0646R		BZ	EXTERR	IF NEITHER EXT NOR ENT,ERR	ASM05990
0608R	4510 161AR		CLH	ONE,PASS	ON PASS1 SET BITS OF	ASM06000
060CR	4230 061ER		BNE	CHKDEF	MULTI ENTRY/EXTRN&ERROR F	ASM06010
0610R	0857		LHR	R,HOLD		ASM06020
0612R	4650 15C6R		OH	R,XNBITS	SET CURRENT LINES EXT/ENT	ASM06030

0616R	405A 0000	STH	R,0(P)	IN CASE BOTH AN EXTRNENTRY	ASM06040		
061AR	4300 0646R	B	EXTERR	2ND TIME TO SEE THIS SYMBOL	ASM06050		
061ER	C470 0200	CHKDEF	NHI	HOLD,DEFBIT	IF THIS SYMBOL DEFINED	ASM06060	
0622R	4330 063ER	BZ	CHKENT	CHECK TO SEE IF ENTRY	ASM06070		
0626R	C450 0400	NHI	R,EXTBIT	IF UNDEFINED SYMBOL =ENTRY	ASM06080		
062AR	4230 0632R	BNZ	**8	SET F ERROR AND TREAT AS	ASM06090		
062ER	D210 156CR	STB	ONE,ERRORF	AN EXTRN	ASM06100		
0632R	D20A 0001	STB	ZERO,1(P)	ZERO VALUE FOR EXTRN'S	ASM06110		
0636R	D20A 0003	STB	ZERO,3(P)	LEAVE CODE EXT-UND	ASM06120		
063AR	4300 0662R	B	COMMA		ASM06130		
063ER	C450 0800	CHKENT	NHI	R,ENTBIT	IF DEFINED SYMBOL IS AN EXTRN	ASM06140	
0642R	4230 0662R	BNZ	COMMA	F ERROR,IF ENTRY OK	ASM06150		
0646R	D210 156CR	EXTERR	STB	ONE,ERRORF	ASM06160		
064AR	4300 0662R	B	COMMA		ASM06170		
064ER	4850 15E6R	* EXT-ENTRY ENTER	LH	R,SS	OPERAND SYMBOL NOT FOUND IN SYMTAB SET EXTRN AND UNDEF BITS	ASM06180 ASM06190	
0652R	4650 15C6R	OH	R,XNBITS	SET EXTRN/ENTRY AND UNDEF	ASM06200		
0656R	4050 15E6R	STH	R,SS	EXTERNALLY DEFINED IF	ASM06210		
065AR	C860 15E6R	LHI	REG,SS	NEEDED TO INSERT SYMBOL SS	ASM06220		
065ER	41D0 1116R	BAL	RETURN,INSRTS	NOT DEFINED THIS ASSEMBLY	ASM06230		
0662R	C580 002C	COMMA	CLHI	CHAR,C','	ASM06240		
0666R	4330 05E4R	BE	COMA		ASM06250		
066AR	4300 0CBER	B	ENDCHK	CHECK END AND PRINT	ASM06260		
066ER	41F0 0ED4R	***** * ASSEMBLER DIRECTIVES (PSEUDO-OP) * ASSEMBLE SOURCE STATEMENT OF THE FORM: *(LABEL) ORG (AN EXPRESSION) (COMMENT) *****	ORG	BAL	EXIT,MAXLIM	UPDATE MAXLOC IF NECESSARY	ASM06270 ASM06280 ASM06290 ASM06300 ASM06310 ASM06320
0672R	41E0 117CR	BAL	LINK,AEXP	GET VALUE OF OPERAND	ASM06330		
0676R	4330 0682R	BZ	ORGOK		ASM06360		

067AR	D210 156CR	STB	ONE,ERRORF	"F" FORMAT ERR IF UNDEFINED	ASM06370
067ER	0861	LHR	REG,ONE	FORCE ORG ERR TO RELZERO	ASM06380
0680R	07CC	XHR	VALUE,VALUE	FOR UNDEFINED OPERANDS	ASM06390
0682R	4060 1686R	ORGOK STH	REG,MODE	SET MODE	ASM06400
0686R	C4C0 FFFE	NHI	VALUE,X'FFFE'	FORCE ORIGIN LOW & EVEN	ASM06410
068AR	40C0 168ER	STH	VALUE,LOC	SET NEW LOCATION COUNTER	ASM06420
068ER	41F0 0ED4R	BAL	EXIT,MAXLIM		ASM06430
0692R	41F0 09E8R	BAL	EXIT,EDITC	PACK PRINT RECORD,POST LABEL	ASM06440
0696R	4300 0CBER	B	ENDCHK	CHKEND AND PRINT	ASM06450
				*****	ASM06460
				* ASSEMBLER DIRECTIVES (PSEUDO-OP) EQU	ASM06470
				* ASSEMBLE SOURCE STATEMENT OF THE FORM:	ASM06480
				*(LABEL) EQU (AN EXPRESSION) (COMMENT)	ASM06490
				*****	ASM06500
069AR	41E0 117CR	EQU	BAL LINK,AEXP	EVALUATE EXPRESSION	ASM06510
069ER	4850 15F0R	LH	R,LS	CHECK FOR PRESENCE OF LABEL	ASM06520
06A2R	4210 06AAR	BM	**8	OK,LABEL FIELD HAD SYMBOL	ASM06530
06A6R	D210 156CR	STB	ONE,ERRORF		ASM06540
06AAR	4860 15B4R	LH	REG,FRF	WAS OPERAND DEFINED	ASM06550
06AER	4330 06BER	BZ	DEFEQU	DEFINE THIS LABELS SYMBOL	ASM06560
06B2R	D210 156CR	STB	ONE,ERRORF	ERR TO EQUATE TO UNDEFINED	ASM06570
06B6R	C650 0200	OHI	R,DEFBIT	SET UNDEF LABEL SYMBOL	ASM06580
06BAR	4050 15F0R	STH	R,LS		ASM06590
06BER	4860 15B8R	DEFEQU	LH REG,RELFLG	FETCH RELBIT OF VALUE	ASM06600
06C2R	41F0 09F0R	BAL	EXIT,EDITC1	INSERT LABEL IN SYMTAB	ASM06610
06C6R	4300 0CBER	EQU LAB	B ENDCHK		ASM06620
				*****	ASM06630
				* ASSEMBLER DIRECTIVES (PSEUDO-OP) DO	ASM06640
				* ASSEMBLE SOURCE STATEMENT OF THE FORM:	ASM06650
				*(LABEL) DO (AN EXPRESSION) (COMMENT)	ASM06660
				*****	ASM06670
06CAR	41E0 044AR	DO	BAL LINK,SCAN	OPERAND MUST NOT BE FRF OR REL	ASM06680
06CER	0855	LHR	R,R	TEST FRF	ASM06684
06D0R	4230 06DAR	BNZ	ZERVAL	ZERO VALUE FOR UNDEFINES	ASM06685
06D4R	08CC	LHR	VALUE,VALUE	CHECK FOR PROPER 0 OR POS VALUE	ASM06690

06D6R 4310  
06DCR 06DCR  
06DAR 07CC  
06DCR C6C0  
H000  
06E0R 40C0  
168CR  
06E4R 4300  
0CBER

BNM ZERVAL\*2 OK  
ZERVAL XHR VALUE,VALUE DON'T ASSEMBLE NEXT LINE  
OHI VALUE,X'R000' FLAG NEG = "DO" MODE  
STH VALUE,DOC SET DO COUNTER = VALUE OF OPERAND  
R ENDCHK CHECK END,PRINT, GOTO DOLINE

ASM06700  
ASM06730  
ASM06740  
ASM06750  
ASM06760

\*\*\*\*\*  
\* ASSEMBLER DIRECTIVES (PSEUDO-OP) DS  
\* ASSEMBLE SOURCE STATEMENT OF THE FORM:  
\*(LABEL) DS (AN EXPRESSION) (COMMENT)  
\*\*\*\*\*  
\* VALUE OF DS EXPRESSION UPDATES LOC COUNTER. LABEL,  
\* IF ANY, IS INSERTED IN SYMTAB WITH LOC COUNTER VALUE  
\* PR ALREADY PACKED WITH CURRENT LOC & R IF NECESSARY  
\*\*\*\*\*

ASM06780  
ASM06790  
ASM06800  
ASM06810  
ASM06820  
ASM06830  
ASM06840  
ASM06850  
ASM06860  
ASM06870

06E8R 41E0  
044AR  
06ECR 0855  
06EER 4230  
01D4R  
06F2R 41F0  
0ED4R  
06F6R 4AC0  
168ER  
06FAR 085C  
06FCR 0451  
06FER 0AC5  
0700R 40C0  
168ER  
0704R 41F0  
0ED4R  
0708R 4300  
0CC6R

DS BAL LINK,SCAN OPERAND MUST NOT BE REL/FRF  
LHR R,R IF OPERAND UNDEF,RESERVE 4  
BNZ RESERR+4 RESERVE 4BYTES FOR UNDEFINED  
RESERV BAL EXIT,MAXLIM  
AH VALUE,LOC UPDATE CURRENT LOCATION COUNTER  
LHR R,VALUE OPERAND MUST BE FORCED EVEN  
NHR R,ONE  
AHR VALUE,R  
STH VALUE,LOC  
BAL EXIT,MAXLIM  
R PRINTR CANNOT DO NORMAL ENDCHK

ASM06880  
ASM06890  
ASM06900  
ASM06910  
ASM06920  
ASM06930  
ASM06950  
ASM06960  
ASM06970  
ASM06980

\*\*\*\*\*  
\* ASSEMBLER DIRECTIVES (PSEUDO-OP) DC  
\* ASSEMBLE SOURCE STATEMENT OF THE FORM:  
\*(LABEL) DC EXPRESSION,EXPRESSION,...,EXPRESSION  
\*\*\*\*\*

ASM06990  
ASM07000  
ASM07010  
ASM07020  
ASM07030  
ASM07040  
ASM07050  
ASM07060  
ASM07070  
ASM07071  
ASM07072  
ASM07080  
ASM07100

070CR 01F3  
070ER C890  
001C  
0712R 4090  
15AER  
0716R 4020  
15CER  
071AR C850  
0008  
071ER 4050  
158ER  
0722R 41E0  
117CR  
0726R CA60  
0008

DC BALR EXIT,GET  
LHI COUNT,28 FORCE AEXP TO DO DC C',E',  
STH COUNT,TYPECN AND D' CONSTANTS  
STH TWO,AFIELD LEAVE RFCF A ZERO  
LHI R,8 SET PC=8 BETWEEN OPERANDS  
STH R,PC  
BAL LINK,AEXP  
AHI REG,8 REG=0 ABS/REG=1 REL



072AR	4060	STH	REG,PC	SET PC=8 ABS/PC=9 REL	ASM07110
	158ER				
072ER	40C0	EDITDC	STH	VALUE,W1	PACK THE VALUE TO EDIT
	15DCR				ASM07120
0732R	C5B0	CLHI	CHAR,C','	IF NOT AT COMMA THIS IS	ASM07130
	002C				
0736R	4330	BE	**12	LAST OPERAND OF DC.	ASM07140
	0742R				
073AR	41E0	BAL	LINK,CHKEND	SET FORMAT ERROR IF END	ASM07150
	0C2CR				
073ER	D210	STB	ONE,ERRORF	IS NOT SPACE OR CR.	ASM07160
	156CR				
0742R	41F0	BAL	EXIT,EDIT		ASM07170
	0A30R				
0746R	C5B0	CLHI	CHAR,C','		ASM07180
	002C				
074AR	4330	BE	DC-2		ASM07190
	070CR				
074ER	C850	LHI	R,12		ASM07200
	000C				
0752R	4050	STH	R,TYPECN	REESTABLISH NORM TYPE CONSTANT	ASM07210
	15AER				
0756R	4000	STH	ZERO,AFIELD	RESET AFIELD FLAG	ASM07220
	15CER				
075AR	4300	B	DOLINE		ASM07230
	011CR				
					ASM07240
				* ASSEMBLER DIRECTIVES (PSEUDO-OP)	END
				* ASSEMBLE SOURCE STATEMENT OF THE FORM:	
				* (LABEL) END (AN EXPRESSION=TRANSFER ADDRESS)	
					ASM07250
					ASM07260
					ASM07270
					ASM07280
				* THE "END" ROUTINE PROCESSES THE END OF FUNCTION	ASM07290
				* TO DUMP THE SYMBOL TABLE,CLEAR ALL UNDEFINED VALUES,	ASM07300
				* GETS THE REFERENCE/DEFINITION ADDRESSES FOR GLOBALS	ASM07310
				* OUT TO TAPE,SENDS THE MAXIMUM RELOCATABLE LOCATION,	ASM07320
				* PUNCH A TRANSFER ADDRESS,IF PRESENT IN OPERAND,	ASM07330
				* SWITCH THE TAPE MODE TO RELOCATABLE IF NECESSARY,	ASM07340
				* SENDS THE END OF TAPE CONTROL ITEM TO PUNCH THE LAST	ASM07350
				* PUNCH BUFFER, AND GOES EITHER TO DO THE NEXT PASS	ASM07360
				* OR IF THIS IS THE LAST PASS OF THE ASSEMBLY ISSUES	ASM07370
				* AN END OF JOB SVC TO OS	ASM07380
					ASM07390
					ASM07400
075ER	4850	END	LH	R,EOL	
	15D4R				
0762R	4230	BNZ	**12	SKIP IF NO OPERAND	ASM07410
	076ER				
0766R	41E0	BAL	LINK,AEXP	FETCH VALUE OF TRANSFER AD	ASM07420
	117CR				
076AR	40C0	STH	VALUE,XADRS	RELFLG HOLDS REL OF XADRS	ASM07430
	1636R				
				* LABEL,IF ANY,AND LOC COUNT IN PR & SYMTAB	ASM07440
076ER	41F0	BAL	EXIT,MAXLIM	SET HIGH ADRS FOR LOADER	ASM07450
	0ED4R				
0772R	41E0	BAL	LINK,CHKOPT	PRINT END STATEMENT	ASM07460
	0098R				
0776R	4850	LH	R,OPSCRT	WAS THIS PASS WRITTEN/	ASM07470

077AR	1626R 4330 07A6R	BZ	NOSCRA	OR READ FROM SCRATCH? NO	ASM07480
077ER	D350 18H5R	LB	R,SCRK+1	FETCH LU USED FOR SCRATCH	ASM07490
0782R	D250 18A5R	STB	R,SRBLK+1	FORCE SR LU = SC LU	ASM07500
0786R	41D0 00D6R	BAL	RETURN,IOWAIT	CHECK STATUS ON LAST I/O	ASM07510
078AR	18BCR	DC	SCWAIT	ISSUE WAIT ON SCRATCH	ASM07520
078CR	18B4R	DC	SCBLK	REISSUE SCRATCH IF TROUBLE	ASM07530
078ER	41D0 00D6R	BAL	RETURN,IOWAIT	ISSUE SCRATCH REWIND	ASM07540
0792R	18BER	DC	SCRWD		ASM07541
0794R	18BER	DC	SCRWD		ASM07542
0796R	4300 07AER	B	STDUMP	DUMP THE SYMBOL TABLE	ASM07550
079AR	D1A0 18FCR	NOLERR LM	10,NOERR	PACK PRINT RECORD WITH	ASM07560
079ER	D0A0 178AR	STM	10,PR	NO ERRORS MESSAGE	ASM07570
07A2R	4300 07D0R	B	PRINIT		ASM07580
07A6R	41F0 092ER	NOSCRA BAL	EXIT,PASCHK	IF NON-LAST PASS,	ASM07590
07AAR	E110 18C2R	SVC	1,BKSPFL	BACKSPACE LUI TO FILE MARK.	ASM07600
07AER	C8E0 07D4R	STDUMP LHI	LINK,NPRPAS		ASM07610
07B2R	41F0 0CFAR	BAL	EXIT,PRNTCK	ON PRINTING PASSES	ASM07620
07B6R	4000 168AR	STH	ZERO,LCOUNT	EJECT FOR ERRMSG & STDUMP	ASM07630
07BAR	48C0 1912R	LH	0,ERRCNT	DID LISTING CONTAIN ERRORS?	ASM07640
07BER	4330 079AR	BZ	NOLERR	NO,	ASM07650
07C2R	E120 18D8R	SVC	2,CNVERT	YES,UNPACK NUMBER INTO MSG	ASM07660
07C6R	0700	XHR	ZERO,ZERO	MAINTAIN REG ZERO=0000	ASM07670
07C8R	D180 1908R	LM	8,YESERR	PACK PRINT RECORD WITH	ASM07680
07CCR	D080 178AR	STM	8,PR	ERROR MESSAGE & COUNT	ASM07690
07D0R	41E0 0D98R	PRINIT BAL	LINK,CHKOPT	PRINT UNLESS"NO PRNT" OPTED.	ASM07700
07D4R	48B0 162AR	NPRPAS LH	NXTSYM,STO	SET POINTER TO 1ST SYMBOL	ASM07710
07D8R	4300 07E0R	B	DUMP+4	SKIP CALL ON CHKOPT	ASM07720
07DCR	41E0 0D98R	DUMP BAL	LINK,CHKOPT	PRINT UNLESS"NOPRNT"	ASM07730
07E0R	0840	LHR	STX,NXTSYM	OR ON PASS 3 OF 3	ASM07740
07E2R	4540 162CR	CLH	STX,STE	END DUMP IF NONE/OR LAST	ASM07750
07E6R	4330	BE	ENDUMP		ASM07760

07EAR	08A6R C850 178CR	LHI	R,PR+2	SET R POINTER FOR PACSYM	ASM07770
07EER	08A4	LHR	P,STX		ASM07780
07FOR	41E0 08EAR	BAL	LINK,PACSYM		ASM07790
07F4R	08B9	LHR	NXTSYM,COUNT	PRINT RECORD/COUNT=NEXTSYM	ASM07800
07F6R	41E0 08C8R	BAL	LINK,UNPACK	UNPACK SYMBOL'S CODE&VALUE	ASM07810
07FAR	41E0 0C48R	BAL	LINK,PACKPR	PACK SYMBOL'S VALUE INTO	ASM07820
07FER	1794R	DC	A,PR+10	PRINT RECORD +10	ASM07830
0800R	D374 0000	LB	HOLD,0(STX)	FETCH CODE OF SYMBOL	ASM07840
0804R	0857	LHR	R,HOLD		ASM07850
0806R	0451	NHR	R,ONE	ISOLATE RELBIT OF SYMBOL	ASM07860
0808R	D365 15D2R	LB	REG,RSPACE(R)	R=0,GET SPACE	ASM07870
080CR	D260 1798R	STB	REG,PR+14	APPEND "R" IF R=1	ASM07880
0810R	0867	LHR	REG,HOLD		ASM07890
0812R	C460 002E	NHI	REG,X'2E'	DUMP THIS LINE IF LOCAL	ASM07900
0816R	4330 07DCR	BZ	DUMP	SINGULARLY DEFINED SYMBOL	ASM07910
081AR	C560 0020	CLHI	REG,X'20'	FORM FLAG TABLE INDEX	ASM07920
081ER	4330 0868R	BE	MLTLOC		ASM07930
0822R	4220 0870R	BP	MLTEXT		ASM07940
0826R	4856 156ER	LH	R,FLAGS-2(REG)	GET 2CHARS TO PRECEDE SYM	ASM07950
082AR	4050 178AR	STRFLG STH	R,PR	STORE IN PRINT RECORD	ASM07960
082ER	0867	LHR	REG,HOLD	GET ORIGINAL CODE	ASM07970
0830R	C460 000E	NHI	REG,X'E'	DUMP THIS LINE IF ITS A	ASM07980
0834R	4330 07DCR	BZ	DUMP	MULTIPLY DEFINED SYMBOL	ASM07990
0838R	C460 000C	NHI	REG,X'C'	IF ONLY THE UNDEFINED BIT	ASM08000
083CR	4330 07DCR	BZ	DUMP	IS SET, FORWARD REFERENCE	ASM08010
0840R	0867	LHR	REG,HOLD		ASM08020
0842R	C460 0012	NHI	REG,X'12'	GET USED&UNDEFBIT	ASM08030
0846R	0562	CLHR	REG,TWO	IF THIS EXTRN/ENTRY IS	ASM08040
0848R	4330 089AR	BE	NOREFX	UNDEF&UNSTRUNG,VALNOT REF	ASM08050
084CR	0857	LHR	R,HOLD	POSITION RELBIT FOR FLIP	ASM08060
084ER	41F0 0F16R	BAL	EXIT,FLIP+4		ASM08070
0852R	D374 0000	LB	HOLD,0(STX)	GET SYMBOL'S CODE AGAIN	ASM08080
0856R	40C0	STH	VALUE,W1		ASM08085

085AR	0472	NHR	HOLD,TWO	IF DEFBIT=0TREAT AS ENTRY	ASM08090
085CR	4230	BNZ	EXTRNS		ASM08100
0860R	0878R C850 0007	LHI	R,7	ENTRY BIT SET,SET PC=7	ASM08110
0864R	4300	B	SETPC	FOR DEFINITION ADRS/VALUE	ASM08120
0868R	087CR C850 4D20	MLTLOC LHI	R,C'M'	LOCAL MULTIPLE DEFINED	ASM08130
086CR	4300	B	STRFLG		ASM08140
0870R	082AR C850 2A4D	MLTEXT LHI	R,C'M'	EXTERNAL MULTIPLE DEFINED	ASM08150
0874R	4300	B	STRFLG		ASM08160
0878R	082AR C850 0006	EXTRNS LHI	R,6	EXTRN BIT SET,SET PC=6	ASM08170
087CR	41F0	SETPC BAL	EXIT,PACKPB-4	PACK PB W/PC=6/7 & VALUE	ASM08200
0880R	0AA6R C850	LHI	R,W1		ASM08210
0884R	15DCR 08A4	LHR	P,STX	RESET POINTER TO SYMBOL	ASM08220
0886R	41E0	BAL	LINK,PACSYM	PACK SYMBOL IN W1,W2,W3	ASM08230
088AR	0BEAR C850 0006	LHI	R,6	6*6=X'C'FOR REFERENCE SYMB	ASM08240
088ER	4A50	AH	R,PC	6*7=X'D'FOR DEFINITION SYM	ASM08250
0892R	158ER 41F0	BAL	EXIT,PACKPB-4	PACK C OR D WITH SYMBOL	ASM08270
0896R	0AA6R 4300 07DCR	B	DUMP		ASM08280
				* FOR UNSTRUNG UNDEFINED EXTRN OR ENTRY SYMBOLS	ASM08290
				* JUST PRINT VALUE DON'T PUNCH REF ADRS ON TAPE	ASM08300
089AR	C850	NOREFX LHI	R,C'M'	SHOW UNSTRUNG UNDEFINED	ASM08310
089ER	2A2A 4050	STH	R,PR	COVER PREVIOUS LETTER	ASM08320
08A2R	178AR 4300	B	DUMP		ASM08330
08A6R	07DCR 4850	ENDUMP LH	R,EOL	IF NO XADRS SKIP TO	ASM08340
08AAR	15D4R 4230	BNZ	ENDREL	SEND END OF TAPE CODE	ASM08350
08AER	08C4R 4850	LH	R,XADRS		ASM08360
08B2R	1636R 4050	STH	R,W1		ASM08370
08B6R	15DCR 4850	LH	R,RELFLG		ASM08380
08BAR	1588R 41F0	BAL	EXIT,FLIP+4	SEND FLIP FOR XADRS IF NEC	ASM08390
08BER	0F16R 41F0	BAL	EXIT,PACKPB-10	PACK XADRS WITH CONTROL	ASM08410
08C2R	0AA0R 0004	DC	4	ITEM 4	ASM08420

08C4R	4010	ENDREL	STH	ONE,MODE	FORCE MODE RELOCATABLE	ASM08430
	1686R					
08C8R	4850		LH	R,MAXLOC		ASM08440
	1690R					
08CCR	4330		BZ	ENDONE		ASM08450
	08DCR					
08D0R	4050		STH	R,LOC	SEND MAXLOC TO TAPE	ASM08460
	168ER					
08D4R	41F0		BAL	EXIT,MAXLIM	AND LEAVE MODE REL ON TAPE	ASM08470
	0ED4R					
08D8R	4300		B	**8		ASM08480
	08E0R					
08DCR	41F0	ENDONE	BAL	EXIT,FLIP		ASM08490
	0F12R					
08E0R	41F0		BAL	EXIT,PACKPB=10	SEND END OF TAPE SIGNAL=1	ASM08500
	0AA0R					
08E4R	0001		DC	1	AND PUNCH LAST BUFFER	ASM08510
08E6R	4850		LH	R,OPRINT	DID LISTING TAKE PLACE?	ASM08520
	1620R					
08EAR	4330		BZ	CHKPAS	CONTINUE	ASM08530
	08F6R					
08EER	4100		BAL	RETURN,IOWAIT	CHECK STATUS OF LAST I/O	ASM08540
	00D6R					
08F2R	18D2R		DC	PRWAIT	WAIT ON LAST LISTED LINE	ASM08550
08F4R	18CAR		DC	PR2BLK	REISSUE IF TROUBLED I/O	ASM08560
08F6R	41F0	CHKPAS	BAL	EXIT,PASCHK	WAS THIS THE FINAL PASS?	ASM08570
	092ER					
08FAR	4300		B	BEGINP	NO,BEGIN NEXT PASS	ASM08580
	0058R					
08FER	4850		LH	R,OPRINT	DID LISTING TAKE PLACE?	ASM08585
	1620R					
0902R	4330		BZ	**12	NO, SKIP EOF	ASM08586
	090ER					
0906R	4100		BAL	RETURN,IOWAIT	WRITE END-OF-FILE TO LISTDV	ASM08587
	00D6R					
090AR	18C6R		DC	PREOF		ASM08588
090CR	18C6R		DC	PREOF		ASM08589
090ER	4850		LH	R,OPUNCH	YES,DID OBJECT OUTPUT OCCUR?	ASM08590
	1622R					
0912R	4330		BZ	EOJ	NO, ISSUE END OF JOB	ASM08600
	092AR					
0916R	4850		LH	R,PBSWCH*2	GET LAST I/O OBJECT BUFFER	ASM08610
	169ER					
091AR	C855		LMI	R,108(R)	GET BLOCK ADDRESS	ASM08620
	006C					
091ER	4050		STH	R,ADRBLK	SET ARG FOR IOWAIT ROUTINE	ASM08630
	0928R					
0922R	4100		BAL	RETURN,IOWAIT	CHECK STATUS OF LAST I/O	ASM08640
	00D6R					
0926R	18D4R		DC	PBWAIT	WAIT FOR LAST OBJECT OUTPUT	ASM08650
0928R	0000	ADRBLK	DC	00000	REISSUE IF TROUBLED I/O	ASM08660
092AR	E130	EOJ	SVC	3,SUSPND	SEND OS END OF JOB REQUEST	ASM08670
	18D6R					
092ER	4860	PASCHK	LH	REG,OPASS	FETCH PRECEDING PASS NUMBER	ASM08680
	161CR					
0932R	4850		LH	R,PASS	FETCH CURRENT PASS NUMBER	ASM08690

```

161AR
0936R 0A51          AHR  R,ONE          UPDATE CURRENT PASS NUMBER      ASM08700
0938R  C556          CLHI R,1(REG)         WAS THIS THE FINAL PASS?      ASM08710
0001
093CR 028F          BLR  EXIT            NON-LAST PASS                    ASM08720
093ER 430F          B    4(EXIT)         FINAL PASS DONE.              ASM08730
0004

*****
* SOURCE CHARACTER ISOLATION ROUTINE
*****
0942R  D3B4          GETCHR LB  CHAR,0(SRX)
0000
0946R 0A41          AHR  SRX,ONE
0948R 030F          BR   EXIT

*****
* SOURCE SYMBOL RETRIEVAL ROUTINE
*****
* IF THE VARIABLE NUMBER OF CHARACTERS IN THE SOURCE
* STATEMENT FORM A SYMBOL RETURN IS MADE TO THE 2ND
* INSTRUCTION FOLLOWING THE CALL ON "SYMBOL"
* WHEN THE VERY FIRST CHAR IS NOT A LETTER RETURN
* TO THE FIRST INSTRUCTION FOLLOWING THE CALL
*****
094AR 0755          SYMBOL XHR  R,R
094CR  C895          LHI  COUNT,10(R)      R=0,CLEAR SYMBOL STORAGE      ASM08890
000A
0950R 4009          CLEARS STH  ZERO,SS-2(COUNT) R=10, CLEAR LABEL STORAGE      ASM08910
15E4R
0954R 0892          SHR  COUNT,TWO
0956R 0595          CLMR COUNT,R
0958R 4230          BNE  CLEARS
0950R
095CR 41F0          BAL  EXIT,LETTER
09BAR
0960R 4300          B    NOTLTR           CHAR WAS NOT HOLDING A LETTER  ASM08960
09A0R
0964R  C860          STRCHR LHI  REG,X'8000' SET BEGINNING OF ENTRY BIT      ASM08970
8000
0968R 4069          STH  REG,SS(COUNT)
15E6R
096CR  D2B9          STB  CHAR,SS+2(COUNT)
15E8R
0970R 0A91          AHR  COUNT,ONE
0972R 4300          B    *+8
097AR
0976R  D2B9          LOOPS STB  CHAR,SS+2(COUNT)
15E8R
097AR 0A91          AHR  COUNT,ONE
097CR 01F3          BALR EXIT,GET
097ER 41E0          BAL  LINK,ALFNUM
09CAR
0982R 4300          B    4(RETURN)
0004
0986R  C595          CLHI COUNT,7(R)
0007
098AR 4280          BL   LOOPS

```

0976R					
098ER	D210	STB	ONE,ERRORF		ASM09090
	156CR				
0992R	01F3	GETEND	BALR EXIT,GET		ASM09100
0994R	41E0	BAL	LINK,ALFNUM		ASM09110
	09CAR				
0998R	430D	B	4(RETURN)		ASM09120
	0004				
099CR	4300	B	GETEND		ASM09130
	0992R				
09A0R	C5B0	NOTLTR	CLHI CHAR,C'.' ALLOW SPECIAL CHARS .,S,@		ASM09140
	002E				
09A4R	4330	BE	STRCHR		ASM09150
	0964R				
09A8R	C5B0	CLHI	CHAR,C'!'		ASM09160
	0024				
09ACR	4330	BE	STRCHR		ASM09170
	0964R				
09B0R	C5B0	CLHI	CHAR,C'!'		ASM09180
	0040				
09B4R	4330	BE	STRCHR		ASM09190
	0964R				
09B8R	030D	BR	RETURN		ASM09200
					ASM09210
			*****		ASM09220
			* IF CHAR IS A LETTER RETURN TO 2ND INSTRUCTION		ASM09230
			* IF CHAR NOT A LETTER RETURN TO 1ST INSTRUCTION		ASM09240
			* FOLLOWING THE CALL TO LETTER		ASM09250
09BAR	C5B0	LETTER	CLHI CHAR,C'A'		ASM09260
	0041				ASM09270
09BER	028F	BLR	EXIT		ASM09280
09C0R	C5B0	CLHI	CHAR,C'Z'+1		ASM09290
	005B				
09C4R	038F	BNLR	EXIT		ASM09300
09C6R	430F	B	4(EXIT)		ASM09310
	0004				ASM09320
			*****		ASM09330
			* IF CHAR IS AN ALPHANUMERIC CHAR RETURN TO 2ND INSTRUCTION		ASM09340
			* FOLLOWING THE CALL TO ALFNUM		ASM09350
09CAR	41F0	ALFNUM	BAL EXIT,LETTER		ASM09360
	09BAR				ASM09370
09CER	4300	B	CHKDIG		ASM09380
	09D6R				ASM09390
09D2R	430E	B	4(LINK)		ASM09400
	0004				ASM09410
09D6R	08FE	CHKDIG	LHR EXIT,LINK		ASM09420
			*****		ASM09430
			* IF CHAR IS NOT A DIGIT RETURN TO THE 1ST INSTRUCTION,		ASM09440
			* IF CHAR IS A DIGIT RETURN TO THE 2ND INSTRUCTION		ASM09450
			* FOLLOWING THE CALL TO DIGIT		
09D8R	C5B0	DIGIT	CLHI CHAR,C'0'		
	0030				
09DCR	028F	BLR	EXIT		
09DER	C5B0	CLHI	CHAR,C'9'+1		
	003A				
09E2R	038F	BNLR	EXIT		
09E4R	430F	B	4(EXIT)		

```

0004
*****
* EDITING THE LINE'S LABEL AND LOCATION COUNTER
*****
09E8R 4860 "EDITC" PACKS"LOC"& THE LETTER "R" INTO PRINT RECORD+2
1686R      EDITC  LH   REG,MODE
09ECR 48C0      LH   VALUE,LOC
168ER
*****
*"EDITC1" EXPECTS "VALUE" PRESET TO VALUE OF OPERAND
* AND "REG" HOLDING RELFLG OF OPERAND'S VALUE
*****
09F0R D356 EDITC1 LB   R,RSPACE(REG)  PACK LETTER'R', IF REG=1
15D2R
09F4R 4880      LH   TEMP,ASSEM      RETURN TO DO NEXT LINE IF IN
1688R
09F8R 4330      BZ   DOLINE          CONDITIONAL ASSEMBLY
011CR
09FCR D250      STB   R,PR+6
1790R
0A00R CD60      SLHL  REG,8          REPOSITION RELBIT, IF ANY
0008
0A04R 4660      OH   REG,LS          "OR" IT INTO LABEL'S CODE
15F0R
0A08R 4060      STH  REG,LS          (USE"REG"BEFOR PACKPR DOES
15F0R
0A0CR 41E0      BAL  LINK,PACKPR      PACK VALUE (LOC)
0C48R
0A10R 178CR     DC   A(PR+2)
0A12R 4850      LH   R,LS          WAS THERE A LABEL?
15F0R
0A16R 031F      BNMR  EXIT          NO,DON'T INSERT ONE
0A18R D2C0      STB  VALUE,LS+3      PACK VALUE OF LABEL
15F3R
0A1CR CCC0      SRHL  VALUE,8
0008
0A20R D2C0      STB  VALUE,LS+1
15F1R
0A24R 40F0      STH  EXIT,EDITCX+2    SAVE LINKAGE REGISTER
0A2ER
0A28R 41D0      BAL  RETURN,INSERT    INSERT IN SYMBOL'S TABLE
1004R
0A2CR 4300      EDITCX B   0000
0000
*****
* EDITING THE LINE'S ASSEMBLED OBJECT PRINT & PUNCH DATA
*****
0A30R 40F0 "EDIT" PACKS PRINT RECORD W/CONTENTS W1 FOR PC= 8,9
0A96R      * PRINTS THE LINE AND PACKS THE PUNCH BUFFER.
0A34R 48C0      * FOR PC= A OR B, W2 GETS PACKED FOR PRINTING OF SECOND
      * LINE.  ADDITIONALLY, THE LOCATION COUNTER GETS UPDATED.
*****
EDIT  STH  EXIT,EDITX+2
*****
LH   VALUE,W1
*****

```

ASM09460  
 ASM09470  
 ASM09480  
 ASM09490  
 ASM09500  
 ASM09510  
 ASM09520  
 ASM09530  
 ASM09540  
 ASM09550  
 ASM09560  
 ASM09570  
 ASM09580  
 ASM09590  
 ASM09600  
 ASM09610  
 ASM09620  
 ASM09630  
 ASM09640  
 ASM09650  
 ASM09660  
 ASM09670  
 ASM09680  
 ASM09690  
 ASM09700  
 ASM09710  
 ASM09720  
 ASM09730  
 ASM09740  
 ASM09750  
 ASM09760  
 ASM09770  
 ASM09780  
 ASM09790  
 ASM09800  
 ASM09810  
 ASM09820



0A38R	15DCR 41E0	BAL	LINK,PACKPR	PACK PRINT RECORD	ASM09830
0A3CR	0C48R 1792R	DC	A(PR+8)		ASM09840
0A3ER	4850	LH	R,PC		ASM09850
0A42R	158ER C550	CLHI	R,X'A'		ASM09860
0A46R	000A 4330	BE	**12		ASM09870
0A4AR	0A52R C550	CLHI	R,X'B'		ASM09880
0A4ER	000B 4230	BNE	EDIT1		ASM09890
0A52R	0A6AR 41E0	BAL	LINK,PRINT	PRINT FIRST LINE WHEN A	ASM09900
0A56R	0CCAR 4850	LH	R,LOC	SECOND LINE WILL FOLLOW	ASM09910
0A5AR	168ER 0A52	AHR	R,TWO	NEVER APPEND "R" OR "F"	ASM09920
0A5CR	4050	STH	R,LOC	BUMP LOC COUNTER BY TWO	ASM09930
0A60R	168ER 48C0	LH	VALUE,W2	GET 2ND LINE INFO	ASM09940
0A64R	15DER 41E0	BAL	LINK,PACKPR		ASM09950
0A68R	0C48R 1792R	DC	A(PR+8)		ASM09960
0A6AR	41F0	EDIT1 BAL	EXIT,PACKPB	PACK THE PUNCH BUFFER	ASM09970
0A6ER	0AAAR 4850	LH	R,LOC	BUMP LOC COUNTER BY TWO	ASM09980
0A72R	168ER 0A52	AHR	R,TWO		ASM09990
0A74R	4050	STH	R,LOC		ASM10000
0A78R	168ER 4850	LH	R,PC	IF PC ODD, PC=9,B	ASM10010
0A7CR	158ER CC50	SRHL	R,1	PACK "R" FOR RELOCATABLE	ASM10020
0A80R	0001 4280	BC	PACKR	IF PC EVEN,PC=8,A	ASM10030
0A84R	0A98R 4860	LH	REG,FRF	PACK "F" IF FORWARD	ASM10040
0A8BR	15B4R D366	LB	REG,FSPACE(REG)	REFERENCE FLAG SET	ASM10050
0A8CR	15D0R D260	STORE STB	REG,PR+12		ASM10060
0A90R	1796R 41E0	BAL	LINK,PRINT		ASM10070
0A94R	0CCAR 4300	EDITX B	0000		ASM10080
0A98R	0000 C860	PACKR LHI	REG,C'R'		ASM10090
0A9CR	0052 4300	B	STORE		ASM10100
	0A8CR				ASM10110
				*****	ASM10120
				* PACKING THE LINE'S ASSEMBLED BINARY OBJECT PUNCH DATA	ASM10130
				*****	

0AA0R 485F 0000	LH	R,0 (EXIT)	FETCH PUNCH CODE	ASM10140
0AA4R 0AF2	AHR	EXIT,TWO	BUMP LINKAGE BEYOND DC	ASM10150
0AA6R 4050 158ER	STH	R,PC	SET PC (PUNCH CODE)	ASM10160
0AAAR 4850 1622R	PACKPB LH	R,OPUNCH	EXIT, IF NO PNCH SELECTED	ASM10170
0AAER 033F	BZR	EXIT		ASM10180
0AB0R 4850 1688R	LH	R,ASSEM	RETURN IF IN CONDITIONAL	ASM10190
0AB4R 033F	BZR	EXIT	ASSEMBLY	ASM10200
0AB6R 4850 161CR	LH	R,OPASS	EXIT ALSO IF THIS IS NOT	ASM10210
0ABAR 4550 161AR	CLH	R,PASS	THE LAST PASS OF THIS ASSEMBLY.	ASM10220
0ABER 023F	BNER	EXIT		ASM10230
0AC0R 4850 158ER	LH	R,PC	FETCH PUNCH CODE	ASM10240
0AC4R D3A5 157ER	LB	P,CONTRL(R)	FETCH NUMBER OF DATA ITEMS	ASM10250
0AC8R 40A0 1590R	STH	P,ITEMS	SAVE NUMBER OF DATA ITEMS	ASM10260
0ACCR 087A	LHR	HOLD,P	SAVE NUMBER OF DATA ITEMS	ASM10270
0ACER 4AA0 1788R	AH	P,PBX	ADD PUNCH BUFFER INDEX TO	ASM10280
0AD2R C5A0 0000	CLHI	P,208	SEE IF BUFFER HAS ROOM	ASM10290
0AD6P 4280 0AE2R	BL	GOPACK	YES, GO PACK WHAT IS NECESSARY	ASM10300
0ADAR 41E0 0B6CR	BAL	LINK,PUNCH	NO, PUNCH THE FILLED BUFFER	ASM10310
0ADER 4850 158ER	LH	R,PC	FETCH PC AGAIN	ASM10320
0AE2R 41D0 0B1FR	GOPACK BAL	RETURN,PACK4	PACK PC AS CONTROL ITEM	ASM10330
0AE6R 4510 158ER	CLH	ONE,PC	IF PC = ONE THIS IS END	ASM10340
0AEAR 4230 0AF2R	BNE	**8		ASM10350
0AEER 41E0 0B6CR	BAL	LINK,PUNCH	PUNCH THE LAST BUFFER	ASM10360
0AF2R 4870 1590R	LH	HOLD,ITEMS		ASM10370
0AF6R 033F	BZR	EXIT	EXIT IF PC = 0,1,2,3,OR E	ASM10380
0AF8R 4860 150CR	LH	REG,W1		ASM10390
0AFCR 41E0 0B52R	BAL	LINK,PACHAF	PACK A HALFWORD ONLY FOR	ASM10400
0B00R C570 0008	CLHI	HOLD,8	PC=4,5,6,7,8,9	ASM10410
0B04R 028F	BLR	EXIT	PACK 2 HALFWORDS ONLY FOR	ASM10420
0B06R 4860 150ER	LH	REG,W2	PC=A OR B	ASM10430
0B0AR 41E0 0B52R	BAL	LINK,PACHAF		ASM10440
0B0ER C570	CLHI	HOLD,12		ASM10450

000C					
0B12R 028F		BLR	EXIT	PACK 3 HALFWORDS ONLY FOR	ASM10460
0B14R 4860		LH	REG,W3	PC=C,D, OR F	ASM10470
15E0R					
0B18R 41E0		BAL	LINK,PACHAF	PCVC,D, OR F	ASM10480
0B52R					
0B1CR 030F		BR	EXIT		ASM10490
					ASM10500
					ASM10510
					ASM10520
					ASM10530
0B1ER 48A0		PACK4 LH	P,PBX	PBX=INDEX INTO 208 4BITS	
1788R					
0B22R 088A		LHR	TEMP,P	UPDATE PBX BY ONE	ASM10540
0B24R 0A81		AHR	TEMP,ONE		ASM10550
0B26R 4080		STH	TEMP,PBX		ASM10560
1788R					
0R2AR CCA0		SRHL	P,1	IS PBX EVEN OR ODD	ASM10570
0001					
0B2ER 4280		BC	PACK1	IF ODD, LEAVE RIGHTJUSTIFIED	ASM10580
0B36R					
0B32R CD50		SLHL	R,4	LEFT JUSTIFY WITHIN BYTE	ASM10590
0004					
0B36R CCA0	PACK1	SRHL	P,1	DOES PBX=LEFT OR RIGHT BYTE	ASM10600
0001					
0B3AR 4280		BC	PACK2	LEAVE 4BITS RIGHTJUSTIFIED	ASM10610
0B42R					
0B3ER CD50		SLHL	R,8	LEFT JUSTIFY WITHIN HAFWRD	ASM10620
0008					
0B42R 0AAA	PACK2	AHR	P,P	DOUBLE TO GET HALFWORD ADRS	ASM10630
0B44R 4AA0		AH	P,PBSWCH	PBSWCH=PB1 OR PB2	ASM10640
169CR					
0B48R 465A		OH	R,4(P)	COPY 4BITS INTO	ASM10650
0004					
0B4CR 405A		STH	R,4(P)	PUNCH BUFFER	ASM10660
0004					
0B50R 030D		BR	RETURN		ASM10670
					ASM10680
					ASM10690
					ASM10700
0B52R C890		PACHAF LHI	COUNT,4		
0004					
0B56R 0856	LOOPP	LHR	R,REG	FETCH HALFWORD TO PACK	ASM10710
0B58R CC50		SRHL	R,12	ISOLATE LEFTMOST 4BITS	ASM10720
000C					
0B5CR 41D0		BAL	RETURN,PACK4	PACK 4BITS	ASM10730
0B1ER					
0B60R CD60		SLHL	REG,4	REPOSTION NEXT 4BITS	ASM10740
0004					
0B64R 0B91		SHR	COUNT,ONE		ASM10750
0B66R 4230		BNZ	LOOPP	DONE? NO,LOOP	ASM10760
0B56R					
0B6AR 030E		BR	LINK	YES,EXIT	ASM10770
					ASM10780
					ASM10790
					ASM10800
					ASM10810
					ASM10820

0B6CR	C880 FFFF	PUNCH	LHI	TEMP,-1	START CHKSUM WITH -1	ASM10830
0B70R	48A0 169CR		LH	P,PBSWCH	FETCH FILLED BUFFER ADRS	ASM10840
0B74R	085A		LHR	R,P	SETUP BXLE LIMITS TO	ASM10850
0B76R	0862		LHR	R+1,TWO	CHECKSUM ALL DATA EXCEPT	ASM10860
0B78R	C875 0066		LHI	R+2,102(R)	SEQNUM.	ASM10870
0B7CR	4785 0000		XH	TEMP,0(R)	XH WITH SEQNUM.	ASM10880
0B80R	4785 0004	CHKSUM	XH	TEMP,4(R)	CHECK SUM FILLED BUFFER.	ASM10890
0B84R	C150 0B80R		BXLE	R,CHKSUM		ASM10900
0B88R	408A 0002		STH	TEMP,2(P)	STORE CHECKSUM.	ASM10910
0B8CR	4880 169ER		LH	TEMP,PBSWCH+2	FETCH NEXT BUFFER'S ADRS	ASM10920
0B90R	C858 006C		LHI	R,108(TEMP)		ASM10930
0B94R	4050 0RA6R		STH	R,BLKADR		ASM10940
0B98R	4850 1684R		LH	R,PSTART	1ST TIME THRU EACH PASS	ASM10950
0B9CR	4330 0BA8R		BZ	SETP	PSTART=0,SKIP I/O CHECK	ASM10960
0BA0R	41D0 00D6R		BAL	RETURN,IOWAIT	CHECK STATUS ON LAST I/O	ASM10970
0BA4R	18D4R		DC	PBWAIT		ASM10980
0BA6R	0000	BLKADR	DC	00000	REISUE PB1/PB2 + 108	ASM10990
0BA8R	4010 1684R	SETP	STH	ONE,PSTART	SET PSTART NON-ZERO	ASM11000
0BACR	E11A 006C		SVC	1,108(P)	ISSUE BOSS SVC TO PUNCH	ASM11010
0BB0R	40A0 169ER		STH	P,PBSWCH+2	REVERSE PUNCH BUFFERS	ASM11020
0BB4R	4080 169CR		STH	TEMP,PBSWCH	STATUS	ASM11030
0BB8R	485A 0000		LH	R,0(P)	FETCH OLD SEQNUM	ASM11040
0BBCR	0851		SHR	R,ONE	DECREMENT RECORD SEQNUM	ASM11050
0BBER	4058 0000		STH	R,0(TEMP)	NEWSEQ=OLDSEQ-1	ASM11060
0BC2R	41D0 00BAR		BAL	RETURN,ZEROPB	* ZERO NEXT PUNCH BUFFER EXCEPT FOR PB SEQNUM AND PBX	ASM11070
0BC6R	030E		BR	LINK		ASM11080
					*****	ASM11090
					* UTILITY ROUTINE - UNPACK	ASM11100
					*****	ASM11110
					*"UNPACK" UNPACKS THE SYMBOL AT "P" SUCH THAT	ASM11120
					* HOLD = RIGHT HALF OF CODE	ASM11130
					* REG = LEFT HALF OF CODE	ASM11140
					* VALUE = V1 AND V2, FOR INSTRUCTIONS=OP CODE&EXTENSION	ASM11150
0BC8R	D37A 0000		UNPACK	LB	HOLD,0(P)      FETCH SYMBOL'S CODE BYTE	ASM11160
						ASM11170

0BCCR	0867	LHR	REG,HOLD	SAVE IT IN SECOND PLACE	ASM11180
0BCER	C470	NHI	HOLD,X'0F'	ISOLATE RIGHT HALF OF CODE	ASM11190
	000F				
0BD2R	C460	NHI	REG,X'F0'	ISOLATE LEFT HALF OF CODE	ASM11200
	00F0				
0BD6R	CC60	SRHL	REG,4		ASM11210
	0004				
0BDAR	48CA	LH	VALUE,0(P)	FETCH V1	ASM11220
	0000				
0BDER	CDC0	SLHL	VALUE,8	LEFT JUSTIFY V1	ASM11230
	0008				
0BE2R	488A	LH	TEMP,2(P)	FETCH V2	ASM11240
	0002				
0BE6R	928C	STBR	TEMP,VALUE	PACK V2 WITH V1	ASM11250
0BE8R	030E	BR	LINK	EXIT	ASM11260
					ASM11270
					ASM11280
					ASM11290
					ASM11300
					ASM11310
					ASM11320
					ASM11330
					ASM11340
					ASM11350
					ASM11360
0BEAR	C880	PACSYM	LHI	TEMP,X'2020'	
	2020				
0BEER	4085	STH	TEMP,0(R)	FILL 12 BYTES W/SPACES	ASM11370
	0000				
0BF2R	4085	STH	TEMP,2(R)		ASM11380
	0002				
0BF6R	4085	STH	TEMP,4(R)		ASM11390
	0004				
0BFAR	D38A	LB	TEMP,2(P)	STORE 1ST CHAR INTO	ASM11400
	0002				
0BFER	D285	STB	TEMP,0(R)	1ST BYTE OF W1 OR PR+2	ASM11410
	0000				
0C02R	0875	LHR	HOLD,R		ASM11420
0C04R	C89A	LHI	COUNT,4(P)		ASM11430
	0004				
0C08R	4889	LOOPAC	LH	TEMP,0(COUNT)	ASM11440
	0000				
0C0CR	021E	BMR	LINK		ASM11450
0C0ER	D389	LB	TEMP,0(COUNT)		ASM11460
	0000				
0C12R	0888	LHR	TEMP,TEMP		ASM11470
0C14R	4330	BZ	ALIGN		ASM11480
	0C28R				
0C18R	D285	STB	TEMP,1(R)		ASM11490
	0001				
0C1CR	0A51	AHR	R,ONE		ASM11500
0C1ER	0A91	AHR	COUNT,ONE		ASM11510
0C20R	C59A	CLHI	COUNT,9(P)		ASM11520
	0009				
0C24R	4280	BL	LOOPAC		ASM11530
	0C08R				
0C28R	0A91	ALIGN	AHR	COUNT,ONE	ASM11540

```

0C2AR 030E          BR      LINK
*****
0C2CR C5B0          CHKEND CLHI  CHAR,X'20'      CHKEND RETURNS TO 1ST
0020
0C30R 433E          BE      4(LINK)      INSTRUCTION FOLLOWING
0004
0C34R C5B0          CHKCR  CLHI  CHAR,X'0D'      CALLING SEQUENCE FOR
000D
0C38R 433E          BE      4(LINK)      ERROR FLAGGING IF CHAR
0004
0C3CR C5B0          CLHI  CHAR,X'0A'      NOT A PROPER END OF
000A
0C40R 023E          BNER  LINK      SPACE,CARRIAGE RETURN,
0C42R 01F3          BALR  EXIT,GET      OR LINEFEED AND CARRIAGE
0C44R 4300          B      CHKCR      RETURN.
0C34R
*****
* UTILITY ROUTINE - PACKPR
*****
* "PACKPR" CONVERTS THE 4 HEX DIGITS IN VALUE TO ASCII
* AND PACKS THEM IN THE 4 BYTES AT POINTER "R"
0C48R C890          PACKPR LHI  COUNT,12      SETUP SHIFT COUNT TODO 4
000C
0C4CR 485E          LH     R,0(LINK)      SETUP POINTER INTO PR
0000
0C50R 086C          REPEAT LHR  REG,VALUE      OBTAIN HEX DIGITS
0C52R CC69          SRHL  REG,0(COUNT)    POSITION DIGITS IN LSD
0000
0C56R C460          NHI   REG,X'F'      ISULATE SINGLE DIGIT
000F
0C5AR CA60          AHI   REG,C'0'      CONVERT TO ASCII NUMERIC
0030
0C5ER C560          CLHI  REG,C'9'+1
003A
0C62R 4280          BL     **8          IF ALPHA, ADD 7
0C6AR 0C6AR          AHI   REG,7
0007
0C66R CA60          STB   REG,0(R)      PACK 7BIT ASCII IN BYTE
0000
0C6ER 0A51          AHR   R,ONE        UPDATE BYTE ADDRESS
0C70R C890          SHI   COUNT,4      DONE?
0004
0C74R 4380          BNL   REPEAT      NO, DO AGAIN
0C50R
0C78R 430E          B      2(LINK)
0002
*****
* UTILITY ROUTINE - PAGING
*****
0C7CR 4850          PAGING LH  R,PAGE      INCREMENT PAGE NUMBER
189CR
0C80R 9365          LBR   REG,R
0C82R 0A61          AHR   REG,ONE
0C84R C560          CLHI  REG,X'3A'
003A

```

ASM11550  
 ASM11560  
 ASM11570  
 ASM11580  
 ASM11590  
 ASM11600  
 ASM11610  
 ASM11620  
 ASM11630  
 ASM11640  
 ASM11650  
 ASM11660  
 ASM11670  
 ASM11680  
 ASM11690  
 ASM11700  
 ASM11710  
 ASM11720  
 ASM11730  
 ASM11740  
 ASM11750  
 ASM11760  
 ASM11770  
 ASM11780  
 ASM11790  
 ASM11800  
 ASM11810  
 ASM11820  
 ASM11830  
 ASM11840  
 ASM11850  
 ASM11860  
 ASM11870  
 ASM11880  
 ASM11890  
 ASM11900

0CB8R	4280	BL	OKPAGE		ASM11910
0CB8R	0CB8R				
0CB8R	CA50	AHI	R,X'0100'	INCREMENT IND DIGIT	ASM11920
	0100				
0C90R	C650	OHI	R,X'1000'	NEEDED TO SUPPRESSED 0	ASM11930
	1000				
0C94R	C450	NHI	R,X'FFF0'		ASM11940
	FFF0				
0C98R	C550	CLHI	R,X'3A30'	LAST PAGE WAS 99	ASM11950
	3A30				
0C9CR	4280	BL	SETPAG		ASM11960
	0CB2R				
0CA0R	D360	LB	REG,PAGE-1	GET HUNDREDTHS POSITION	ASM11970
	189BR				
0CA4R	0A61	AHR	REG,ONE	INCREMENT BY 100	ASM11980
0CA6R	C660	OHI	REG,X'0010'	FORCE ASCII 21 TO ASCII 31	ASM11990
	0010				
0CAAR	D260	STB	REG,PAGE-1	FIRST TIME THRU AFTER PG.99	ASM12000
	189BR				
0CAER	C850	LHI	R,X'3030'	RESTART PAGING AT X00	ASM12010
	3030				
0CB2R	4050	SETPAG	STH	R,PAGE	STORE TENS AND UNITS DIGITS
	189CR				ASM12020
0CB6R	0309	BR	COUNT		ASM12030
0CB8R	D260	OKPAGE	STB	REG,PAGE+1	ASM12040
	189DR				
0CBCR	0309	BR	COUNT		ASM12050
					ASM12060
					ASM12070
					ASM12080
					ASM12090
0CBER	41E0	ENDCHK	BAL	LINK,CHKEND	PRINTF IF NOT PROPR END
	0C2CR				
0CC2R	D210	PRINTF	STB	ONE,ERRORF	ASM12100
	156CR				
0CC6R	C8E0	PRINTR	LHI	LINK,DOLINE	ASM12110
	011CR				
					ASM12120
					ASM12130
					ASM12140
					ASM12150
0CCAR	4850	PRINT	LH	R,ASSEM	RETURN TO DO NEXT LINE IF
	1688R				
0CCER	033E	BZR	LINK		IN CONDITIONAL ASSEMBLY
0CD0R	4850	LH	R,OPRINT		WAS "NOPRINT" OPTION OPTED?
	1620R				ASM12161
0CD4R	033E	BZR	LINK		YES, RETURN WITHOUT PRINTING
0CD6R	0350	LB	R,SR		GET SOURCE'S FIRST CHAR
	179AR				ASM12163
0CDAR	4050	STH	R,SVCHR1		SAVE IT FOR LIST ROUTINE
	1654R				ASM12164
0CDER	D350	LB	R,ERRORS		IF SYMBOL TABLE OVERFLOWED
	156ER				ASM12170
0CE2R	0855	LHR	R,R		PRINT LINE WITH"S" ERROR
0CE4R	4230	BNZ	PRNTER		UNLESS"NOPRINT"WAS OPTED.
	0DACR				ASM12180
0CE8R	D350	LB	R,PR+1		WAS THERE A SEQNUM ERROR
	178BR				ASM12200

0CECR	C550		CLHI	R,X'23'	YES,IF 1ST CHAR IS NUMBER SIGN	ASM12210
	0023					
0CF0R	4330		BE	PRNTER	PRINT UNLESS"NOPRNT"OPTED.	ASM12220
	0DACR					
0CF4R	0788		XHR	TEMP,TEMP	FLAG=NO TITLE ERROR.	ASM12230
0CF6R	C8F0		LHI	EXIT,PRPASS	PRNTCK EXIT FOR PRINT PASSES	ASM12240
	0D12R					
0CFAR	4510	PRNTCK	CLH	ONE,PASS	PRINT OR NON-PRINT PASS?	ASM12250
	161AR					
0CFER	4230		BNE	CHKTWO		ASM12260
	0D0AR					
0D02R	4510		CLH	ONE,OPASS		ASM12270
	161CR					
0D06R	033F		BER	EXIT	PRINT IF 1ST PASS OF PASS1	ASM12280
0D08R	030E		BR	LINK	EXIT,1ST PASS OF PASS2 OR3	ASM12290
0D0AR	4520	CHKTWO	CLH	TWO,PASS		ASM12300
	161AR					
0D0ER	023E		BNER	LINK	EXIT, 3RD PASS OF PASS3	ASM12310
0D10R	030F		BR	EXIT	GO TO PRPASS/STDUMP+8	ASM12320
0D12R	C5E0	PRPASS	CLHI	LINK,EJECT+4	DID TITLE ROUTINE CALL?	ASM12330
	0518R					
0D16R	4230		BNE	*+8	IF SO, PRPASS&NO ERRORS,SET	ASM12340
	0D1ER					
0D1AR	4000		STH	ZERO,LCOUNT	SIGNAL TO FORMFEED& TITLE.	ASM12350
	168AR					
0D1ER	4850	CHKLCT	LH	R,LCOUNT	IF ZERO,TOF,PAGE AND TITLE.	ASM12360
	168AR					
0D22R	4230		BNZ	CHKERR	LINE AWAITING OUTPUT	ASM12370
	0D56R					
0D26R	4850		LH	R,LSTART	IF LSTART NON-ZERO,	ASM12380
	1682R					
0D2AR	4330		BZ	TITLED	I/O & PROCEED NEEDS CHECKING	ASM12390
	0D3AR					
0D2ER	4000		STH	ZERO,LSTART	RESET LSTART=0	ASM12400
	1682R					
0D32R	41D0		BAL	RETURN,IOWAIT	CHECK LAST LINE EACH PAGE	ASM12410
	0D06R					
0D36R	18D2R		DC	PRWAIT	WAIT FOR LIST DEVICE	ASM12420
0D38R	18CAR		DC	PR2BLK	REISSUE LAST LINE/IF TROUBLE	ASM12430
0D3AR	41D0	TITLED	BAL	RETURN,IOWAIT	WAIT& CHECK ON STATUS	ASM12440
	0D06R					
0D3ER	18ACR		DC	TRBLK	WRITE TOF,HEADER & SKIP LINE	ASM12480
0D40R	18ACR		DC	TRBLK	REISSUE I/O WAIT IF TROUBLE	ASM12490
0D42R	4020		STH	TWO,LCOUNT	SET LINE COUNTER=TWO	ASM12500
	168AR					
0D46R	4190		BAL	COUNT,PAGING	BUMP PAGE NO. BY 1	ASM12510
	0C7CR					
0D4AR	0888		LHR	TEMP,TEMP	WAS THIS A TITLE LINE ERROR?	ASM12520
0D4CR	4230		BNZ	CHKERR	YES,LIST SOURCE TITLE,ALSO	ASM12530
	0D56R					
0D50R	C5E0		CLHI	LINK,EJECT+4	DID TITLE ROUTINE CALL PRINT	ASM12540
	0518R					
0D54R	033E		BER	LINK	IF SO GO BACK TO DOLINE	ASM12550
0D56R	C890	CHKERR	LHI	COUNT,-14		ASM12560
	FFF2					
0D5AR	0788		XHR	TEMP,TEMP	FLAG=0000 FOR NO ERRORS ON LINE	ASM12570



0D5CR 4859	AGAIN	LH	R,ERRORS(COUNT)	FETCH LOWEST PRIORITY ERR	ASM12580
156R					
0D60R 9365		LBR	REG,R	SEPARATE FLAG FROM LETTER	ASM12590
0D62R C450		NHI	R,X'0100'		ASM12600
0100					
0D66R 4330		BZ	NOFLG	THIS FLAG NOT SET	ASM12610
0D74R					
0D6AR 0881		LHR	TEMP,ONE	INDICATES AT LEAST ONE ERROR.	ASM12620
0D6CR D260		STB	REG,PR	PACK PR WITH ERROR LETTER	ASM12630
178AR					
0D70R D209		STB	ZERO,ERRORS(COUNT)	CLEAR FLAGS THAT WERE SET.	ASM12640
156R					
0D74R 0A92	NOFLG	AHR	COUNT,TWO	GO CHECK ERROR OF HIGHER	ASM12650
0D76R 4320		BNP	AGAIN	PRIORITY, UNLESS DONE.	ASM12660
0D5CR					
0D7AR 4A80		AH	TEMP,ERRCNT	ADD CUMULATIVE COUNT TO 1/0	ASM12670
1912R					
0D7ER 4080		STH	TEMP,ERRCNT	UPDATE TOTAL ERRORS THIS PASS	ASM12680
1912R					
0D82R 4150		BAL	R,LIST	PRINT ONE LINE	ASM12690
0DCAR					
0D86R C890		LHI	COUNT,14	IN CASE THIS STATEMENT HAS	ASM12700
000E					
0D8AR 41F0		BAL	EXIT,BLKOUT	SUCCESSIVE LINES GENERATED	ASM12710
015ER					
0D8ER C850		LHI	R,X'00'	BLANK OUT 14 BYTES AND	ASM12720
000D					
0D92R D250		STB	R,SR-1	PACK TERMINAL OR	ASM12730
1799R					
0D96R 030E		BR	LINK		ASM12740
0D98R 0788	CHKOPT	XHR	TEMP,TEMP	NOT ERROR IN TITLE LINE.	ASM12750
0D9AR 4850		LH	R,OPRINT	ONLY ON "NOPRNT" OPTION	ASM12760
1620R					
0D9ER 033E		BZR	LINK	RETURN WITHOUT PRINTING	ASM12770
0DA0R D350		LB	R,SR	GET SOURCE'S FIRST CHAR	ASM12771
179AR					
0DA4R 4050		STH	R,SVCHR1	SAVE IT FOR LIST ROUTINE	ASM12772
1654R					
0DA8R 4300		B	CHKLCT	GO CHECK LINE COUNT,ETC.	ASM12780
0D1ER					
0DACR C5E0	PRNTER	CLHI	LINK,EJECT+4	WHEN LINE IN ERROR WAS TITLE	ASM12790
0518R					
0DB0R 4230		BNE	PRNTR1	DON'T ZERO LINECOUNT AND	ASM12800
0DBER					
0DB4R C880		LHI	TEMP,C'0'	SET NO TABBING SIGNAL AND	ASM12801
002A					
0DB8R 4080		STH	TEMP,SVCHR1	SAVE IT FOR LIST ROUTINE	ASM12802
1654R					
0DBCR 0881		LHR	TEMP,ONE	SET FLAG TO REPRINT SOURCE	ASM12810
0DBER 08DE	PRNTR1	LHR	RETURN,LINK	SAVE LINK FOR JOINING LOGIC	ASM12820
0DC0R 41E0		BAL	LINK,PRNTCK=4	USE LINK TO COME BACK	ASM12830
0CF6R					
0DC4R 08ED		LHR	LINK,RETURN	HERE,ON NON-PRINTING PASSES.	ASM12840
0DC6R 4300		B	CHKLCT		ASM12850
0D1ER					
*****					ASM12860

\* LIST ROUTINE TABULATES FREE FORMATTED SOURCE  
 \*\*\*\*\*  
 \* REDEFINE REGISTERS FOR LIST ROUTINE

0004	SRX	EQU	SRX	4	ASM12870
0005	PRX	EQU	R	5	ASM12880
0006	APSFLG	EQU	REG	6	ASM12890
0007	COL1	EQU	HOLD	7	ASM12900
0008	TEMP	EQU	TEMP	8	ASM12910
0009	APS	EQU	COUNT	9	ASM12920
000A	SPC	EQU	P	10	ASM12930
000B	CHAR	EQU	CHAR	11	ASM12940
000C	CARTN	EQU	VALUE	12	ASM12950
000D	COMNT	EQU	RETURN	13	ASM12960
000E	POS	EQU	LINK	14	ASM12970
000F	EXIT	EQU	EXIT	15	ASM12980
0DCAR	D040	LIST	STM	4,SAVREG	ASM12990
0DCER	15FAR				ASM12960
	4850	LH	R,LSTART	1ST TIME THRU EACH PASS	ASM12970
	1682R				ASM12980
0DD2R	4330	BZ	SETL	LSTART=0,SKIP I/O CHECK	ASM12990
0DDER	4100	BAL	RETURN,IOWAIT	CHECK PRIOR I/O PROCEED	ASM13000
0DD6R	00D6R				ASM13010
0DDAR	18D2R	DC	PRWAIT	WAIT & CHECK LAST I/O PROCEED	ASM13020
0DDCR	18CAR	DC	PR2BLK	REISSUE ON BAD STATUS	ASM13030
0DDER	4010	SETL	STM	ONE,LSTART	ASM13040
	1682R			SET LSTART NON-ZERO	ASM13050
0DE2R	D140	LM	4,LMDATA	INITIALIZE GEN REGS	ASM13060
	164ER				ASM13070
0DE6R	D370	LB	COL1,SR	GET LINE'S 1ST CHAR	ASM13080
	179AR				ASM13090
0DEAR	01F3	NEWCHR	BALR	EXIT,GET	ASM13100
0DECR	05BC		CLHR	CHAR,CARTN	ASM13110
0DEER	4330		BE	EOLCHR	ASM13120
	0EAER			THIS IS END OF LINE	ASM13130
0DF2R	0589	CLHR	CHAR,APS		ASM13140
0DF4R	4230	BNE	CHKSPC		ASM13150
	0E00R				ASM13160
0DF8R	08DD	LHR	COMNT,COMNT	HAS THE COMENT STARTED	ASM13170
0DFAR	4230	BNZ	MOVCHR	DON'T SET APS FLAG	ASM13180
	0E06R				ASM13190
0DFER	0761	XHR	APSFLG,ONE	REVERSE APSFLAG	ASM13200
0E00R	058A	CHKSPC	CLHR	CHAR,SPC	ASM13210
0E02R	4330		BE	SPACE	ASM13220
	0E18R			IF AT A SPACE.	ASM13230
0E06P	D2B5	MOVCHR	STB	CHAR,0(PRX)	ASM13240
	0000			MOVE CHAR INTO I/O BUFFER	ASM13250
0E0AR	0A51	AHR	PRX,ONE	INCREMENT INDEX INTO PR2	ASM13260
0E0CR	C550	*CLHI	PRX,PR2+88		ASM13270
	1844R				ASM13280
0E10R	4380	*BNL	POSSEQ	GO POSITION SEQNUM	ASM13290
	0FA2R				ASM13300
0E14R	4300	B	NEWCHR	OTHERWISE CONTINUE	ASM13310
	0DEAR				ASM13320
0E18R	0866	SPACE	LHR	APSFLG,APSFLG	ASM13330
0E1AR	4230		BNZ	MOVCHR	ASM13340
				PRINT AS IS, IF CHAR=SPACE	ASM13350
				AND ITS WITHIN APOSTROPHE'S	ASM13360

0E06R					
0E1ER 0884		LHR	TEMP,SRX	SAVE POINTER IN CASE NOTAB	ASM13280
0E20R 01F3	SCANSP	BALR	EXIT,GET	GET NEXT CHAR AFTER SPACE	ASM13290
0E22R 05BA		CLHR	CHAR,SPC	SCAN OVER SPACES/BUMPING SRX	ASM13300
0E24R 4330		BE	SCANSP		ASM13310
0E20R					
0E28R 05BC		CLHR	CHAR,CARTN	END OF LINE IF	ASM13320
0E2AR 4330		BE	EOLCHR	CARRIAGE RETURN.	ASM13330
0EAER					
0E2ER C550		CLHI	PRX,PR2+16*	DON'T TABULATE 1ST 16 BYTES	ASM13340
17FCR					
0E32R 4280		BL	NOTAB		ASM13350
0E76R					
0E36R C570		CLHI	COL1,C'0'	DON'T TABULATE COMMENT LINES	ASM13360
002A					
0E3AR 4330		BE	NOTAB		ASM13370
0E76R					
0E3ER C8E0		LHI	POS,24+PR2*	TABULATE OPERATION FIELD	ASM13380
1804R					
0E42R 055E		CLHR	PRX,POS	TO COLUMN 25	ASM13390
0E44R 4280		BL	TAB		ASM13400
0E80R					
0E48R C8E0		LHI	POS,30+PR2	TABULATE OPERAND FIELD	ASM13410
180AR					
0E4CR 055E		CLHR	PRX,POS	TO COLUMN 31	ASM13420
0E4ER 4280		BL	TAB		ASM13430
0E80R					
0E52R C8E0		LHI	POS,46+PR2	TABULATE COMMENT FIELD	ASM13440
181AR					
0E56R 055E		CLHR	PRX,POS	TO COLUMN 47	ASM13450
0E58R 4280		BL	TAB-2		ASM13460
0E7ER					
0E5CR C540		CLHI	SRX,PR+8*	HIT SEQNUM,YET	ASM13470
17E2R					
0E60R 4380		BNL	TAB2	YES,SPACE OVER TO SEQNUM	ASM13480
0E94R					
0E64R 08DD		LHR	COMNT,COMNT	IF THIS SPACE WITHIN COMMENT	ASM13490
0E66R 4230		BNZ	NOTAB	DON'T TAB, SEND SPACE AS IS	ASM13500
0E76R					
0E6AR 08D1		LHR	COMNT,ONE	SET COMNT FLAG STARTED	ASM13510
				* SEND ONE SPACE AND THEN SEND 1ST CHAR OF COMMENT	ASM13520
				* DELETE LEADING SPACES OF COMMENT BY NOT RESETTING SRX	ASM13530
0E6CR 02A5		STB	SPC,0 (PRX)		ASM13540
0000					
0E70R 0A51		AHR	PRX,ONE		ASM13550
0E72R 4300		B	MOVCHR		ASM13560
0E06R					
0E76R 0848	NOTAB	LHR	SRX,TEMP		ASM13570
0E78R 08BA		LHR	CHAR,SPC		ASM13580
0E7AR 4300		B	CHKSRX		ASM13590
0E8CR					
				* SEND SPACES UNTIL AT NEXT COLUMN	ASM13600
0E7ER 08D1		LHR	COMNT,ONE	SET COMMENT FLAG	ASM13610
0E80R 02A5	TAB	STB	SPC,0 (PRX)		ASM13620
0000					
0E84R 0A51		AHR	PRX,ONE		ASM13630



1690R						
0EECR 4560	M1	CLH	REG,SAVLOC			ASM13970
15C8R						
				* IF CURRENT LOC AFTER THIS LINE'S ASSEMBLY IS NOT THE		ASM13980
				* SAME AS AT ITS BEGINNING GO TO M2 TO SEND LOAD PROGRAM		ASM13990
				* AND FLIP FIRST IF MODE CHANGED.		ASM14000
0EF0R 4230		BNE	M2			ASM14010
0EFCR						
				* IF THIS LINE'S ASSEMBLY CAUSED NO CHANGE IN LOCATION		ASM14020
				* COUNTER AND NO CHANGE IN MODE, EXIT		ASM14030
0EF4R 4880		LH	TEMP,FLIPS	WITH LOCS SAME,CHECK MODES		ASM14040
1692R						
0EF8R 0558		CLHR	R,TEMP	IF MODES SAME,RNOT=TEMP		ASM14050
0EFAR 023F		BNER	EXIT	LOCS SAME,MODES SAME EXIT		ASM14060
0EFCR 40F0	M2	STH	EXIT,M2END+2	SAVE RETURN ADDRESS		ASM14070
0F10R						
0F00R 4060		STH	REG,W1	SEND NEW LOCATION		ASM14080
15DCR						
0F04R 41F0		BAL	EXIT,FLIP	SEND FLIP IF NECESSARY		ASM14090
0F12R						
0F08R 41F0		BAL	EXIT,PACKPB-10	PACK THE PUNCH BUFFER		ASM14100
0AA0R						
0F0CR 0005		DC	5	WITH CONTROL ITEM 5		ASM14110
0F0ER 4300	M2END	B	0000			ASM14120
0000						
				*****		ASM14130
				* ASSEMBLY MODE CONTROL (ABSOLUTE/RELOCATABLE) ROUTINE		ASM14140
				*****		ASM14150
0F12R 4850	FLIP	LH	R,MODE	HERE TO FLIP ON MODE'S REL		ASM14160
1686R						
0F16R 40F0		STH	EXIT,EXITFP+2			ASM14170
0F32R						
0F1AR 0451		NHR	R,ONE	IS RIGHTMOST RELBIT SET?		ASM14180
0F1CR 4880		LH	TEMP,FLIPS	GET RECORDED TAPE MODE		ASM14190
1692R						
0F20R 0558		CLHR	R,TEMP	IF NOT EQUAL FLIP UNNECESSARY		ASM14200
0F22R 023F		BNER	EXIT			ASM14210
0F24R 0751		XHR	R,ONE	IF EQUAL,REVERSE TAPE MODE		ASM14220
0F26R 4050		STH	R,FLIPS	AND SEND FLIP TO TAPE		ASM14230
1692R						
0F2AR 41F0	SEND3	BAL	EXIT,PACKPB-10	PACK PUNCH BUFFER		ASM14240
0AA0R						
0F2ER 0003		DC	3	WITH CONTROL ITEM 3		ASM14250
0F30R 4300	EXITFP	B	0000	EXIT		ASM14260
0000						
				*****		ASM14310
				* SYMBOL TABLE AND OP TABLE SYMBOL SEARCH ROUTINE		ASM14320
				*****		ASM14330
				* THIS ROUTINE PERFORMS A BINARY SEARCH THROUGH THE TABLE		ASM14340
				* WHOSE LIMITS ARE AT (P) AND (P+2) FOR A MATCHING		ASM14350
				* SYMBOL TO THAT IN EITHER LS OR SS TEMPORARY STORAGE		ASM14360
				* IF NO MATCH IS FOUND,SEARCH RETURNS TO THE FIRST		ASM14370
				* INSTRUCTION FOLLOWING THE CALL WITH (P) & A(NEXT)		ASM14380
				* CONTAINING A POINTER TO THE NEW SYMBOL'S INSERTION PT.		ASM14390
				* IF A MATCH IS FOUND,RETURN IS TO THE SECOND INSTRUCTION		ASM14400
				* FOLLOWING THE CALL WITH (P) & A(NEXT) CONTAINING		ASM14410

\* A POINTER TO THE BEGINNING OF THAT SYMBOL'S ENTRY.

\*\*\*\*\*

0F34R	48AD	SEARCH	LH	P,0(RETURN)	GET POINTER TO TABLE POINTERS	ASM14420
	0000					ASM14430
0F38R	486D		LH	REG,2(RETURN)	GET POINTER TO TEMP STORAGE	ASM14440
	0002					ASM14450
0F3CR	CAD0		AHI	RETURN,4	BUMP RETURN BEYOND TWO DC'S	ASM14460
	0004					ASM14470
0F40R	485A		LH	R,2(P)	INITIALIZE END POINTER	ASM14480
	0002					ASM14490
0F44R	48AA		LH	P,0(P)	INITIALIZE CURRENT POINTER	ASM14500
	0000					ASM14510
0F48R	40A0		STH	P,NEXT	AND"INSERT"POINTER=ORIGIN	ASM14520
	1634R					ASM14530
0F4CR	05A5		CLHR	P,R	DOES TABLE HAVE ANY	ASM14540
0F4ER	033D		BER	RETURN	ENTRIES AT ALL?	ASM14550
0F50R	088A		LHR	TEMP,P	INITIALIZE TOP=TABLE&ORIGIN	ASM14560
0F52R	40A0	COMPAR	STH	P,NEXT	UPDATE"INSERT" POINTER	ASM14570
	1634R					ASM14580
0F56R	C890		LHI	COUNT,4	KEEP COUNT ASIDE FROMP&REG	ASM14590
	0004					ASM14600
0F5AR	487A		LH	HOLD,2(P)	FETCH 1ST CHAR IN TABLENTRY	ASM14610
	0002					ASM14620
0F5ER	48C6		LH	VALUE,2(REG)	FETCH 1ST CHAR OF SYMBOL	ASM14630
	0002					ASM14640
0F62R	C470		NHI	HOLD,X'FF00'	MASK OFF EXTRANEIOUS INFO	ASM14650
	FF00					ASM14660
0F66R	C4C0		NHI	VALUE,X'FF00'		ASM14670
	FF00					ASM14680
0F6AR	057C		CLHR	HOLD,VALUE	DOES IT MATCH 1ST CHAR OF	ASM14690
0F6CR	4330		BE	MATCH1	SYMBOL IN TEMPORARY STORAGE	ASM14700
	0F80R					ASM14710
0F70R	4280	NOTEQL	BL	MOVEUP	LOW?GO ALPHABETICALLY HIGH	ASM14720
	0FC4R					ASM14730
0F74R	05A8	MOVEDN	CLHR	P,TEMP	HIGH?GO ALPHABETICALLY LOW	ASM14740
0F76R	033D		BER	RETURN	IF AT CURRENT TOP RETURN	ASM14750
0F78R	085A		LHR	R,P	SET NEW BOTTOM LIMIT=P	ASM14760
0F7AR	0AA8		AHR	P,TEMP	ADD CURRENT BOTTOM&TOP	ASM14770
0F7CR	4300		B	DIVIDE	AND GO HALVE	ASM14780
	0FE0R					ASM14790
0F80R	487A	MATCH1	LH	HOLD,4(P)	FETCH 2ND/3RD TABLE CHARS	ASM14800
	0004					ASM14810
0F84R	4210		BM	EOTE	UNLESS END OF TABLE ENTRY	ASM14820
	0FB8R					ASM14830
0F88R	4576		CLH	HOLD,4(REG)	DO THEY MATCH TEMP SYMBOL'S	ASM14840
	0004					ASM14850
0F8CR	4230		BNE	NOTEQL	NO	ASM14860
	0F70R					ASM14870
0F90R	0A92		AHR	COUNT,TWO	YES, BUMP COUNT TO 6	ASM14880
0F92R	487A		LH	HOLD,6(P)	FETCH 4TH/5TH TABLE CHARS	ASM14890
	0006					ASM14900
0F96R	4210		BM	EOTE	UNLESS END OF TABLE ENTRY	ASM14910
	0FB8R					ASM14920
0F9AR	4576		CLH	HOLD,6(REG)	DO THEY MATCH TEMP SYMBOL'S	ASM14930
	0006					ASM14940
0F9ER	4230		BNE	NOTEQL	NO	ASM14950

0F70R					
0FA2R 0A92		AHR	COUNT,TWO	YES,BUMP COUNT TO 8	ASM14760
0FA4R 487A		LH	HOLD,8(P)	FETCH 6TH TABLE ENTRY CHAR	ASM14770
0008					
0FA8R 4210		BM	EOTE	UNLESS END OF TABLE ENTRY	ASM14780
0FB8R					
0FACR 4576		CLH	HOLD,8(REG)	DOES IT MATCH TEMP SYNBOL'S	ASM14790
0008					
0FB0R 4230		BNE	NOTEQL	NO	ASM14800
0F70R					
0FB4R 4300	MATCH	B	4(RETURN)	YES, RETURN W/MATCH AT "P"	ASM14810
0004					
0FB8R 4090	EOTE	STH	COUNT,**6	HIT END OF TABLE ENTRY	ASM14820
0FB8R					
0FBCR 4876		LH	HOLD,0000(REG)	IS TEMP SYMBOL ALSO AT END	ASM14830
0000					
0FC0R 4330		BZ	MATCH	IF SO, HAVE A MATCH AT "P"	ASM14840
0FB4R					
0FC4R 0AA9	* TABLE ENTRY HAD TOO FEW CHARS				ASM14850
0FC6R 487A	MOVEUP	AHR	P,COUNT	MOVE P BEYOND LOW ENTRY	ASM14860
0000	ADJUST	LH	HOLD,0(P)		ASM14870
0FCAR 4210		BM	**10		ASM14880
0FD4R					
0FCER 0AA2		AHR	P,TWO		ASM14890
0FD0R 4300		B	ADJUST	FIND -BEGINNING OF ENTRY	ASM14900
0FC6R					
0FD4R 40A0		STH	P,NEXT	UPDATE"INSERT" POINTER	ASM14910
1634R					
0FD8R 05A5		CLHR	P,R	REACHED END OF TABLE LEFT	ASM14920
0FDAR 033D		BER	RETURN	RETURN W/NO MATCH	ASM14930
	* SET NEW TOP LIMIT FROM WHICH WE ARE MOVING HIGHER				ASM14940
0FD4R		LHR	TEMP,P		ASM14950
0FDCR 088A		AHR	P,R	ADD CURRENT TOP&BOTTOM	ASM14960
0FDER 0AA5		BC	NEGADR	CARRY RESULTS FROM NEG ADRD	ASM14970
0FE0R 4280	DIVIDE				ASM14980
0FF8R		SRHL	P,2	DIVIDE BY 4,MULTIPLY BY 2	ASM14990
0FE4R CCA0					ASM15000
0002					
0FE8R 0AAA	ADDADR	AHR	P,P	MIDWAY POINT HALFWORD ADRS	ASM15010
0FEAR 487A	ADJUST	LH	HOLD,0(P)	FIND NEAREST LOWER	ASM15020
0000					ASM15030
0FEER 4210		BM	COMPAR	BEGINNING OF ENTRY	ASM15040
0F52R					
0FF2R 0BA2		SHR	P,2	BACKUP A HALFWORD	ASM15050
0FF4R 4300		B	ADJUST		ASM15060
0FEAR					
0FF8R CCA0	NEGADR	SRHL	P,2	HALVE FOR MIDWAY POINT BUT	ASM15070
0002					ASM15080
0FFCR C6A0		OHI	P,X'4000'	OR CARRY BIT BACK IN	ASM15090
4000					ASM15100
1000R 4300		B	ADDADR		
0FE8R					
*****					ASM15070
* POSTING THE LABEL FIELD SYMBOL IN THE SYMBOL TABLE					ASM15080
*****					ASM15090
* THIS ROUTINE POSTS THE LABEL OF A SOURCE STATEMENT					ASM15100

```

* IF THE SYMBOL'S ALREADY IN THE TABLE,IT MAY HAVE BEEN
* ENTERED AS AN ENTRY,EXTRN, AS A PREVIOUS LABEL OR
* AS A PREVIOUS OPERAND SYMBOL, OR AS IT WAS
* ENCOUNTERED IN SAME POSITION ON A PREVIOUS PASS
*****
1004R 40D0      INSERT  STH  RETURN,INSRTX+2
      117AR
1008R 41D0      BAL    RETURN,SEARCH  AND OBTAIN POINTER
      0F34R
100CR 162AR     DC     STO
100ER 15F0R     DC     LS
1010R 4300      B      INSRTL      NOTFOUND,GQ INSERT AT NEXT
      110ER
1014R 41E0      BAL    LINK,UNPACK  UNPACK SYMBOL IN TABLE
      0BC8R
1018R 0857      LHR   R,HOLD
101AR 0380      LB    TEMP,LS
      15F0R
101ER 0482      NHR   TEMP,TWO
1020R 4330      BZ    LABDEF
      1034R

* FERROR ALREADY SET,SYMBOL IN LABEL OF EQU OR ORG UNDEFINED
1024R 0452      NHR   R,TWO      IS IT DEFINED IN TABLE
* SYMBOL USED THIS LABEL HAS DEFINED VALUE IN TABLE
* DON'T CHANGE IT TO PRESENT UNDEFINED VALUE
1026R 4330      BZ    INSRTX      YES,DEFINED IN TABLE
      1178R

* UNDEFINED LABEL ALREADY IN TABLE AS UNDEFINED
102AR 0461      NHR   REG,ONE      USED BIT SET?
* DON'T DISTURB STRING INDICATED BY USED BIT BEING SET
102CR 4230      BNZ   INSRTX      IF SO,LEAVE ALONE
      1178R
1030R 4300      B      INSRT3      GO UPDATE NEW UNDEFINED VALUE
      10F0R
1034R 0452      LABDEF NHR  R,TWO      IS IT DEFINED INTABLE?
1036R 4330      BZ    INSRT2      SYMBOL. CHANGE VALUE IF "M"
      10A2R

* PLACE FORMAT ERROR ON EQU TO UNDEFINEDS ON PASS1
* THAT GET DEFINED ON PASS 2
103AR 05F0      CLHI  EXIT,EQULAB
      06C6R
103ER 4230      BNE   **16
      104ER
1042R 4510      CLH   ONE,PASS
      161AR
1046R 4330      BE    **8
      104ER
104AR 0210      STB   ONE,ERRORF
      156CR
104ER 0461      NHR   REG,ONE      IF USED BIT NOT SET
1050R 4330      BZ    INSRT3      SYMBOL UNDEF BUT NEVER USED
      10F0R

* UNSTRING
1054R 40C0      STH   VALUE,W1      SYMBOL'S VALUE=REFERENCE
      15DCR
1058R 0857      LHR   R,HOLD      POSITION RELBIT OF VALUE

```

```

ASM15110
ASM15120
ASM15130
ASM15140
ASM15150
ASM15160
ASM15170
ASM15180
ASM15190
ASM15200
ASM15210
ASM15220
ASM15230
ASM15240
ASM15250
ASM15260
ASM15270
ASM15280
ASM15290
ASM15300
ASM15310
ASM15320
ASM15330
ASM15340
ASM15350
ASM15360
ASM15370
ASM15380
ASM15390
ASM15400
ASM15410
ASM15420
ASM15430
ASM15440
ASM15450
ASM15460
ASM15470
ASM15480
ASM15490

```



105AR	41F0 0F16R	BAL	EXIT,FLIP+4	SYMBOL BEING DEFINED,	ASM15500
105ER	41F0 0AA0R	BAL	EXIT,PACKPB-10	PACK REF ADDRESS AND	ASM15520
1062R	0006	DC	6	ITS LOADER CONTROL ITEM 6	ASM15530
1064R	D350 15F0R	LB	R,LS	POSITION RELBIT OF DEFADRS	ASM15540
1068R	41F0 0F16R	BAL	EXIT,FLIP+4	SEND FLIP ON RELBIT	ASM15550
106CR	48C0 15F0R	LH	VALUE,LS		ASM15560
1070R	48A0 1634R	LH	P,NEXT	REESTABLISH POINTER IOSYM	ASM15570
1074R	485A 0000	LH	R,0(P)		ASM15580
1078R	C450 1C00	NHI	R,X'1C00'	USED,EXTRN,ENTRY, BITS	ASM15590
107CR	06C5	Ohk	VALUE,R	00=FWREF,10=ENTRY,01=EXTRN	ASM15600
107ER	D2C0 15DCR	STB	VALUE,W1	V1	ASM15610
1082R	40CA 0000	STH	VALUE,0(P)	SET NEW CODE & V1	ASM15620
1086R	D350 15F3R	LB	R,LS+3	V2	ASM15630
108AR	D25A 0003	STB	R,3(P)		ASM15640
108ER	D250 15DDR	STB	R,W1+1		ASM15650
1092R	41F0 0AA0R	BAL	EXIT,PACKPB-10	SEND DEFINITION ADDRESS AND	ASM15670
1096R	0007	DC	7	AND ITS CONTROL ITEM 7	ASM15680
1098R	41F0 0AA0R	BAL	EXIT,PACKPB-10	OUTPUT CHAIN REFERENCE	ASM15700
109CR	0002	DC	2		ASM15701
109ER	4300 1178R	B	INSRTX		ASM15710
10A2R	485A 0000	INSRT2 LH	R,0(P)	DEFINED SYMBOL FOUND	ASM15720
10A6R	C450 01FF	NHI	R,X'1FF'	MASK RELBIT & VALUE1	ASM15730
10AAR	4860 15F0R	LH	REG,LS	GET NEW VALUE	ASM15740
10AER	C460 01FF	NHI	REG,X'1FF'	MASK RELBIT & VALUE1	ASM15750
10B2R	D38A 0003	LB	TEMP,3(P)	GET 2ND HALF OF VALUE	ASM15760
10B6R	D3C0 15F3R	LB	VALUE,LS+3		ASM15770
10BAR	C470 0004	NHI	HOLD,4	SHOW""ERROR IF DEFINED EXTRN	ASM15780
10BER	4330 10C6R	BZ	*+8		ASM15790
10C2R	D210 156AR	STB	ONE,ERRORM	DON'T SET M BIT,YET	ASM15800
10C6R	0556	CLHR	R,REG	DO VALUES MATCH?	ASM15810
10C8R	4230	BNE	MERROR	NO,MULTIPLE DEFINED ERROR	ASM15820

10CCR	058C	CLHR	TEMP,VALUE		ASM15830
10CER	4330	BE	INSRTX		ASM15840
10D2R	1178R				
10D2R	D210	MERROR	STB	ONE,ERRORM SET MULTIPLE DEFINED	ASM15850
10D6R	156AR				
10D6R	488A	LH	TEMP,0(P)	STORE NEW VALUE& RELBIT	ASM15860
10DAR	0000				
10DAR	C480	NHI	TEMP,X'FE00'	SAVE CODE BITS IN TABLE	ASM15870
10DER	FE00				
10DER	0668	OHR	REG,TEMP	MESH CODE &NEW VALUE	ASM15880
10E0R	C660	OHI	REG,X'2000'	SET MULTIPLE DEFINED BIT	ASM15890
10E4R	2000				
10E4R	406A	STH	REG,0(P)	POST IN TABLE	ASM15900
10E8R	0000				
10E8R	D2CA	STB	VALUE,3(P)		ASM15910
10ECR	0003				
10ECR	4300	B	INSRTX	EXIT WHEN P=LAST HW STORED.	ASM15920
10F0R	1178R				
10F0R	4850	INSRT3	LH	R,LS DEFINING A SYMBOL PREVIOUSLY	ASM15930
10F4R	15F0R				
10F4R	487A	LH	HOLD,0(P)	FETCH CODE IN TABLE	ASM15940
10F8R	00C0				
10F8R	C470	NHI	HOLD,X'0C00'	ISOLATE ENTRY/EXTRN BITS	ASM15950
10FCR	0C00				
10FCR	0657	OHR	R,HOLD	COPY 00=FRF,01=EXT,10=ENT	ASM15960
10FER	405A	STH	R,0(P)	ENTRY BIT&USED BT, EXT	ASM15970
1102R	0000				
1102R	D350	LB	R,LS+3	SET VALUE=DEFINITION ADRS	ASM15980
1106R	15F3R				
1106R	D25A	STB	R,3(P)		ASM15990
110AR	0003				
110AR	4300	B	INSRTX		ASM16000
110ER	1178R				
110ER	C860	INSRTL	LHI	REG,LS POINT TO SYMBOL IN LABEL S	ASM16010
1112R	15F0R				
1112R	4300	B	GETHWS	SKIP RESETTING EXIT &REG	ASM16020
111ER	111ER				
					ASM16030
					ASM16040
					ASM16050
					ASM16060
1116R	40D0	INSRTS	STH	RETURN,INSRTX*2	ASM16070
111AR	117AR				
111AR	C860	LHI	REG,SS	POINT TO SYMBOL IN SS	ASM16070
111ER	15E6R				
111ER	C896	GETHWS	LHI	COUNT,4(REG) ASSUME 2 HALFWORDS TO PACK	ASM16080
1122R	0004				
1122R	4859	LOOPIN	LH	R,0(COUNT) START WITH 3RD HW IN LS/SS	ASM16090
1126R	0000				
1126R	4330	BZ	ENDSYM		ASM16100
112AR	1134R				
112AR	0A92	AHR	COUNT,TWO		ASM16110
112CR	C596	CLHI	COUNT,10(REG)		ASM16120
1130R	000A				
1130R	4280	BL	LOOPIN		ASM16130
1122R	1122R				

1134R	0B96	ENDSYM	SHR	COUNT,REG	ISOLATE NO. OF HWS	ASM16140
1136R	4A90		AH	COUNT,STE	ADD TO CURRENT END SYMTAB	ASM16150
	162CR					
113AR	4590		CLH	COUNT,SIZE	IF NOT LOW, NO ROOM LEFT	ASM16160
	162ER					
113ER	4280		BL	INSRT	GOTO INSERT AT P=NEXT	ASM16170
	114AR					
1142R	D210		STB	ONE,ERRORS	SET SYMBOL TABLE OVERFLOW	ASM16180
	156ER					
1146R	4300		B	PRINTR	PRINT UNLESS"NOPRNT"	ASM16190
	0CC6R					
					* IF"LS"OR"SS"CONTAINS SYMBOL OF LESS THAN 6 CHARS	ASM16200
					* ZERO INDICATES END OF HALFWORDS NECESSARY TO INSERT.	ASM16210
114AR	48A0	INSRT	LH	P,STE		ASM16220
	162CR					
114ER	4090		STH	COUNT,STE		ASM16230
	162CR					
1152R	488A	LOOPST	LH	TEMP,0(P)	MOVE SYMBOLS UP TO NEW STE	ASM16240
	0000					
1156R	4089		STH	TEMP,0(COUNT)		ASM16250
	0000					
115AR	0BA2		SHR	P,TWO		ASM16260
115CR	0B92		SHR	COUNT,TWO		ASM16270
115ER	45A0		CLH	P,NEXT	PASSED POINT OF INSERTION	ASM16280
	1634R					
1162R	4380		BNL	LOOPST	NOT YET,LOOP	ASM16290
	1152R					
1166R	0AA2	LOOPRG	AHR	P,TWO	1ST TIME,ADJUST P=NEXT	ASM16300
1168R	4886		LH	TEMP,0(REG)	FETCH TEMP STORAGE IN LS/SS	ASM16310
	0000					
116CR	408A		STH	TEMP,0(P)	STORE IN SYMTAB	ASM16320
	0000					
1170R	0A62		AHR	REG,TWO	INCREMENT TEMP STORAGE INDEX	ASM16330
1172R	05A9		CLHR	P,COUNT	ANY CELLS LEFT TO FILL?	ASM16340
1174R	4280		BL	LOOPRG	YES,LOOP	ASM16350
	1166R					
1178R	4300	INSRTX	B	0000		ASM16360
	0000					
					*****	ASM16370
					* OPERAND'S EXPRESSION EVALUATOR ROUTINE - AEXP	ASM16380
					*****	ASM16390
					* AN EXPRESSION IS ONE ITEM OR A COMBINATION OF ITEMS	ASM16400
					* LINKED IN THE SOURCE BY A PLUS OR MINUS CHARACTER	ASM16410
					* EACH ITEM CONSISTS OF A SINGLE SYMBOL, DECIMAL NUMBER,	ASM16420
					* THE LOCATION COUNTER "*" ASTERISK SYMBOL, OR CONSTANT	ASM16430
					* SUCH AS:	ASM16440
				X'9ADE'	HEXIDECIMAL CONSTANT	ASM16450
				C','	INSTRUCTION CHARACTER	ASM16460
				H'32767'	DECIMAL CONSTANT	ASM16470
				A( AN EXPRESSION)	"DC" ADDRESS CONSTANT	ASM16480
				C'CHAR STRING'	"DC" CHARACTER CONSTANT	ASM16490
					"DC" FLOATING POINT	ASM16500
				E'1234.5678E-09'	SINGLE PRECISION	ASM16510
				D'1234.5678E-09'	DOUBLE PRECISION	ASM16520
					*****	ASM16530
117CR	40E0	AEXP	STH	LINK,AEXPEX+2		ASM16540

129ER					
1180R	C890	* CLEAR	V1,V2,RELFLG,REL CNT,FRF,MFLAG,PAREN,AND VALUE		ASM16550
	0012	LHI	COUNT,ENDATA-DATA		ASM16560
1184R	4009	CLEAR	STH	ZERO,DATA(COUNT) CLEAR	ASM16570
	1580R				
1188R	0892		SHR	COUNT,TWO	ASM16580
118AR	4310		BNM	CLEAR	ASM16590
	1184R				
118ER	07CC		XHR	VALUE,VALUE	ASM16600
1190R	4850		LH	R,EOL	ASM16610
	15D4R				
1194R	4230		BNZ	SETREL=2	RETURN W/ABSOLUTE ZERO=VALUE
	1292R				
1198R	41F0		BAL	EXIT,SIGN	IF NEG,SET MFLAG
	141AR				
119CR	01F3	* RETURN HERE TO BYPASS SIGN BETWEEN ITEMS OF AN EXPRESSION			ASM16640
		BYPASS	BALR	EXIT,GET	BYPASS SIGN
		* RETURN HERE AFTER PASSING A( FOR ADDRESS CONSTANTS			ASM16650
119ER	07CC	ITEM	XHR	VALUE,VALUE	INITIALIZE VALUE & COUNT=0
11A0R	0799		XHR	COUNT,COUNT	ASM16670
11A2R	4090		STH	COUNT,ITMFRF	ZERO LOCAL ITEM'S FRF FLAG
	15C4R				ASM16680
11A6R	4090		STH	COUNT,REL CNT	ZERO RELOCATABILITY OF ITEM
	15B6R				ASM16690
11AAR	4100		BAL	RETURN,CNSTNT	IS ITEM A CONSTANT
	1394R				ASM16700
11AER	4300		B	NOTCON	NO, BRANCH
	12A0R				ASM16710
11B2R	01F3		BALR	EXIT,GET	YES, GET HERE TO BYPASS
11B4R	40C0	ITEND	STH	VALUE,V1	STORE VALUE OF ONE ITEM
	15BCR				ASM16720
11B8R	4850		LH	R,ITMFRF	WAS IT A FORWARD REF
	15C4R				ASM16730
11BCR	4230		BNZ	CHECKF	IT WAS A FWREF
	11E0R				ASM16740
11C0R	4850		LH	R,FRF	THIS ITEM NOT FWREF,PROCEED
	15B4R				ASM16750
11C4R	4330		BZ	CHECKM	IF NO ITEMS AT ALL ARE FRF
	1238R				ASM16760
11C8R	4510		CLH	ONE,OPASS	FOR PASS1 OF 1,ANY USE OF
	161CR				ASM16770
11CCR	4330		BE	SETERR	FRF/EXTRN IN AN EXP, IS F
	11D8R				ASM16780
11D0R	4850		LH	R,EXTFLG	WAS FRF ITEM AN EXTRN?
	15B2R				ASM16790
11D4R	4330		BZ	XPAREN	NO,DON'T COVER U ERR WITH F ERR
	1268R				ASM16800
11D8R	0210	SETERR	STB	ONE,ERRORF	FRF/EXTRN USED IN EXP
	156CR				ASM16810
11DCR	4300		B	XPAREN	
	1268R				ASM16820
11E0R	4850	CHECKF	LH	R,FRF	THIS ITEM FRF,WAS ANY OTHER?
	15B4R				ASM16830
11E4R	4230		BNZ	XPAREN	IGNORE MULTIPLE FRF'S
	1268R				ASM16840

11E8R 4010 1584R	STH	ONE,FRF	SET THE OVERALL FRF OF EXP	ASM16870
11ECR 4850 1580R	LH	R,COMBO	IS THIS ITEM THE FIRST	ASM16880
11FOR 4230 1214R	BNZ	CHECKP	NO, THERE WERE OTHER ITEMS	ASM16890
11F4R 4510 161CR	CLH	ONE,OPASS	IS THIS PASS 1 OF 1	ASM16900
11F8R 4330 1208R	BE	STRFRF	YES, GO DIRECT TO STORE FWREF	ASM16910
11FCR 4850 15B2R	LH	R,EXTFLG	NO, WAS THIS FEREV AN EXTRN	ASM16920
1200R 4230 1208R	BNZ	STRFRF	YES GO INDIRECT TO STORE IT	ASM16930
1204R D210 1564R	STB	ONE,ERRORU	SET UERR FOR FRF'S ON PASS 2/3	ASM16940
1208R 48C0 15BCR	STRFRF LH	VALUE,V1	ENTIRE VALUE OF EXP=FRF	ASM16950
120CR 4860 15B6R	LH	REG,RELCNT	ONLY AND ITS RELBIT	ASM16960
1210R 4300 1260R	B	SETVAL		ASM16970
1214R 4510 161CR	CHECKP CLH	ONE,OPASS	ITEM=FRF COMBINED W/CONSTANTS	ASM16980
1218R 4230 1224R	BNE	CHECKX	IS THIS PASS 1 OF 1	ASM16990
121CR D210 156CR	STB	ONE,ERRORF	ON PASS1OF 1,FORMAT ERROR	ASM17000
1220R 4300 1208R	B	STRFRF	USE FRF VALUE ONLY FOR STRING	ASM17010
1224R 4850 15B2R	CHECKX LH	R,EXTFLG	ON PASS 2/3FRF=EXTRNDON'T SET U ERR	ASM17020
1228R 4230 1208R	BNZ	STRFRF	* ON PASS 2/3 IF FRF IS AN EXTRN DON'T SET U ERROR BUT STRING REF ADRS DISCARD PRECEDING VALUES	ASM17030 ASM17040
122CR 4300 1204R	B	STRFRF-4	* FOR NON-EXTRN FRF'S ON PASS 2/3 SET UERROR GO SET U ERROR	ASM17050 ASM17060
1230R 4010 1580R	BYPAS1 STH	ONE,COMBO	* FIRST TIME THRU BYPAS1 /SET COMBO TO SHOW MULTIPLE ITEMS * IN EXPRESSION	ASM17070 ASM17080 ASM17090
1234R 4300 119CR	B	BYPASS		ASM17100
1238R 48C0 15BAR	CHECKM LH	VALUE,V2	* HERE TO COMBINE PRECEDING ACCUMULATED ITEM VALUES * WITH CURRENT ITEM'S VALUE AND THEIR RELOCATABILITY BITS FETCH ACCUMULATED VALUE OF EXP	ASM17110 ASM17120 ASM17130
123CR 4860 1588R	LH	REG,RELFLG		ASM17140
1240R 4880 158ER	LH	TEMP,MFLAG	SUBTRACT OR ADD THIS ITEM?	ASM17150
1244R 4330 1258R	BZ	ADD4		ASM17160
			* IF MFLAG SET=1,SUBTRACT CURRENT ITEM'S VALUE IN V1 * FROM EXPRESSION'S ACCUMULATED VALUE IN V2	ASM17170 ASM17180

			* AND LIKEWISE TO THEIR RESPECTIVE RELOCATABILITY BITS	ASM17190
			* * SUBTRACT RELCNT(ITEM'S RELBIT) FROM RELFLG(COMBINED RELBITS)	ASM17200
1248R	4000	STH	ZERO,MFLAG CLEAR MINUS FLAG	ASM17210
	158R			
124CR	48C0	SH	VALUE,V1 SUBTRACT THIS ITEM'S VALUE	ASM17220
	158CR			
1250R	4860	SH	REG,RELCNT AND RELOCATABILITY	ASM17230
	1586R			
1254R	4300	B	SETVAL	ASM17240
	1260R			
			* IF MFLAG RESET=0, ADD CURRENT ITEM'S VALUE IN V1	ASM17250
			* TO EXPRESSION'S ACCUMULATED VALUE IN V2	ASM17260
			* AND LIKEWISE TO THEIR RESPECTIVE RELOCATABILITY BITS	ASM17270
			* ADD RELCNT(ITEM'S RELBIT) TO RELFLG(COMBINED RELBITS)	ASM17280
1258R	4AC0	ADD4 AH	VALUE,V1 ADD THIS ITEM'S VALUE	ASM17290
	158CR			
125CP	4A60	AH	REG,RELCNT AND RELOCATABILITY	ASM17300
	1586R			
1260R	40C0	SETVAL STH	VALUE,V2 STORE ACCUMULATED VALUES	ASM17310
	158AR			
1264R	4060	STH	REG,RELFLG AND ABS/REL COMBO FLAG	ASM17320
	1588R			
1268R	41E0	XPAREN BAL	LINK,CHKPAR IF ) CLEAR PAREN FLAG	ASM17330
	1406R			
126CR	C8E0	LHI	LINK,BYPAS1	ASM17340
	1230R			
1270R	41F0	BAL	EXIT,SIGN DOES EXPRESSION CONTINUE?	ASM17350
	141AR			
1274R	030F	BR	LINK GOTO BYPAS1 IF EXP CONTINUES	ASM17360
1276P	4850	LH	R,PAREN RETURN HERE IF NON + OR -	ASM17370
	15C0R			
127AR	4330	BZ	**8 NEED FOR ) NULLIFIED,SKIP F	ASM17380
	1282R			
127ER	0210	STB	ONE,ERRORF F ERR IF MISSING RIGHT )	ASM17390
	156CR			
1282R	0866	LHR	REG,REG REG CONTAINS ABS/REL COMBO	ASM17400
1284R	4330	BZ	SETREL IF ZERO, ABSOLUTE VALUE	ASM17410
	1294R			
1288R	0561	CLHR	REG,ONE IF ONE, RELOCATABLE VALUE	ASM17420
128AR	4330	BE	SETREL	ASM17430
	1294R			
			* SET R ERROR, IF EXPRESSION COMBINED ILLEGAL REL+REL	ASM17440
			* OR ABSOLUTE VALUE - RELOCATABLE VALUE	ASM17450
128ER	0210	STB	ONE,ERRORR	ASM17460
	1562R			
1292R	0766	XHR	REG,REG TREAT ABS/REL ERROR AS ABS	ASM17470
1294R	4060	SETREL STH	REG,RELFLG SAVE RELOCATABILITY OF EXP	ASM17480
	1588R			
1298R	4850	LH	R,FRF RETURN WITH LOADING FRF	ASM17490
	1584R			
129CR	4300	AEXPEX B	0000	ASM17500
	0000			
			*****	ASM17510
12A0R	41D0	NOTCON BAL	RETURN,SYMBOL IS ITEM A SYMBOL	ASM17520
	094AR			
12A4R	4300	B	NOTSYM NO,BRANCH	ASM17530

12A8R	1354R 41D0 0F34R	BAL	RETURN,SEARCH	IF NOT IN TABLE	ASM17540
12ACR	162AR	DC	STO		ASM17550
12AER	15E6R	DC	SS		ASM17560
12B0R	4300 1324R	B	NOTEND	SYMBOL'S 1ST USE ANYWHERE	ASM17570
		* SYMBOL BEING USED IN PRESENT OPERAND EXISTS IN SYMTAB			ASM17580
		* AT POINTER "P"			ASM17590
12B4R	41E0 0BC8R	BAL	LINK,UNPACK	OBTAIN CODES & VALUE	ASM17600
12B8R	0857	LHR	R,HOLD		ASM17610
12BAR	0452	NHR	R,TWO	ISOLATE DEFINITION BIT	ASM17620
12BCR	4230 12CAR	BNZ	UNDEF		ASM17630
		* SYMBOL IS DEFINED RETURN TO "AEXP" W/VALUE & RELCNT			ASM17640
12C0R	0471	USEVAL	NHR HOLD,ONE	ISOLATE VALUE'S RELBIT	ASM17650
12C2R	4070 15B6R	STH	HOLD,RELCNT	IF RELBIT SET, RELFLG GETS ET	ASM17660
12C6R	4300 11B4R	B	ITEND	BRANCH TO STORE VALUE	ASM17670
		* SYMBOL BEING USED IN PRESENT OPERAND WAS FOUND IN			ASM17680
		* SYMBOL TABLE BUT STILL UNDEFINED.			ASM17690
		* SYMBOL'S CODES & VALUES UNPACKED			ASM17700
12CAR	4010 15C4R	UNDEF	STH ONE,ITMFRF	SET THIS ITEM'S FRF FLAG	ASM17710
12CER	0857	LHR	R,HOLD	GET CODE OF UNDEFINED SYMBOL	ASM17720
12D0R	C450 0004	NHI	R,4	IS THE UNDEFINED SYMBOL AN	ASM17730
12D4R	4330 12DCR	BZ	**8	AN EXTRN	ASM17740
12D8R	4010 15B2R	STH	ONE,EXTFLG	IF SO,SET THE EXTRN FLAG	ASM17750
12DCR	4850 15B4R	LH	R,FRF	WERE ANY OTHER ITEMS FRFS	ASM17760
12E0R	4230 1314R	BNZ	DONTST	DON'T STRING MULTIPLE FRFS	ASM17770
		* DON'T THREAD SYMBOL'S PRESENT USAGE IN STRING IF			ASM17780
		* ITS NOT BEING USED IN AN AFIELD OPERAND			ASM17790
12E4R	4850 15CER	LH	R,AFIELD		ASM17800
12E8R	4330 1314R	BZ	DONTST		ASM17810
		* IF USED IN AFIELDAND USED BIT SET,CONTINUE THREADING			ASM17820
		* IF USED IN AFIELD BUT USED BIT NOT SET , START STRING			ASM17830
12ECR	0461	NHR	REG,ONE	USED BIT SET?	ASM17840
12EER	4230 1302R	BNZ	CHAIN		ASM17850
12F2R	485A 0000	LH	R,0(P)		ASM17860
12F6R	C650 1000	OHI	R,USED BIT	SET USED BIT	ASM17870
12FAR	405A 0000	STH	R,0(P)		ASM17880
12FER	07CC	XHR	VALUE,VALUE		ASM17890
1300R	0777	XHR	HOLD,HOLD		ASM17900

				* SYMBOL BEING USED AT CURRENT LOC+RFCF IN CURRENT MODE	ASM17910
1302R	0471	CHAIN	NHR	HOLD,ONE ISOLATE RELBIT	ASM17920
1304R	4070		STH	HOLD,RELCNT	ASM17930
	15B6R				
1308R	40C0		STH	VALUE,V1 SAVE OLD REFADRS AS VALUE	ASM17940
	15BCR				
130CR	41E0		BAL	LINK,PACREF	ASM17950
	13E0R				
1310R	4300		B	ITEND+4	ASM17960
	11B8R				
				* IF THIS SYMBOL HAS NEVER BEEN STRUNG PACK A SINGLE	ASM17970
				* REFADRS BUT LEAVE USED BIT CLEAR	ASM17980
1314R	0461	DONTST	NHR	REG,ONE USED BIT SET?	ASM17990
1316R	4230		BNZ	USEVAL USE LAST REFADRS AS VALUE	ASM18000
	12C0R				
				* USE LAST REFERENCE ADDRESS IN TABLE	ASM18010
131AP	41E0		BAL	LINK,PACREF	ASM18020
	13E0R				
131ER	07CC	NOREF	XHR	VALUE,VALUE RETURN TO AEXP WITH ZERO VALUE	ASM18030
1320R	4300		B	ITEND	ASM18040
	11B4R				
				* SYMBOL BEING USED IN OPERAND NOT IN TABLE YET	ASM18050
				* INSERT AT POINTER "P"	ASM18060
1324R	4850	NOTFND	LH	R,SS	ASM18070
	15E6R				
1328R	4010		STH	ONE,ITMFRF	ASM18080
	15C4R				
132CR	0650		OHI	R,DEFBIT SET UNDEFINED BIT	ASM18090
	0200				
1330R	4050		STH	R,SS	ASM18100
	15E6R				
1334R	4860		LH	REG,AFIELD IF SYMBOL NOT USED IN AFIELD	ASM18110
	15CER				
1338R	4330		BZ	NOUSE SKIP SETTING USED BIT	ASM18120
	1344R				
133CR	0650		OHI	R,USEDDBT SET BIT INDICATING STRING	ASM18130
	1000				
1340R	4050		STH	R,SS	ASM18140
	15E6R				
1344R	08A0	NOUSE	LHI	P,SS PACK REFADRS IN TEMP.SS	ASM18150
	15E6R				
1348R	41E0		BAL	LINK,PACREF	ASM18160
	13E0R				
134CR	41D0		BAL	RETURN,INSRTS INSERT SYMBOL IN TABLE	ASM18170
	1116R				
1350R	4300		B	NOREF VAL=0000F& REL COMBO LEFT	ASM18180
	131ER				
				* IF NONREF,INSERT ZERO FOR VALUE OF SYMBOL FIRST TIME	ASM18190
				*****	ASM18200
1354R	41F0	NOTSYM	BAL	EXIT,DIGIT IS ITEM A DECIMAL NUMBER?	ASM18210
	0908R				
1358R	4300		B	NOTDIG NO,BRANCH	ASM18220
	136ER				
135CR	41E0		BAL	LINK,DEC RETURN W/CHAR=1ST NON-DEC	ASM18230
	1442R				
1360P	0597		CLHR	COUNT,HOLD COUNT MUST=5 OR LESS	ASM18240



1362R	4280	BL	**8		ASM18250
1366R	0210	STB	ONE,ERROR	TRUNCATION ERROR	ASM18260
136AR	4300	B	ITEND	GOTO STORE DECIMAL VALUE	ASM18270
136ER	0580	NOTDIG	CLHI	CHAR,C'..'	IS ITEM=LOCATION COUNTER
1372R	4330	BE	LOCNTR		ASM18290
1376R	D210	STB	ONE,ERRORF		ASM18300
137AR	4300	B	ITEND		ASM18320
137ER	48C0	LOCNTR	LH	VALUE,LOC	LOCATION COUNTER SYMBOL
1382R	4850	LH	R,MODE	GET MODE'S RELBIT	ASM18340
1386R	4020	STH	TWO,ASTFLG	NON-ZERO MEANS LOC COUNTER USED	ASM18350
138AR	4050	STH	R,RELCNT	ADD TO COUNT OF REL'S	ASM18360
138ER	01F3	BALR	EXIT,GET	BYPASS "A"	ASM18370
1390R	4300	B	ITEND		ASM18380
	1184R				ASM18390
				* IF HERE,"CHAR"HOLDS INVALID CHAR, FORMAT INCORRECT	ASM18400
				* BY PRESETTING "TYPECN" TO 28 THE "DC" ROUTINE FORCES	ASM18410
				* "AEXP" TO ACCEPT & EVALUATE ALL CONSTANTS EXCEPT THE	ASM18420
				* TWO CHARACTER C'..' CONSTANT	ASM18430
				* THE MAIN FLOW MAINTAINS "TYPECN" SET TO 12,SUCH THAT	ASM18440
				* ALL OTHER SOURCE STATEMENT OPERANDS ALLOW ONLY	ASM18450
				* DECIMAL,H'...',X'...',AND TWO CHARACTER C'..' CONSTANTS	ASM18460
1394R	D384	CNSTNT	LB	TEMP,0(SRX)	GET SECOND CHAR OF ITEM
1398R	086B	LHR	REG,CHAR	OF OPERAND'S EXPRESSION	ASM18480
139AR	CD60	SLHL	REG,8		ASM18490
139ER	9286	STBR	TEMP,REG	PACK 1ST & 2ND CHARS	ASM18500
13A0R	4850	LH	R,TYPECN	TYPECN=28/DC OR=12/OTHER	ASM18510
13A4R	C850	LOOPCN	SHI	R,4	1ST INDEX=24/DC OR=8/ELSE
13A8R	021D	BMR	RETURN	IT'S NOT A CONSTANT	ASM18530
13AAR	4565	CLH	REG,CTABLE(R)	IF NO MATCH IN CTABLE	ASM18540
13AER	4230	BNE	LOOPCN		ASM18550
13B2R	4855	LH	R,CTABLE+2(R)	GET ADDRESS OF ROUTINE	ASM18560
13B6R	01F3	BALR	EXIT,GET	GET ITEM'S 2ND CHAR AGAIN	ASM18570
13B8R	086B	LHR	REG,CHAR	SAVE 1ST QUOTE OR PAREN	ASM18580
13BAR	01F3	BALR	EXIT,GET	GET 1ST CHAR AFTER QUOTE	ASM18590
13BCR	05B6	CLHR	CHAR,REG	IF 2ND SAME AS 1ST TREAT AS	ASM18600
13BER	033D	BER	RETURN	NULL CONSTANT:V=0,F ERROR	ASM18610

13C0R	0799	XHR	COUNT,COUNT		ASM18620
13C2R	41E5	BAL	LINK,0(R)		ASM18630
	0000				
				* FOR H' OF EXP GOTO DEC & RETURN W/1ST NON-DEC CHR	ASM18640
				* FOR C' OF EXP GOTO CHRCON& RETURN W/1ST NON-CHR CHR	ASM18650
				* FOR X' OF EXP GOTO HEX & RETURN W/1ST NON-HEX CHR	ASM18660
				* FOR A( OF"DC" GOTO ADRS & RETURN TO ITEM	ASM18670
				* FOR C' OF"DC" GOTO DCCHR & GOTO EDITDC FOR LAST HW	ASM18680
				* FOR D' OF"DC" GOTO FLOATD & RETURN TO EDITDC AT END	ASM18690
				* FOR E' OF"DC" GOTO FLOATE & RETURN TO EDITDC	ASM18700
13C6R	0597	ENDCON	CLHR COUNT,HOLD	TRUNCATION ERROR CHECK PT	ASM18710
13C8R	4280	BL	**8		ASM18720
	13D0R				
13CCR	0210	STB	ONE,ERRORT	SET "T" TRUNCATION ERROR	ASM18730
	1566R				
13D0R	C5H0	CLHI	CHAR,X'27'	1ST NON-DEC/CHR/HEX=QUOTE	ASM18740
	0027				
13D4R	433D	BE	4(RETURN)	IF END=QUOTE,GOTO ITEND+4	ASM18750
	0004				
13D8R	0210	STB	ONE,ERRORF	ELSE,"F" ERROR&GOTO ITEND	ASM18760
	156CR				
13DCR	430D	B	6(RETURN)	DON'T BYPASS NON-QUOTE	ASM18770
	0006				
				* "PACREF" PACKS CURRENT LOCATION COUNTER AS ADJUSTED BY	ASM18780
				* RFCF WITH CURRENT MODE AS ABSOLUTE OR RELOCATABLE	ASM18790
				* VALUE FOR SYMBOL'S REFERENCE ADDRESS	ASM18800
				*****	ASM18810
13F0R	48C0	PACREF	LH VALUE,LOC	REFADRS=CURRENT LOC + RFCF	ASM18820
	168FR				
13E4R	4AC0	AH	VALUE,RFCF	RFCF=0/DC,RFCF=2/AFIELD	ASM18830
	15CCR				
13E8R	D2CA	STB	VALUE,3(P)	PACK RIGHT HALF OF VALUE	ASM18840
	0003				
13ECR	CCC0	SRHL	VALUE,8		ASM18850
	0008				
13F0R	D2CA	STB	VALUE,1(P)	PACK LEFT HALF OF VALUE	ASM18860
	0001				
13F4R	D35A	LB	R,0(P)	SET CURRENT MODE AS RELBIT:	ASM18870
	0000				
13F8R	C450	NHI	R,X'FE'	CLEAR OLD REFADRS RELBIT	ASM18880
	00FE				
13FCR	4650	OH	R,MODE	RELBIT OF SYMBOL'S NEW	ASM18890
	1686R				
1400R	D25A	STB	R,0(P)	REFADRS	ASM18900
	0000				
1404R	030E	BR	LINK		ASM18910
					ASM18920
1406R	4850	CHKPAR	LH R,PAREN	IS C')' STILL EXPECTED	ASM18930
	15C0R				
140AR	033E	BZR	LINK	NO,RETURN	ASM18940
140CR	C5B0	CLHI	CHAR,C')'	IS PRESENT CHAR = )	ASM18950
	0029				
1410R	023E	BNER	LINK	NOTYET,LEAVE PAREN FLAGSET	ASM18960
1412R	4000	STH	ZERO,PAREN	YES,CLEAR PAREN FLAG	ASM18970
	15C0R				
1416R	01F3	BALR	EXIT,GET	BYPASS RIGHT PAREN	ASM18980

1418R 030E		BR	LINK		ASM18990
141AR C5B0	SIGN	CLHI	CHAR,C1=1	IF SIGN=PLUS,LEAVE MINUS	ASM19000
002B					ASM19010
141ER 033F		BER	EXIT	FLAG AS IS & GOTO BYPASS+	ASM19020
1420R C5B0		CLHI	CHAR,C1=1	IF SIGN NEITHER +OR-	ASM19030
002D					
1424R 423F		BNE	2(EXIT)	RETURN W/O BYPASSING CHAR	ASM19040
0002					
1428R 4850	SIGN1	LH	R,MFLAG	IF SIGN IS MINUS, SET	ASM19050
158ER					
142CR 0751		XHR	R,ONE	MINUS FLAG.	ASM19060
142ER 4050		STH	R,MFLAG		ASM19070
158ER					
1432R 030F		BR	EXIT	AND RETURN TO BYPASS MINUS	ASM19080
					ASM19090
					ASM19100
1434R 4010	ADRS	STH	ONE,PAREN	THIS MEANS TO EXPECT A RIGHT)	ASM19110
15C0R					
1438R 41F0		BAL	EXIT,SIGN	IS THERE A SIGN INSIDE (	ASM19120
141AR					
143CR 01F3		BALR	EXIT,GET	BYPASS PLUS OR MINUS	ASM19130
143ER 4300		B	ITEM		ASM19140
119ER					
1442R C870	DEC	LHI	HOLD,6	LEGAL COUNT=5 DIGITS	ASM19150
0006					ASM19160
1446R 41F0		BAL	EXIT,SIGN	IS THERE A SIGN INSIDE'	ASM19170
141AR					
144AR 01F3		BALR	EXIT,GET	BYPASS+/-,SET MFLAG IF -	ASM19180
144CR 41F0	LOOPD	BAL	EXIT,DIGIT		ASM19190
09D8R					
1450R 430E		B	0(LINK)	RETURN W/1ST NON=DEC-DIGIT	ASM19200
0000					
1454R 088C		LHR	TEMP,VALUE	MULTIPLY BY TEN	ASM19210
1456R C580		CLHI	TEMP,3277	DO NOT ALLOW OVERFLOW	ASM19220
00CD					
145AR 4380		BNL	ERRDEC	BUT KEEP PASSING DIGITS	ASM19230
147ER					
145ER 0A88		AHR	TEMP,TEMP	2N	ASM19240
1460R 0A88		AHR	TEMP,TEMP	4N	ASM19250
1462R 0AC8		AHR	VALUE,TEMP	4N+1N=5N	ASM19260
1464R 0ACC		AHR	VALUE,VALUE	2*5N=10N	ASM19270
1466R CACB		AHI	VALUE,-X'30'(CHAR)		ASM19280
FFD0					
146AR 4310		BNM	OK	IF GREATER THAN-32768 +32767	ASM19290
1482R					
146ER C5C0		CLHI	VALUE,X'8000'		ASM19300
8000					
1472R 4230		BNE	ERRDEC		ASM19310
147ER					
1476R 4850		LH	R,MFLAG		ASM19320
158ER					
147AR 4230		BNZ	OK		ASM19330
1482R					
147ER D210	ERRDEC	STB	ONE,ERRORT		ASM19340

```

1566R
1482R 01F3 OK BALR EXIT,GET ASM19350
1484R 0A91 AHR COUNT,ONE ASM19360
1486R 4300 B LOOPD ASM19370
144CR
*****
* "HEX" ROUTINE CONVERTS AND PACKS "VALUE" WITH UP TO 4 ASM19380
* ASCII CODED HEXIDECIMAL SOURCE CHARS PRECEDING A ASM19390
* NONHEX CHAR. COUNT MUST BE 4 OR LESS ASM19400
* LEGAL COUNT+1 STORED IN HOLD FOR COMPARISON ON RETURN ASM19410
* UNDER ERROR CONDITIONS, THE RIGHTMOST 4 HEX DIGITS ASM19420
* PRECEDING A TERMINATING ' OR NON-HEX CHAR ARE ASSEMBLED. ASM19430
* ASM19440
148AR C870 HEX LHI HOLD,5 LEGAL NO.+1 OF HEX DIGITS ASM19450
0005
148ER 41F0 BAL EXIT,SIGN IS THERE A SIGN INSIDE' ASM19460
141AR
1492R 01F3 BALR EXIT,GET BYPASS+/-,SET MFLAG IF - ASM19470
1494R C5B0 LOOPHX CLHI CHAR,C'F'+1 ASM19480
0047
1498R 038E BNLR LINK RETURN WITH 1ST NONHEX CHR ASM19490
149AR C5B0 CLHI CHAR,C'9'+1 ASM19500
003A
149ER 4280 BL HEXNUM ASM19510
14ACR
14A2R C5B0 CLHI CHAR,C'A' ASM19520
0041
14A6R 028E BLR LINK RETURN IF NONHEX BETWEEN9A ASM19530
14A8R C8B0 SHI CHAR,7 CONVERT HEX ALPHA CHAR ASM19540
0007
14ACR C5B0 HEXNUM CLHI CHAR,C'0' ASM19550
0030
14B0R 028E BLR LINK RETURN IF ABSOLUTELY NOTHX ASM19560
14B2R CDC0 SLHL VALUE,+4 MOVE PREVIOUS HEX LEFT ASM19570
0004
14B6R C6CB OHI VALUE,-X'30'(CHAR) COPY CHAR W/O ASCII CODE ASM19580
FFD0
14BAR 0A91 AHR COUNT,ONE KEEP TRACK NUMBER OF HEX ASM19590
14BCR 01F3 BALR EXIT,GET GET NEXT HEX CHAR ASM19600
14BER 4300 B LOOPHX ASM19610
1494R
*****
* CHARACTER CONSTANT IN AN INSTRUCTION AFIELD ASM19620
* UNDER ERROR CONDITIONS, THE RIGHTMOST 2 CHARACTERS ASM19630
* PRECEDING A TERMINATING ' OR CARRIAGE RETURN ASM19640
* ARE ASSEMBLED AND TRUNCATION "T" ERROR IS SET ASM19650
* ASM19660
14C2R 0872 CHRCON LHR HOLD,TWO ASM19670
14C4R 08CB LHR VALUE,CHAR PLACE FIRST CHAR AFTER SINGLE ASM19680
14C6R 01F3 BALR EXIT,GET ASM19690
14C8R C5B0 LOOPCH CLHI CHAR,X'27' ASM19700
0027
14CCR 033E BER LINK EXIT ON NEXT SINGLE QUOTE ASM19710
14CER C5B0 CLHI CHAR,X'0D' ASM19720
000D
14D2R 033E BER LINK ASM19730
14D4R 0A91 AHR COUNT,ONE ASM19740
14D6R CDC0 SLHL VALUE,8 ASM19750

```

0008					
14DAR 06CB		OHR	VALUE,CHAR		ASM19760
14DCR 01F3		BALR	EXIT,GET		ASM19770
14DER 4300		B	LOOPCH		ASM19780
14C8R					
*****					
* CHARACTER STRING IN A DC OPERAND					
*****					
* CHAR = 1ST CHAR AFTER 1ST QUOTE					
14E2R C850	DCCHR	LHI	R,8	CONTROL ITEM = 8 2BYTESABS	ASM19790
0008					ASM19800
14E6R 4050		STH	R,PC	PUNCH CODE = 8 FOR ALL HWS	ASM19810
158ER					ASM19820
14EAR 4300		B	FIRST1		ASM19830
14F6R					
14EER 40C0	LOOPDC	STH	VALUE,W1	STORE NEXT TWO CHARS FOR EDIT	ASM19840
15DCR					
14F2R 41F0		HAL	EXIT,EDIT	PACK PR AND PB	ASM19850
0A30R					
14F6R 08CB	FIRST1	LHR	VALUE,CHAR		ASM19860
14F8R CDC0		SLHL	VALUE,8		ASM19870
0008					ASM19880
14FCR C6C0		OHI	VALUE,X'20'	COPY IN SPACE	ASM19890
0020					
1500R 41E0		BAL	LINK,ONECHR	GET NEXT CHAR UNLESS END	ASM19900
1530R					ASM19910
1504R 01F3		BALR	EXIT,GET	BYPASS TERMINAL QUOTE	ASM19920
1506R 4300		B	EDITDC	EDIT LAST HALFWORD	ASM19930
072ER					
150AR 92BC		STBR	CHAR,VALUE	REPLACE SPACE WITH NEXTCHR	ASM19940
150CR 41E0		BAL	LINK,ONECHR	GET NEXT CHAR UNLESS END	ASM19950
1530R					
1510R 01F3		BALR	EXIT,GET	BYPASS TERMINAL QUOTE	ASM19960
1512R 4300		B	EDITDC	EDIT LAST HALFWORD	ASM19970
072ER					
1516R 4890		LH	COUNT,RELCNT	CLEARED INITIALLY BY ITEM	ASM19980
15B6R					
151AR 0A92		AHR	COUNT,TWO		ASM19990
151CR 4090		STH	COUNT,RELCNT	DCCHR DOES TO AEXP ROUTINE	ASM20000
15B6R					
1520R C590		CLHI	COUNT,62	EDIT 62ND HW AND TRUNCATE	ASM20010
003E					
1524R 4280		BL	LOOPDC	IF NOT AT END LOOP	ASM20020
14EER					
1528R D210		STB	ONE,ERRORT	SET T ERROR FOR PRINT	ASM20030
1566R					
152CR 4300		B	EDITDC	EDIT LAST HALFWORD WITH"T"	ASM20040
072ER					
1530R 01F3	ONECHR	BALR	EXIT,GET		ASM20050
1532R C5B0		CLHI	CHAR,X'27'	IF QUOTE RETURN TO BYPASS	ASM20060
0027					
1536R 033E		BER	LINK	AND EDIT LAST HALFWORD	ASM20070
1538R C5B0		CLHI	CHAR,X'0D'	IF CARRIAGE RETURN,SKIP	ASM20080
000D					
153CR 423E		BNE	6(LINK)	BYPASS BUT END WITH F ERROR	ASM20090
0006					

1540R	D210	STB	ONE,ERRORF	ENDING CR CAME BEFORE '	ASM20100
	156CR				
1544R	430E	B	2(LINK)	RETURN TO GO TO EDIT LAST	ASM20110
	0002				

```
*****
*          ***** DATA AREA *****
*****
```

\* TABLES

```
*****
* TABLE OF ASSEMBLER DIRECTIVE ROUTINE ADDRESSES
```

1548R	070ER	DTABLE	DC	DC	CODE=X'E0'	ASM20170
154AR	06CAR		DC	DO	CODE=X'E1'	ASM20180
154CR	06E8R		DC	DS	CODE=X'E2'	ASM20190
154ER	075ER		DC	END	CODE = X'D3'	ASM20200
1550R	0508R		DC	ENTRY	CODE=X'E4'	ASM20210
1552R	069AR		DC	EQU	CODE=X'E5'	ASM20220
1554R	0500R		DC	EXTRN	CODE=X'E6'	ASM20230
1556R	0540R		DC	OPT	CODE=X'E7'	ASM20240
1558R	066ER		DC	ORG	CODE=X'E8'	ASM20250
155AR	04ECR		DC	TITLE	CODE=X'E9'	ASM20260
155CR	0118R		DC	PAUSE	CODE=X'EA'	ASM20270
155ER	22EAR		DC	DB	CODE=X'EB'	ASM20280

```
* THE "END" CODE=X'D3' AND THE "IF" CODE = X'D0'
* THE "IF" AND "END" ROUTINES ARE REACHED PRIOR
* SWITCHING THROUGH "DTABLE"
```

```
*****
* TABLE OF LINE ERROR FLAGS
```

```
* ERROR FLAGS SET DURING EACH SOURCE LINE'S EVALUATION.
* E=EVENESS REQUIRED OF R1 OR R2 FIELD OF INSTRUCTION
* R=RELOCATABILITY OF AN EXPRESSION IN OPERAND ILLEGAL.
* U=UNDEFINED SYMBOL WITHIN AN OPERAND'S EXPRESSION
* T=TRUNCATING EXCESSIVE DIGITS IN CONSTANT NECESSITATED
* O=OPERATION MNEMONIC INVALID
* M=MULTIPLE DEFINED SYMBOL USED IN LABEL
* F=FORMAT ERROR
* S=SYMBOL TABLE OVERFLOW ASSIGNED HIGHEST PRIORITY
```

1560R	0045	ERRORE	DC	X'0045'	ASM20410
1562R	0052	ERRORR	DC	X'0052'	ASM20420
1564R	0055	ERRORU	DC	X'0055'	ASM20430
1566R	0054	ERRORT	DC	X'0054'	ASM20440
1568R	004F	ERRORO	DC	X'004F'	ASM20450
156AR	0040	ERRORM	DC	X'0040'	ASM20460
156CR	0046	ERRORF	DC	X'0046'	ASM20470
156ER	0053	ERRORS	DC	X'0053'	ASM20480

```
* AS A PARTICULAR TYPE ERROR OCCURS THE APPROPRIATE FLAG
* ABOVE GET SET EQUAL TO ONE. IF MULTIPLE ERRORS ARE
* FOUND, THE PRINT ROUTINE PACKS THE ERROR OF HIGHEST
* PRIORITY.
* ERRORS FROM HIGH TO LOWEST PRIORITY=S,F,M,O,T,U,R,E.
```

```
*****
* THE FLAGS TABLE CONTAINS SYMBOL TABLE DUMP ERROR
```

```
* FLAGS FOR CONDITIONS MENTIONED IN THE COMMENT FIELD
```

1570R	5520	FLAGS	DC	C'U'	BITS 0010 FORWARD REFERENCE	ASM20500
1572R	2A3E		DC	C'>'	BITS 0100 DEFINED EXTRN	ASM20510
1574R	2A20		DC	C'*'	BITS 0110 UNDEFINED EXTRN	ASM20520
1576R	2A20		DC	C'>'	BITS 1000 DEFINED ENTRY	ASM20530

1578R 2A3C	DC	C'<'	BITS 1010 UNDEFINED ENTRY	ASM20620
157AR 442A	DC	C'D*	BOTH EXT/ENTRY,DEFINED	ASM20630
157CR 442A	DC	C'D*	BOTH EXT/ENTRY,UNDEFINED	ASM20640
*****				
* CONTROL ITEM TABLE CONTAINS NUMBER OF DATA ITEMS FOLLOWING				
* EACH SPECIFIC LOADER CONTROL ITEM IN PC				
157ER 0000	CONTRL DC	0,0	ITEMS 0,1,2,3	ASM20670
0000				ASM20680
1582R	DO	3	ITEMS 4,5,6,7,8,19	ASM20690
1582R 0404	DC	X'0404'		ASM20700
1584R 0404	DC	X'0404'		ASM20700
1586R 0404	DC	X'0404'		ASM20700
1588R 0808	DC	X'0808'	ITEMS A&B	ASM20710
158AR 0C0C	DC	X'0C0C'	ITEMS C&D	ASM20720
158CR 000C	DC	X'000C'	ITEMS E&F	ASM20730
158ER 0000	PC DC	0	PUNCH CONTROL ITEM	ASM20740
1590R 0000	ITEMS DC	0	NUMBER ITEMS TO FOLLOW CONTROL	ASM20750
*****				
* CTABLE CONTAINS SWITCH ADDRESSES FOR DC ROUTINES				
1592R 4327	CTABLE DC	X'4327',CHRCON	IF C' SWITCH TO 1 OR 2 CHR	ASM20760
14C2R				ASM20780
1596R 4827	DC	X'4827',DEC	IF H' SWITCH TO DECIMAL	ASM20790
1442R				
159AR 5827	DC	X'5827',HEX	IF X' SWITCH TO HEXIDECIMAL	ASM20800
148AR				
159ER 4128	DC	X'4128',ADRS	IF A ( SWITCH TO ADDRESS CN	ASM20810
1434R				
15A2R 4327	DC	X'4327',DCCHR	IF C' SWITCH TO CHR STRING	ASM20820
14E2R				
15A6R 4527	DC	X'4527',A(FLOATE)	SINGLE PRECISION FLTPOINT	ASM20830
1EAAR				
15AAR 4427	DC	X'4427',A(FLOATD)	DOUBLE PRECISION FLTPOINT	ASM20840
1ER2R				
* TYPE CONSTANT(TYPECN) SET=12 TO ALLOW EXPRESSION EVALUATOR				
* ROUTINE TO ASSEMBLE ONLY INSTRUCTION CONSTANTS				
* WHEN TYPECN SET = 28 AEXP ASSEMBLES DC CONSTANTS				
15AER 000C	TYPECN DC	12		ASM20850
*****				
* TEMPORARY STORAGE FOR AEXP - EXPRESSION EVALUATOR ROUTINE				
*****				
15B0R	DATA EQU	*		ASM20910
15B0R 0000	COMBO DC	0	SET,EXP HAS MORE THAN 1ITEM	ASM20920
15B2R 0000	EXTFLG DC	0		ASM20930
15B4R 0000	FRF DC	0	FORWARD REFERENCE FLAG	ASM20940
15B6R 0000	RELCNT DC	0	COUNT OF REL COMBINATIONS	ASM20950
15B8R 0000	RELFLG DC	0	=1 IF VALUE RELOCATABLE	ASM20960
15BAR 0000	V2 DC	0	HOLDS TEMP ITEM FOR AEXP	ASM20970
15BCR 0000	V1 DC	0	HOLDS TEMP ITEM FOR AEXP	ASM20980
15BER 0000	MFLAG DC	0		ASM20990
15C0R 0000	PAREN DC	0		ASM21000
15C2R 0000	ASTFLG DC	0		ASM21010
15C2R	ENDATA EQU	**2		ASM21020
15C4R 0000	ITMFRF DC	0	HOLDS FRF OF INDIVIDUAL ITEM	ASM21030
*****				
* TEMPORARIES AND FLAGS				
*****				
				ASM21050
				ASM21060
				ASM21070

```

* XNBITS HOLDS CODE TO DIFFERENTIATE BETWEEN ENTRY OR EXTRN
15C6R 0000 XNBITS DC 0 X'0A00' OR X'0600'
15C8R 0000 SAVLOC DC 0
15CAR 0000 SAVFRF DC 0 SAVE CELL
* RFCF IS A FLAG FOR ADJUSTING PROPER REFERENCE ADDRESSES
* RFCF = 0; REFADRS = LOCATION COUNTER
* RFCF = 2; REFADRS = LOCATION COUNTER + 2
15CCR 0000 RFCF DC 0
* AFIELD IS A FLAG TO THE AEXP ROUTINE TO ALLOW
* STRINGING OF REFERENCE ADDRESSES ONLY THROUGH
* HALFWORD ADDRESSABLE CELL LOCATIONS
15CER 0000 AFIELD DC 0
* FSPACE AND RSPACE ARE INDEXED INTO FOR APPROPRIATE CHARACTER
* SPACES FOR AN ABSOLUTE DEFINED VALUE
* F FOR A FORWARD REFERENCE AND R FOR A RELOCATABLE VALUE
15D0R 2046 FSPACE DC X'2046'
15D2R 2052 RSPACE DC X'2052'
15D4R 0000 EOL DC 0 END OF LINE SIGNAL
15D6R 0000 FLTFLG DC 0 SINGLE PREC.=0,DOUBLE = 1
* SAVE,PFLAG,W1,W2,W3,W4,W5 REQUIRED SEQUENTIALLY FOR FLOAT
* SAVE USED BY MAIN FLOW AND FLOATING POINT ROUTINE
15D8R 0000 SAVE DC 0
* FLAG FOR PERIOD IN DC E' AND D' CONSTANTS
15DAR 0000 PFLAG DC 0
* W1/W2 HOLD(S) ASSEMBLED OBJECT DATA
* W1,2,3,4,5 WORDS REQUIRED FOR FLOATING CONVERSION
15DCR 0000 W1 DC 0
15DER 0000 W2 DC 0
15E0R 0000 W3 DC 0
15E2R 0000 W4 DC 0
15E4R 0000 W5 DC 0
* TEMPORARY SYMBOL STORAGE
15E6R SS DS 10
* TEMPORARY LABEL STORAGE
15F0R LS DS 10
15FAR SAVREG DS 24
1612R 0100 DBCONS DC X'0100',X'FF00',X'0080',X'FF80'
FF00
0080
FF80

*****
* ASSUMED OPTIONS= PASS2,PRINT,PUNCH,STOP,NOSCRATCH,NOSEQ,FLOAT
161AR 0001 PASS DC 1
161CR OPTION
161CR 0002 OPASS DC X'0002' PASS1=1,PASS2=2,PASS3=3
161ER 0001 OPAUSE DC X'0001' STOP=1,GO=0
1620R 0001 OPRINT DC X'0001' PRINT=1,NOPRINT=0
1622R 0001 OPUNCH DC 1 =0,NOPUNCH,=1 PUNCH
1624R 0001 OPNFLT DC 1 =1KEEP FLOAT,=0OVERLAY
1626R 0000 OPSCRT DC 0 0=NOSCRATCH,1=SCRATCH
1628R 0000 * OPSQNM DC 0 =0NOSQNUMCHK,=1CHECK SQNUM

*****
* POINTERS
*****
* POINTERS TO SYMBOL TABLE BOUNDARIES
162AR 23E4R STO DC SYMTAB

```

```

ASM21080
ASM21090
ASM21100
ASM21110
ASM21120
ASM21130
ASM21140
ASM21150
ASM21160
ASM21170
ASM21180
ASM21190
ASM21200
ASM21210
ASM21220
ASM21230
ASM21240
ASM21250
ASM21260
ASM21270
ASM21280
ASM21290
ASM21300
ASM21310
ASM21320
ASM21330
ASM21340
ASM21350
ASM21360
ASM21370
ASM21380
ASM21390
ASM21400
ASM21410
ASM21420
ASM21430
ASM21440

ASM21450
ASM21460
ASM21470
ASM21480
ASM21490
ASM21500
ASM21510
ASM21520
ASM21530
ASM21540
ASM21550
ASM21560
ASM21570
ASM21580
ASM21590
ASM21600

```



162CR 23E4R	STE DC SYMTAB		ASM21610
162EP 3FFF	* POINTER TO MAXIMUM LIMIT SYMBOL TABLE CAN EXPAND TO SIZE DC X'3FFF'		ASM21620
1630R 1918R	* POINTERS TO OP TABLE BOUNDARIES OPORG DC OPTAB		ASM21630
1632R 1EA8R	OPEND DC OPTAB		ASM21640
	* NEXT HOLDS A POINTER TO A SYMBOL TABLE ENTRY		ASM21650
	* EITHER THAT OF A MATCHING SYMBOL OR AS A POINT OF		ASM21660
	* INSERTION, FOR A SYMBOL ABSENT FROM THE TABLE		ASM21670
1634R 0000	NEXT DC 0		ASM21680
1636R 0000	XADRS DC 0	HOLDS TRANSFER ADDRESS OF "END"	ASM21690
	*****		ASM21700
	* INITIALIZATION DATA		ASM21710
	*****		ASM21720
	* PASS & OPTION CONTROL INITIALIZATION DATA		ASM21730
1638R 0001	OPTS DC 1,X'8000',2,1,1,1,1,0,0,SYMTAB,SYMTAB		ASM21740
8000			ASM21750
0002			ASM21760
0001			
0001			
0001			
0000			
0000			
23E4R			
23E4R			
	* LIST ROUTINE'S REGISTER INITIALIZATION DATA		
164ER 178AR	LMDATA DC PR,PR2,0,SR,0,X'27',X'20',0,X'0D',0		ASM21770
17ECR			ASM21780
0000			
179AR			
0000			
0027			
0020			
0000			
0000			
0000			
1654R	SVCHR1 EQU LMDATA+6	HOLDS SOURCE'S FIRST CHAR	ASM21785
	* EACH PASS INITIALIZES GENERAL REGISTERS AND DATA BELOW		ASM21790
	* WITH CONTENTS OF PSETUP		ASM21800
1662R 0000	PSETUP DC 0,1,2,0,0,0,0,0,C'	',PB1,PB2,-1,C' 1'	ASM21810
0001			
0002			
0000	*		
0000			
0000			
0000			
0000			
2020			
2020			
2020			
2020			
16A0R			
1714R			
FFFF			
2031			

```

* THIS ORDERED SET OF PASS INITIALIZED DATA MUST
* RESIDE DIRECTLY BELOW PB1
* LSTART USED =0, PASS STARTED, NO LISTING I/O YET.
* LSTART = 1, LISTING I/O & PROCEED IN PROGRESS
1682R 0000 LSTART DC 0000 LISTING START FLAG
* PSTART USED =0: PASS STARTED, NO PUNCHING I/O ET.
* PSTART =1 PUNCHING I/O & PROCEED IN PROGRESS
1684R 0000 PSTART DC 0000 PASS START FLAG
1686R 0001 MODE DC 1
* ASSEM = 0 FOR CONDITIONAL NON-ASSEMBLY MODE
* ASSEM = NON-ZERO FOR UNCONDITIONAL ASSEMBLY
1688R 0002 ASSEM DC 2 =2 UNCONDITIONAL ASSEMBLY
168AR 0000 *LCOUNT DC 0
168CR 0000 DOC DC 0 HOLDS OPERAND OF "DO" OP
168ER 0000 LOC DC 0 LOCATION COUNTER
1690R 0000 MAXLOC DC 0 MAX LOCATION TO BE ASSEMBLED
* FLIPS = 0 MEANS EVEN NUMBER OF FLIPS SENT TO TAPE / RELOCATABLE
* FLIPS =1 MEANS ODD NUMBER OF FLIPS SENT TO TAPE/ABSOLUTE MODE
1692R 0000 FLIPS DC 0
1694R *SEQNUM DS 8 SEQUENCE # OF SOURCE LINE
169CR 16A0R PBSWCH DC PB1,PB2
1714R

*****
* I/O BUFFERS
*****
16A0R PB1 DS 108 PUNCH BUFFER ONE
170CR 3002 PB1BLK DC X'3002',STATUS,PB1,PB1+107 BINARY/WRITE/PROCEED
0000
16A0R
170BR

1714R PB2 DS 108 PUNCH BUFFER TWO
1780R 3002 PB2BLK DC X'3002',STATUS,PB2,PB2+107 BINARY/WRITE/PROCEED
0000
1714R
177FR

* PBX IS CLEARED WHENEVER THE PUNCH BUFFER IS CLEARED.
* PBX IS AN INDEX INTO THE PB'S 208 4BIT DATA POSITIONS
1788R 0000 PBX DC 0 PUNCH BUFFER INDEX
178AR PR DS 16 PRINT RECORD
179AR SR DS 80 MAX SOURCE LINE LENTH=80BYTFS
17E8R SREND EQU *-2
17EAR 0D0A DC X'0D0A'
17ECR PR2 DS 96 LISTING OUTPUT BUFFER
184CR 0D0A DC X'0D0A'

*****
184ER 0C00 + FFEED DC X'0C00',0,0,0 FORM FEED + NULL CHARS
0000 X'0A0A',X'0A0A',X'0A0A',X'0A0A'
0000
0000

1856R *HEADER DO 3
1856R 2020 *DC X'2020'
1858R 2020 *DC X'2020'
185AR 2020 *DC X'2020'
185CR *TR DS 56 TITLE RECORD
1894R 2020 *DC C PAGE
5041

```

```

ASM21820
ASM21830
ASM21840
ASM21850
ASM21860
ASM21870
ASM21880
ASM21890
ASM21900
ASM21910
ASM21920
ASM21930
ASM21940
ASM21950
ASM21960
ASM21970
ASM21980
ASM21990
ASM22000
ASM22010
ASM22020
ASM22030
ASM22040
ASM22050
ASM22060
ASM22070
ASM22080
ASM22090
ASM22100
ASM22110
ASM22120
ASM22130
ASM22140
ASM22150
ASM22160
ASM22170
ASM22180
ASM22190
ASM22199
ASM22200
ASM22210
ASM22210
ASM22210
ASM22210
ASM22220
ASM22230

```

Address	Code	DC	CI	Listing Page Number	ASM
4745					
2020					
189CR 2031	PAGE	DC	C1 11	LISTING PAGE NUMBER	ASM22240
189ER 0A0D		DC	X'0A0D'	EXTRA LINE FEED AFTER TITLE	ASM22250
*****					ASM22270
* OS SVC PARAMETER CONTROL BLOCKS					ASM22280
*****					ASM22290
* LOGICAL UNIT ASSIGNMENT					ASM22300
* SOURCE INPUT DEVICE = LOGICAL UNIT # 1					ASM22310
* BINARY OUTPUT DEVICE = LOGICAL UNIT # 2					ASM22320
* LISTING DEVICE = LOGICAL UNIT # 3					ASM22330
* SCRATCH DEVICE = LOGICAL UNIT # A					ASM22340
0000	STATUS	EQU	0		
18A0R 0005	SIZES	DC	5,R		ASM22350
0005					ASM22360
18A4R 4801	SRBLK	DC	X'4801',STATUS,SR,SR+79	ASCII/READ/WAIT	ASM22370
0000					
179AR					
17E9R					
18ACR 2803	TRBLK	DC	X'2803',STATUS,FFEEED,HEADER+73		ASM22380
0000					
184ER					
189FR					
18B4R 2004	SCBLK	DC	X'2004',STATUS,SR,SR+79	ASCII/WRITE/PROCEED	ASM22400
0000					
179AR					
17E9R					
18BCR 0804	SCWAIT	DC	X'0804'	WAIT ON LU4,SCRATCH	ASM22410
18BER C004	SCRWD	DC	X'C004',STATUS	REWIND LU4	ASM22420
0000					
18C2R 8201	BKSPFL	DC	X'8201',STATUS	BACKSPACE LUI TO FILE MARK	ASM22430
0000					
18C6R 8803	PREOF	DC	X'8803',STATUS	WRITE END-OF-FILE TO LISTOV	ASM22435
0000					
18CAR 2003	PR2BLK	DC	X'2003',STATUS,PR2,PR2+97	ASCII/WRITE/PROCEED	ASM22440
0000					
18LER					
17E9R					
184DR					
18D2R 0803	PRWAIT	DC	X'0803'	WAIT ON LU3,LIST DEVICE	ASM22450
18D4R 0802	PBWAIT	DC	X'0802'	WAIT ON LU2,OBJECT DEVICE	ASM22460
18D6R 0001	SUSPND	DC	1		ASM22470
18D8R 0006	CNVERT	DC	6,ERRCNT		ASM22480
1912R					
*****					ASM22490
* MESSAGE BUFFERS					ASM22500
*****					ASM22510
18DCR 0007	PASBLK	DC	7,6,C'PASS 1'		ASM22520
0006					
5041					
5353					
2031					
18E5R	PASNUM	EQU	*-1		ASM22530
18E6R 0006	ERBLK1	DC	6,A(ERBLK2+4)		ASM22540
18EER					
18EAR 0007	ERBLK2	DC	7,14,0,0,C' I/O ERROR'		ASM22550
000E					

```

0000
0000
2049
2F4F
2045
5252
4F52
18FCR 4E4F NOERR DC C'NO ERRORS',X'0A0D' ASM22560
2045
5252
4F52
5320
0A0D
1908R 2121 YESERR DC X'2121',C' ERRORS' ASM22570
2045
5252
4F52
5320
1912R 0000 ERRCNT DC 0,0,X'0A0D' ASM22580
0000
0A0D

***** ASM22590
* MNEMONIC OP CODE TABLE FOR INTERDATA ASSEMBLERS. ASM22600
***** ASM22610
* FOR EACH MNEMONIC, BYTE1 CONTAINS CODE ASM22620
* BYTE 2 CONTAINS MACHINE OP CODE FOR INSTRUCTIONS ASM22630
* BYTE 4 CONTAINS EXTENSION FOR EXTENDED INSTRUCTIONS ASM22640
* BYTE 3 CONTAINS FIRST LETTER OF MNEMONIC ASM22650
* BYTES 5,6,ETC.,CONTINUE WITH REMAINING LETTERS OF MNEMONIC ASM22660
* LAST BYTE, IF SUPERFLUOUS, HAS ZERO FILLER ASM22670
* MACHINE INSTRUCTION ENTRY CODE BYTES START WITH X'8' ASM22680
* OPTION STATEMENT MNEMONICS' CODE BYTE START WITH X'9' ASM22690
* ASSEMBLER DIRECTIVE MNEMONICS' ENTRIES HAVE ASM22700
* CODE BYTE STARTING WITH X'E,D, OR C' ASM22710
* BEGINNING OF EACH ENTRY DETERMINED BY NEGATIVE CODE ASM22720
* OF FIRST HALFWORD ASM22730
***** ASM22740
1918R 82EA OPTAB DC X'82EA',X'4180' A ASM22750
4180
191CR 8265 DC X'8265',X'4100',C'BL' ABL ASM22760
4100
424C
1922R 824E DC X'824E',X'4100',X'4348' ACH ASM22770
4100
4348
1928R 880E DC X'880E',X'4100',X'4348',X'5200' ACHR ASM22780
4100
4348
5200
1930R 82E4 DC X'82E4',X'4180',C'CT',X'4500' ACTE ASM22790
4180
4354
4500
1938R 826A DC X'826A',X'4180',X'4500' AE ASM22800
4180
4500

```

193ER	882A 41C0 4552	DC	X'882A',X'41C0',X'4552'	AER	ASM22810
1944R	824A 4100 4800	DC	X'824A',X'4100',X'4800'	AH	ASM22820
194AR	82CA 4100 4849	DC	X'82CA',X'4100',X'4849'	AHI	ASM22830
1950R	8261 4100 484D	DC	X'8261',X'4100',C'HM'	AHM	ASM22840
1956R	880A 4100 4852	DC	X'880A',X'4100',X'4852'	AHR	ASM22850
195CR	826A 4100 4853	DC	X'826A',X'4100',C'HS'	AHS	ASM22860
1962R	82DF 4100 4900	DC	X'82DF',X'4100',X'4900'	AI	ASM22870
1968R	889F 4100 4952	DC	X'889F',X'4100',X'4952'	AIR	ASM22880
196ER	8826 4100 4953	DC	X'8826',X'4100',C'IS'	AIS	ASM22890
1974R	83D5 4100 4C00	DC	X'83D5',X'4100',X'4C00'	AL	ASM22900
197AR	8264 4100 544C	DC	X'8264',X'4100',C'TL'	ATL	ASM22910
1980R	8343 4200	DC	X'8343',X'4200'	B	ASM22920
1984R	8241 4200 414C	DC	X'8241',X'4200',X'414C'	BAL	ASM22930
198AR	8261 4200 414C 4900	DC	X'8261',X'4200',C'AL',X'4900'	BALI	ASM22940
1992R	8801 4200 414C 5200	DC	X'8801',X'4200',X'414C',X'5200'	BALR	ASM22950
199AR	8342 4280 4300	DC	X'8342',X'4280',X'4300'	BC	ASM22960
19A0R	8C02 4280 4352	DC	X'8C02',X'4280',C'CR'	BCR	ASM22970
19A6R	8D20 4280 4353	DC	X'8D20',X'4280',C'CS'	BCS	ASM22980
19ACR	8343	DC	X'8343',X'4230',X'4500'	BE	ASM22990

4230 4500 19B2R 8C03	DC	X'8C03',X'4230',C'ER' BER	ASM23000
4230 4552 19B8R 8D22	DC	X'8D22',X'4230',C'ES' BES	ASM23010
4230 4553 19BER 8822	DC	X'8822',X'4200',C'FB',X'5300' BFBS	ASM23020
4200 4642 5300 19C6R 8243	DC	X'8243',X'4200',X'4643' BFC	ASM23030
4200 4643 19CCR 8263	DC	X'8263',X'4200',C'FC',X'4900' BFCI	ASM23040
4200 4643 4900 19D4R 8803	DC	X'8803',X'4200',X'4643',X'5200' BFCR	ASM23050
4200 4643 5200 19DCR 8823	DC	X'8823',X'4200',C'FF',X'5300' BFFS	ASM23060
4200 4646 5300 19E4R 82DC	DC	X'82DC',X'4200',C'IM' BIM	ASM23070
4200 4940 19EAR 8342	DC	X'8342',X'4280',X'4C00' BL	ASM23080
4280 4C00 19F0R 8C02	DC	X'8C02',X'4280',C'LR' BLR	ASM23090
4280 4C52 19F6R 8D20	DC	X'8D20',X'4280',C'LS' BLS	ASM23100
4280 4C53 19FCR 8342	DC	X'8342',X'4210',X'4D00' BM	ASM23110
4210 4D00 1A02R 8C02	DC	X'8C02',X'4210',C'MR' BMR	ASM23120
4210 4D52 1A08R 8D20	DC	X'8D20',X'4210',C'MS' BMS	ASM23130
4210 4D53 1A0ER 8343	DC	X'8343',X'4280',C'NC' BNC	ASM23140
4280 4E43 1A14R 8C03	DC	X'8C03',X'4280',C'NC',X'5200' BNCR	ASM23150
4280 4E43 5200 1A1CR 8D22	DC	X'8D22',X'4280',C'NC',X'5300' BNCS	ASM23160

	4280				
	4E43				
	5300				
1A24R	8342	DC	X'8342',X'4230',X'4E45'	BNE	ASM23170
	4230				
	4E45				
1A2AR	8C02	DC	X'8C02',X'4230',C'NE',X'5200'	BNER	ASM23180
	4230				
	4E45				
	5200				
1A32R	8D20	DC	X'8D20',X'4230',C'NE',X'5300'	BNES	ASM23190
	4230				
	4E45				
	5300				
1A3AR	8343	DC	X'8343',X'4280',X'4E4C'	BNL	ASM23200
	4280				
	4E4C				
1A40R	8C03	DC	X'8C03',X'4280',C'NL',X'5200'	BNLR	ASM23210
	4280				
	4E4C				
	5200				
1A48R	8D22	DC	X'8D22',X'4280',C'NL',X'5300'	BNLS	ASM23220
	4280				
	4E4C				
	5300				
1A50R	8343	DC	X'8343',X'4210',X'4E4D'	BNM	ASM23230
	4210				
	4E4D				
1A56R	8C03	DC	X'8C03',X'4210',C'NM',X'5200'	BNMR	ASM23240
	4210				
	4E4D				
	5200				
1A5ER	8D22	DC	X'8D22',X'4210',C'NM',X'5300'	BNMS	ASM23250
	4210				
	4E4D				
	5300				
1A66R	8343	DC	X'8343',X'4220',X'4E50'	BNP	ASM23260
	4220				
	4E50				
1A6CR	8C03	DC	X'8C03',X'4220',C'NP',X'5200'	BNPR	ASM23270
	4220				
	4E50				
	5200				
1A74R	8D22	DC	X'8D22',X'4220',C'NP',X'5300'	BNPS	ASM23280
	4220				
	4E50				
	5300				
1A7CR	8342	DC	X'8342',X'4230',X'4E5A'	BNZ	ASM23290
	4230				
	4E5A				
1A82R	8C02	DC	X'8C02',X'4230',C'NZ',X'5200'	BNZR	ASM23300
	4230				
	4E5A				
	5200				
1A8AR	8D20	DC	X'8D20',X'4230',C'NZ',X'5300'	BNZS	ASM23310
	4230				

4E5A 5300 1A92R 8342	DC	X'8342',X'4240',X'4F00' BO	ASM23320
4240 4F00 1A98R 8C02	DC	X'8C02',X'4240',C'OR' BOR	ASM23330
4240 4F52 1A9ER 8D20	DC	X'8D20',X'4240',C'OS' BQS	ASM23340
4240 4F53 1AA4R 8342	DC	X'8342',X'4220',X'5000' BP	ASM23350
4220 5000 1AAAR 8C02	DC	X'8C02',X'4220',C'PR' BPR	ASM23360
4220 5052 1AB0R 8D20	DC	X'8D20',X'4220',C'PS' BPS	ASM23370
4220 5053 1AB6R 8C03	DC	X'8C03',X'4200',X'5200' BR	ASM23380
4200 5200 1ABCR 8D22	DC	X'8D22',X'4200',X'5300' BS	ASM23390
4200 5300 1AC2R 8820	DC	X'8820',X'4200',C'TB',X'5300' BTBS	ASM23400
4200 5442 5300 1ACAR 8242	DC	X'8242',X'4200',X'5443' BTC	ASM23410
4200 5443 1AD0R 8262	DC	X'8262',X'4200',C'TC',X'4900' BTCI	ASM23420
4200 5443 4900 1AD8R 8802	DC	X'8802',X'4200',X'5443',X'5200' BTCR	ASM23430
4200 5443 5200 1AE0R 8821	DC	X'8821',X'4200',C'TF',X'5300' BTFS	ASM23440
4200 5446 5300 1AF8R 82C0	DC	X'82C0',X'4200',X'5848' BXH	ASM23450
4200 5848 1AEER 82C1	DC	X'82C1',X'4200',X'584C',X'4500' BXLE	ASM23460
4200 584C 4500 1AF6R 8343	DC	X'8343',X'4230',X'5A00' BZ	ASM23470
4230 5A00 1AFCR 8C03	DC	X'8C03',X'4230',C'ZR' BZR	ASM23480



4230 5A52 1B02R 8022	DC	X'8022',X'4230',C'ZS' BZS	ASM23490
4230 5A53 1B08R 82E9	DC	X'82E9',X'4380' C	ASM23500
4380 1B0CR 8269	DC	X'8269',X'4380',X'4500' CE	ASM23510
4380 4500 1B12R 8829	DC	X'8829',X'43C0',X'4552' CER	ASM23520
43C0 4552 1B18R 8249	DC	X'8249',X'4300',X'4800' CH	ASM23530
4300 4800 1B1ER 82C9	DC	X'82C9',X'4300',C'HI' CHI	ASM23540
4300 4849 1B24R 82E1	DC	X'82E1',X'4300',C'HN' CHN	ASM23550
4300 484E 1B2AR 8809	DC	X'8809',X'4300',C'HR' CHR	ASM23560
4300 4852 1B30R 82D4	DC	X'82D4',X'4300',C'LB' CLB OF MOD5	ASM23570
4300 4C42 1B36R 8825	DC	X'8825',X'4300',C'LB',X'5200' CLBR	ASM23580
4300 4C42 5200 1B3ER 8245	DC	X'8245',X'4300',X'4C48' CLH	ASM23590
4300 4C48 1B44R 82C5	DC	X'82C5',X'4300',X'4C48',X'4900' CLHI	ASM23600
4300 4C48 4900 1B4CR 8805	DC	X'8805',X'4300',X'4C48',X'5200' CLHR	ASM23610
4300 4C48 5200 1B54R 82E3	DC	X'82E3',X'4380',C'OS',X'4500' COSE	ASM23620
4380 4F53 4500 1B5CR EB00	*DC	X'EB00',X'4400',X'4200' DB	ASM23625
4400 4200 1B62R E000	DC	X'E000',X'4400',X'4300' DC	ASM23630
4400 4300 1B68R 826D	DC	X'826D',X'4480',X'4500' DE	ASM23640
4480 4500			

1B6ER	882D 44C0 4552	DC	X'882D',X'44C0',X'4552'	DER	ASM23650
1B74R	824D 4480 4800	DC	X'824D',X'4480',X'4800'	DH	ASM23660
1B7AR	880D 4480 4852	DC	X'880D',X'4480',X'4852'	DHR	ASM23670
1B80R	E100 4400 4F00	DC	X'E100',X'4400',X'4F00'	DO	ASM23680
1B86R	E200 4400 5300	DC	X'E200',X'4400',X'5300'	DS	ASM23690
1B8CR	D300 4500 4E44	DC	X'D300',X'4500',C'ND'	END	ASM23700
1B92R	E400 4500 4E54	DC	X'E400',X'4500',X'4E54',X'5259'	ENTRY	ASM23710
1B9AR	5259 8895 4500 5053 5200	DC	X'8895',X'4500',C'PS',X'5200'	EPSR	ASM23720
1BA2R	C500 4500 5155	DC	X'C500',X'4500',X'5155'	EQU	ASM23730
1BA8R	8894 4500 5842 5200	DC	X'8894',X'4500',C'XB',X'5200'	EXBR	ASM23740
1BB0R	E600 4500 5854 524E	DC	X'E600',X'4500',X'5854',X'524E'	EXTRN	ASM23750
1BB8R	9801 4600 4C4F 4154	DC	X'9801',X'4600',C'LOAT'	FLOAT	ASM23760
1BC0R	82D8 4600 4E44	DC	X'82D8',X'4600',C'ND'	FND	ASM23770
1BC6R	8898 4600 4E44 5200	DC	X'8898',X'4600',C'ND',X'5200'	FNDR	ASM23780
1BCER	9200 4700 4F00	DC	X'9200',X'4700',X'4F00'	GO	ASM23790
1RD4R	D000 4900 4600	DC	X'D000',X'4900',X'4600'	IF	ASM23800
1BDAR	82E8 4C80	DC	X'82E8',X'4C80'	L	ASM23810

1BDER	9F00 4C00 4142	DC	X'9F00',X'4C00',X'4142'	LAB=	ASM23820
1BE4R	82D3 4C00 4200	DC	X'82D3',X'4C00',X'4200'	LB	ASM23830
1BEAR	8893 4C00 4252	DC	X'8893',X'4C00',X'4252'	LBR	ASM23840
1BF0R	8825 4C00 4353	DC	X'8825',X'4C00',C'CS'	LCS	ASM23850
1BF6R	8268 4C80 4500	DC	X'8268',X'4C80',X'4500'	LE	ASM23860
1BF8R	8828 4CC0 4552	DC	X'8828',X'4CC0',X'4552'	LER	ASM23870
1C02R	8248 4C00 4800	DC	X'8248',X'4C00',X'4800'	LH	ASM23880
1C08R	82C8 4C00 4849	DC	X'82C8',X'4C00',X'4849'	LHI	ASM23890
1C0ER	8808 4C00 4852	DC	X'8808',X'4C00',X'4852'	LHR	ASM23900
1C14R	8268 4C00 4900	DC	X'8268',X'4C00',X'4900'	LI	ASM23910
1C1AR	8824 4C00 4953	DC	X'8824',X'4C00',C'IS'	LIS	ASM23920
1C20R	82D1 4C00 4D00	DC	X'82D1',X'4C00',X'4D00'	LM	ASM23930
1C26R	83C2 4C00 5053 5700	DC	X'83C2',X'4C00',X'5053',X'5700'	LPSW	ASM23940
1C2ER	826C 4D80 4500	DC	X'826C',X'4D80',X'4500'	ME	ASM23950
1C34R	882C 4DC0 4552	DC	X'882C',X'4DC0',X'4552'	MER	ASM23960
1C3AR	824C 4D80 4800	DC	X'824C',X'4D80',X'4800'	MH	ASM23970
1C40R	880C 4D80 4852	DC	X'880C',X'4D80',X'4852'	MHR	ASM23980
1C46R	82DC 4D80 4855	DC	X'82DC',X'4D80',C'HU'	MHU	ASM23990
1C4CR	889C	DC	X'889C',X'4D80',C'HU',X'5200'	MHUR	ASM24000

4080				
4855				
5200				
1C54R	8209	DC	X'8209',X'4D00',C'OV' MOV	ASM24010
	4000			
	4F56			
1C5AR	8899	DC	X'8899',X'4D00',C'OV',X'5200' MOVR	ASM24020
	4000			
	4F56			
	5200			
1C62R	8244	DC	X'8244',X'4E00',X'4800' NH	ASM24030
	4E00			
	4800			
1C68R	82C4	DC	X'82C4',X'4E00',X'4849' NHI	ASM24040
	4E00			
	4849			
1C6ER	8804	DC	X'8804',X'4E00',X'4852' NHR	ASM24050
	4E00			
	4852			
1C74R	82E4	DC	X'82E4',X'4E00',C'HS' NHS	ASM24060
	4E00			
	4853			
1C7AR	8342	DC	X'8342',X'4E00',X'4F50' NOP	ASM24070
	4E00			
	4F50			
1C80R	9600	DC	X'9600',X'4E00',X'4F50',X'4E43',X'4800' NOPNCH	ASM24080
	4E00			
	4F50			
	4E43			
	4800			
1C8AR	8C02	DC	X'8C02',X'4E00',X'4F50',X'5200' NOPR	ASM24090
	4E00			
	4F50			
	5200			
1C92R	9400	DC	X'9400',X'4E00',X'4F50',X'524E',X'5400' NOPRNT	ASM24100
	4E00			
	4F50			
	524E			
	5400			
1C9CR	82DE	DC	X'82DE',X'4F00',X'4300' OC	ASM24110
	4F00			
	4300			
1CA2R	889E	DC	X'889E',X'4F00',X'4352' OCR	ASM24120
	4F00			
	4352			
1CA8R	8246	DC	X'8246',X'4F00',X'4800' OH	ASM24130
	4F00			
	4800			
1CAER	82C6	DC	X'82C6',X'4F00',X'4849' OHI	ASM24140
	4F00			
	4849			
1CB4R	8806	DC	X'8806',X'4F00',X'4852' OHR	ASM24150
	4F00			
	4852			
1CBAR	8266	DC	X'8266',X'4F00',C'HS' OHS	ASM24160
	4F00			

4853 1CC0R C700 4F00 5054	DC	X'C700',X'4F00',X'5054' OPT	ASM24170
1CC6R C800 4F00 5247	DC	X'C800',X'4F00',X'5247' ORG	ASM24180
1CCCR 9001 5000 4153 5331	DC	X'9001',X'5000',X'4153',X'5331' PASS1	ASM24190
1CD4R 9002 5000 4153 5332	DC	X'9002',X'5000',X'4153',X'5332' PASS2	ASM24200
1CDCR 9003 5000 4153 5333	* DC	X'9003',X'5000',X'4153',X'5333' PASS3	ASM24210
1CE4R CA00 5000 4155 5345	DC	X'CA00',X'5000',C'AUSE' PAUSE	ASM24220
1CECR 82E7 5000 4F50	DC	X'82E7',X'5000',C'OP' POP	ASM24230
1CF2R 9401 5000 5249 4E54	DC	X'9401',X'5000',X'5249',X'4E54' PRINT	ASM24240
1CFAR 82E6 5000 5348	DC	X'82E6',X'5000',C'SH' PSH	ASM24250
1D00R 9601 5000 554E 4348	DC	X'9601',X'5000',X'554E',X'4348' PUNCH	ASM24260
1D08R 82D7 5200 4200	DC	X'82D7',X'5200',X'4200' RB	ASM24270
1D0ER 8267 5200 424C	DC	X'8267',X'5200',C'BL' RBL	ASM24280
1D14R 8897 5200 4252	DC	X'8897',X'5200',X'4252' RBR	ASM24290
1D1AR 82DB 5200 4400	DC	X'82DB',X'5200',X'4400' RD	ASM24300
1D20R 889B 5200 4452	DC	X'889B',X'5200',X'4452' RDR	ASM24310
1D26R 82D9 5200 4800	DC	X'82D9',X'5200',X'4800' RH	ASM24320
1D2CR 8899	DC	X'8899',X'5200',C'HR' RHR	ASM24330

5200 4852 1D32R 82EF	DC	X'82EF',X'5280',X'4C00' RL	ASM24340
5280 4C00 1D38R 82EB	DC	X'82EB',X'5280',C'LL' RLL	ASM24350
5280 4C4C 1D3ER 82EE	DC	X'82EE',X'5280',X'5200' RR	ASM24360
5280 5200 1D44R 82EA	DC	X'82EA',X'5280',C'RL' RRL	ASM24370
5280 524C 1D4AR 8266	DC	X'8266',X'5200',C'TL' RTL	ASM24380
5200 544C 1D50R 82EB	DC	X'82EB',X'5380' S	ASM24390
5380 1D54R 824F	DC	X'824F',X'5300',X'4348' SCH	ASM24400
5300 4348 1D5AR 880F	DC	X'880F',X'5300',X'4348',X'5200' SCHR	ASM24410
5300 4348 5200 1D62R 9A01	DC	X'9A01',X'5300',C'CR',X'5400' SCRT	ASM24420
5300 4352 5400 1D6AR 826B	DC	X'826B',X'5380',X'4500' SE	ASM24430
5380 4500 1D70R 882B	DC	X'882B',X'53C0',X'4552' SER	ASM24440
53C0 4552 1D76R 824B	DC	X'824B',X'5300',X'4800' SH	ASM24450
5300 4800 1D7CR 82CB	DC	X'82CB',X'5300',X'4849' SHI	ASM24460
5300 4849 1D82R 880B	DC	X'880B',X'5300',X'4852' SHR	ASM24470
5300 4852 1D88R 826B	DC	X'826B',X'5300',C'HS' SHS	ASM24480
5300 4853 1D8ER 82E2	DC	X'82E2',X'5380',C'IN',X'4500' SINE	ASM24490
5380 494E 4500 1D96R 83E2	DC	X'83E2',X'5300',C'IN',X'5400' SINT	ASM24500
5300 494E 5400			

1D9ER	8827 5300 4953	DC	X'8827',X'5300',C'IS' SIS	ASM24510
1DA4R	82EF 5380 4C41	DC	X'82EF',X'5380',C'LA' SLA	ASM24520
1DAAR	82CF 5300 4C48 4100	DC	X'82CF',X'5300',X'4C48',X'4100' SLHA	ASM24530
1DB2R	82CD 5300 4C48 4C00	DC	X'82CD',X'5300',X'4C48',X'4C00' SLHL	ASM24540
1DBAR	82ED 5380 4C4C	DC	X'82ED',X'5380',C'LL' SLL	ASM24550
1DC0R	8891 5300 4C4C 5300	DC	X'8891',X'5300',C'LL',X'5300' SLLS	ASM24560
1DC8R	9C01 5300 5143 4848	*DC	X'9C01',X'5300',C'QCHK' SQCHK	ASM24570
1DD0R	8264 5380 5145	DC	X'8264',X'5380',C'QE' SQE	ASM24580
1DD6R	8824 5300 5145 5200	DC	X'8824',X'5300',C'QE',X'5200' SQER	ASM24590
1DDER	82EE 5380 5241	DC	X'82EE',X'5380',C'RA' SRA	ASM24600
1DE4R	82CE 5300 5248 4100	DC	X'82CE',X'5300',X'5248',X'4100' SRHA	ASM24610
1DECR	82CC 5300 5248 4C00	DC	X'82CC',X'5300',X'5248',X'4C00' SRHL	ASM24620
1DF4R	82EC 5380 524C	DC	X'82EC',X'5380',C'RL' SRL	ASM24630
1DFAR	8890 5300 524C 5300	DC	X'8890',X'5300',C'RL',X'5300' SRLS	ASM24640
1E02R	82DD 5300 5300	DC	X'82DD',X'5300',X'5300' SS	ASM24650
1E08R	889D 5300 5352	DC	X'889D',X'5300',X'5352' SSR	ASM24660

1E0ER	82E0 5380 5400	DC	X'82E0',X'5380',X'5400'	ST	ASM24670
1E14R	82D2 5300 5442	DC	X'82D2',X'5300',X'5442'	STB	ASM24680
1E1AR	8892 5300 5442 5200	DC	X'8892',X'5300',X'5442',X'5200'	STBR	ASM24690
1E22R	8260 5380 5445	DC	X'8260',X'5380',X'5445'	STE	ASM24700
1E28R	8240 5300 5448	DC	X'8240',X'5300',X'5448'	STH	ASM24710
1E2ER	8260 5300 5449	DC	X'8260',X'5300',C'TI'	STI	ASM24720
1E34R	82D0 5300 544D	DC	X'82D0',X'5300',X'544D'	STM	ASM24730
1E3AR	9201 5300 544F 5000	DC	X'9201',X'5300',X'544F',X'5000'	STOP	ASM24740
1E42R	82E1 5300 5643	DC	X'82E1',X'5300',C'VC'	SVC	ASM24750
1E48R	82C3 5400 4849	DC	X'82C3',X'5400',C'HI'	THI	ASM24760
1E4ER	C900 5400 4954 4C45	DC	X'C900',X'5400',C'TITLE'	TITLE	ASM24770
1E56R	8264 5400 524E	DC	X'8264',X'5400',C'RN'	TRN	ASM24780
1E5CR	8824 5400 524E 5200	DC	X'8824',X'5400',C'RN',X'5200'	TRNR	ASM24790
1E64R	8890 5500 4E43 4800	DC	X'8890',X'5500',X'4E43',X'4800'	UNCH	ASM24800
1E6CR	82D6 5700 4200	DC	X'82D6',X'5700',X'4200'	WB	ASM24810
1E72R	8896 5700 4252	DC	X'8896',X'5700',X'4252'	WBR	ASM24820
1E78R	82DA 5700 4400	DC	X'82DA',X'5700',X'4400'	WD	ASM24830



1E7ER	889A 5700 4452	DC	X'889A',X'5700',X'4452'	WDR	ASM24840
1E84R	82D8 5700 4800	DC	X'82D8',X'5700',X'4800'	WH	ASM24850
1E8AR	8898 5700 4852	DC	X'8898',X'5700',C'HR'	WHR	ASM24860
1E90R	8247 5800 4800	DC	X'8247',X'5800',X'4800'	XH	ASM24870
1E96R	82C7 5800 4849	DC	X'82C7',X'5800',X'4849'	XHI	ASM24880
1E9CR	8807 5800 4852	DC	X'8807',X'5800',X'4852'	XHR	ASM24890
1EA2R	8267 5800 4853	DC	X'8267',X'5800',C'HS'	XHS	ASM24900
1EA8R	8000	OPTABE DC	X'8000'		ASM24910
			*****		ASM24920
			* FLOATING POINT DATA CONSTANT CONVERSION ROUTINES		ASM24930
			*****		ASM24940
			* SINGLE PRECISION FLOATING POINT DC E' CONSTANT		ASM24950
1EAAR	4000	FLOATE STH	ZERO,FLTFLG	SINGLE PRECISION FLAG	ASM24960
	15D6R				
1EAER	4300	B	FLOAT		ASM24970
	1EB6R				
			* DOUBLE PRECISION FLOATING POINT DC D' CONSTANT		ASM24980
1EB2R	4010	FLOATD STH	ONE,FLTFLG	DOUBLE PRECISION FLAG	ASM24990
	15D6R				
			* MFLAG WAS CLEARED BY AEXP ROUTINE		ASM25000
			* CLEAR SAVE,PFLAG,W1,W2,W3,W4,W5		ASM25010
1EB6R	C890	FLOAT	LHI	COUNT,12	ASM25020
	000C				
1EBAR	4009	FLOAT2 STH	ZERO,SAVE(COUNT)		ASM25030
	15D8R				
1EBER	0892	SHR	COUNT,TWO		ASM25040
1EC0R	4310	BNM	FLOAT2		ASM25050
	1EBAR				
1EC4R	41F0	BAL	EXIT,SIGN	CHECK FOR LEADING + OR -	ASM25060
	141AR				
1EC8R	01F3	BALR	EXIT,GET	BYPASS SIGN IF RETURNED HERE	ASM25070
1ECAR	4850	LH	R,MFLAG		ASM25080
	158ER				
1ECER	4330	BZ	**B		ASM25090
	1ED6R				
1ED2R	C850	LHI	R,X'8000'		ASM25100
	8000				
1ED6R	4050	STH	R,V1	SET RESULT'S SIGN	ASM25110
	15BCR				
1EDAR	41F0	FLOAT1 BAL	EXIT,DIGIT	IS PRESENT CHAR A DIGIT?	ASM25120
	09D8R				
1EDER	4300	B	EXPO	RETURN HERE IF NOT A DIGIT	ASM25130

1EE2R	41F0	BAL	EXIT,MULTAD	DIGIT IS REGISTER CHAR	ASM25140
	1F92R				
1EFFAR	4850	LH	R,PFLAG	PERIOD FLAG NOT SET =0	ASM25150
	15DAR				
1EEAR	4330	BZ	FLOAT3		ASM25160
	1EF8R				
1EEER	4890	LH	COUNT,SAVE	FETCH NO. OF PLACES	ASM25170
	15D8R				
1EF2R	0891	SHR	COUNT,ONE	TO THE RIGHT OF DECIMAL PT.	ASM25180
1EF4R	4090	STH	COUNT,SAVE		ASM25190
	15D8R				
1EF8R	01F3	FLOAT3	BALR	EXIT,GET	GET NEXT CHARACTER
1EFAR	4300		B	FLOAT1	RETURN FOR NEXT DIGIT
	1EDAR				ASM25200
1EEFR	C580	EXPO	CLHI	CHAR,C'..'	IS THIS NON-DEC CHAR A "."
	002E				ASM25220
1F02R	4230	BNE	POSSE		NO, POSSIBLE "E"
	1F16R				ASM25230
1F06R	4850	LH	R,PFLAG	WAS A PERIOD ALREADY FOUND	ASM25240
	15DAR				
1F0AR	4230	BNZ	NUMEND	END NUMBER ON SECOND PERIOD	ASM25250
	1F4AR				
1F0ER	4010	STH	ONE,PFLAG	SET 1ST PERIOD FOUND FLAG	ASM25260
	15DAR				
1F12R	4300	B	FLOAT3		ASM25270
	1EF8R				
1F16R	C580	POSSE	CLHI	CHAR,C'E'	IS THIS THE EXPONENT'S "E"
	0045				ASM25280
1F1AR	4230	BNE	NUMEND		ASM25290
	1F4AR				
1F26R	01F3	BALR	EXIT,GET	BYPASS "E" FOR EXPONENT	ASM25300
1F28R	0799	XHR	COUNT,COUNT		ASM25310
1F2AR	4090	STH	COUNT,MFLAG		ASM25320
	15BER				
1F26R	07CC	XHR	VALUE,VALUE		ASM25330
1F28R	41E0	BAL	LINK,DEC	RETURN W/1ST NON-DEC CHAR	ASM25340
	1442R				
1F2AR	4850	LH	R,MFLAG		ASM25350
	15BER				
1F30R	4330	BZ	**10		ASM25360
	1F3AR				
1F34R	C7CC	XHI	VALUE,X'FFFF'	NEGATIVE EXPONENT	ASM25370
	FFFF				
1F38R	0AC1	AHR	VALUE,ONE		ASM25380
1F3AR	C590	CLHI	COUNT,3		ASM25390
	0003				
1F3ER	4280	BL	**8		ASM25400
	1F46R				
1F42R	0210	STB	ONE,ERRORT	TRUNCATION ERROR	ASM25410
	1556R				
1F46R	4300	B	**6		ASM25420
	1F4CR				
1F4AR	07CC	NUMEND	XHR	VALUE,VALUE	ASM25430
1F4CR	4AC0		AH	VALUE,SAVE	ADJUST EXP DUETO DECIMALPT
	15D8R				ASM25440

1F50R	40C0 15D8R	STH	VALUE,SAVE		ASM25450
* READJUST EXPONENT					
1F54R	41F0 21D8R	BAL	EXIT,FIXEXP		ASM25460 ASM25470
1F58R	C5B0 0027	CLHI	CHAR,X'27'	PROPER ENDING QUOTE?	ASM25480
1F5CR	4230 1F66R	BNE	**10		ASM25490
1F60R	01F3	BALR	EXIT,GET	BYPASS QUOTE	ASM25500
1F62R	4300 1F66R	B	**8		ASM25510
1F66R	D210 156CR	STB	ONE,ERRORF	IMPROPER END,NO QUOTE	ASM25520
1F6AR	C850 000A	LHI	R,X'A'	PUNCH CODE = X'A' FOR	ASM25530
1F6ER	4050 158ER	STH	R,PC	4 BYTES ABSOLUTE	ASM25540
1F72R	4850 15D6R	LH	R,FLTFLG	WHICH PRECISION WAS REQUESTED	ASM25550
1F76R	4330 0732R	BZ	EDITDC+4	SINGLE PRECISION	ASM25560
1F7AR	41F0 0A30R	BAL	EXIT,EDIT	DOUBLE PRECISION	ASM25570
1F7ER	4850 15E0R	LH	R,W3	AFTER EDITING 1ST HALF	ASM25580
1F82R	4050 15DCR	STH	R,W1	MOVE SECOND HALF TO EDIT	ASM25590
1F86R	4850 15E2R	LH	R,W4		ASM25600
1F8AR	4050 15DER	STH	R,W2		ASM25610
1F8ER	4300 0732R	B	EDITDC+4		ASM25620
*****					
* REDEFINE REGISTERS FOR FLOATING POINT CONVERSION					
0000	R0	EQU	0		ASM25630
0001	A1	EQU	1		ASM25640
0002	A2	EQU	2		ASM25650
0003	A3	EQU	3		ASM25660
0004	A4	EQU	4		ASM25670
0005	A5	EQU	5		ASM25680
0006	B1	EQU	6		ASM25690
0007	B2	EQU	7		ASM25700
0008	B3	EQU	8		ASM25710
0009	B4	EQU	9		ASM25720
000A	B5	EQU	10		ASM25730
000B	Z1	EQU	11		ASM25740
000C	Z2	EQU	12		ASM25750
000D	Z3	EQU	13		ASM25760
000E	Z4	EQU	14		ASM25770
000F	Z5	EQU	15		ASM25780
0004	BACK	EQU	A4		ASM25790
0003	SHIFT	EQU	A3		ASM25800
*****					
* "MULTAD" MULTIPLIES THE ACCUMULATED CONVERTED					
					ASM25810
					ASM25820
					ASM25830
					ASM25840

\* ARGUMENT DOUBLE OR SINGLE PRECISION  
 \* FLOATING POINT NUMBER IN THE ASSEMBLER'S W1,W2,  
 \* W3,W4, W5 BY FLOATING POINT 10, ADJUSTS THE EXPONENT  
 \* IF NECESSARY TO MAINTAIN NORMALIZATION, AND PROCEEDS  
 \* TO ADD IN THE NEXT DIGIT (CONTAINED IN REGISTER CHAR)  
 MULTAD STH EXIT,LEAVE+2

ASM25850  
 ASM25860  
 ASM25870  
 ASM25880  
 ASM25890  
 ASM25900

1F92R 40F0  
 228AR  
 1F96R 4040  
 227ER  
 1F9AR 4080  
 2282R  
 1F9ER 4100  
 2014R

STH SRX,SAVSRX+2  
 STH CHAR,SAVCHR+2  
 BAL R0,MULTA

ASM25910  
 ASM25920  
 ASM25930

\* R1,W5 HOLD RESULT OF MULTIPLICATION  
 \* PREPARE EXIT FROM MULTAD

ASM25940  
 ASM25950  
 ASM25960

1FA2R C800  
 2272R  
 1FA6R 4860  
 2282R  
 1FAAR C460  
 000F  
 1FAER 0330  
 1FB0R 4830  
 1SDCR  
 1FB4R 9320  
 1FB6R C430  
 7F00  
 1FBAR 0810  
 1FBR 0830  
 4100  
 1FC0R 4000  
 2144R

LHI R0,RESTOR  
 LH B1,SAVCHR+2 FETCH NEXT DIGIT  
 NHI B1,X'F' MASK OFF ASCII 3  
 BZR R0 GOTO RESTORE REGS&RETURN  
 LH SHIFT,W1  
 LBR A2,SHIFT SAVE HIGH FRACTION W/O EXP  
 NHI SHIFT,X'7F00' ISOLATE EXPONENT  
 LHR A1,SHIFT SAVE EXPONENT  
 SHI SHIFT,X'4100' SUBTRACT EXP-1  
 ROUND1 STH R0,OUT+2

ASM25970  
 ASM25980  
 ASM25990  
 ASM26000  
 ASM26010  
 ASM26020  
 ASM26030  
 ASM26040  
 ASM26050

\* GET UP NEXT DIGIT IN FLOATING PT FORMAT

ASM26060  
 ASM26070

1FD0R 0D60  
 0000  
 1FD2R 0777  
 1FD4R 0780  
 1FD6R 0790  
 1FDEP 07AA  
 1FD8R 0E30  
 0008  
 1FDAR 4310  
 1FE4R  
 1FE0R 0860  
 0100  
 1FE2R 4300  
 2120R  
 1FE4R 4340  
 2100R  
 1FE6R 43AC  
 1584R  
 1FE8R 4E00  
 1402R  
 1FE0R 4E00  
 1560R  
 1FE2R 4070

SLHL B1,4  
 XHR B2,B2  
 XHR B3,B3  
 XHR B4,B4  
 XHR B5,B5  
 SRHA SHIFT,B SHIFT RT (EXP-1) HEX DIGITS  
 BNM MULTZD  
 OHI B1,X'4100' UNLESS (EXP-1) WAS NEGATIVE  
 FMULTI  
 MULTZD BAL BACK,SHIFTR SHIFT DIGIT RIGHT  
 AH B5,W5 ADDTO PRESENT ACCUMULATION  
 ACH B4,W4  
 ACH B3,W3  
 ACH B2,W2

ASM26080  
 ASM26090  
 ASM26100  
 ASM26110  
 ASM26120  
 ASM26130  
 ASM26140  
 ASM26150  
 ASM26160  
 ASM26170  
 ASM26180  
 ASM26190  
 ASM26200

	15DER				
1FF4R	0E62	ACHR	B1,A2	ADDIN HIGH FRACTION W/O EXPONENT	ASM26210
1FF6R	0836	LHR	SHIFT,B1		ASM26220
1FF8R	C430	NHI	SHIFT,X'0F00'	CHECK FOR CARRY	ASM26230
	0F00				
1FFCR	4330	BZ	GMULTI	NO CARRY	ASM26240
	212CR				
2000R	CA10	AHI	A1,X'0100'	ADD 1 TO SAVED EXPONENT	ASM26250
	0100				
2004R	4210	BM	OVEADD	ANDSHIFT RIGHT 1 HEXDIGIT	ASM26260
	214ER				
2008R	C830	LHI	SHIFT,1		ASM26270
	0001				
200CR	4140	BAL	BACK,SHIFTR		ASM26280
	2160R				
2010R	4300	B	GMULTI		ASM26290
	212CR				
					ASM26300
				* DOUBLE & A HALF PRECISION MULTIPLY SUBROUTINE	ASM26310
				* MULTIPLIES THE FLOATING POINT NUMBER IN THE FIVEA-REGS	ASM26320
				* BY THE FLOATING POINT NUMBER IN THE FIVE W LOCATIONS	ASM26330
				* LOADED INTO THE 5 BREGS, THE RESULT IS STORED	ASM26340
				* BACK INTO THE FIVE (W) WORD LOCATIONS	ASM26350
2014R	0755	MULTA	XHR A5,A5	MULTIPLY BY 10**+1	ASM26360
2016R	4815	MULTIP	LH A1,CONST(A5)	LOAD POWER OF 10 REQUESTED	ASM26370
	228ER				
201AR	4825	LH	A2,CONST+2(A5)	AS A5 INDEXES INTO CONST	ASM26380
	2290R				
201ER	4835	LH	A3,CONST+4(A5)	A5=0/10**+1,A5=10**-1	ASM26390
	2292R				
2022R	4845	LH	A4,CONST+6(A5)	A5=20/10**+10,A5=30/10**-10	ASM26400
	2294R				
2026R	4855	LH	A5,CONST+8(A5)		ASM26410
	2296R				
202AR	4000	STH	R0,OUT+2		ASM26420
	2144R				
202ER	48B0	LH	Z1,W1		ASM26430
	15DCR				
2032R	48C0	LH	Z2,W2		ASM26440
	15DER				
2036R	48D0	LH	Z3,W3		ASM26450
	15E0R				
203AR	48E0	LH	Z4,W4		ASM26460
	15E2R				
203ER	48F0	LH	Z5,W5		ASM26470
	15E4R				
				* ADD EXPONENTS OF ZREG MULTIPLICAND & AREG MULTIPLIER	ASM26480
2042R	086B	LHR	B1,Z1		ASM26490
2044R	C460	NHI	B1,X'7F00'		ASM26500
	7F00				
2048R	0871	LHR	B2,A1		ASM26510
204AR	C470	NHI	B2,X'7F00'		ASM26520
	7F00				
204ER	0A67	AHR	B1,B2		ASM26530
2050R	4060	STH	B1,EXP		ASM26540
	228CR				

```

* MOVE Z1-Z5 ONE HEX DIGIT LEFT
2054R 9388      LBR  Z1,Z1      REMOVE EXPONENT FROM HIGH FRACTION
2056R 0960      LHI  B1,4
      0004
2058R 0A7F      ADD   AHR  Z5,Z5      SHIFT LEFT BY 4 ADDS
205CR 0E8E      ACHR  Z4,Z4
205ER 0E00      ACHR  Z3,Z3
2060R 0ECC      ACHR  Z2,Z2
2062R 0E88      ACHR  Z1,Z1
2064R 0860      SHI  B1,1
      0001
2066R 0230      BNZ  ADD
      205AD

* MULTIPLY THE ZREGS BY THE AREGS
206CR 0766      XHR  B1,B1      CLEAR RESULTANT BREGS
206ER 0777      XHR  B2,B2
2070R 0788      XHR  B3,B3
2072R 0799      XHR  B4,B4
2074R 07AA      XHR  B5,B5
2076R 0800      LHI  R0,71
      0047
2078R 0C50      (MULTI) SRHL  A5,1
      0001
207ER 4380      BFC  8,AMULTI
      208CR
2082R 0AAF      AHR  B5,Z5      BIT SET,ADDIN NUMBER
2084R 0E9E      ACHR  B4,Z4
2086R 0E8D      ACHR  B3,Z3
2088R 0E7C      ACHR  B2,Z2
208AR 0E6B      ACHR  B1,Z1
208CR 0C40      AMULTI SRHL  B5,1
      0001
2090R 0C90      SRHL  B4,1
      0001
2094R 4380      BFC  8,**+8
      209CR
2098R 06A0      OHI  B5,X'8000'
      8000
209CR 0C80      SRHL  B3,1
      0001
20A0R 4380      BFC  8,**+8
      20A8R
20A4R 0690      OHI  B4,X'8000'
      8000
20A8R 0C70      SRHL  B2,1
      0001
20ACR 4380      BFC  8,**+8
      20B4R
20B6R 0680      OHI  B3,X'8000'
      8000
20B8R 0C60      SRHL  B1,1
      0001
20BBR 4380      BFC  8,**+8
      20C0R
20BCR 0670      OHI  B2,X'8000'
      8000

```

ASM26550  
 ASM26560  
 ASM26570  
  
 ASM26580  
 ASM26590  
 ASM26600  
 ASM26610  
 ASM26620  
 ASM26630  
  
 ASM26640  
  
 ASM26650  
 ASM26660  
 ASM26670  
 ASM26680  
 ASM26690  
 ASM26700  
 ASM26710  
  
 ASM26720  
 ASM26730  
  
 ASM26740  
 ASM26750  
 ASM26760  
 ASM26770  
 ASM26780  
 ASM26790  
  
 ASM26800  
  
 ASM26810  
  
 ASM26820  
  
 ASM26830  
  
 ASM26840  
  
 ASM26850  
  
 ASM26860  
  
 ASM26870  
  
 ASM26880  
  
 ASM26890  
  
 ASM26900  
  
 ASM26910

20C0R	CC40 0001	BMULTI	SRHL	A4,1		ASM26920
20C4R	4380		BFC	8,++8		ASM26930
20C8R	20CCR C650 8000		OHI	A5,X'8000'		ASM26940
20CCR	CC30 0001		SRHL	A3,1		ASM26950
20D0R	4380		BFC	8,++8		ASM26960
20D8R	20D8R C640 8000		OHI	A4,X'8000'		ASM26970
20D8R	CC20 0001		SRHL	A2,1		ASM26980
20DCR	4380		BFC	8,++8		ASM26990
20E0R	20E4R C630 8000		OHI	A3,X'8000'		ASM27000
20E4R	CC10 0001		SRHL	A1,1		ASM27010
20E8R	4380		BFC	8,++8		ASM27020
20ECR	20F0R C620 8000		OHI	A2,X'8000'		ASM27030
20F0R	CB00 0001	CMULTI	SHI	R0,1		ASM27040
20F4R	4310 207AR		BNM	DMULTI		ASM27050
20F8R		* NORMALIZE RESULT IN BREGS				ASM27060
20F8R	4810 228CR	ENDMUL	LH	A1,EXP		ASM27070 ASM27080
20FCR	08B6		LHR	Z1,B1	GET EXPONENT	ASM27090
20FER	4330 2142R		BZ	OUT	EXIT IF ANSWER ZERO	ASM27100
2102R	C830 0001		LHI	SHIFT,1		ASM27110
2106R	C4R0 0F00		NHI	Z1,X'0F00'	SHIFT RIGHT REG'D FOR CARRY	ASM27120
210AR	4230 2118R		BNZ	EMULTB	YES	ASM27130
210ER	0733		XHR	SHIFT,SHIFT		ASM27140
2110R	CB10 0100		SHI	A1,X'0100'	ADJUST EXPONENT	ASM27150
2114R	4280 2146R		BL	UNDADD	UNDERFLOW	ASM27160
2118R	4140 2160R	EMULTB	BAL	BACK,SHIFTR		ASM27170
211CR	CA10 4000	* EMULTI	AHI	A1,X'4000'		ASM27180 ASM27190
2120R	4280 214ER		BC	OVEADD		ASM27200
2124R	CA10 8000		AHI	A1,X'8000'		ASM27210
2128R	4210		BM	UNDADD		ASM27220

```

2146R
212CR 0661  GMULTI  OHR  B1,A1
212ER 4060  FMULTI  STH  B1,W1          STORE RESULT OF MULTIPLY
150CR
2132R 4070          STH  B2,W2          ASM27250
150ER
2136R 4080          STH  B3,W3          ASM27260
1509R
213AR 4090          STH  B4,W4          ASM27270
15E2R
213ER 40A0          STH  B5,W5          ASM27280
15E4R
2142R 40B0  OHR  B  0000          RETURN          ASM27290
0000
* UNDERFLOW - SET RESULT TO ZERO
2146R 0766  UNDAOD  XHR  B1,B1          ASM27300
2148R 0777          XHR  B2,B2          ASM27310
214AR 4300          B  SETREG          ASM27320
2156R          ASM27330
* OVERFLOW - SET RESULT TO MAXIMUM NUMBER
214ER 0860  OVEAOD  LHI  B1,X'7FFF'          ASM27340
7FFF          ASM27350
2152R 0870          LHI  B2,X'FFFF'          ASM27360
FFFF
2156R 0887  SETREG  LHR  B3,B2          ASM27370
2158R 0897          LHR  B4,B2          ASM27380
215AR 08A7          LHR  B5,B2          ASM27390
215CR 4000          B  FMULTI          ASM27400
2162R
* SHIFT RESULTANT HEX DIGITS RIGHT BY COUNT NUMBER
2160R 0510  SHIFTR  CLHI  SHIFT,18          ASM27410
0012          ASM27420
2164R 4280          BL  SHIFT4          ASM27430
2170R
2168R 07AA          XHR  B5,B5          ASM27440
216AR 0799          XHR  B4,B4          ASM27450
216CR 0788          XHR  B3,B3          ASM27460
216ER 0777          XHR  B2,B2          ASM27470
2170R 0766          XHR  B1,B1          ASM27480
2172R 0755          BR  BACK          ASM27490
2174R 0744  SHIFTR  SHI  SHIFT,4          ASM27500
0004
2178R 4210          BM  SHIFT1          ASM27510
2180R
* SHIFT B-REGS RIGHT BY MULTIPLES OF 4 HEX DIGITS
217CR 0800          LHR  B5,B4          ASM27520
217ER 080F          LHR  B4,B3          ASM27530
2180R 0807          LHR  B3,B2          ASM27540
2182R 0876          LHR  B2,B1          ASM27550
2184R 0706          XHR  B1,B1          ASM27560
2186R 4300          B  SHIFT4          ASM27570
2190R
218AR 0A30  SHIFTR  AHI  SHIFT,3          ONLY DECREMENT BY 1  ASM27580
0003
218ER 0214          BPCR  1,BACK          ASM27600
* SHIFT B-REGS RIGHT BY SINGLE HEX DIGITS          ASM27610

```



2190R	CCA0 0004	SRHL	B5,4		ASM27620
2194R	0809	LHR	R0,B4		ASM27630
2196R	C400 000F	NHI	R0,X'F'		ASM27640
219AR	CD00 000C	SLHL	R0,12		ASM27650
219ER	06A0	OHR	B5,R0		ASM27660
21A0R	CC90 0004	SRHL	B4,4		ASM27670
21A4R	0808	LHR	R0,B3		ASM27680
21A6R	C400 000F	NHI	R0,X'F'		ASM27690
21AAR	CD00 000C	SLHL	R0,12		ASM27700
21AER	0690	OHR	B4,R0		ASM27710
21B0R	CC80 0004	SRHL	B3,4		ASM27720
21B4R	0807	LHR	R0,B2		ASM27730
21B6R	C400 000F	NHI	R0,X'F'		ASM27740
21BAR	CD00 000C	SLHL	R0,12		ASM27750
21BER	0680	OHR	B3,R0		ASM27760
21C0R	CC70 0004	SRHL	B2,4		ASM27770
21C4R	0806	LHR	R0,B1		ASM27780
21C6R	C400 000F	NHI	R0,X'F'		ASM27790
21CAR	CD00 000C	SLHL	R0,12		ASM27800
21CER	0670	OHR	B2,R0		ASM27810
21D0R	CC60 0004	SRHL	B1,4		ASM27820
21D4R	4300	B	SHIFT4		ASM27830
2174R					
		*****			ASM27840
		* "FIXEXP" ADJUSTS THE RESULT BY ABSORBING THE			ASM27850
		* EXPONENT IN COMBINATION WITH THE SCALE FACTOR OF THE			ASM27860
		* DIGIT POSITIONS TO THE RIGHT OF THE DECIMAL POINT			ASM27870
21D8R	40F0	FIXEXP	STH	EXIT,LEAVE+2	ASM27880
	228AR				
21DCR	4040	STH	SRX,SAVSRX+2		ASM27890
	227ER				
21E0R	40B0	STH	CHAR,SAVCHR+2		ASM27900
	2282R				
21E4R	4870	LOOPX	LH	B2,SAVE	ASM27910
	15D8R				
21E8R	4330	BZ	ROUND	IF ZERO,EXPONENT ABSORBED	ASM27920
	2230R				
21ECR	4210	BM	FIXNEG		ASM27930
	2218R				
21F0R	CB70	SHI	B2,10	DECREASE BY 10	ASM27940
	000A				
21F4R	4310	BNM	FIXPOS	OK IF STILL POS	ASM27950
	2208R				

21F8R	CA70 0009	AHI	B2,9	ONLY DECREASE BY 1	ASM27960
21FCR	4070 1508R	STH	B2,SAVE		ASM27970
2200R	4100 2014R	BAL	0,MULTA	MULTIPLY BY 10***+1	ASM27980
2204R	4300 21E4R	B	LOOPX		ASM27990
2208R	C850 0014	FIXPOS LHI	A5,20	MULTIPLY BY 10***+10	ASM28000
220CR	4070 1508R	FIXEXC	STH	B2,SAVE	ASM28010
2210R	4100 2016R	BAL	0,MULTIP		ASM28020
2214R	4300 21E4R	B	LOOPX		ASM28030
2218R	C850 001E	FIXNEG LHI	A5,30	MULTIPLY BY 10**+10	ASM28040
221CR	CA70 002A	AHI	B2,10	INCREASE EXP BY +10	ASM28050
2220R	4320 220CR	BNP	FIXEXC		ASM28060
2224R	C870 0039	SHI	B2,9	ONLY INCREASE BY 1	ASM28070
2228R	C850 000A	LHI	A5,10	MULTIPLY BY 10**+1	ASM28080
222CR	4300 220CR	B	FIXEXC		ASM28090
2230R	C850 0001	ROUND LHI	B1,1	ROUNDING BIT	ASM28100
2234R	C800 2266R	LHI	R0,FXSIGN	PREPARE RETURN	ASM28110
2238R	4810 150CR	LH	A1,W1	FETCH HIGH FRAC & EXPO	ASM28120
223CR	9321	LBR	A2,A1	HIGH FRACTION W/O EXP	ASM28130
223ER	C410 7F00	NHI	A1,X'7F00'	ISOLATE EXPONENT	ASM28140
2242R	4870 1506R	LH	B2,FLTFLG	ROUND SINGLE OR DOUBLE?	ASM28150
2246R	4230 225AR	BNZ	DOUBLE		ASM28160
224AR	C830 0500	LHI	SHIFT,X'0500'	POSITIONS ROUNDING BIT	ASM28170
224ER	4870 15E0R	LH	B2,W3		ASM28180
2252R	4310 226CR	BNM	FXSIGN		ASM28190
2256R	4300 1F00R	B	ROUND1		ASM28200
225AR	C830 0000	DOUBLE LHI	SHIFT,X'0000'	POSITIONS ROUNDING BIT	ASM28210
225ER	4870 15E4R	LH	B2,W5	IF BIT ZERO SET,ROUND	ASM28220
2262R	4210 1FC0R	BM	ROUND1		ASM28230
2266R	4860	FXSIGN LH	B1,W1	SET IN SIGN BIT HELD IN	ASM28240

226AR	15DCR 4660	OH	B1,V1	BIT ZERO OF V1	ASM28250	
226ER	15BCR 4060	STH	B1,W1		ASM28260	
2272R	15DCR 0700	RESTOR	XHR	ZERO,ZERO	ASM28270	
2274R	C810 0001	LHI	ONE,1		ASM28280	
2278R	C820 0002	LHI	TWO,2		ASM28290	
227CR	C840 0000	SAVSRX	LHI	SRX,0	ASM28300	
2280R	C880 0000	SAVCHR	LHI	CHAR,0000	ASM28310	
2284R	C830 0942R	LHI	GET,GETCHR		ASM28320	
2288R	4300 0000	LEAVE	B	0000	ASM28330	
228CR	0000	EXP	DC	0	ASM28340	
228ER	41A0 0000	CONST	DC	X'41A0',0,0,0,0 10**1	ASM28350	
2298R	0000 0000 0000 4019 9999 9999 9999	DC	X'4019',X'9999',X'9999',X'9999',X'999A'	10**-1	ASM28360	
22A2R	4925 40BE 4000 0000	DC	X'4925',X'40BE',X'4000',0,0	10**+10	ASM28370	
22ACR	386D F37F 675E F6EA DF5B	DC	X'386D',X'F37F',X'675E',X'F6EA',X'DF5B'	10**-10	ASM28380	
22B6R	0855	TEST2	LHR	R,R	IF NOT DEFINED,JUST CHOP	ASM28389
22B8R	4230		BNZ	CHOP		ASM28390
22BCR	22E6R 0866		LHR	REG,REG	IF NOT ABSOLUTE,JUST CHOP	ASM28410
22BER	4230 22E6R		BNZ	CHOP		ASM28420
22C2R	4850 15C2R	LH	R,ASTFLG	LOCATION COUNTER * USED?	ASM28430	
22C6R	0A55	AHR	R,R	DOUBLE FOR KEY TO CONSTANTS	ASM28440	
22C8R	08CC	LHR	VALUE,VALUE	POS OR NEG EXPRESSION?	ASM28450	
22CAR	4210 22DAR	BM	TEST2B		ASM28460	
22CER	45C5 1612R	CLH	VALUE,DBCONS(R)	R=0 FOR NO *,R=4 IF * USED	ASM28470	
22D2R	4380 22E2R	BNL	SETTER	SET T ERROR,NOT 0-FF/0-7F	ASM28480	

22D6R	4300		B	CHOP		ASM28490
	22E6R					
22DAR	45C5	TEST2H	CLH	VALUE,DBCONS+2(R)	NOT FFFF-FF00/FFFF-FF80	ASM28500
	1614R					
22DFR	4380		BNL	CHOP		ASM28510
	22E6R					
22E2R	0210	SETTER	STB	ONE,ERRORT		ASM28520
	1566R					
22E6R	93CC	CHOP	LBR	VALUE,VALUE		ASM28530
22E8R	030E		BR	LINK		ASM28540
						ASM28544
						ASM28545
						ASM28546
						ASM28547
						ASM28548
						ASM28549
						ASM28550
						ASM28551
						ASM28552
						ASM28553
						ASM28554
						ASM28555
						ASM28556
						ASM28557
						ASM28560
22EAR	41D0	DB	BAL	RETURN,ENDDB	SEE IF ENDING DB * STATEMENT	
	23B4R					
22EER	4850		LH	R,LOC	IF LOCATION COUNTER	ASM28580
	168FR					
22F2R	0C50		SRHL	R,1	EVEN,LINE STARTS W/EVEADRS	ASM28590
	0001					
22F6R	4280		BC	ODDSTR	ODD,LINE STARTS W/ODD ADRS	ASM28600
	2338R					
22FAR	4000		STH	ZERO,DBFLAG	0= EVEN BYTE ADRS START	ASM28610
	23E2R					
22FER	41E0	EVEN	BAL	LINK,SCAN	EVALUATE A SINGLE OPERAND	ASM28620
	044AR					
2302R	41E0		BAL	LINK,TEST2	CHECK FOR ERROR,IF ERROR VAL=ZERO	ASM28630
	22B6R					
2306R	02C0		STB	VALUE,SAVBYT	SAVE EVEN BYTE FOR PUNCH BUFFER	ASM28640
	23E0R					
230AR	4850		LH	R,LOC	LOCATION IS EVEN	ASM28650
	168ER					
230ER	0A51		AHR	R,ONE		ASM28660
2310R	4050		STH	R,LOC	MAKE IT NEXT HIGHEST ODD	ASM28670
	168ER					
2314R	4850		LH	R,DBFLAG	WAS START EVEN?	ASM28680
	23E2R					
2318R	4330		BZ	QUESTO	YES, GO QUESTO	ASM28690
	2380R					
						ASM28700
231CR	0890		LHI	COUNT,4	STARTED ODD,ATTACH EVEN BYTE	ASM28710
	0004					
2320R	41E0		BAL	LINK,PACKPR+4		ASM28720
	0C4CR					
2324R	1794R		DC	PR+10		ASM28730
2326R	05B0		CLHI	CHAR,C','		ASM28740

232AR	002C 4230 0CHFR	BNE	ENDCHK	PRINT AND GET NEXT SOURCE LINE	ASM28750
232FR	01F3	BALR	EXIT,GET	BYPASS COMMA	ASM28760
2330R	41E0 0CCAR	BAL	LINK,PRINT	PRINT AND RETURN FOR NEXT	ASM28770
2334R	4300 233CR	B	ODD	OPERAND THIS SAME LINE.	ASM28780
2338R	4010 23E2R	ODDSTR	STH	ONE,DBFLAG	ASM28790
233CR	41E0 044AR	ODD	BAL	LINK,SCAN	EVALUATE A SINGLE OPERAND ASM28800
2340R	41E0 22B6R	BAL	LINK,TEST2	CHECK FOR ERROR, ZERO VALUE	ASM28810
2344R	46C0 23E0R	LAST1	OH	VALUE,SAVBYT	ATTACH TO LAST EVEN BYTE ASM28820
2348R	40C0 15DCR	STH	VALUE,W1		ASM28830
234CR	41F0 0AA0R	BAL	EXIT,PACKPB-10	PACK TWO BYTES IN PUNCH BUFFER	ASM28850
2350R	0008	DC	B	WITH LOADER ITEM B	ASM28860
2352R	48C0 15DCR	LH	VALUE,W1	GET SAME TWO BYTES AGAIN	ASM28870
2356R	4850 168ER	LH	R,LOC	LOCATION ODD	ASM28880
235AR	0A51	AHR	R,ONE		ASM28890
235CR	4050 168ER	STH	R,LOC	MAKE IT NEXT HIGHEST EVEN	ASM28900
2360R	4850 23E2R	LH	R,DBFLAG	WAS START EVEN?	ASM28910
2364R	4330 239CR	BZ	QUESTE	YES GO QUESTE	ASM28920
				* HERE, HAVE AN ODD ADRESSED BYTE PUNCHED BUT NOT YET	ASM28930
				* PRINTED	ASM28940
2368R	C890 0004	LHI	COUNT,4	PACK ODD BYTE IN RIGHT HALF	ASM28950
236CR	41E0 0C4CR	BAL	LINK,PACKPR+4	OF VALUE INTO PRINT RECORD	ASM28960
2370R	1792R	DC	PR+8 ?	AT PR+8.	ASM28970
2372R	C580 002C	CLHI	CHAR,C','		ASM28980
2376R	4236 0CBER	BNE	ENDCHK	PRINT & GET NEXT SOURCE LINE	ASM28990
237AR	01F3	BALR	EXIT,GET	BYPASS COMMA	ASM29000
237CR	4300 22FER	B	EVEN	DON'T PRINT UNTIL EVEN ATTACHED.	ASM29010
2380R	C580 002C	QUESTO	CLHI	CHAR,C','	ASM29020
2384R	4230 238ER	BNE	SINGLE		ASM29030
2388R	01F3	BALR	EXIT,GET	BYPASS COMMA	ASM29040
238AR	4300 233CR	B	ODD	GO TO ATTACH ODD BYTE TO EVEN	ASM29050
238ER	C890 0004	SINGLE	LHI	COUNT,4	STARTED EVEN AND THIS EVEN ASM29060
2392R	41E0	BAL	LINK,PACKPR+4	BYTE STANDS ALONE	ASM29070

2396R	0C4CR 1792R	DC	PR*8			ASM29080
2398R	4300 0C8ER	B	ENDCHK	PRINT AND GET NEXT SOURCE LINE		ASM29090
239CR	41E0 0C48R	QUESTE	BAL	LINK,PACKPR	PACK EVEN/ODD PAIR	ASM29100
		* HERE, PACK AN EVEN/ODD PAIR IN PRINT RECORD.				ASM29110
23A0R	1792R	DC	PR*8			ASM29120
23A2R	C5B0 002C	CLHI	CHAR,C*,'	IF NO COMMA, THIS IS END LINE		ASM29130
23A6R	4230 0C8ER	BNE	ENDCHK	PRINT & GET NEXT SOURCE LINE		ASM29140
23AAR	01F3	BALR	EXIT,GET	BYPASS COMMA		ASM29150
23ACR	41E0 0CCAR	BAL	LINK,PRINT	PRINT BUT GET NEXT OPERAND		ASM29160
23B0R	4300 22FER	B	EVEN	THIS SAME LINE.		ASM29170
23B4R	C5B0 002A	ENDDB	CLHI	CHAR,C*,'	1ST CHAR =*?	ASM29180
23B8R	0230	BNER	RETURN	CONTINUE PROCESSING DB'S.		ASM29185
23BAR	01F3	BALR	EXIT,GET	GET CHAR AFTER ASTERISK		ASM29190
23BCR	41E0 0C2CR	BAL	LINK,CHKEND	SEE IF SPACE, OR CARRTN.		ASM29195
23C0R	4300 230AR	B	ENDDB2	THIS IS ENDING DB STATEMENT		ASM29196
23C4R	4850 168ER	LH	R,LOC	FETCH CURRENT LOCATION		ASM29220
23C8R	CC50 0001	SRHL	R,1	IF LOCATION EVEN, ALL DATA		ASM29230
23CCR	4380 0CC6R	BFC	8,PRINTR	PROPER END ASSURED.		ASM29240
23D0R	4020 23E2R	STH	TWO,DBFLAG	IF ODD, SEND ZERO FILLER OF		ASM29250
23D4R	07CC	XHR	VALUE,VALUE	ONE BYTE WITH THIS LOCATION		ASM29260
23D6R	4300 2344R	B	LAST1	AND PUNCH LAST DATA BYTE.		ASM29270
23DAR	0B42	ENDDB2	SHR	SRX,TWO	BACKUP SOURCE INDEX	ASM29271
23DCR	01F3	BALR	EXIT,GET	GET ASTERISK AGAIN INTO CHAR		ASM29272
23DER	0300	BR	RETURN	CONTINUE PROCESSING DB'S		ASM29273
23E0R	0000	SAVBYT	DC	0	SAVES EVEN ADDRESS BYTE CONTENTS	ASM29280
23E2R	0000	DBFLAG	DC	0	0=EVEN START; 1=ODD START 2=LAST1	ASM29290
		* VALUE OF "DB" OPERAND MUST NOT BE FORWARD REFERENCE.				ASM29300
		* RELOCATABLE, OR OUTSIDE THE RANGE X'00' THRU X'FF'				ASM29310
23E4R		SYMTAB	END	PASS1	SYMBOL TABLE STARTS HERE.	ASM29320

NO ERRORS

A1	0001
A2	0002
A3	0003
A4	0004
A5	0005
ADD	205AR
ADD4	1258R
ADDADR	0FE8R
ADJUST	0FC6R
ADJUST	0FEAR
ADRBLK	0928R
ADRS	1434R
AEXP	117CR
AEXPEX	129CR
AFIELD	15CER
AGAIN	0D5CR
ALFNUM	09CAR
ALIGN	0C28R
AMULTI	208CR
APS	0009
APSEFLG	0006
ASSEM	1688R
ASTFLG	15C2R
B1	0006
B2	0007
B3	0008
B4	0009
B5	000A
BACK	0004
BADVAL	0442R
BEGIN	003AR
BEGIN1	003ER
REGINP	0058R
BKSPFL	18C2R
BLKADR	0BA6R
BLKOUT	015ER
BMULTI	20C0R
BOTHBF	040CR
BYPAS1	1230R
BYPASS	119CR
CARTN	000C
CHAIN	1302R
CHAR	000B
CHECKF	11E0R
CHECKM	1238R
CHECKP	1214R
CHECKX	1224R
CHKCR	0C34R
CHKDEF	061ER
CHKDIG	09D6R
CHKDOC	0242R
CHKEND	0C2CR
CHKENT	063ER
CHKERR	0D56R
CHKLCT	0D1ER

CHKOPT	0D98R
CHKPAR	1406R
CHKPAS	08F6R
CHKSPC	0F00P
CHKSRX	0F8CR
CHKSUM	0B80R
CHKTWO	0D0AR
CHOP	22E6R
CHRCON	14C2R
CLEAR	1184R
CLEAR5	0950P
CLRERR	0120R
CLRPB	00C2R
CMULTI	20F0R
CNSTNT	1394R
CONVERT	18D8R
COLI	0007
COMA	05E4R
COMAND	0292R
COMBO	15B0R
COMMA	0662R
COMNT	000D
COMPAR	0F52R
CONST	228ER
CONTRL	157ER
COUNT	0009
CTABLE	1592R
DATA	15B0R
DB	22EAR
DBCONS	1612R
DBFLAG	23E2R
DC	070ER
DCCHR	14F2R
DEC	1442R
DECD0C	0256R
DEFBIT	0200
DEFERU	068ER
DIGIT	09D8R
DIVIDE	0FE0R
DMULTI	207AR
DO	06CAR
DOC	168CR
DOLINE	011CR
DONTST	1314R
DOUBLE	225AR
DS	06E8R
DTABLE	1548R
DUMP	07DCR
EDIT	0A30R
EDIT1	0A6AR
EDITC	09E8R
EDITC1	09F0R
EDITCX	0A2CR
EDITDC	072ER
EDITX	0A94R
EJECT	0514R



EMULTB 2118R  
EMULTI 211CR  
END 075ER  
ENDATA 15C2R  
ENDCHK 08ER  
ENDCON 13C6R  
ENDD8 23B4R  
ENDD82 23DAR  
ENDMUL 20F8R  
ENDONE 08DCR  
ENDREL 08C4R  
ENDSCN 0464R  
ENDSYM 1134R  
ENDUMP 08A6R  
ENTBIT 0800  
ENTER 064ER  
ENTRY 05D8R  
EOJ 092AR  
EOL 15D4R  
EOLCHR 0EAER  
EOTE 0FB8R  
EQU 069AR  
EQU LAB 06C6R  
ERBLK1 18E6R  
ERBLK2 18EAR  
ERRCNT 1912R  
ERRDEC 147ER  
ERORE 1560R  
ERRORF 156CR  
ERRORM 156AR  
ERRORO 1568R  
ERRORR 1562R  
ERRORS 156ER  
ERRORT 1566R  
ERRORU 1564R  
EVEN 22FER  
EVENR1 04A2R  
EVENR2 04A6R  
EXIT 000F  
EXITFP 0F30R  
EXP 228CR  
EXPO 1EFER  
EXTBIT 0400  
EXTENT 0C00  
EXTERR 0646R  
EXTFLG 1582R  
EXTRN 05D0R  
EXTRNS 0878R  
FFEED 184ER  
FINDSP 016ER  
FINEND 039AR  
FIRST1 14F6R  
FIXEXC 220CR  
FIXEXP 21D8R  
FIXNEG 2218R  
FIXPOS 2208R

FLAGS	1570R
FLIP	0F12R
FLIPS	1692R
FLOAT	1FH6R
FLOAT1	1FDAR
FLOAT2	1EBAR
FLOAT3	1EF8R
FLOATD	1EB2R
FLOATE	1EAAR
FLTFLG	15D6R
FMULTI	212ER
FORWRD	042ER
FRF	15B4R
FSPACE	15D0R
FXSIGN	2266R
GET	0003
GETCHR	0942R
GETEND	0992R
GETHWS	111ER
GMULTI	212CR
GO	0068R
GOPACK	0AE2R
HEADER	1856R
HEX	148AR
HEXNUM	14ACR
HOLD	0007
IF	04C6R
IFEND	04B6R
INERR	0188R
INSERT	1004R
INSRT	114AR
INSRT2	10A2R
INSRT3	10F0R
INSRTL	110ER
INSRTS	1116R
INSRTX	1178R
INSTR	02DER
IOWAIT	00D6R
ITEM	119ER
ITEMS	1590R
ITEND	11B4R
ITMFRF	15C4R
LAB	0568R
LABDEF	1034R
LABEL	027ER
LAST1	2344R
LBONLY	01C4R
LCOUNT	168AR
LEAVE	2288R
LETTER	09BAR
LINK	000E
LIST	0DCAR
LM	0ECER
LMDATA	164ER
LOC	168ER
LOCNTR	137ER

LOOPAC	0C08R
LOOPB	0188R
LOOPCH	14C8R
LOOPCN	13A4R
LOOPD	144CR
LOOPDC	14EER
LOOPHX	1494R
LOOPIN	1122R
LOOPP	0B56R
LOOPRG	1166R
LOOPS	0976R
LOOPSM	008AR
LOOPSQ	01E6R
LOOPST	1152R
LOOPTR	04F8R
LOOPX	21E4R
LS	15F0R
LSTART	1682R
M1	0EECR
M2	0EFCR
M2END	0F0ER
MATCH	0FB4R
MATCH1	0F80R
MAXLIM	0ED4R
MAXLOC	1690R
MERROR	10D2R
MFLAG	158ER
MLTEXT	0870R
MLTLOC	0868R
MODE	1686R
MOVCHR	0E06R
MOVEON	0F74R
MOVEUP	0FC4R
MULTA	2014R
MULTAD	1F92R
MULTIP	2016R
MULTZO	1FE0R
NEGADR	0FF8R
NEWCHR	0DEAR
NEXT	1634R
NOERR	18FCR
NOFLG	0D74R
NOLERR	079AR
NOREF	131ER
NOREFX	089AR
NOSCRA	07A6R
NOSCRT	014ER
NOTAB	0E76R
NOTCON	12A0R
NOTDIG	136ER
NOTEQL	0F70R
NOTFND	1324R
NOTLTR	09A0R
NOTSYM	1354R
NOUSE	1344R

NPRPAS 0704R  
NTITLE 051CR  
NUMEND 1F4AR  
NXTSYM 000B  
ODD 233CR  
ODDSTR 2338R  
OK 1482R  
OKPAGE 0CB8R  
ONE 0001  
ONECHR 1530R  
ONEWRD 037ER  
ONEWRX 0390R  
OPASS 161CR  
OPAUSE 161ER  
OPEND 1632R  
OPERR 01CCR  
OPNFLT 1624R  
OPORG 1630R  
OPRINT 1620R  
OPSCRT 1626R  
OPSQNM 1628R  
OPT 0540R  
OPT2 0596R  
OPTAB 1918R  
OPTABE 1EA8R  
OPTEND 0536R  
OPTERR 0532R  
OPTION 161CR  
OPTPRT 05BAR  
OPTS 1638R  
OPUNCH 1622R  
ORG 066ER  
ORGOK 0682R  
OUT 2142R  
OVEADD 214ER  
P 000A  
PACHAF 0B52R  
PACK1 0B36R  
PACK2 0B42R  
PACK4 0B1ER  
PACKPH 0AAAAR  
PACKPR 0C48R  
PACKR 0A98R  
PACREF 13E0R  
PACSYM 0BEAR  
PAGE 189CR  
PAGING 0C7CR  
PAREN 15C0R  
PASBLK 18DCR  
PASCHK 092ER  
PASMSG 0046R  
PASNUM 18E5R  
PASS 161AR  
\* PASS1 0000R  
\* PASS22 0020R  
\* PASS23 0028R

\* PASS33 0034R  
PAUSE 0118R  
PB1 16A0R  
PB1BLK 170CR  
PB2 1714R  
PB2BLK 1780R  
PBSWCH 169CR  
PBWAIT 1804R  
PBX 1788R  
PC 158ER  
PFLAG 150AR  
POS 000E  
POSSE 1F16R  
POSSEQ 0EA2R  
PR 178AR  
PR2 17ECR  
PR2BLK 18CAR  
PREOF 18C6R  
PRINIT 07D0R  
PRINT 0CCAR  
PRINTF 0CC2R  
PRINTR 0CC6R  
PRNTCK 0CFAR  
PRNTER 0DACR  
PRNTR1 0DBER  
PROCES 0264R  
PRPASS 0D12R  
PRWAIT 1802R  
PRX 0005  
PSETUP 1662R  
PSTART 1684R  
PUNCH 0B6CR  
QUESTE 239CR  
QUESTO 2380R  
QUESTS 023AR  
R 0005  
R0 0000  
R1 0468R  
R1R2 0496R  
R2 048AR  
R2END 049ER  
READ 021ER  
READIN 0156R  
REG 0006  
RELBIT 0100  
RELCNT 1586R  
RELFLG 1588R  
REPEAT 0C50R  
RESERR 01D0R  
RESERV 06F2R  
RESTOR 2272R  
RETURN C000  
RFCF 15CCR  
ROUND 2230R  
ROUND1 1FC0R  
RSPACE 15D2R

SAVBYT 23E0R  
SAVCHR 2280R  
SAVE 15D8R  
SAVFRF 15CAR  
SAVLOC 15C8P  
SAVREG 15FAR  
SAVSRX 227CR  
SCAN 044AR  
SCANSP 0E20R  
SCBLK 18B4R  
SCRWD 18BER  
SCWAIT 18BCR  
SEARCH 0F34R  
SEND3 0F2AR  
SEQNUM 1694R  
SETERR 1108R  
SFTL 0DDER  
SETOPT 05B2R  
SETP 0BA8R  
SETPAG 0CB2R  
SETPC 0B7CR  
SETREG 2156R  
SETREL 1294R  
SETTER 22E2R  
SETVAL 1260R  
SHIFT 0003  
SHIFT1 218AR  
SHIFT4 2174R  
SHIFTR 2160R  
SHORTX 03E0R  
SIGN 141AR  
SIGN1 1428R  
SINGLE 238ER  
SIZE 162ER  
SIZES 18A0R  
SPACE 0E18R  
SPACER 0186R  
SPC 000A  
SPECOP 03AAR  
SQCHK 01DCR  
SQERR 0200R  
SR 179AR  
SRBLK 18A4R  
SREND 17E8R  
SRX 0004  
SS 15E6R  
STATUS 0000  
STDONE 00B6R  
STDUMP 07AER  
STE 162CR  
STO 162AR  
STORE 0A8CR  
STRCHR 0964R  
STRFLG 082AR  
STRFRF 1208R  
STX 0004

SUSPND	18D6R
SVCALL	00DER
SVCHR1	1654R
SWITCH	04E4R
SWTCH1	04DCR
SYMBOL	094AR
SYMTAB	23E4R
TAB	0E80R
TAB2	0E94R
TEMP	0008
TFST	043CR
TEST2	22B6R
TEST2B	22DAR
TITLE	04ECR
TITLED	0D3AR
TR	185CR
TRBLK	18ACR
TRYSPC	01A4R
TWO	0002
TWOWRD	032ER
TWOWRX	0336R
TYPECN	15AER
UNDADD	2146R
UNDEF	12CAR
UNDEFX	0600
UNPACK	0BC8R
UPDATE	0212R
USEDRT	1000
USEVAL	12C0R
V1	15BCR
V2	15BAR
VALUE	000C
W1	15DCR
W2	15DER
W3	15E0R
W4	15E2R
W5	15E4R
XADRS	1636R
XENTDF	0E00
XNBITS	15C6R
XPAREN	1268R
YESERR	1908R
Z1	000B
Z2	000C
Z3	000D
Z4	000E
Z5	000F
ZERO	0000
ZEROPB	00BAR
ZERVAL	06DAR

\* OS TEXT EDITOR (OS TIDE), PROGRAM NUMBER 03-027 IS AN  
 \* ON-LINE, INTERACTIVE TEXT EDITING PROGRAM THAT ASSISTS  
 \* THE USER WITH THE CREATION, EXAMINATION, AND MANIPULATION  
 \* OF TEXTUAL INFORMATION SUCH AS SOURCE PROGRAMS.  
 \*  
 \* THE USER DIRECTS OS TIDE THROUGH KEYBOARD COMMANDS  
 \* EDITING THE TEXTUAL INFORMATION WITHIN TIDE'S TEXT BUFFER  
 \* UNTIL A HARD COPY OF THE TEXT IS OBTAINED(LISTING/TAPE).  
 \*  
 \* PRIOR TO EXECUTION, FOUR LOGICAL UNITS MUST BE  
 \* ASSIGNED TO PHYSICAL DEVICE ADDRESSES.  
 \*  
 \* ASSIGN LOGICAL UNIT #01 TO A SOURCE INPUT DEVICE  
 \* OS TIDE READS FROM LU #01 FOR:  
 \* APPEND NUMBER OF LINES(A#)  
 \* INSERT NUMBER OF LINES (I#)  
 \* SKIP(S) AND SKIP NUMBER OF LINES(S#)  
 \* REPRODUCE NUMBER OF LINES (R#)  
 \*  
 \* ASSIGN LOGICAL UNIT #02 TO A BINARY OUTPUT DEVICE  
 \* OS TIDE WRITES SOURCE TAPES TO LU #02 FOR  
 \* OUTPUT (O) DIRECTIVE  
 \* REPRODUCE (R) DIRECTIVE  
 \* ASSIGN LOGICAL UNIT #03 TO A LIST OUTPUT DEVICE  
 \* OS TIDE WRITES A LISTING OF THE TEXT TO LU#03 FOR  
 \* PRINT (P) DIRECTIVE  
 \* ASSIGN LOGICAL UNIT NUMBER 05 TO A KEYBOARD/PRINTER  
 \* DEVICE FOR OS TIDE'S OPERATOR CONSOLE RESPONSES.  
 \*  
 \* OS TIDE LOCATION X'0008'R AND X'000A'R RELOCATABLE  
 \* CONTAIN POINTERS TO THE BOTTOM AND THE TOP OF  
 \* ITS TEXT BUFFER OF 2000 BYTES.  
 \* THE USER MAY CHANGE THESE POINTERS TO REPOSITION AND/OR  
 \* EXPAND THE SIZE OF THE TEXT BUFFER WITHIN TIDE.

00000030  
 00000040  
 00000050  
 00000060  
 00000070  
 00000080  
 00000090  
 00000100  
 00000110  
 00000120  
 00000130  
 00000140  
 00000150  
 00000160  
 00000170  
 00000180  
 00000190  
 00000200  
 00000210  
 00000220  
 00000230  
 00000240  
 00000250  
 00000260  
 00000270  
 00000280  
 00000290  
 00000300  
 00000310  
 00000320  
 00000330  
 00000340  
 00000350  
 00000360  
 00000370  
 00000380  
 00000400  
 00000410  
 00000420  
 00000430  
 00000440  
 00000450  
 00000460  
 00000470  
 00000480  
 00000490  
 00000500  
 00000510  
 00000520  
 00000530  
 00000540  
 00000550  
 00000560

0000R 4300    ORG    B    START            INITIALIZE TEXT EDITOR  
           0070R  
 0004R 4300    RESTRT B    COMODE        RESTART  
           00DAR  
 \*  
 \*  
 0008R 093ER   BONTXT DC    BONBUF        TEXT BUFFER  
 000AR 110CR   ENDTXT DC    ENDBUF        ADDRESS DELIMITERS  
 \*  
 0000            CHAR    EQU    0            REGISTER ASSIGNMENTS  
 0001            R1     EQU    1  
 0002            R2     EQU    2  
 0002            LENGTH EQU    2  
 0003            COUNT EQU    3  
 0004            FROM    EQU    4  
 0004            COLUMN EQU    4  
 0005            ONE     EQU    5  
 0006            THRU    EQU    6  
 0007            TO     EQU    7  
 0008            R8     EQU    8  
 0009            LINPTR EQU    9            POINTS TO CHAR IN A LINE



000A	R10	EQU	10		00000570
000B	R11	EQU	11		00000580
000C	R12	EQU	12		00000590
000D	R13	EQU	13		00000600
000E	R14	EQU	14		00000610
000F	R15	EQU	15		00000620
	*				00000630
000CR 093DR	OPNPTR	DC	BGNBUF-1	START OF OPEN LINE	00000640
000ER 093DR	ENDPTR	DC	BGNBUF-1	END OF LAST LINE	00000650
	*				00000660
0010R	LINE	DS	86	WORK AREA FOR TEXT	00000670
	*				00000680
0066R	COMAND	DS	2	COMMAND STORAGE AREA	00000690
0068R	ARG1	DS	2	STORAGE FOR FIRST ARG	00000700
006AR	ARG2	DS	2	DITTO FOR SECOND ARG	00000710
006CR	TABFLG	DS	2	TABULATE FLAG	00000720
006ER 0000	LCOUNT	DC	0	LINE COUNT	00000730
	*				00000740
	*				00000750
	*				00000760
0070R 4810	START	LH	R1,BGNTXT	TEST TEXT BUFFER LIMITS	00000770
0008R					
0074R 4820		LH	R2,ENDTXT	IF REDEFINED LIMITS OVER-	00000780
000AR					
0078R C510		CLHI	R1,ORG	WRITE TIDE, FORCE BUFFER	00000790
0000R					
007CR 4330		BE	FORCE	TO PROGRAM DEFINED LOCS.	00000800
0092R					
0080R 4280		BL	LETXT	BGNTXT LT ORG	00000810
00A6R					
0084R C510		CLHI	R1,END		00000820
093CR					
0088R 4280		BL	FORCE	BGNTXT <END	00000830
0092R					
008CR 0512		CLHR	R1,R2		00000840
008ER 4280		BL	START2	BGNTXT < ENDTXT	00000850
0088R					
	*				00000860
	*			FORCE TEXT BUFFER LIMITS	00000870
	*			TO THOSE DEFINED IN TIDE	00000880
	*				00000890
	*				00000900
0092R C810	FORCE	LHI	R1,END+2		
093ER					
0096R 4010		STH	R1,BGNTXT		00000910
0008R					
009AR C810		LHI	R1,END+1999		00000920
110BR					
009ER 4010		STH	R1,ENDTXT		00000930
000AR					
00A2R 4300		B	START2		00000940
0088R					
	*				00000950
	*			BGNTXT IS LT ORG	00000960
00A6R 0512	LETXT	CLHR	R1,R2		00000970
00A8R 4280		BL	**8		00000980
00B0R					

00ACR	4300 0092R	B	FORCE	FORCE LIMITS	00000990
00B0R	C520 0000R	CLHI	R2,ORG		00001000
00B4R	4380 0092R	BNL	FORCE		00001010
		*			00001020
		*			00001030
00B8R	4810 0008R	START2 LH	R1,BGNTXT	INITIALIZE TIDE	00001040
00BCR	C850 0001	LHI	ONE,1		00001050
00C0R	0815	SHR	R1,ONE		00001060
00C2R	4010 000CR	STH	R1,OPNPTR	AND HOUSEKEEPING	00001070
00C6R	4010 000ER	STH	R1,ENDPTR		00001080
00CAR	E110 080AR	SVC	1,CRLF	CARRIAGE RETURN	00001090
		*			00001100
00CER	E110 08E2R	SVC	1,TIDE		00001110
		*			00001120
		* RESET TABULATE FLAG			00001130
00D2R	C810 0732R	NOTAB LHI	R1,NTB	REMOVE TABULATE FLAG	00001140
00D6R	4010 006CR	STH	R1,TABFLG		00001150
		*			00001160
		*			00001170
		* ENTER COMMAND MODE			00001180
		*			00001190
00DAR	C800 005F	COMODE LHI	CHAR,X,5F		00001200
00DER	D200 0928R	STB	CHAR,MESS3	SET MODE FLAG	00001210
00E2R	E110 08EAR	SVC	1,MODE	SHOW MODE	00001220
00E6R	C850 0001	LHI	ONE,1	REESTABLISH ONE=1 ON RESTARTS	00001225
		*			00001230
00EAR	0733	XHR	COUNT,COUNT	ARGUMENT COUNT	00001240
00ECR	4030 0066R	STH	COUNT,COMAND	CLEAR COMMAND	00001250
00F0R	4030 0068R	STH	COUNT,ARG1	RESET ARGUMENTS	00001260
00F4R	4030 006AR	STH	COUNT,ARG2		00001270
00F8R	4030 006ER	STH	COUNT,LCOUNT	LINES/PAGE COUNT	00001280
		*			00001290
		*			00001300
00FCR	41D0 07FAR	BAL	R13,RDLINE	READ LINE FROM KEYBOARD	00001310
0100R	0829	LHR	LENGTH,LINPTR	SAVE LENGTH	00001320
0102R	0711	XHR	R1,R1	ZERO R1	00001330

0104R 0799	XHR	LINPTR,LINPTR	ZERO LINE POINTER	00001340
0106R 41F0	BAL	R15,SEARCH	SEARCH FOR COMMAND	00001350
0234R				
010AR 0264R	DC	COMTBL,COMADR		00001360
0280R				
010ER 4300	B	ERROR	ERROR RETURN	00001370
052ER				
0112R 40A0	STH	R10,COMAND	SAVE COMMAND ROUTINE ADR	00001380
0066R				
				00001390
				00001400
				00001410
				00001420
				00001430
				00001440
				00001450
				00001460
0116R 41F0	BAL	R15,SEARCH		00001470
0234R				
011AR 0288R	DC	ARGTBL,ARGADR		00001480
02CAR				
011ER 4300	B	CM1	COMMAND NOT IN ARG TABLE	00001490
012AR				
0122R C8E0	LHI	R14,PLUS+4	COMMAND IN ARG TABLE	00001500
01A2R				
0126R 4300	B	GT1		00001510
0230R				
				00001520
012AR D309	CM1	LB	CHAR,LINE+1(LINPTR)	00001530
0011R				
012ER C500	CLHI	CHAR,X'0D'	CR	00001540
000D				
0132R 4330	BE	PROCES		00001550
013ER				
0136R C500	CLHI	CHAR,X'20'	SPACE	00001560
0020				
013AR 4230	BNE	ERROR		00001570
052ER				
				00001580
013ER 0592	PROCES	CLHR	LINPTR,LENGTH	00001590
0140R 4280		BL	PLUS	00001600
019ER				
0144R 4820	LH	R2,COMAND		00001610
0066R				
0148R 0302	BR	R2	B ROUTINE	00001620
				00001630
				00001640
				00001650
				00001660
				00001670
				00001680
				00001690
				00001700
014AR 0800	CNVDTB	SHR	CHAR,CHAR	
014CR D3A9	DTB1	LB	R10,LINE(LINPTR)	
0010R				
0150R C4A0	NHI	R10,X'F'	FIRST NON-NUMERIC	
000F				
0154R 0800	LHR	CHAR,CHAR		
0156R 4330	BZ	DTB3	CHAR WILL CONTAIN	
017ER				

LINE POINTER  
NOT INCREMENTED  
IF COMMAND A NUMERIC  
IT IS IN ARG TABLE TOO

\* SUBROUTINE CONVERTS DECIMAL FROM LINE  
\* TO BINARY IN CHAR

\*  
\* CNVDTB SHR CHAR,CHAR DECIMAL TO BINARY  
\* DTB1 LB R10,LINE(LINPTR) CONVERSION EXIT ON

015AR	CDB0		SLHL	R11,2	BINARY NUMBER	00001710
	0002					
015ER	0AB0		AHR	R11,CHAR		00001720
0160R	CDB0		SLHL	R11,1		00001730
	0001					
0164R	0ABA		AHR	R11,R10		00001740
0166R	080B		LHR	CHAR,R11		00001750
0168R	D3A9	DTB2	LB	R10,LINE+1(LINPTR)		00001760
	0011R					
016CR	C5A0		CLHI	R10,X'30'		00001770
	0030					
0170R	028F		BTCR	8,R15		00001780
0172R	C5A0		CLHI	R10,X'3A'		00001790
	003A					
0176R	038F		BFCR	8,R15		00001800
0178R	0A95		AHR	LINPTR,ONE		00001810
017AR	4300		B	DTB1		00001820
	014CR					
		*				00001830
017ER	92A0	DTB3	STBR	R10,CHAR		00001840
0180R	92AB		STBR	R10,R11		00001850
0182R	4300		B	DTB2		00001860
	0168R					
		*				00001870
		*				00001880
0186R	4880	CURLIN	LH	R8,OPNPTR	CURRENT LINE ARG	00001890
	000CR					
018AR	4300		B	LSTLIN+4		00001900
	0192R					
		*				00001910
		*				00001920
018ER	4880	LSTLIN	LH	R8,ENDPTR	LAST LINE ARG	00001930
	000ER					
0192R	08DE		LHR	R13,R14	CURLIN CONTINUES HERE	00001940
0194R	41E0		BAL	R14,CRCNT	FIND LINE NUMBER	00001950
	01F6R					
0198R	08ED		LHR	R14,R13		00001960
019AR	080A		LHR	CHAR,R10	LINE NUM IN R10 TO CHAR	00001970
019CR	030F		BR	R15	RETURN	00001980
		*				00001990
		*				00002000
019ER	41E0	PLUS	BAL	R14,GETARG	POSITIVE MODIFIER	00002010
	0222R					
01A2R	4A01		AH	CHAR,ARG1(R1)		00002020
	0068R					
01A6R	4001		STH	CHAR,ARG1(R1)		00002030
	0068R					
01AAR	4300		B	PLUS		00002040
	019ER					
		*				00002050
		*				00002060
01AER	41E0	MINUS	BAL	R14,GETARG	NEGATIVE MODIFIER	00002070
	0222R					
01B2R	48A1		LH	R10,ARG1(R1)		00002080
	0068R					
01B6R	0BA0		SHR	R10,CHAR		00002090

01B8R 40A1	STH	R10,ARG1(R1)		00002100
0068R				
01BCR 4300	B	MINUS		00002110
01AER				
				00002120
				00002130
				00002140
				00002150
01C0R 0595	DLMIT1	CLHR LINPTR,ONE		
01C2R 4280	BL	**8		
01C6R 4230	BNE	PROCES		00002160
013ER				
01CAR D309	LB	CHAR,LINE(LINPTR)		00002170
0010R				
01CER C500	CLHI	CHAR,X'20'	SPACE	00002180
0020				
01D2R 4230	BNE	ERROR	NO ERROR	00002190
052ER				
01D6R 0835	LHR	COUNT,ONE	ARGUMENT COUNT IS ONE	00002200
01D8R 4300	B	PROCES		00002210
013ER				
				00002220
				00002230
01DCR 0535	DLMIT2	CLHR COUNT,ONE	COMMA DELIMITER. TWO ARGS	00002240
01DER 4280	BL	**8		00002250
01E6R 4230	BNE	ERROR	ERROR	00002260
052ER				
01E6R 0A35	AHR	COUNT,ONE	ARGUMENT COUNT IS TWO	00002270
01E8R C810	LHI	R1,2		00002280
0002				
01ECR 4300	B	PROCES		00002290
013ER				
				00002300
				00002310
01FOR 0A95	DLMIT3	AHR LINPTR,ONE	INCREMENT CHAR POINTER	00002320
01F2R 4300	B	PROCES	NO ARGS COUNT ZERO	00002330
013ER				
				00002340
				00002350
				00002360
				00002370
				00002380
				00002390
01F6R 4580	CRCNT	CLM RB,BGNTXT	RB := OPNPTR/ENDPTR	
0008R				
01FAR 4280	BL	ERROR	ERROR	00002400
052ER				
01FER 07AA	XHR	R10,R10	CLEAR CR COUNT	00002410
0200R 48B0	LH	R11,BGNTXT		00002420
0008R				
0204R 08B5	SHR	R11,ONE		00002430
0206R D30B	CR1	LB CHAR,0(R11)		00002440
0000				
020AR C500	CLHI	CHAR,X'0D'		00002450
0000				
020ER 4230	BNE	CR2	NO CR	00002460
0214R				

0212R	0AA5	AHR	R10,ONE	CR COUNT	LINE NUMBER	00002470
0214R	0AB5	CR2	AHR	R11,ONE		00002480
0216R	05B8	CLHR	R11,R8	R8-ENDPTR	OR OPNPTR	00002490
0218R	4280	BL	CR1			00002500
	0206R					
021CR	4330	BE	CR2-2			00002510
	0212R					
0220R	030E	BR	R14	RETURN	R10 LINE NUMBER	00002520
						00002530
						00002540
0222R	0A95	GETARG	AHR	LINPTR,ONE	PROCESS ARGUMENT	00002550
0224R	41F0	BAL	R15,SEARCH	ELEMENTS		00002560
	0234R					
0228R	02B8R	DC	ARGTBL,ARGADR			00002570
	02CAR					
022CR	4300	B	ERROR	ERROR	RETURN	00002580
	052ER					
						00002590
0230R	01FA	GT1	BALR	R15,R10	B ROUTINE ADDR IN R10	00002600
0232R	030E	BR	R14			00002610
						00002620
						00002630
						00002640
						00002650
						00002660
						00002670
0234R	488F	SEARCH	LH	R8,0(R15)	TABLE START ADDR	00002680
	0000					
0238R	48AF	LH	R10,2(R15)	ROUTINE	START ADDR	00002690
	0002					
023CR	D3B9	LB	R11,LINE(LINPTR)			00002700
	0010R					00002710
0240R	D3C8	SR1	LB	R12,0(R8)		00002720
	0000					00002730
0244R	05BC	CLHR	R11,R12			00002740
0246R	4330	BE	SR2	CHAR IS	IN TABLE	00002750
	025CR					
						00002760
024AR	C5C0	CLHI	R12,X'7F'	END	OF TABLE	00002770
	007F					00002780
024ER	433F	BE	4(R15)			00002790
	0004					
0252R	0A85	AHR	R8,ONE	PICK	NEXT BYTE	00002800
0254R	CAA0	AHI	R10,2	PICK	UP NEXT HALF WORD	00002810
	0002					
0258R	4300	B	SR1	CONTINUE	TABLE SEARCH	00002820
	0240R					
						00002830
025CR	48AA	SR2	LH	R10,0(R10)	ROUTINE ADDR IN R10	00002840
	0000					00002850
0260R	430F	B	8(R15)	RETURN		00002860
	0008					
						00002870
						00002880
						00002890
0264R	2A2E	COMTBL	DC	C'+.0123456789'		00002900

	3031			
	3233			
	3435			
	3637			
	3839			
0270R	4143	DC	C'ACDEIKLNOPRS'	00002870
	4445			
	494B			
	4C4E			
	4F50			
	5253			
027CR	5421	DC	X'5421',X'5E7F' END OF TABLE	00002880
	5E7F			
	*			
	*		COMMAND ROUTINE ADDRESSES	00002890
	*			00002900
0280R	02EER	COMADR DC	LNUMBR,LNUMBR,LNUMBR,LNUMBR	00002910
	02EER			00002920
	02EER			
	02EER			
0288R	02EER	DC	LNUMBR,LNUMBR,LNUMBR,LNUMBR	00002930
	02EER			
	02EER			
	02EER			
0290R	02EER	DC	LNUMBR,LNUMBR,LNUMBR,LNUMBR	00002940
	02EER			
	02EER			
	02EER			
0298R	0334R	DC	APPEND,CHANGE,DELETE,ENDJOB	00002950
	03B8R			
	0404R			
	08CCR			
02A0R	03BER	DC	INSERT,KILL,LIST,NOTAB	00002960
	0454R			
	0458R			
	00D2R			
02A8R	0474R	DC	OUTPUT,PRINT,REPROD,SKIP	00002970
	0468R			
	04C4R			
	04D0R			
02B0R	04FAR	DC	TAB,PAUSE,BYTCNT,ERROR	00002980
	08D0R			
	0506R			
	052ER			
	*			00002990
	*		TABLE OF ARGUMENTS	00003000
	*			00003010
02B8R	3031	ARGTBL DC	C'0123456789'	00003020
	3233			
	3435			
	3637			
	3839			
02C2R	2A2E	DC	C'*.+- ,',X'0D7F'	00003030
	2B2D			
	202C			
	0D7F			

	*				00003040
	*	ARGUMENT ROUTINE ADDRESSES			00003050
	*				00003060
02CAR	014AR	ARGADR	DC	CNVDTB,CNVDTB,CNVDTB,CNVDTB	00003070
	014AR				
	014AR				
	014AR				
02D2R	014AR		DC	CNVDTB,CNVDTB,CNVDTB,CNVDTB	00003080
	014AR				
	014AR				
	014AR				
02DAR	014AR		DC	CNVDTB,CNVDTB,CURLIN,LSTLIN	00003090
	014AR				
	0186R				
	018ER				
02E2R	019ER		DC	PLUS,MINUS,DLIMIT1,DLIMIT2	00003100
	01AER				
	01COR				
	01DCR				
02EAR	01F0R		DC	DLIMIT3,ERROR	00003110
	052ER				
	*				00003120
	*				00003130
	*				00003140
02EER	48A0	LNUMBR	LH	R10,ARG1	OPEN ARGUMENT LINE
	0068R				00003150
02F2R	4320		BNP	ERROR	ERROR
	052ER				00003160
02F6R	41F0		BAL	R15,FINDLN	FIND THE LINE
	0536R				00003170
02FAR	4010		STH	R1,OPNPTR	NEW OPEN LINE
	000CR				00003180
	*				00003190
02FER	4880	LSTOPN	LH	R8,OPNPTR	O L POINTER IN R8
	000CR				00003200
0302R	4580		CLH	R8,SGNTXT	
	0008R				00003210
0306R	4280		BL	COMODE	RETURN TO COMMAND MODE
	00DAR				00003220
030AR	41E0		BAL	R14,CRCNT	GET LINE NUMBER
	01F6R				00003230
030ER	41F0		BAL	R15,CNVBT0	CONVERT TO DECIMAL
	056AR				00003240
0312R	41E0		BAL	R14,GETCR	GET END OF LINE
	055AR				00003250
0316R	C800		LHI	CHAR,X'20'	ENTER SPACE CHAR AFTER
	0020				00003260
031AR	D200		STB	CHAR,LINE+4	DECIMAL LINE NUMBER
	0014R				00003270
031ER	4840		LH	FROM,OPNPTR	START OF LINE
	000CR				00003280
0322R	0868		LHR	THRU,R8	END LINE ADDRESS
	0324R		LHI	TO,LINE+5	FIRST OPEN ADDR IN LINE
	0015R				00003300
0328R	41F0		BAL	R15,MOVXTI	MOVE OPEN LINE TO
	05C4R				00003310



```

*
*
032CR E110 LISTLN SVC 1,LINER 00003320
08FAR 00003330
0330R 4300 B COMODE RETURN TO COMMAND MODE 00003340
00DAR 00003350
*
*
* APPEND LINES 00003360
* 00003370
* 00003380
* 00003390
0334R 07FF APPEND XHR R15,R15 00003400
0336R 40F0 AI STH R15,AIFLG+2 00003410
0348R 0348R
033AR C800 LHI CHAR,C+0 00003420
002A
033ER D200 STB CHAR,MESS3 SET MODE FLAG 00003430
0928R
0342R 41F0 BAL R15,ONEARG 00003440
05F4R
*
* 00003450
0346R C8F0 AIFLG LHI R15,0 00003460
0000
034AR 4330 BZ A1-2 APPEND 00003470
035AR 035AR
*
* INSERT TEXT LINES INTO 00003480
* 00003490
034ER 4820 LH R2,OPNPTR BUFFER EITHER
000CR
0352R 4520 CLH R2,BGNTXT FROM THE KEYBOARD OR 00003500
000AR
0356R 4280 BL ERROR FROM INPUT DEVICE 00003510
052ER
*
* 00003520
035AR 07CC XHR R12,R12 INITIALIZE ITERATION COUNT 00003530
035CR 0535 A1 CLHR COUNT,ONE COUNT IS 1 FOR INPUT 00003540
* DEVICE SPECIFIED 00003550
035ER 4230 BNE A2 COUNT NOT 1, INPUT TTY 00003560
0372R 0372R
*
* COUNT 0,2 FOR 00003570
* TTY INPUT 00003580
0362R 45C0 CLH R12,ARG1 ITERATIONS COMPLETE 00003590
0068R
0366R 4380 BNL LSTOPN YES, LIST OPEN LINE 00003600
02FER
* APPEND OR INSERT FROM SOURCE INPUT DEVICE 00003610
* 00003620
036AR 41D0 BAL R13,NOKBRD NON-KEYBOARD INPUT
07DAR
036ER 4300 B A3 00003630
037AR
*
* 00003640
* 00003650
0372R E110 A2 SVC 1,MODE 00003660
08EAR
*
* EDIT MODE PRINT ASTERISK 00003670
0376R 41D0 BAL R13,RDLINE READ LINE FROM TTY 00003680
07FAR
*
* 00003690

```

037AR	4820	A3	LH	R2,ENDPTR	ENDPTR + LINPTR	00003700
	000ER					
037ER	0A29		AHR	R2,LINPTR	TOTAL BUFFER USED	00003710
0380R	48F0		LH	R15,AIFLG+2		00003720
	0348R					
0384R	4230		BNZ	S3		00003730
	03C4R					
0388R	4520		CLH	R2,ENDTXT	BUFFER OVERFLOW WITH LINE	00003740
	000AR					
038CR	4280		BL	**8		00003750
	0394R					
0390R	4230		BNE	OVRFLW	OVERFLOW OF TEXT BUFFER	00003760
	0672R					
		*			NO OVERFLOW	00003770
0394R	C840		LHI	FROM,LINE	SET TEXT POINTERS TO	00003780
	0010R					
0398R	C869		LHI	THRU,LINE(LINPTR)	APPEND LINE TO END	00003790
	0010R					
039CR	4870		LH	TO,ENDPTR	OF LAST LINE	00003800
	000ER					
03A0R	0817		LHR	R1,TO	IN TEXT BUFFER	00003810
03A2R	0A75		AHR	TO,ONE	APPENDED LINE IS O L	00003820
03A4R	4070		STH	TO,OPNPTR	SET POINTER	00003830
	000CR					
03A8R	41F0		BAL	R15,MOVTXT	APPEND LINE	00003840
	05C4R					
03ACR	0A19		AHR	R1,LINPTR	TO TEXT	00003850
03AER	4010		STH	R1,ENDPTR	POINTS TO END OF LAST LINE	00003860
	000ER					
03B2R	0AC5		AHR	R12,ONE	INCREMENT ITERATION COUNT	00003870
03B4R	4300		B	A1	CONTINUE APPENDING	00003880
	035CR					
		*				00003890
		*				00003900
		*				00003910
		*	CHANGE	CHANGE LINES		00003920
		*				00003930
03B8R	41D0		CHANGE	BAL	R13,DELETE+4	DELETE TEXT
	040RR					
		*				00003950
03BCR	0733		XHR	COUNT,COUNT	FROM TEXT BUFFER	00003960
					FORCE COUNT =0 AFTER DELETE	00003970
		*				00003980
		*				00003990
		*	INSERT	INSERT LINES		00004000
		*				00004010
03BER	0AF5		INSERT	LHR	R15,ONE	00004020
03COR	4300		B	AI		
	0336R					
		*				00004030
		*				00004040
03C4R	4520		S3	CLH	R2,ENDTXT	TEST FOR OVERFLOW
	000AR					00004050
03C8R	4380		BNL	OVRFLW	OVERFLOW OF TEXT BUFFER	00004060
	0672R					
		*			SET POINTERS	00004070
		*			IN TEXT BUFFER	00004080

03CCR 4840	LH	FROM,OPNPTR	TO MOVE TEXT FROM	00004090
000CR				
03D0R 4860	LH	THRU,ENDPTR	OPEN LINE THRU END	00004100
000FR				
03D4R 4870	LH	TO,OPNPTR	OF LAST LINE	00004110
000CR				
	*		ALLOW ROOM TO INSERT	00004120
03D8R 0A79	AHR	TO,LINPTR	NEW LINE BEFORE O L	00004130
	*		R1 POINTS TO O L	00004140
03DAR 0817	LHR	R1,TO	POSITION CHANGED	00004150
03DCR 41F0	BAL	R15,MOVTXT		00004160
05C4R				
03E0R C840	LHI	FROM,LINE	INSERT LINE	00004170
0010R				
03E4R C869	LHI	THRU,LINE-1(LINPTR)	IN TEXT BUFFER	00004180
000FR				
03E8R 4870	LH	TO,OPNPTR		00004190
000CR				
	*			00004200
03ECR 41F0	BAL	R15,MOVTXT		00004210
05C4R				
03F0R 4010	STH	R1,OPNPTR	OPEN LINE POINTER	00004220
000CR				
03F4R 4820	LH	R2,ENDPTR		00004230
000ER				
03F8R 0A29	AHR	R2,LINPTR		00004240
03FAR 4020	STH	R2,ENDPTR	SET NEW END POINTER	00004250
000ER				
03FER 0ACS	AHR	R12,ONE		00004260
0400R 4300	B	A1	CONTINUE CHANGE,INSERT	00004270
035CR				
	*			00004280
	*			00004290
	*	DELETE LINE		00004300
	*			00004310
0404R C8D0	DELETE LHI	R13,LSTOPN	DELETE	00004320
02FER				
0408R 41F0	DELCHG BAL	R15,TWOARG	CHANGE COMMAND ENTERS HERE	00004370
0606R				
040CR 0848	LHR	FROM,R8	R8 = END OF ARG2 OR O L	00004380
040ER 0A45	AHR	FROM,ONE	END OF ARG2 OR O L +1	00004390
0410R 4540	CLH	FROM,ENDPTR		00004400
000ER				
	*		IF END OF LAST LINE	00004410
	*		DELETED IS END OF	00004420
	*		LAST LINE IN BUFFER	00004430
0414R 4380	BNL	OPNPRV	OPEN PREVIOUS LINE	00004440
043AR				
0418R 48A0	LH	R10,ARG1	LOAD R10 W ARG1	00004450
0068R				
041CR 41F0	BAL	R15,FINDLN	FIND LINE	00004460
0536R				
	*		NOW RETURN FROM FINDLN	00004470
	*		R1 = ARG1 START	00004480
	*		R8 = ARG1 END	00004490
0420R 0871	LHR	TO,R1		00004500

0422R	4860		LH	THRU,ENDPTR		00004510
	000ER					
		*			O L PTR NOW START	00004520
0426R	4070		STH	TO,OPNPTR	OF FIRST DELETED LINE	00004530
	000CR					
042AR	08E6		LHR	R14,THRU	ENDPTR	00004540
042CR	0AE7		AHR	R14,TO	+ START ARG1 LINE	00004550
042ER	0BE4		SHR	R14,FROM	- END ARG2 LINE +1	00004560
0430R	40E0		STH	R14,ENDPTR	SET NEW ENDPTR	00004570
	000ER					
0434R	41F0		BAL	R15,MOVTXT	MOVE TEXT	00004580
	05C4R					
0438R	030D		BR	R13	LIST OPEN LINE	00004590
		*			OR RETURN TO INSERT	00004600
		*			FOR CHANGE COMMAND	00004610
		*				00004620
		*				00004630
		*				00004640
043AR	48A0	OPNPRV	LH	R10,ARG1	OPEN AND LIST PREVIOUS LINE	00004650
	006RR					
043ER	055A		CLHR	ONE,R10	ARG LT OR =1	00004650
0440R	4380		BNL	KILL	B START	00004660
	0454R					
0444R	0BA5		SHR	R10,ONE		00004670
0446R	41F0		BAL	R15,FINDLN	FIND PREV LINE	00004680
	0536R					
044AR	4010		STH	R1,OPNPTR	R1 START OF PREV LINE	00004690
	000CR					
044ER	4080		STH	R8,ENDPTR	R8 END OF PREV LINE	00004700
	000ER					
0452R	030D		BR	R13	LIST OPEN LINE	00004710
		*				00004720
		*				00004730
		*			K COMMAND KILL TEXT BUFFER	00004740
		*				00004750
0454R	4300	KILL	B	START	INITIALIZE TEXT EDITOR	00004760
	0070R					
		*				00004770
		*				00004780
		*			LIST OPEN LINE	00004790
		*				00004800
0458R	C810	LIST	LHI	R1,X'2805'	KEYBOARD OUTPUT	00004810
	2805					
045CR	4010		STH	R1,DEVOUT	LINES WITH	00004820
	090AR					
0460R	C8D0		LHI	R13,5	SET LINE POINTER	00004830
	0005					
		*			TEXT IS 5 CHAR	00004840
		*			FROM START OF LINE	00004850
0464R	4300		B	PR1		00004860
	047AR					
		*				00004870
		*				00004880
		*			PRINT LINE	00004890
		*				00004900
0468R	C810	PRINT	LHI	R1,X'2803'	LIST DEVICE	00004910
	2803					

046CR	4010	STH	R1,DEVOUT	OUTPUT DEVICE	00004920
	090AR				
0470R	4300	B	OPUT		00004930
	047AR				
		*			00004940
		*			00004950
		* PUNCH TEXT			00004960
0474R	41F0	OUTPUT BAL	R15,SETBIN	INPUT DEVICE SPECIFIED	00004980
	067AR				
		*			00004990
0478R	07DD	OPUT XHR	R13,R13	CLEAR LINE POINTER	00005000
		*			00005010
047AR	41F0	PR1 BAL	R15,TWOARG	VALIDATE ARGS	00005020
	0606R				
047ER	48B0	LH	R11,ARG1		00005030
	0068R				
0482R	08AB	PR2 LHR	R10,R11	LOAD NEXT LINE ARG	00005040
0484R	41F0	BAL	R15,FINDLN	GET LINE	00005050
	0536R				
0488R	0841	LHR	FROM,R1	R1 = START OF LINE	00005060
048AR	0868	LHR	THRU,R8	R8 = END OF LINE	00005070
048CR	C87D	LHI	TO,LINE(R13)	R13 = 0 FOR P = 5 FOR L	00005080
	0010R				
0490R	0881	LHR	R8,R1	LOAD OPEN LINE	00005090
0492R	081D	LHR	R1,R13	SAVE LINE POINTER	00005100
0494R	082B	LHR	R2,R11	SAVE ARG1	00005110
0496R	41E0	BAL	R14,CRCNT	DETERMINE	00005120
	01F6R				
049AR	41F0	BAL	R15,CNVBTD	LINE NUMBER	00005130
	056AR				
049ER	C800	LHI	CHAR,X:20'	SPACE	00005140
	0020				
04A2R	D200	STB	CHAR,LINE+4	AFTER LINE NUMBER	00005150
	0014R				
04A6R	08B2	LHR	R11,R2	RESTORE	00005160
04A8R	08D1	LHR	R13,R1		00005170
04AAR	41F0	BAL	R15,MOVTXT		00005180
	05C4R				
		*			00005190
04AER	41F0	PR3 BAL	R15,PUTOUT		00005200
	068CR				
04B2R	0AB5	AHR	R11,ONE		00005210
04B4R	45B0	CLH	R11,ARG2	TEXT OUTPUT COMPLETE	00005220
	006AR				
04B8R	4280	BL	PR2	NO	00005230
	0482R				
04BCR	4330	BE	PR2	NO	00005240
	0482R				
04C0R	4300	B	COMODE		00005250
	00DAR				
		*			00005260
		*			00005270
		*	REPRODUCE INPUT		00005280
		*			00005290
04C4R	41F0	REPROD BAL	R15,SETBIN	INPUT DEVICE SPECIFIED	00005300
	067AR				

04C8R	C8B0	*	LHI	R11,PUTOUT	SET BRANCH	00005310
	068CR					00005320
04CCR	4300		B	SKIP+4		00005330
	04D4R					
		*				00005340
		*				00005350
		*		SKIP INPUT LINES		00005360
		*				00005370
04D0R	C8B0	SKIP	LHI	R11,SK2	SET BRANCH	00005380
	04E8R					
		*				00005390
04D4R	41F0		BAL	R15,ONEARG	R12 IS ARG1	00005400
	05F4R					
04D8R	07AA		XHR	R10,R10		00005410
04DAR	0535		CLHR	COUNT,ONE		00005420
04DCR	4330		BE	SK1		00005430
	04E2R					
04E0R	08CA		LHR	R12,R10	SET ARG TO ZERO	00005440
04E2R	41D0	SK1	BAL	R13,NOKBRD	GET INPUT LINE	00005450
	07DAR					
		*			B TO PUTOUT FOR R	00005460
		*			AND SK2 FOR S	00005470
04E6R	01FB		BALR	R15,R11		00005480
		*				00005490
		*				00005500
04E8R	08CC	SK2	LHR	R12,R12		00005510
04EAR	4330		BZ	SK1	NO ARG SKIP UNTIL BK	00005520
	04E2R					
04EER	0AA5		AHR	R10,ONE		00005530
04FOR	05AC		CLHR	R10,R12		00005540
04F2R	4280		BL	SK1		00005550
	04E2R					
		*			LIST LAST LINE SKIPPED OR	00005560
04F6R	4300		B	LISTLN	REPRODUCED	00005570
	032CR					
		*				00005580
		*				00005590
		*		SET TABULATE FOR PRINT AND LIST		00005600
		*				00005610
04FAR	C810	TAB	LHI	R1,TB	SET TABULATE FLAG	00005620
	0750R					
04FER	4010		STH	R1,TABFLG	FOR PRINT	00005630
	006CR					
0502R	4300		B	COMODE	RETURN TO COMMAND MODE	00005640
	00DAR					
		*				00005650
		*				00005660
		*		SHOW NUMBER OF BYTES USED		00005670
		*				00005680
0506R	48A0	BYTCNT	LH	R10,ENDPTR	SHOW NUMBER	00005690
	000ER					
050AR	45A0		CLH	R10,BGNTXT	CONSUMED	00005700
	0008R					
050ER	4280		BL	ERROR	IN TEXT BUFFER	00005710
	052ER					

0512R	4880	LH	R8,BGNTXT	START OF TEXT BUFFER	00005720
	0008R				
0516R	0BA8	SHR	R10,R8	DIFFERENCE	00005730
0518R	0AA5	AHR	R10,ONE	PLUS ONE GIVES BYTE COUNT	00005740
051AR	41F0	BAL	R15,CNVBTD	CONVERT BIN TO DEC	00005750
	056AR				
051ER	C8F0	LHI	R15,X'0D'		00005760
	000D				
0522R	D2F0	STB	R15,LINE+4	PRINT NUMBER	00005770
	0014R				
0526R	E110	SVC	1,LINEB	ON TTY	00005780
	08FAR				
052AR	4300	B	COMODE		00005790
	00DAR				
					00005800
					00005810
				ERROR ROUTINE PRINT QUESTION MARK	00005820
					00005830
052ER	E110	ERROR SVC	1,ERR		00005840
	08F2R				
0532R	4300	B	COMODE	RETURN TO COMMAND MODE	00005850
	00DAR				
					00005860
					00005870
				* FIND START AND END OF LINE WHOSE NUMBER IS IN R10	00005880
					00005890
0536R	4880	FINDLN LH	R8,BGNTXT	FIND BEGINNING	00005900
	0008R				
053AR	4580	CLH	R8,ENDPTR		00005910
	000ER				
053ER	4380	BNL	ERROR		00005920
	052ER				
0542R	0818	FN1 LHR	R1,R8	AND ENDING	00005930
0544R	41E0	BAL	R14,GETCR	OF ARGUMENT LINE	00005940
	055AR				
0548R	0BA5	SHR	R10,ONE	R1= START R8= END	00005950
054AR	033F	BFCR	3,R15	B IF DIFF =0	00005960
054CR	4580	CLH	R8,ENDPTR		00005970
	000ER				
0550R	4380	BNL	ERROR	ERROR	00005980
	052ER				
0554R	0A85	AHR	R8,ONE		00005990
0556R	4300	B	FN1		00006000
	0542R				
					00006010
					00006020
				FIND END OF A LINE	00006030
					00006040
055AR	D308	GETCR LB	CHAR,0(R8)	R8 POINTS TO LINE	00006050
	0000				
055ER	C500	CLHI	CHAR,X'0D'		00006060
	0000				
0562R	033F	BFCR	3,R14	R8 END OF LINE	00006070
0564R	0A85	AHR	R8,ONE	NO CR	00006080
0566R	4300	B	GETCR	CONTINUE SEARCH	00006090
	055AR				

	*				00006100
	*				00006110
	*		BINARY TO DECIMAL CONVERSION		00006120
	*		MAX 4 DEC DIGITS		00006130
	*		CONVERT NUMBER IN R10 TO DECIMAL		00006140
	*				00006150
056AR	C5A0	CNVBTD	CLHI R10,10000	R10 CONTAINS BINARY	00006160
	2710				
056ER	4380		BNL ERROR	NUMBER ERROR RETURN	00006170
	052ER				
0572R	088R		SHR R11,R11	INITIALIZE REGISTERS TO	00006180
0574R	08CC		SHR R12,R12	ZERO	00006190
0576R	C8D0	CNV1	LHI R13,C'0'	SET REGISTER TO DECIMAL	00006200
	0030				
057AR	08EC		LHR R14,R12		00006210
057CR	0AEF		AHR R14,R14		00006220
057ER	48EF		LH R14,BASE(R14)	LOAD DECIMAL BASE TABLE	00006230
	05BCR				
0582R	05AE	CNV2	CLHR R10,R14		00006240
0584R	4280		BL CNV3		00006250
	0590R				
0588R	08AF		SHR R10,R14		00006260
058AR	0AD5		AHR R13,ONE		00006270
058CR	4300		B CNV2		00006280
	0582R				
	*				
0590R	D2DC	CNV3	STB R13,LINE(R12)	STORE DEC DIGIT IN BUFFER	00006290
	0010R				00006300
0594R	C5C0		CLHI R12,3		00006310
	0003				
0598R	031F		BFCR 1,R15	BRANCH IF R12 NOT LOWER	00006320
059AR	C5D0		CLHI R13,C'0'		00006330
	0030				
059ER	4230		BNE CNV4		00006340
	05B4R				
05A2R	088R		LHR R11,R11		00006350
05A4R	4230		BNZ CNV5		00006360
	05B6R				
05A8R	C8D0		LHI R13,C' '	SPACE CHAR	00006370
	0020				
	*				
	*			IF PREV AND CURRENT CHAR	00006380
				ZERO, STORE SPACE	00006390
05ACR	D2DC		STB R13,LINE(R12)		00006400
	0010R				
05B0R	4300		B CNV5		00006410
	05B6R				
	*				
05B4R	08BD	CNV4	LHR R11,R13	SAVE PREV DIGIT	00006420
05B6R	0AC5	CNV5	AHR R12,ONE		00006430
05B8R	4300		B CNV1		00006440
	0576R				00006450
	*				
05BCR	03ER	BASE	DC 1000	DECIMAL	00006460
05BER	0064		DC 100	BASE	00006470
05C0R	000A		DC 10	TABLE	00006480
05C2R	0001		DC 1		00006490
					00006500



```

*
*
* SUBROUTINE MOVES TEXT
*
05C4R 0574  MOVVXT  CLHR  TO, FROM      MOVE TEXT
05C6R 033F          BFCR  3, R15      UP OR DOWN IN
05C8R 4380          BNL   M2        TEXT BUFFER
                                00006510
                                00006520
                                00006530
                                00006540
                                00006550
05CCR 03A4  M1     LB    R10,0(FROM)  THREE POINTERS USED      00006580
                                0000
05D0R 02A7          STB   R10,0(TO)   FROM BEGINNING POINT    00006590
                                0000
05D4R 0A75          AHR   TO, ONE     THRU ENDING POINT      00006600
05D6R 0140          BXLE  FROM, M1    TO DESTINATION POINT   00006610
                                05CCR
05DAR 030F          BR    R15
                                00006620
                                00006630
                                00006640
05DCR 0A76  M2     AHR   TO, THRU
05DER 0B74          SHR   TO, FROM
05E0R 03A6  M3     LB    R10,0(THRU)
                                00006660
                                0000
05E4R 02A7          STB   R10,0(TO)
                                00006670
                                0000
05E8R 0B75          SHR   TO, ONE
05EAR 0B65          SHR   THRU, ONE
05ECR 0564          CLHR  THRU, FROM
05EER 4380          BNL   M3
                                00006680
                                00006690
                                00006700
                                00006710
05F2R 030F          BR    R15
                                00006720
                                00006730
                                00006740
                                00006750
                                00006760
                                00006770
                                00006780
                                00006790
*
*
*           FOR ONE ARG COMMANDS COUNT IS ONE
*
05F4R 0535  ONEARG CLHR  COUNT, ONE   VALIDATE FOR
05F6R 028F          BTCR  8, R15      ONE ARGUMENT B IF LT
05F8R 4230          BNE   ERROR
                                00006800
05FCR 48C0          LH    R12, ARG1    LOOK AT ARG ONE  COUNT=1
                                00006810
                                0068R
0600R 4210          BM    ERROR        NO NEGATIVE ARGS
                                052ER
0604R 030F          BR    R15         RETURN
                                00006820
                                00006830
                                00006840
                                00006850
                                00006860
                                00006870
*
*
*           FOR TWO ARG COMMANDS COUNT IS TWO
*
0606R 4820  TWOARG LH    R2, ENDPTR   VALIDATE A TWO
                                000ER
060AR 4520          CLH   R2, BGNTXT   ARGUMENT COMMAND      00006880
                                0008R
060ER 4280          BL    ERROR        ERROR BUFFER EMPTY    00006890
                                052ER
0612R 0833          LHR   COUNT, COUNT
0614R 4230          BNZ   TW1         COUNT NOT =0          00006900
                                0634R
0618R 05F0          CLHI  R15, DELCHG+4 FOR CHANGE/DELETE NO ARGS 00006910

```

061CR	040CR 4330 0646R	BE	TW1A	MEANS USE OPEN LINE	0000691
0620R	C830 0002	LHI	COUNT,2	COUNT=0 FORCE =2	0000692
0624R	4050 0068R	STH	ONE,ARG1	SET ARG1 =1	0000693
0628R	4880 000ER	LH	R8,ENDPTR	END OF LAST LINE TO R8	0000694
062CR	41E0 01F6R	BAL	R14,CRCNT	GET LINE NUMBER	0000695
0630R	40A0 006AR	STH	R10,ARG2		0000696
0634R	0535	TW1	CLHR	COUNT,ONE	0000697
0636R	4230 0656R	BNE	TW2	COUNT =2	0000698
063AR	48A0 0068R	LH	R10,ARG1	FOR P,O,L,D,C COMMANDS, SET	0000698
063ER	40A0 006AR	STH	R10,ARG2	ARG2=ARG1=SINGLE ARG PRESENT	0000698
0642R	4300 0656R	B	TW2		0000698
0646R	4880 000CR	TW1A	LH	R8,OPNPTR	COUNT=0, CHANGE OR DELETE
		*		COUNT CR TO FIND O L	0000700
064AR	41E0 01F6R	BAL	R14,CRCNT	NUMBER	0000701
064ER	40A0 0068R	STH	R10,ARG1	NUMBER OF OPEN LINE	0000702
0652R	40A0 006AR	STH	R10,ARG2	IN R10 SET ARG1=ARG2	0000703
0656R	4880 0068R	TW2	LH	R8,ARG1	COUNT =2
065AR	4320 052ER	BNP	ERROR	NO NEGATIVE ARGS	0000705
065ER	48A0 006AR	LH	R10,ARG2		0000706
0662R	05A8	CLHR	R10,R8		0000707
0664R	4280 052ER	BL	ERROR		0000708
0668R	083F	LHR	COUNT,R15	SAVE EXIT ADDRESS	0000709
		*		R10 SET TO	0000710
		*		NUMBER OF OPEN OR ARG2 LINE	0000711
		*		ON ENTRY	0000712
066AR	41F0 0536R	BAL	R15,FINDLN		0000713
		*		AFTER FINDLN	0000714
		*		R1 = START	0000715
		*		R8 = END	0000716
		*		OF THE OPEN OR ARG2 LINE	0000717
066ER	08F3	LHR	R15,COUNT		0000718
0670R	030F	BR	R15		0000719
		*			0000720
		*			0000721
		*		PRINT EXCLAMATION POINT TEXT BUFFER OVERFLOW	0000722
		*			0000723

0672R	E110	OVRFLW	SVC	1,OVFL		00007240
	0902R					
0676R	4300	B	COMODE		TO COMMAND MODE	00007250
	000AR					
		*				00007260
		*				00007270
		*				00007280
067AR	C8E0	SETBIN	LHI	R14,X'2802'	PAPER TAPE OUTPUT	00007290
	2802					
067ER	40E0		STH	R14,DEVOUT		00007300
	090AR					
0682R	C8E0		LHI	R14,NTB		00007310
	0732R					
0686R	40E0		STH	R14,TABFLG	NO TABBING ON	00007320
	006CR					
068AR	030F		BR	R15	OUTPUT	00007330
		*				00007340
		*				00007350
		*				00007360
068CR	0799	PUTOUT	XHR	LINPTR,LINPTR	INITIALIZE LINE POINTER	00007370
068ER	4830		LH	COUNT,COMAND		00007380
	0066R					
0692R	0530		CLHI	COUNT,LIST	IF COMMAND L	00007390
	0453R					
0696R	4230		BNE	**8		00007400
	069ER					
069AP	C890		LHI	LINPTR,5	SKIP OVER NUMBER	00007410
	0005					
069ER	0329		LB	R2,LINE(LINPTR)	SAVE FIRST LINE CHAR	00007420
	0010R					
		*				00007430
06A2R	D309	NEWCHR	LB	CHAR,LINE(LINPTR)		00007440
	0010R					
06A6R	0500		CLHI	CHAR,X'0D'	IS CHAR CR?	00007450
	0000					
06AAR	4330		BE	EOL	YES	00007460
	077CR					
06AER	0500		CLHI	CHAR,X'20'	IS CHAR BLANK	00007470
	0020					
06B2R	4330		BE	SPACE	YES	00007480
	06C4R					
06B6R	0A95	NC1	AHR	LINPTR,ONE	BUMP LINE POINTER	00007490
06B8R	0590		CLHI	LINPTR,80	LINE LIMIT	00007500
	0050					
06BCR	4380		BNL	EOL	YES	00007510
	077CR					
06C0R	4300		B	NEWCHR	NO	00007520
	06A2R					
		*				00007530
		*				00007540
06C4R	0869	SPACE	LHR	THRU,LINPTR	SAVE LINE PTR	00007550
06C6R	0520		CLHI	R2,X'2A'	A * COMMENT CARD	00007560
	002A					
06CAR	4330		BE	NTB	YES, NO TABS	00007570
	0732R					
06CER	4870		LH	TO,TABFLG		00007571

06D2R	006CR C570 0732R		CLHI	TO,NTB	IF NO TAB SET,	0000757
06D6R	0337		BFCR	3,TO	GOTO NTB	0000757
06D8R	0A95	SCAN	AHR	LINPTR,ONE	SCAN	0000758
06DAR	0309 0010R		LB	CHAR,LINE(LINPTR)	OVER	0000759
06DER	C500 0020		CLHI	CHAR,X*20	SPACES	0000760
06E2R	4330 0738R		BE	SQUEEZ	DELETE EXTRA SPACES	0000761
06E6R	C500 000D	* SPAC1	CLHI	CHAR,X*0D	CR?	0000762 0000763
06EAR	4330 077CR		BE	EOL		0000764
06EER	4870 006CR		LH	TO,TABFLG	TABFLG IS NTB OR TB	0000765
06F2R	48E0 0066R		LH	R14,COMAND	TO SEE IF 'LIST'	0000766
06F6R	C830 0008		LHI	COUNT,8	OPERATION FOR PRINT	0000767
06FAR	C5E0 0458R		CLHI	R14,LIST		0000768
06FER	4230 0706R		BNE	**8		0000769
0702R	CA30 0005		AHI	COUNT,5	ADJUST	0000770
0706R	0593		CLHR	LINPTR,COUNT		0000771
0708R	0287		BTCR	8,TO		0000772 0000773 0000774
070AR	C830 000E	* .	LHI	COUNT,14	OPERAND FOR PRINT	0000775
070ER	C5E0 0458R		CLHI	R14,LIST		0000776
0712R	4230 071AR		BNE	**8		0000777
0716R	CA30 0005		AHI	COUNT,5	ADJUST	0000778
071AR	0539		CLHR	COUNT,LINPTR		0000779
071CR	0387		BFCR	8,TO		0000780 0000781
071ER	C830 0024	* .	LHI	COUNT,36	COMMENT FOR PRINT	0000782
0722R	C5E0 0458R		CLHI	R14,LIST		0000783
0726R	4230 072ER		BNE	**8		0000784
072AR	CA30 0005		AHI	COUNT,5	ADJUST	0000785
072ER	0593		CLHR	LINPTR,COUNT		0000786
0730R	0287		BTCR	8,TO		0000787
0732R	0896	* NTB	LHR	LINPTR,THRU	RESTORE	0000788
0734R	4300 06B6R		B	NC1	NEXT CHARACTER	0000789

	*				00007900
	*				00007910
0738R	0849	SQUEEZ	LHR	COLUMN,LINPTR	MOVE LINE RIGHT
073AR	D304		LB	CHAR,LINE+1(COLUMN)	
	0011R				00007920
073ER	D204		STB	CHAR,LINE(COLUMN)	
	0010R				00007940
0742R	C500		CLHI	CHAR,X'0D'	
	000D				00007950
0746R	4330		BE	SCAN+2	END OF LINE
	06DAR				00007960
074AR	0A45		AHR	COLUMN,ONE	
074CR	4300		B	SQUEEZ+2	
	073AR				00007970
					00007980
	*				00007990
	*				00008000
0750R	C840		TB	LHI	COLUMN,78
	004E				POINT TO END OF LINE
0754R	D304		LB	CHAR,LINE(COLUMN)	
	0010R				00008020
0758R	D204		STB	CHAR,LINE+1(COLUMN)	
	0011R				00008030
	*				MOVE LINE LEFT
075CR	0549		CLHR	COLUMN,LINPTR	
075ER	4330		BE	TB1	
	0768R				00008040
0762R	0845		SHR	COLUMN,ONE	
0764R	4300		B	TB+4	LOOP
	0754R				00008070
					00008080
	*				00008090
0768R	C800		TB1	LHI	CHAR,X'20'
	0020				GENERATE SPACE
076CR	D209		STB	CHAR,LINE(LINPTR)	INSERT
	0010R				00008100
	*				MOVE TEXT UNTIL LINPTR
0770R	0A95		AHR	LINPTR,ONE	EQUALS TAB COUNT
0772R	0593		CLHR	LINPTR,COUNT	
0774R	4280		BL	TB	
	0750R				00008120
0778R	4300		B	NEWCHR	PICK UP NEW CHAR
	06A2R				00008130
	*				00008140
	*				00008150
	*				END OF LINE
	*				00008170
	*				00008180
	*				00008190
	*				00008200
077CR	C800		EOL	LHI	CHAR,X'0D'
	000D				OUTPUT CR
0780R	D209		STB	CHAR,LINE(LINPTR)	INSURANCE
	0010R				00008210
0784R	E110		SVC	1,DEVOUT	OUTPUT LINE
	090AR				00008230
0788R	48E0		LH	R14,DEVOUT+2	TEST STATUS
	090CR				00008240
078CR	4210		BTC	1,IOERR	
	089AR				00008250
0790R	4820		LH	R2,COMAND	
					00008260

0794R	0066R C520		CLHI R2,PRINT	IF COMM L OR P	00008270
0798R	0468R 4330		BE CKLCNT	CHECK LINE/PG COUNT	00008280
079CR	07A6R C520		CLHI R2,LIST	BUT NOT ON	00008290
07A0R	0458R 4330 07A6R		BE CKLCNT	R AND O COMMANDS	00008300
07A4R	030F	*	BR R15	RETURN	00008310
		*			00008320
		*			00008330
		*			00008340
		*			00008350
07A6R	4820 006ER	CKLCNT	LH R2,LCOUNT	ON L CHECK	00008360
07AAR	0A25		AHR R2,ONE	LINE COUNT	00008370
07ACR	4020 006ER		STH R2,LCOUNT	LCOUNT SET =0 IN COMODE	00008380
07B0R	C520 0036		CLHI R2,54	LINE COUNT LIMIT OF 54	00008390
07B4R	028F		BTCR 8,R15	NO	00008400
		*			00008410
		*			00008420
07B6R	C830 000C		LHI COUNT,12		00008430
07BAR	C800 000D	LFS	LHI CHAR,X'0D'	AT END OF PAGE	00008440
07BER	D200 0010R		STB CHAR,LINE		00008450
07C2R	E110 090AR		SVC 1,DEVOUT		00008460
07C6R	48E0 090CR		LH R14,DEVOUT+2		00008470
07CAR	4210 089AR		BTC 1,IOERR		00008480
07CER	0B35 07D0R		SHR COUNT,ONE		00008490
	4230 07BAR		BNZ LFS		00008500
07D4R	4030 006ER		STH COUNT,LCOUNT	ZERO LCOUNT	00008510
07D8R	030F		BR R15		00008520
		*			00008530
		*			00008540
		*			00008550
		*			00008560
		*			00008570
07DAR	C8F0 4801	NOKBRD	LHI R15,X'4801'	PAPER TAPE INPUT	00008580
07DER	40F0 0912R		STH R15,DEVIN		00008590
07E2R	C8F0 0D0D		LHI R15,X'0D0D'		00008600
07E6R	40F0 0060R		STH R15,LINE+80		00008610
07EAR	E110		SVC 1,DEVIN		00008620

07EER	0912R 48E0	LH	R14,DEVIN+2	TEST STATUS	00008630
07F2R	0914R 4210	BTC	1,IOERR		00008640
07F6R	089AR 4300	B	SCANNER-2		00008650
	0816R				
	*				00008660
	*				00008670
	* KEYBOARD LINE INPUT				00008680
	*				00008690
07FAR	C8F0 4805	RDLINE LHI	R15,X'4805'	KEYBOARD INPUT	00008700
07FER	40F0	STH	R15,DEVIN		00008710
0802R	0912R C8F0	LHI	R15,X'0D0D'		00008720
0806R	0D0D 40F0	STH	R15,LINE+80		00008730
080AR	0060R E110	SVC	1,DEVIN	READ LINE	00008740
	0912R				
080ER	48F0	LH	R15,DEVIN+2		00008750
0812R	0914R 4210	BTC	1,LSTOPN	LINE BREAK,SHOW OPEN LINE	00008760
	02FER				
0816R	0799	XHR	LINPTR,LINPTR	RESET LINE POINTER	00008780
0818R	D309	SCANNER LB	CHAR,LINE(LINPTR)		00008790
	0010R				00008800
081CR	0A95	AHR	LINPTR,ONE		00008810
081ER	C500	CLHI	CHAR,X'0A'	IS CHAR LF OPEN LINE	00008820
	000A				
0822R	4330	BE	LNFEED	FOLLOWING CURR OPN LINE	00008830
	0880R				
0826R	C590	CLHI	LINPTR,82		00008840
	0052				
082AR	4380	BNL	ERROR		00008850
	052ER				
082ER	C500	CLHI	CHAR,X'0D'	CR	00008860
	000D				
0832R	4230	BNE	SCANNER	CONTINUE SCAN	00008870
	0818R				
0836R	0595	* FOUND CARRIAGE RETURN			00008880
0838R	4330	SCN1 CLHR	LINPTR,ONE	AT FIRST CHARACTER ?	00008890
	0850R	BE	ENDSC		00008900
083CR	D379	LB	TO,LINE-2(LINPTR)		00008910
	000ER				
0840R	C570	CLHI	TO,X'20'	SPACE?	00008920
	0020				
0844R	023D	BTCR	3,R13	NO	00008930
0846R	0895	SHR	LINPTR,ONE		00008940
0848R	D209	STB	CHAR,LINE-1(LINPTR)	DELETE TRAILING SPACES	00008950
	000FR				
084CR	4300	B	SCN1		00008960

0836R					00008970	
0850R	D370	ENDSC	LB	TO,MESS3	MODE FLAG	00008980
0854R	0928R C570		CLHI	TO,X'2A'	YES CR AND IN E MODE	00008990
0858R	002A 033D		BFCR	3,R13	EXIT	00009000
						00009010
						00009020
						00009030
						00009040
						00009050
						00009060
085AR	4880	BKSPAC	LH	R8,OPNPTR	OPEN	
	000CR					
085ER	0885	BK1	SHR	R8,ONE	AND	00009070
0860R	4580		CLH	R8,BGNTXT	LIST PREV LINE	00009080
	0008R					
0864R	4280		BL	ERROR	NO PREV LINE	00009090
	052ER					
0868R	4330		BE	BK2	PREV LINE AT BGNTXT	00009100
	0878R					
086CR	D308		LB	CHAR,-1(R8)		00009110
	FFFF					
0870R	C500		CLHI	CHAR,X'0D'	CR	00009120
	000D					
0874R	4230		BNE	BK1	CONTINUE SEARCH FOR CR	00009130
	085ER					
0878R	4080	BK2	STH	R8,OPNPTR	CR FOUND. SET POINTER	00009140
	000CP					
087CR	4300		B	LSTOPN	LIST OPEN LINE	00009150
	02FER					
						00009160
						00009170
						00009180
						00009190
						00009200
						00009210
						00009220
0880R	4880	LNFEED	LH	R8,OPNPTR		
	000CR					
0884R	41E0		BAL	R14,GETCR	FIND END OF O L	00009230
	055AR					
0888R	4580		CLH	R8,ENDPTR		00009240
	000ER					
088CR	4380		BNL	ERROR	ERROR	00009250
	052ER					
0890R	0A85		AHR	R8,ONE		00009260
0892R	4080		STH	R8,OPNPTR	SET NEW O L POINTER	00009270
	000CR					
0896R	4300		B	LSTOPN	LIST OPEN LINE	00009280
	02FER					
						00009290
						00009300
						00009310
						00009320
						00009330
						00009340
089AR	07DD					
089CR	08FE					

•  
•  
• OPENS AND LISTS LINE PRECEDING  
• CURRENT OPEN LINE

•  
•  
• LINE FEED ROUTINE ADJUSTS POINTERS  
• OPENS AND LISTS LINE  
• FOLLOWING CURRENT OPEN LINE

•  
• I/O TRANSFER ERROR  
• R14 = ERROR STATUS  
IOERR XHR R13,R13  
LHR R15,R14



089ER	CDE0	SLHL	R14,4		00009350
	0004				
08A2R	CCF0	SRHL	R15,12		00009360
	000C				
08A6R	C6F0	OHI	R15,X'30'	CONVERT HEX TO ASCII	00009370
	0030				
08AAR	C5F0	CLHI	R15,X'3A'		00009380
	003A				
08AER	4280	BL	**8		00009390
	08B6R				
08B2R	CAF0	AHI	R15,7		00009400
	0007				
08B6R	D2FD	STB	R15,MESS6+6(R13)		00009410
	0936R				
08BAR	0AD5	AHR	R13,ONE		00009420
08BCR	C5D0	CLHI	R13,4		00009430
	0004				
08C0R	4280	BL	IOERR+2	LOOP	00009440
	089CR				
		* PRINT ERROR MESSAGE			00009450
08C4R	E110	SVC	1,ERRIO		00009460
	091AR				
08C8R	4300	B	COMODE	RETURN TO COMMAND MODE	00009470
	00DAR				
		*			00009480
		* END OF JOB			00009490
		*			00009500
08CCR	E130	ENDJOB	SVC 3,0		00009510
	0000				
		*			00009520
		* PAUSE, RETURN TO OS			00009530
		* CONTINUE RESTARTS TIDE AT ORG +4			00009540
		*			00009550
08D0R	E120	PAUSE	SVC 2,**8		00009560
	08D8R				
08D4R	4300	B	RESTR		00009570
	0004R				
08D8R	0001	DC	1		00009580
					00009590
		*			00009600
		*			00009610
		*			00009620
		*			00009630
		* IO CALL PARAMETER BLOCKS			00009640
		*			00009650
08DAR	2805	CRLF	DC X'2805',0,MESS1,MESS1		
	0000				
	0926R				
	0926R				
08E2R	2805	TIDE	DC X'2805',0,MESS2,MESS2+4		00009660
	0000				
	0922R				
	0926R				
08EAR	2805	MODE	DC X'2805',0,MESS3,MESS3+1		00009670
	0000				
	0928R				
	0929R				

08F2R	2805	ERR	DC	X'2805',0,MESS4,MESS4+2	00009680
	0000				
	092AR				
	092CR				
08FAR	2805	LINEB	DC	X'2805',0,LINE,LINE+79	00009690
	0000				
	0010R				
	005FR				
0902R	2805	OVFL	DC	X'2805',0,MESS5,MESS5+1	00009700
	0000				
	092ER				
	092FR				
090AR	2805	DEVOUT	DC	X'2805',0,LINE,LINE+79	00009710
	0000				
	0010R				
	005FR				
0912R	4805	DEVIN	DC	X'4805',0,LINE,LINE+79	00009720
	0000				
	0010R				
	005FR				
091AR	2805	ERRIO	DC	X'2805',0,MESS6,MESS6+10	00009730
	0000				
	0930R				
	093AR				
	*				
	*				
	* MESSAGES				00009740
	*				00009750
0922R	5449	MESS2	DC	X'5449',X'4445',X'0D00'	00009760
	4445				00009770
	0D00				00009780
0926R		MESS1	EQU	MESS2+4	
0928R	000D	MESS3	DC	X'000D'	00009790
092AR	203F	MESS4	DC	X'203F',X'0D00'	00009800
	0D00				00009810
092ER	210D	MESS5	DC	X'210D'	
0930R	494F	MESS6	DC	C'IOERR',0,0,X'0D0D'	00009820
	4552				00009830
	5220				
	0000				
	0000				
	000D				
	*				
	*				
093CR	0000	END	DC	0	00009840
					00009850
					00009860
					00009870
					00009880
093ER		BGNBUF	DS	2000	00009890
110CR		ENDBUF	EQU	*-2	00009900
110ER			END	ORG	00009910
A1	035CR				00009920
A2	0372R				
A3	037AR				
AI	0336R				
AIFLG	0346R				

TEXT BUFFER

APPEND 0334R  
ARG1 0068R  
ARG2 006AR  
ARGADR 07CAR  
ARGTBL 02B8R  
BASE 05BCR  
BGNBUF 093ER  
BGNTXT 0008R  
BK1 085ER  
BK2 0878R  
BKSPAC 085AR  
BYTCNT 0506R  
CHANGE 03B8R  
CHAR 0000  
CKLCNT 07A6R  
CM1 012AR  
CNV1 0576R  
CNV2 0582R  
CNV3 0590R  
CNV4 05B4R  
CNV5 05B6R  
CNVBD 056AR  
CNVDTB 014AR  
COLUMN 0004  
COMADR 0280R  
COMAND 0066R  
COMODE 00DAR  
COMTRL 0264R  
COUNT 0003  
CR1 0206R  
CR2 0214R  
CRCNT 01F6R  
CRLF 08DAR  
CURLIN 0186R  
DELCHG 0408R  
DELETE 0404R  
DEVIN 0912R  
DEVOUT 090AR  
DLMIT1 01C0R  
DLMIT2 01DCR  
DLMIT3 01FOR  
DTB1 014CR  
DTB2 0168R  
DTB3 017ER  
END 093CR  
ENDBUF 110CR  
ENDJOB 08CCR  
ENDPTR 000ER  
ENDSC 0850R  
ENDTXT 000AR  
EOL 077CR  
ERR 08F2R  
ERRIO 091AR  
ERROR 052ER  
FINDLN 0536R  
FN1 0542R

FORCE	0092R
FROM	0004
GETARG	0222R
GETCR	055AR
GT1	0230R
INSERT	03BER
IOERR	089AR
KILL	0454R
LCOUNT	006ER
LENGTH	0002
LETXT	00A6R
LFS	07BAR
LINE	0010R
LINEB	08FAR
LINPTR	0009
LINUM	0490R
LIST	0458R
LISTLN	032CR
LNFEED	0880R
LNUMBR	02EER
LSTLIN	018ER
LSTOPN	02FER
M1	05CCR
M2	05DCR
M3	05E0R
MESS1	0926R
MESS2	0922R
MESS3	0928R
MESS4	092AR
MESS5	092ER
MESS6	0930R
MINUS	01AER
MODE	08EAR
MOVTXT	05C4R
NC1	06B6R
NEWCHR	06A2R
NOKBRD	07DAR
NOTAB	00D2R
NTB	0732R
ONE	0005
ONEARG	05F4R
OPNPRV	043AR
OPNPTR	000CR
OPUT	0478R
ORG	0000R
OUTPUT	0474R
OVFL	0902R
OVRFLW	0672R
PAUSE	08D0R
PLUS	019ER
PR1	047AR
PR2	0482R
PR3	04AER
PRINT	0468R
PROCES	013ER
PUTOUT	068CR

R1	0001
R10	000A
R11	000B
R12	000C
R13	000D
R14	000E
R15	000F
R2	0002
R8	0008
RDLINE	07FAR
REPROD	04C4R
RESTR	0004R
S3	03C4R
SCAN	06D8R
SCANNER	0818R
SCN1	0836R
SEARCH	0234R
SETBIN	067AR
SK1	04E2R
SK2	04E8R
SKIP	04D0P
SPAC1	06E6R
SPACE	06C4R
SQUEEZ	0738R
SR1	0240R
SR2	025CR
START	0070R
START2	0088R
TAB	04FAR
TABFLG	006CR
TB	0750R
TB1	0768R
THRU	0006
TIDE	08E2R
TO	0007
TW1	0634R
TW1A	0646R
TW2	0656R
TWOARG	0606P

0000	3	OS16MT	EQU	0	OSL0000
0001	4	DOS	EQU	1	OSL0000
	5	**LOADER			OSL0001
	6	**03030			OSL0002

8	*		OSL00040
9	*		OSL00050
10	*		OSL00060
11	*		OSL00070
12	*	*****	OSL00080
13	*		OSL00090
14	*	THIS LOADER OPERATES INTERACTIVELY, ACCEPTING	OSL00100
15	*	COMMANDS AND PARAMETERS FROM LOGICAL UNIT 5.	OSL00110
16	*	LOADED PROGRAMS MAY BE RELOCATABLE, AND MAY	OSL00120
17	*	REFERENCE COMMON OR EXTERNAL LABELS. FORWARD	OSL00130
18	*	REFERENCES ARE PERMITTED WITHIN PROGRAMS. THIS	OSL00140
19	*	LOADER IS DESIGNED TO OPERATE UNDER THE BOSS	OSL00150
20	*	SYSTEM WITH LU 5 ASSIGNED TO THE CONSOLE I/O	OSL00160
21	*	DEVICE. OTHER DEVICES ARE REFERENCED IN LOADER	OSL00170
22	*	COMMANDS BY THEIR LOGICAL-UNIT-NUMBERS WHICH	OSL00180
23	*	SHOULD BE ASSIGNED BEFORE EXECUTING THE LOADER.	OSL00190
24	*		OSL00200
25	*	OPERATING INSTRUCTIONS AND INPUT SPECS FOLLOW.	OSL00210
26	*	THIS EDITION OF THE LOADER INCLUDES OPERATOR-	OSL00220
27	*	COMMANDS FOR HANDLING A LIBRARY OF RE-	OSL00230
28	*	LOCATABLE PROGRAMS.	OSL00240
29	*		OSL00250
30	*	*****	OSL00260

32 \* OSL002  
33 \* LOADER-OPERATING INSTRUCTIONS OSL002  
34 \* OSL003  
35 \* OSL003  
36 \* LOAD THE LINKING LOADER AT THE BOTTOM OF USER-CORE BY USING THE OSL003  
37 \* OPERATING SYSTEM'S RESIDENT LOADER. OSL003  
38 \* ASSIGN LOGICAL UNITS TO ANY DEVICES THE LINKING LOADER MAY USE. OSL003  
39 \* THESE NORMALLY INCLUDE: OSL003  
40 \* A BINARY INPUT DEVICE OSL003  
41 \* AN ASCII OUTPUT DEVICE OSL003  
42 \* LU 5 FOR LOADER COMMANDS (NORMALLY CONSOLE) OSL003  
43 \* EXECUTE THE LINKING-LOADER AT ITS ORIGIN (TRANSFER ADDRESS) OSL003  
44 \* THE MESSAGE "LOADER" IS LOGGED TO INDICATE THAT A COMMAND IS OSL004  
45 \* EXPECTED FROM THE OPERATOR OSL004  
46 \* INPUT THE DESIRED COMMAND (SEE OPERATOR-COMMANDS, PAGE 3) OSL004  
47 \* WHEN THE MESSAGE "LOADER" IS OUTPUT, THE COMMAND HAS BEEN OSL004  
48 \* EXECUTED AND IS COMPLETE, ERROR-CONDITIONS ARE INDICATED OSL004  
49 \* BY SELF-EXPLANATORY MESSAGES. OSL004  
50 \* ADDITIONAL COMMANDS MAY BE ENTERED WHENEVER THE MESSAGE "LOADER" OSL004  
51 \* HAS BEEN LOGGED. OSL004  
52 \* IF CONTROL IS RETURNED TO THE OPERATING SYSTEM (EITHER BY THE OSL004  
53 \* RETURN COMMAND OR BY THE TERMINATION OF A USER PROGRAM) OSL004  
54 \* THE LOADER MAY BE RESTARTED AT ITS ORIGIN. OSL005  
55 \* IF AN ILLEGAL COMMAND IS DETECTED, "CMD-ERR" IS LOGGED. OSL005  
56 \* OSL005  
57 \* FOR HANDLING A PROGRAM-LIBRARY FILE: OSL005  
58 \* OSL005  
59 \* LIB UNIT SHOULD BE ANY DEVICE CAPABLE OF SEQUENTIAL READ, OSL005  
60 \* WRITE, BACKSPACE, AND REWIND. OSL005  
61 \* OSL005  
62 \* TO CREATE A NEW LIBRARY TAPE: OSL005  
63 \* 1 REWIND LIB UNIT. OSL005  
64 \* 2 WRITE A FILE-MARK ON THE LIB UNIT. OSL006  
65 \* 3 COPY LOADER ON TO LIB UNIT FROM A RELOCATABLE TAPE OSL006  
66 \* THIS TAPE MAY THEN BE USED TO LOAD THE OS LOADER WITH OSL006  
67 \* THE BOSS RESIDENT LOADER, PRIOR TO EXECUTING THE OS LOADER OSL006  
68 \* TO LOAD ADDITIONAL LIBRARY PROGRAMS OSL006  
69 \* OSL006  
70 \* TO ADD PROGRAMS TO LIBRARY: OSL006  
71 \* 1 WRITE AN APPROPRIATE LABEL ON THE LIB DEVICE (THIS MAY BE OSL006  
72 \* OMITTED IF THE PROGRAM IS ALREADY LABELED) OSL006  
73 \* 2 COPY THE PROGRAM ON THE LIB DEVICE. THE OUT COMMAND MAY BE OSL006  
74 \* USED TO ADD ABSOLUTE LOAD-MODULES TO THE LIBRARY, BUT THE LIB OSL007  
75 \* DEVICE MUST NOT BE USED AS AN INPUT UNIT IN THE SAME OPERATION OSL007  
76 \* NOTE: LABEL AND OUT COMMANDS DESTROY ANY USER PROGRAM LOADED OSL007  
77 \* IMMEDIATELY ABOVE THE LIB LOADER. OSL007  
78 \* OSL007  
79 \* TO LOAD A PROGRAM FROM THE LIBRARY: OSL007  
80 \* 1 FIND THE PROGRAM OSL007  
81 \* 2 LOAD FROM THE LIB DEVICE OSL007  
82 \* 3 EDIT OR FIND AND LINK IF THIS PROGRAM REFERENCES OTHER LIBRARY OSL007  
83 \* PROGRAMS OSL007  
84 \* OSL008  
85 \* LIBRARY-FILE DETAILS: OSL008  
86 \* OSL008  
87 \* ANY LIBRARY-PROGRAM WHICH MAY BE CALLED (FOUND) BY THE OSL008



88	*	OPERATOR MUST BE LABELED, EITHER BY BEING ASSEMBLED WITH A LABEL OR BY HAVING A LABEL ADDED WHEN IT IS COPIED INTO THE LIBRARY.	OSL00840
89	*		OSL00850
90	*		OSL00860
91	*		OSL00870
92	*	A FILE-MARK MUST FOLLOW THE LAST PROGRAM ON THE TAPE. THE LABEL, COPY, AND OUT OPERATIONS SEARCH FORWARD FOR THIS MARK BEFORE WRITING, THEN WRITE OVER IT, AND THEN WRITE ANOTHER FILE-MARK AFTER WRITING THE LAST RECORD. BEFORE USING THESE DIRECTIVES WITH A NEW TAPE, AN INITIAL FILE-MARK SHOULD BE WRITTEN (REWIND, EOF).	OSL00880
93	*		OSL00890
94	*		OSL00900
95	*		OSL00910
96	*		OSL00920
97	*		OSL00930
98	*		OSL00940
99	*		OSL00950
100	*	FOR HANDLING OS/16-MT CORE-IMAGE OVERLAYS:	OSL00960
101	*		OSL00970
102	*	OVERLAY UNIT SHOULD BE A RANDOM ACCESS DEVICE. IF A DRUM ITS PSEUDO-SECTOR SIZE SHOULD BE 256 BYTES.	OSL00980
103	*		OSL00990
104	*		OSL01000
105	*	OVERLAY DIRECTORY MAY BE INITIALIZED WITH THE DIRECT CMD.	OSL01010
106	*		OSL01020
107	*	TO ADD A PROGRAM TO THE OVERLAY FILE:	OSL01030
108	*	1 FIND THE PROGRAM IF ON A LIBRARY FILE	OSL01040
109	*	2 USE THE OL CMD TO ESTABLISH THE OVERLAY NAME AND LU OF THE DIRECTORY	OSL01050
110	*		OSL01060
111	*	3 LINK IN ALL PROGRAM SEGMENTS OF THE OVERLAY	OSL01070
112	*	4 INSERT THE PROGRAM IN THE OVERLAY FILE	OSL01080
113	*	NOTE: OVERLAYS ARE LOADED INTO MEMORY IMMEDIATELY ABOVE THE OS LOADER. ANY PROGRAMS PREVIOUSLY LOADED IN THAT AREA WILL BE DESTROYED.	OSL01090
114	*		OSL01100
115	*		OSL01110
116	*		OSL01120
117	*	OVERLAYS MAY BE DELETED FROM THE OVERLAY FILE VIA THE DELETE CMD.	OSL01130
118	*		OSL01140
119	*	A MAP OF THE OVERLAY FILE MAY BE OBTAINED VIA THE MD COMMAND.	OSL01150
120	*		OSL01160
121	*	TABLE OF LEGAL LOADER-COMMANDS	OSL01170
122	*		OSL01180
123	*	CMD: ARG1: ARG2: MEANING:	OSL01190
124	*		OSL01200
125	*	BIAS BBBB - SET BIAS TO BBBB	OSL01210
126	*	ORG BBBB - SET BIAS TO BBBB	OSL01220
127	*	ORG - - SET BIAS TO END OF LOADER	OSL01230
128	*	BC LLLL - SET LENGTH OF BLANK COMMON	OSL01240
129	*	LC LLLL - SET LENGTH OF LABELLED COMMON	OSL01250
130	*	OUT LU LABEL OUTPUT LABELED LOAD-MODULE TO LU	OSL01260
131	*	OUT LU - OUTPUT LOAD-MODULE TO LU,	OSL01270
132	*	XOUT - - END LOAD-MODULE, WRITE LAST RECORD	OSL01280
133	*	GO - - TRANSFER TO LOADED PROGRAM	OSL01290
134	*	MAP LU - OUTPUT MEMORY-MAP TO LU.	OSL01300
135	*	LABEL LU NAME WRITE LABEL-RECORD ON FILE LU	OSL01310
136	*	FIND LU NAME POSITION LU TO START OF NAMED PROG.	OSL01320
137	*	COPY LULU - COPY ONE PROGRAM FROM LU A TO LU B	OSL01330
138	*	COPY LULU NAME FIND AND COPY NAMED PROGRAM	OSL01340
139	*	DUPE LULU - DUPLICATE UNTIL EOF OR DEV END	OSL01350
140	*	DUPE LULU NAME DUPLICATE UNTIL NAME IS READ	OSL01360
141	*	LOAD LU - PURGE TABLE, LOAD ONE PROG FROM LU	OSL01370
142	*	LOAD LU NAME FIND AND LOAD A PROGRAM	OSL01380
143	*	LINK LU - LOAD ONE PROG FROM LU, LINKING	OSL01390

144	*	LINK	LU	NAME	FIND AND LINK NAMED PROGRAM	OSL014
145	*	EDIT	LU	-	LOAD ONLY REFERENCED PROGS FROM LU	OSL014
146	*	EDIT	LU	NAME	FIND NAMED LABEL AND EDIT	OSL014
147	*	REWIND	LU	-	REWIND LU	OSL014
148	*	EOF	LU	-	WRITE A FILE-MARK ON LU.	OSL014
149	*	TABLE	LU	-	WRITE TABLE-OF-CONTENTS TO LU.	OSL014
150	*	OVLY	BBBB	-	SET 'DEAD SPACE' TO BBBB	OSL014
151	*	DIRECT	LUSZ	-	INIT DIRECTORY ON LU, SET SIZE TO SZ	OSL014
152	*	OLAY	LU	NAME	SET UP FOR OVERLAY ON LU	OSL014
153	*	INSERT	-	-	INSERT OVERLAY IN FILE	OSL014
154	*	DELETE	LU	NAME	DELETE NAME FROM OVERLAY DIRECTORY	OSL015
155	*	CMPRES	LU	-	COMPRESS OVERLAY DIRECTORY & FILE	OSL015
156	*	MD	LU	-	MAP MOST RECENTLY USED DIR TO LU	OSL015
157	*	MD	LULU	-	MAP DIRECTORY ON 1ST LU TO 2ND LU	OSL015
158	*	END	-	-	EXIT TO THE OS.	OSL015
159	*	LOADER	INPUT	FORMATS		OSL015
160	*					OSL015
161	*	108-BYTE	BINARY	RECORD		OSL015
162	*	BYTES	0-1	SEQUENCE NUMBER -1, -2, -3, ETC.		OSL015
163	*	BYTES	2-3	CHECKSUM XH -1, EVERY OTHER HALF-WORD IN THE BUFFER.		OSL015
164	*	BYTES	4-107	CONTAIN CONTROL-ITEMS OPTIONALLY FOLLOWED BY DATA		OSL016
165	*	ITEMS	AS	FOLLOWS: A=ADRS, D=DATA, R=REL (IF REL MODE) CC=CHAR 7-BIT		OSL016
166	*	CITM:	DATA:	MEANING:		OSL016
167	*					OSL016
168	*	00	-	-	READ NEXT RECORD	OSL016
169	*	01	-	-	END OF PROGRAM	OSL016
170	*	02	-	-	DEFINE CHAIN	OSL016
171	*	03	-	-	TOGGLE ABS/REL MODE	OSL016
172	*	04	AAAA		TRANSFER-ADRS=AAAA RELATIVE	OSL016
173	*	05	AAAA		RELATIVE PROG LOAD-ADDRESS	OSL016
174	*	06	AAAA		RELATIVE REF (OR THREAD) ADDRESS	OSL017
175	*	07	AAAA		RELATIVE DEF ADDRESS	OSL017
176	*	08	DDDD		TEXT ABSOLUTE 1 HW	OSL017
177	*	09	DDDD		TEXT RELATIVE 1 HW	OSL017
178	*	0A	DDDD	DDDD	TEXT ABSOLUTE 2 HW	OSL017
179	*	0B	DDDD	DDDD	TEXT 1HW ABS, 1HW REL	OSL017
180	*	0C	CC	CC CC CC CC CC	REF SYMBOL	OSL017
181	*	0D	CC	CC CC CC CC CC	DEF SYMBOL	OSL017
182	*	0F	CC	CC CC CC CC CC	PROGRAM LABEL	OSL017
183	*	E0	CC	CC CC CC CC CC SSSS	DEFINE NAMED COMMON BLOCK, SIZE=SSSS	OSL017
184	*	E1	CC	CC CC CC CC CC AAAAR	REFERENCE NAMED COMMON, DISPLACEMENT	OSL018
185	*	E2	CC	CC CC CC CC CC AAAAR	DDDD LOAD COMMON TEXT 1 HW	OSL018
186	*	E3	CC	CC CC CC CC CC AAAAR	DDDD DDDD LOAD COM TEXT 2 HW	OSL018
187	*	E4	-	-	RESET SEQUENCE NR	OSL018
188	*					OSL018
189	*					OSL018
190	*	OVERLAY	DIRECTORY	FORMATS		OSL018
191	*					OSL018
192	*	256-BYTE	SECTORS			OSL018
193	*	1ST	DIRECTORY	SECTOR		OSL018
194	*	BYTES	0-1	NEXT AVAILABLE SECTOR		OSL019
195	*	BYTES	2-3	TOTAL NUMBER OF OVERLAYS		OSL019
196	*	BYTES	4-253	CONTAIN UP TO 25 OVERLAY DIRECTORY ENTRIES FORMATTED		OSL019
197	*	AS	FOLLOWS:			OSL019
198	*			BYTES 0-5 OVERLAY NAME		OSL019
199	*			BYTES 6-7 ABSOLUTE LOAD ADDRESS		OSL019



11F6R						
0030R	4210	246	BM	RPMSG	IF ERR STST, REPORT STAT & PAUSE	OSL024
	0FBAR					
0034R	4800	247	LH	PAR,LOGSWT	IF LOG SPECIFIED	OSL024
	0654R					
0038R	4330	248	BZ	NOLOG	SKIP BRANCH	OSL024
	0048R					
003CR	E110	249	SVC	1,LOGBL	ECHO MSG	OSL024
	1236R					
0040R	4800	250	LH	PAR,LOGBL+2	TEST STATUS	OSL024
	1238R					
0044R	4210	251	BM	LOGGER		OSL024
	0FBAR					
0048R		252	NOLOG	EQU *		OSL024
		253	*			OSL024
		254	*	* THIS ROUTINE SCANS THE COMMAND RECEIVED FROM THE OPERATOR AND		OSL025
		255	*	MOVES THE FIRST TWO CHARACTERS INTO THE ARG REGISTER		OSL025
		256	*	PACKS HEX DIGITS, RIGHT-JUSTIFIED, INTO PR,		OSL025
		257	*	AND MOVES THE LABEL FIELD INTO SYMBUF		OSL025
		258	*			OSL025
		259	*	* TYPICAL INPUT: LABEL 3 PNAME		OSL025
		260	*	* RESULT: (ARG)= LA (PAR)= X'0003' (SYMBUF)=PNAME		OSL025
		261	*			OSL025
0048R	C840	262	LHI	ARG,X'2020'	INITIALIZE SYMBUF AREA	OSL025
	2020					
004CR	4040	263	STH	ARG,SYMBUF	TO BLANKS, IN CASE LABEL	OSL025
	1180R					
0050R	4040	264	STH	ARG,SYMBUF+2	IS LESS THAN SIX BYTES	OSL026
	1182R					
0054R	4040	265	STH	ARG,SYMBUF+4	THIS ALSO SETS UP ARG	OSL026
	1184R					
0058R	40C0	266	STH	ZRO,SYMFLG	RESET SYMFLG FOR INITIAL START	OSL026
	102ER					
005CR	0700	267	XHR	PAR,PAR	ZERO PAR TO RIGHT-JUSTIFY HEX	OSL026
005ER	0755	268	XHR	NDX,NDX	ZERO MOVE-SOURCE-POINTER	OSL026
0060R	0733	269	XHR	AC2,AC2	ZERO MOVE DESTINATION-POINTER	OSL026
0062R	C810	270	LHI	RTN,-1	RTN INDICATES WHICH IS CUR DEST AREA	OSL026
	FFFF					
0066R	0766	271	XHR	KEY,KEY	HEX OPERAND PRESENT SWT	OSL026
0068R	D325	272	CRYTE	LB AC1,BUF(NDX)	FETCH NEXT BYTE	OSL026
	1244R					
006CR	C520	273	CLHI	AC1,CR	IF ITS A CR, THEN ALL IS IN	OSL026
	000D					
0070R	4330	274	BE	CMDPRC	GO PROCESS IT.	OSL027
	00EAR					
0074R	C520	275	CLHI	AC1,SP	IF ITS A NON-BLANK CHARACTER,	OSL027
	0020					
0078R	4230	276	BNE	CHAR	GO MOVE IT	OSL027
	0094R					
007CR	0A1D	277	AHR	RTN,ONE	IF BLANK, SWITCH TO THE NEXT FIELD	OSL027
007ER	0733	278	XHR	AC2,AC2	BY BUMPINT RTN AND RESTETTING AC2	OSL027
0080R	0A5D	279	CRYTE2	AHR	NDX,ONE	OSL027
0082R	091E	280	CHR	RTN,TWO	BUMP MOVE-SOURCE	OSL027
0084R	4380	281	BNL	CMDPRC		OSL027
	00EAR					
0088R	C550	282	CLHI	NDX,40	AFTER 40 CHARS, IF NO CR,	OSL027

0028							
0BCR 4280	283		BL	CBYTE	ASSUME IT'S ALL IN. AND GO		OSL02790
0068R							
090R 4300	284		B	CMDPRC	PROCESS IT.		OSL02800
00EAR							
094R 0811	285	CHAR	LHR	RTN,RTN	IF RTN IS NEGATIVE, MOVE BYTE		OSL02810
096R 4210	286		BM	DIR	TO DIR FIELD. IF POSITIVE, TO		OSL02820
00C4R							
09AR 4220	287		BP	SYM	LABEL FIELD, OTHERWISE, TREAT IT AS		OSL02830
00D6R							
09ER C660	288		OHI	KEY,-1	HEX FOUND		OSL02840
FFFF							
0A2R C520	289		CLHI	AC1,C'6'			OSL02850
0047							
0A6R 4380	290		HNL	CMDERR			OSL02860
010AR							
0AAR C520	291		CLHI	AC1,C'0'	HEX. IF ALPHA, ADD 9 IF NUMERIC,		OSL02870
0040							
0AER 4280	292		BL	CHAR1	JUST MASK OFF THE UPPER THREE BITS		OSL02880
00B6R							
0B2R CA20	293		AHI	AC1,9	SO THAT BY CHAR1, AC1 CONTAINS FOUR-		OSL02890
0009							
0B6R C420	294	CHAR1	NHI	AC1,15	BIT VALUE OF HEX CHARACTER. THIS GETS		OSL02900
000F							
0BAR CD00	295		SLHL	PAR,4	SHIFTED INTO PAR FROM THE RIGHT END.		OSL02910
0004							
0BER 0602	296		OHR	PAR,AC1	THE LAST FOUR DIGITS ARE LEFT IN PAR.		OSL02920
0COR 4300	297		B	CBYTE2			OSL02930
0080R							
0C4R 053E	298	DIR	CLHR	AC2,TWO	ONLY THE FIRST TWO CHARACTERS OF THE		OSL02940
0C6R 4380	299		BNL	CBYTE2	DIRECTIVE ARE USED...IGNORE ANY MORE.		OSL02950
0080R							
0CAR 0A3D	300		AHR	AC2,ONE	COUNT FIRST TWO AND SHIFT THEM INT		OSL02960
0CCR CD40	301		SLHL	ARG,8	ARG FROM THE RIGHT.		OSL02970
0008							
0D0R 0642	302		OHR	ARG,AC1			OSL02980
0D2R 4300	303		B	CBYTE2			OSL02990
0080R							
0D6R 053F	304	SYM	CLHR	AC2,SIX	IGNORE ANY CHARACTERS AFTER THE FIRST		OSL03000
0D8R 4380	305		BNL	CBYTE2	SIX OF THE LABEL FIELD.		OSL03010
0080R							
0DUR D223	306		STB	AC1,SYMBUF(AC2)	SYMBUF AND BUMP AC2.		OSL03020
11B0R							
0E0R 40D0	307		STH	ONE,SYMFLG	SET FLAG TO SHOW SYMBOL IN SYMBUF		OSL03030
102ER							
0E4R 0A3D	308		AHR	AC2,ONE			OSL03040
0E6R 4300	309		B	CBYTE2			OSL03050
0080R							
	310				* DECODE COMMAND BY COMPARING CHARS WITH CMDTAB ENTRIES		OSL03060
	311				* ARG CONTAINS 2 SIGNIFICANT CHARACTER, PAR CONTAINS HEX PARAMETER		OSL03070
0EAR CA50	312	CMDPRC	AHI	NDX,1	IN CASE OF ERROR		OSL03080
0001							
0EER 4050	313		STH	NDX,ECHO+2	SAVE LENGTH OF CMD		OSL03090
1242R							
0F2R C850	314		LHI	NDX,CMDTAB	SET INDEX TO TOP OF TABLE		OSL03100
103AR							

00F6R 4545	315	CML1	CLH	ARG,0(NDX)	TEST THIS ENTRY, IF MATCH, GO TO	OSL03
0000						
00FAR 4330	316		BE	CML2	CML2 TO ENTER COMMAND-ROUTINE.	OSL03
0112R						
00FER CA50	317		AHI	NDX,4	IF NO MATCH, BUMP INDEX, AND	OSL03
0004						
0102R C550	318		CLHI	NDX,ECMD	TRY NEXT ENTRY, IF NO MORE ENTRIES,	OSL03
109AR						
0106R 4280	319		HL	CML1	FALL THRU TO CMDERR.	OSL03
00F6R						
010AR E120	320	CMDEKR	SVC	2,CMERR	TYPE CMD-ERR MESSAGE, REQUEST A	OSL03
1100R						
010ER 4300	321		B	CKDEV		OSL03
0FCER						
0112R 4815	322	CML2	LH	RTN,2(NDX)	ON CMD-MATCH, FETCH COMMAND-ROUTINE	OSL03
0002						
0116R 0301	323		BR	RTN	ADDRESS AND BRANCH	OSL03
	324	* LOADER COMMAND-IMPLEMENTATION ROUTINES				OSL03
	325	*				OSL03
	326	* ON ENTRY TO THESE ROUTINES, PAR CONTAINS THE PACKED HEX				OSL03
	327	* PARAMETERS IF ANY WERE ENTERED				OSL03
	328	* ORG SETS THE BIAS (ORIGIN OF NEXT PROGRAM)				OSL03
0118R 0800	329	ORG	LHR	PAR,PAR	IF ZERO-ARGUMENT, SET BIAS TO	OSL03
011AR 4230	330		BNZ	ORG1	TOP OF LOADER (DEFAULT VALUE)	OSL03
012AR						OSL03
011ER 4860	331		LH	KEY,OUTFLG		OSL03
0FFAR						
0122R 4230	332		BNZ	ORG1		OSL03
012AR						
0126R C800	333		LHI	PAR,PBOT	ELSE, MOVE PAR TO BIAS	OSL03
131CR						
012AR 4000	334	ORG1	STH	PAR,BIAS	AND FETCH NEXT COMMAND	OSL03
0FF2R						
012ER 4000	335		STH	PAR,LOC		OSL03
100BR						
0132R 4300	336		B	LOADER		OSL03
0014R						
	337	*				OSL03
	338	* COMMON-LENGTH SETUP DIRECTIVES				OSL03
0136R 4000	339	LABCOM	STH	PAR,LABLEN	SET LAB-COM LENGTH	OSL03
102AR						
013AR 4300	340		B	LOADER		OSL03
0014R						
013ER 4000	341	BLKCOM	STH	PAR,BLKLEN	SET LENTH OF BLANK COMMON	OSL03
102CR						
0142R 4300	342		B	LOADER		OSL03
0014R						
	343	* WITHIN OLD COMMON				OSL03
	344	* WRITE A FILE-MARK ON LIBRARY FILE, AND BACKSPACE OVER IT				OSL03
	345	*				OSL03
0146R C810	346	CWFM	LHI	RTN,LOADER		OSL03
0014R						
014AR 4000	347		STH	PAR,OLU		OSL03
1020R						
014ER 4820	348	WFM	LH	AC1,OLU		OSL03
1020R						

52R D220	349	STB	AC1,FMB+1		OSL03450
1227R					
56R E110	350	SVC	1,FMB		OSL03460
1226R					
	351	*			OSL03470
	352	* BACK-SPACE A RECORD			OSL03480
	353	*			OSL03490
5AR 4820	354	BSR	LH AC1,OLU		OSL03500
1020R					
5ER D220	355	STB	AC1,BSB+1		OSL03510
1223R					
62R E110	356	SVC	1,BSB		OSL03520
1222R					
66R 0301	357	BR	RTN		OSL03530
	358	*			OSL03540
	359	* SEARCH FORWARD FOR A FILE-MARK AND BACKSPACE OVER IT.			OSL03550
	360	*			OSL03560
68R 4820	361	SFM	LH AC1,OLU		OSL03570
1020R					
6CR D220	362	STB	AC1,SFB+1		OSL03580
122RR					
170R E110	363	SVC	1,SFB		OSL03590
122AR					
174R 4300	364	B	BSR		OSL03600
015AR					
	365	* LABEL FILE			OSL03610
178R 4820	366	LABOPT	LH AC1,OUTFLG	DON'T ALLOW LABEL IF OUT IS IN	OSL03620
OFFAR					
17CR 4230	367	BZ	CMDERR	PROGRESS. BUFFER CUD BE DESTROYED.	OSL03630
010AR					
180R 4820	368	LH	AC1,SYMFLG	DON'T LABEL IF NO LABEL	OSL03640
102ER					
184R 4330	369	BZ	CMDERR	IS INPUT IN COMMAND	OSL03650
010AR					
188R C870	370	LHI	RTX,LOADER		OSL03660
0014R					
18CR D200	371	LABMOD	STB PAR,BOUTDV+1	SAVE OUTPUT UNIT #	OSL03670
1205R					
190R 4000	372	STH	PAR,OLU		OSL03680
1020R					
194R 40C0	373	STH	ZRO,USEQ	ZERO OUTPUT SEQ. NR.	OSL03690
100CR					
198R 4110	374	BAL	RTN,SFM	SPACE FWD TO FILE-MARK, AND BACK OVER	OSL03700
0168R					
19CR 4110	375	BAL	RTN,ZOBUF	ZERO OUTPUT-BUFFER	OSL03710
0EE4R					
11A0R 4820	376	LH	AC1,SYMFLG		OSL03720
102ER					
11A4R 0337	377	BZR	RTX		OSL03730
11A6R C800	378	LHI	PAR,15	PACK OUT A PROGRAM-LABEL CONTROL-ITEM	OSL03740
000F					
11AAR 4110	379	BAL	RTN,OPAK	BY CALLING OPAK	OSL03750
0E1CR					
11AER 0755	380	XHR	NDX,NDX	INITIALIZE BYTE-COUNTER	OSL03760
11BOR D325	381	LAB1	LB AC1,SYMBUF(NDX)	FETCH A BYTE FROM SYMBUF (LABEL)	OSL03770
11BOR					

01B4R	0802	382	LHR	PAR,AC1	MOVE I,, FOUR BITS AT ATIME.	OSL037
01B6R	CC00	383	SRHL	PAR,4	TO THE OUTPUT BUFFER	OSL037
	0004					
01BAR	4110	384	BAL	RTN,OPAK	IN TWO CALLS TO OPAK.	OSL038
	0E1CR					
01BER	0802	385	LHR	PAR,AC1		OSL038
01COR	4110	386	BAL	RTN,OPAK		OSL038
	0E1CR					
01C4R	0A5D	387	AHR	NDX,ONE	BUMP BYTE-COUNTER, AND CONTINUE	OSL038
01C6R	055F	388	CLHR	NDX,SIX	UNTIL SIX BYTES HAVE BEEN MOVED.	OSL038
01C8R	4280	389	BL	LAB1		OSL038
	01B0R					
01CCR	C800	390	LHI	PAR,14		OSL038
	000E					
01D0R	4110	391	BAL	RTN,OPAK		OSL038
	0E1CR					
01D4R	C800	392	LHI	PAR,4		OSL038
	0004					
01D8R	4110	393	BAL	RTN,OPAK		OSL038
	0E1CR					
01DCR	4110	394	BAL	RTN,OUTTXT		OSL039
	0EB0R					
01E0R	4110	395	BAL	RTN,WFM		OSL039
	014ER					
01E4R	0307	396	BR	RTX		OSL039
		397				
		398	* FIND A LABEL			OSL039
01E6R	C810	398	FIND	LHI	RTN,LOADER	OSL039
	0014R					OSL039
01EAR	4820	399	LH	AC1,SYMFLG		OSL039
	102ER					
01EER	4330	400	BZ	CMDERR		OSL039
	010AR					
01F2R	4010	401	FINDER	STH	RTN,FIND3	OSL039
	0254R					
01F6R	D200	402	STB	PAR,BINDV+1	SET INPUT UNIT NUMBER	OSL039
	11FDR					
01FAR	4820	403	LH	AC1,SYMFLG		OSL039
	102ER					
01FER	0331	404	BZR	RTN		OSL040
0200R	4000	405	STH	PAR,OLU		OSL040
	1020R					
0204R	D200	406	STB	PAR,RWD+1	AND REWIND UNIT-NUMBER	OSL040
	121BR					
0208R	E110	407	SVC	1,RWD	REWIND LIB IF REWINDABLE DEVICE	OSL040
	121AR					
020CR	4800	408	LH	PAR,RWD+2		OSL040
	121CR					
0210R	4210	409	BM	INER2+4		OSL040
	0FAAR					
0214R	4110	410	FIND1	BAL	RTN,LREAD	OSL040
	0F30R					
0218R	4300	411	B	PNF		OSL040
	0256R					
021CR	4820	412	LH	AC1,BUF	FETCH SEQUENCE-NUMBER	OSL040
	1244R					
0220R	C520	413	CLHI	AC1,-1	IS IT -17	OSL040



24R	FFFF 4230	414		BNE	FIND1	IF NOT, READ ANOTHER RECORD	OSL04100
	0214R						
28R	0788	415		XHR	ITM,ITM	IF SEQ NR IS -1, THEN LETS FETCH	OSL04110
2AR	4110	416		BAL	RTN,FETCH	THE FIRST CONTROL-ITEM (TO PAR)	OSL04120
	0BEER						
2ER	C500	417		CLHI	PAR,15	IS IT 15?	OSL04130
	000F						
32R	4230	418		BNE	FIND1	IF NOT, NO LABEL SO IGNORE PROGRAM	OSL04140
	0214R						
36R	0755	419		XHR	NDX,NDX	IF LABEL, FETCH FIRST TWO BYTES OF IT	OSL04150
38R	4110	420	FIND2	BAL	RTN,DFETCH	AND COMPARE,A HALF-WORD AT A TIME	OSL04160
	0C54R						
3CR	4545	421		CLH	ARG,SYMBUF(NDX)	WITH SYMBUF	OSL04170
	11B0R						
40R	4230	422		BNE	FIND1	IF NOT EQUAL, GO LOOKSOME MORE	OSL04180
	0214R						
44R	0A5E	423		AHR	NDX,TWO	IF EQUAL, BUMP NDX TO CHECK NEXT HW.	OSL04190
46R	C550	424		CLHI	NDX,5	THRU YET???	OSL04200
	0005						
4AR	4280	425		BL	FIND2		OSL04210
	0238R						
4ER	4110	426		BAL	RTN,BSR	IF THIS IS THE LABEL, BACKSPACE OVER	OSL04220
	015AR						
52R	4300	427		B	*		OSL04230
	0252R						
54R		428	FIND3	EQU	**2		OSL04240
56R	4820	429	PNF	LH	AC1,SYMFLG		OSL04250
	102ER						
5AR	0337	430		BZR	RTX		OSL04260
5CR	D100	431		LM	ONE,SYMBUF		OSL04270
	11B0R						
60R	D000	432		STM	ONE,PNFM		OSL04280
	0270R						
64R	E120	433		SVC	2,PNF2		OSL04290
	026CR						
68R	4300	434		B	LOADER		OSL04300
	0014R						
6CR	0007	435	PNF2	DC	7,16,C'000000 NOT FOUND'		OSL04310
	0010						
	3030						
	3030						
	3030						
	204E						
	4F54						
	2046						
	4F55						
	4E44						
70R		436	PNFM	EQU	PNF2+4		OSL04320
		437	*				OSL04330
		438	* PROGRAM-COPY ROUTINE				OSL04340
		439	*				OSL04350
80R	D200	440	COPY	STB	PAR,C0OUT+1	SET UP OUTPUT DEVICE-NUMBER	OSL04360
	122FR						
84R	D200	441		STB	PAR,COWT+1		OSL04370
	121FR						

0288R	CC00 0008	442		SRHL	PAR,8	SHIFT TO GET INPUT DEVICE-NUMBER	OSL041
028CR	4110 01F2R	443		BAL	RTN,FINDER	FIND REQUESTED PROGRAM	OSL041
0290R	D300 121FK	444		LB	PAR,COWT+1		OSL041
0294R	4000 1020R	445		STH	PAR,OLU		OSL041
0298R	4110 0168R	446		BAL	RTN,SFM	POSITION OUTPUT-FILE TO END	OSL041
029CR	4110 0F30R	447	COPY1	BAL	RTN,LREAD	READ INPUT-RECORD	OSL041
02A0R	4300 02E6R	448		B	COPY4	IF EOF, TERMINATE COPY	OSL041
02A4R	E110 122ER	449		SVC	1,COOUT	WRITE IT OUT, AND WHILE WRITING,	OSL041
02ABR	0788	450		XHR	ITM,ITM	LOOK AT ITS CONTROL-ITEMS TO FIND END	OSL041
02AAR	4110 0BEER	451	COPY2	BAL	RTN,FETCH	MOVE ITEM TO PAR	OSL041
02AER	0500	452		CLHR	PAR,ONE	IF 1, END OF PROGRAM, WRITE	OSL044
02B0R	4330 02E6R	453		BE	COPY4	FINAL FILE-MARK	OSL044
02B4R	0820	454		LHR	AC1,PAR	IF ZERO, GO WAIT FOR WRITE TO BE	OSL045
02B6R	4330 02F6R	455		BZ	COPY5	FINISHED AND READ NEXT RECORD	OSL045
02BAR	CD20 0002	456		SLHL	AC1,2	OTHERWISE, MULT ITEM BY FOUR TO INDEX	OSL045
02BER	C500 000E	457		CLHI	PAR,14	THRU TABLE. IF E, GO FOR SECOND	OSL045
02C2R	4330	458		BE	COPY3	GIGIT	OSL045
02CER	02CER						
02C6R	4A82 109CR	459		AH	ITM,CITTAB+2(AC1)	BUMP ITM BY NUMBER OF DATA-ITEMS.	OSL045
02CAR	4300 02AAR	460		B	COPY2	GO FETCH NEXT CONTROL ITEM.	OSL045
02CER	4110 0BEER	461	COPY3	BAL	RTN,FETCH	ON E ITEMS, FETCH SECOND DIGIT	OSL045
02D2R	0820	462		LHR	AC1,PAR	MOVE IT TO AC1.	OSL045
02D4R	CD20 0002	463		SLHL	AC1,2	MULTIPLY BY FOUR	OSL045
02D8R	4812 10DCR	464		LH	RTN,COMTAB+2(AC1)	AND BUMP ITM-POINTER.	OSL046
02DCR	4210 02AAR	465		BM	COPY2		OSL046
02E0R	0A81	466		AHR	ITM,RTN		OSL046
02E2R	4300 02AAR	467		B	COPY2		OSL046
02E6R	D300 122FR	468	COPY4	LB	PAR,COOUT+1		OSL046
02EAR	4000 1020R	469		STH	PAR,OLU		OSL046
02ELR	4110 014ER	470		BAL	RTN,WFM	WHEN END IS REACHED, WRITE A FILL-	OSL046
02F2R	4300 0014R	471		B	LOADER	MARK AND FETCH NEXT COMMAND.	OSL046
02F6R	E110	472	COPY5	SVC	1,COWT		OSL046

121ER						
2FAR 4800	473		LH	PAR,COOUT+2	TEST OUTPUT DEVICE STATUS	OSL04690
1230R						
2FER 4230	474		BNZ	INER2+4		OSL04700
0FAAR						
302R 4300	475		B	COPY1		OSL04710
029CR						
	476	*				OSL04720
306R D200	477	DUPER	STB	PAR,COOUT+1	DUPLICATION ROUTINE	OSL04730
122FR						
30AR D200	478		STB	PAR,COWT+1	COPIES ALL PROGRAMS UNTIL LABEL	OSL04740
121FR						
30ER D200	479		STB	PAR,OLU+1		OSL04750
1021R						
312R CC00	480		SRIIL	PAR,8		OSL04760
0008						
316R D200	481		STB	PAR,BINDV+1		OSL04770
11FDR						
31AR 4110	482		BAL	RTN,SFM		OSL04780
0168R						
31ER 4110	483	DUP1	BAL	RTN,LREAD		OSL04790
0F30R						
322K 4300	484		B	COPY4		OSL04800
02E6R						
326R 4170	485		BAL	RTX,LTST	CHECK IF LABEL-RECORD	OSL04810
0368R						
32AR 4300	486		B	DUP4	IF NOT	OSL04820
0354R						
32ER 0755	487		XHR	NDX,NDX	IF SO, COMPARE IT WITH SYMBUF	OSL04830
330R 4110	488	DUP3	BAL	RTN,DFETCH		OSL04840
0C54R						
334R 4545	489		CLH	ARG,SYMBUF(NDX)		OSL04850
11B0R						
338R 4230	490		BNE	DUP4		OSL04860
0354R						
33CR 0A5E	491		AHR	NDX,TWO		OSL04870
33ER 055F	492		CLHR	NDX,SIX		OSL04880
340R 4280	493		BL	DUP3		OSL04890
0330R						
344R D300	494		LB	PAR,BINDV+1		OSL04900
11FDR						
348R D200	495		STB	PAR,OLU+1		OSL04910
1021R						
34CR 4110	496		BAL	RTN,BSR	IF EQUAL, BACK OVER IT & STOP	OSL04920
015AR						
350R 4300	497		B	COPY4		OSL04930
02E6R						
354R E110	498	DUP4	SVC	1,COOUT		OSL04940
122ER						
358R E110	499		SVC	1,COWT		OSL04950
121EP						
35CR 4800	500		LH	PAR,COOUT+2	TEST OUTPUT DEVICE STATUS	OSL04960
1230R						
360R 4230	501		BNZ	INER2+4		OSL04970
0FAAR						
364R 4300	502		B	DUP1		OSL04980

031ER	503	*				OSL049
	504	*	LABEL-TESTER ROUTINE			OSL050
	505	*				OSL050
	506	*	LABEL-TESTER ROUTINE...RETURNS TO +4 IF NOLABEL, +8 IF LABEL			OSL050
0368R 4820	507	LTST	LH	AC1,BUF		OSL050
1244R						
036CR C520	508		CLHI	AC1,-1		OSL050
FFFF						
0370R 0237	509		BNER	RTX		OSL050
0372R 0788	510		XHR	ITM,ITM		OSL050
0374R 4110	511		BAL	RTN,FETCH		OSL050
0BEER						
0378R C500	512		CLHI	PAR,15		OSL050
000F						
037CR 4337	513		BE	4(RTX)		OSL050
0004						
0380R 0307	514		BR	RTX		OSL051
	515	*				OSL051
	516	*	LIB-TABLE OF CONTENTS			OSL051
	517	*				OSL051
0382R C820	518	TABCON	LHI	AC1,PRMS		OSL051
113CR						
0386R C830	519		LHI	AC2,PRMS+21		OSL051
1151R						
038AR D200	520		STB	PAR,MAPDEV+1	SET UP OUTPUT LU NUMBER	OSL051
1135R						
038ER CC00	521		SRHL	PAR,8		OSL051
0008						
0392R 4110	522		BAL	RTN,MLOG	OUTPUT HEADING "PROGRAMS"	OSL051
0C76R						
0396R D200	523		STB	PAR,BINDV+1		OSL051
11FDR						
039AR D200	524		STB	PAR,RWD+1		OSL052
121BR						
039ER 4000	525		STH	PAR,OLU		OSL052
1020R						
03A2R E110	526		SVC	1,RWD		OSL052
121AR						
03A6R 4800	527		LH	PAR,RWD+2		OSL052
121CR						
03AAR 4210	528		BM	INER2+4		OSL052
0FAAR						
03AER 4110	529	TAB1	BAL	RTN,LREAD		OSL052
0F30R						
03B2R 4300	530		B	LOADER		OSL052
0014R						
03B6R 4170	531		BAL	RTX,LTST		OSL052
0368R						
03BAR 4300	532		B	TAB1		OSL052
03AER						
03BER 4110	533		BAL	RTN,SFETCH		OSL052
0C1AR						
03C2R C820	534		LHI	AC1,SYMBUF		OSL053
11B0R						
03C6R C830	535		LHI	AC2,SYMBUF+5		OSL053

11B5R ICAR 4110	536	BAL	RTN,MLOG		OSL05320
0C76R ICER 4300	537	B	TAB1		OSL05330
03AER					
	538	*			OSL05340
0D2R 4170	539	OUT	BAL	RTX,LABMOD	LABEL OUTPUT MOD IF POSSIBLE OSL05350
018CR 0D6R 4000	540	STH	ONE,OUTFLG	AND SET OUT FLAG	OSL05360
OFFAR 0DAR 40C0	541	STH	ZRO,OITM	RESET OUTPUT BUFFER-POINTER	OSL05370
100ER 0DER 40C0	542	STH	ZRO,OSEQ	RESET OUTPUT SEQUENCE-NUMBER	OSL05380
100CR 0E2R 4110	543	BAL	RTN,ZOBUF	ZRO TO OUTPUT BUFFER	OSL05390
0EE4R 0E6R 40C0	544	STH	ZRO,OLOC	RESET OUTPUT LOC-FLAG	OSL05400
101AR 0EAR C800	545	LHI	PAR,3	PACK A TOGGLE FIRST	OSL05410
0003 0EER 4110	546	BAL	RTN,OPAK	INTO OUTPUT BUFFER.	OSL05420
0E1CR 0F2R 4300	547	B	LOADER		OSL05430
0014R 0F6R 40C0	548	XOUT	STH	ZRO,OUTFLG	ON XOUT, RESET OUT-FLAG OSL05440
OFFAR 0FAR 4820	549	LH	AC1,BOUADV	MOVE BOUADV TO	OSL05450
1204R 0FER 0220	550	STB	AC1,OLU+1		OSL05460
1021R 002R 4820	551	LH	AC1,PTOP	AND PACK OUT PTOP	OSL05470
OFF4R 006R 4830	552	LH	AC2,COMBOT	AND COMBOT AFTER END ITEM.	OSL05480
1022R 00AR 0800	553	LHR	PAR,ONE	TO SET POINTERS IN OS	OSL05490
00CR 4110	554	BAL	RTN,OFW	PACK IT OUT, AND UNLESS IT	OSL05500
0E58R 010R 4820	555	LH	AC1,OITM	FILLED A RECORD EXACTLY,	OSL05510
100ER 014R 4330	556	BZ	JUSTFM		OSL05520
041CR 018R 4110	557	BAL	RTN,OUTTXT	WRITE OUT THE LAST BUFFER.	OSL05530
0EB0R					
	558	JUSTFM	EQU	*	OSL05540
01CR 4110	559	BAL	RTN,WFM	AND A FINAL EOF	OSL05550
014ER 020R 4300	560	B	GULOAD	EXIT TO RESET TOP-OF-CORE	OSL05560
0000R 024R 0200	561	REWIND	STB	PAR,RWD+1	SET LU PARAMETER IN SVC BLOCK, OSL05570
121BR 028R E110	562	SVC	1,RWD	AND REWIND THE UNIT.	OSL05580
121AR 02CR 4800	563	LH	PAR,RWD+2		OSL05590
121CR 030R 4210	564	BM	INER2+4		OSL05600
0FAAR					

0434R	4300 0014R	565	B	LOADER		OSL056
		566 *				OSL056
0438R	4800 0FFCR	567	GO	LH PAR,XFER	ON GO, TEST TRANSFER-ADDRESS.	OSL056
043CR	C500 131CR	568		CLHI PAR,PBOT	IF BELOW PBOT (AS IN CASE OF NO XFER)	OSL056
0440R	4280 010AR	569		BL CMDERR	GO TO ERR. OTHERWISE, BAL TO USER-	OSL056
0444R	0110	570		BALR RTN,PAR	LOADED TRANSFER-ADDRESS. IF USER PROG	OSL056
0446R	4300 0014R	571		B LOADER	RETURNS VIA RTN, FETCH NEXT COMMAND.	OSL056
044AR	40D0 078ER	572	AMAP	STH ONE,AFLAG	SET AFLAG FOR ALPHABETIC SORT	OSL056
044ER	4300 0456R	573		B MAPGO	B	OSL056
0452R	40C0 078ER	574	MAP	STH ZRO,AFLAG	SET AFLAG FOR ADDRESS SORT	OSL057
0456R	4110 06F8R	575	MAPGO	BAL RTN,BUBBLE	DO SORT	OSL057
045AR	0200 1135R	576		STB PAR,MAPDEV+1	SET MAP-LU IN PBLK	OSL057
045ER	01C0 1032R	577		LM ZRO,ZROCON		OSL057
0462R	4820 123ER	578		LH AC1,LDFLG	TEST IF LOAD HAS BEEN DONE	OSL057
0466R	4230 0476R	579		BNZ MAP0	IF NOT, DON'T PRINT MAP.	OSL057
		580 *				OSL057
046AR	4820 0FF6R	581	RESBOT	LH AC1,SBOT	FETCH SBOT. IF EQUAL TO TOC.	OSL057
046ER	4520 1026R	582		CLH AC1,STOP	TABLE IS EMPTH. LOG NO PROGRAMS	OSL057
0472R	4280 047ER	583		BL MAP1	MESSAGE AND FETCH NEXT COMMAND.	OSL057
0476R	E120 110CR	584	MAP0	SVC 2,NPRG	OTHERWISE, LIST MEMORY-MAP IN	OSL058
047AR	4300 0014R	585		B LOADER	ORDER OF LOADING.	OSL058
047ER	C820 113CR	586	MAP1	LHI AC1,PRMS		OSL058
0482R	C830 1151R	587		LHI AC2,PRMS+21	LIST HEADING 'PROGRAMS:'	OSL058
0486R	4110 0C76R	588		BAL RTN,MLOG		OSL058
048AR	0766	589		XHR KEY,KEY	SET KEY TO 0 TO SEARCH FOR LABELS	OSL058
048CR	4170 04CCR	590		BAL RTX,NEWMAP	PRINT PROGRAM MAP	OSL058
0490R	C820 1152R	591		LHI AC1,EPMS		OSL058
0494R	C830 1161R	592		LHI AC2,EPMS+15	LIST HEADING 'ENTRY-POINTS:'	OSL058
0498R	4110 0C76R	593		BAL RTN,MLOG		OSL058
049CR	0860	594		LHR KEY,ONE	SET KEY TO 1 TO SEARCH FOR DEFS	OSL059
049ER	4170	595		BAL RTX,NEWMAP	PRINT ENTRY-POINT MAP	OSL059

A2R	04CCR C820 1162R	596	LHI	AC1,CBMS		OSL05920
A6R	C830 1172R	597	LHI	AC2,CBMS+16		OSL05930
AAR	4110 0C76R	598	HAL	RTN,MLOG	LIST HEADING 'COMMON-BLOCKS'	OSL05940
ALR	C860 0003	599	LHI	KEY,3	SET KEY TO 3 TO SEARCH FOR DEF-COMS.	OSL05950
B2R	4170 04CCR	600	BAL	RTX,NEWMAP	PRINT COMMON-BLOCK MAP	OSL05960
B6R	C820 1174R	601	LHI	AC1,USMS		OSL05970
BAR	C830 1180R	602	LHI	AC2,USMS+12		OSL05980
BER	4110 0C76R	603	BAL	RTN,MLOG	LIST HEADING 'UNDEFINED SYMBOLS'	OSL05990
C2R	086E	604	LHR	KEY,TWO	SET KEY TO 2 TO SEARCH FOR REFS.	OSL06000
C4R	4170 04CCR	605	BAL	RTX,NEWMAP	LIST UNDEFINED EXTERNAL SYMBOLS	OSL06010
C8R	4300 0014R	606	B	LOADER	AND FETCH NEXT COMMAND	OSL06020
4CCR	4850	607	* THIS ROUTINE LISTS MEMORY-MAPS ON MAPDEV			OSL06030
	1026R	608	NEWMAP	LH	NOX,STOP	NOX POINTS AT CURRENT TABLE-ENTRY
4DOR	40C0 1012R	609	STH	ZRO,MADA		OSL06050
4D4R	0733	610	NEWLIN	XHR	AC2,AC2	AC2 POINTS AT NEXT CHAR POSITION OUT
4D6R	C820 0003	611	NEWSYM	LHI	AC1,3	BEFORE MOVING EACH SYMBOL, MOVE 3
4DAR	C800 0020	612	LHI	PAR,X*20*		OSL06080
4DER	D203 1244R	613	MA1	STB	PAR,BUF(AC2)	OUTPUT AS FOUR VALUES AND FOUR SYMB-
4E2R	0A3D	614	AHR	AC2,ONE		OSL06100
4E4R	0B2D	615	SHR	AC1,ONE		OSL06110
4E6R	4220 04DER	616	BP	MA1		OSL06120
4EAR	4110 0CA4R	617	BAL	RTN,CSRCH	REGISTER, KEY, AS SET BY CALLING PROG.	OSL06130
4EER	4300 056ER	618	B	EMAP2	IF TABLE SEARCHED TO BOTTOM, END MAP.	OSL06140
4F2R	056E	619	CLHR	KEY,TWO	OTHERWISE, IF KEY IS 2 PRINT SYMBOL	OSL06150
4F4R	4330 051CR	620	BE	MSYM	AND NO VALUE. IF KEY IS NOT 2, FEICH	OSL06160
4F8R	C810 051CR	621	LHI	RTN,MSYM	THE VALUE OF THE SYMBOL (AT NOX+6)	OSL06170
4FCR	4805 0006	622	LH	PAR,6(NOX)	UNPAK IT AND MOVE IT TO THE	OSL06180
500R	0722	623	WRTV	XHR	AC1,AC1	PRINT-BUFFER AND
502R	E120 10FCR	624	SVC	2,UNPAK		OSL06200
506R	D302 1002R	625	MA	LB	PAR,TMP1(AC1)	AND THEN INTO BUFFER
50AR	D203 1244R	626	STB	PAR,BUF(AC2)		OSL06220

050ER	0A3D	627		AHR	AC2,ONE		OSL06
0510R	0A2D	628		AHR	AC1,ONE		OSL06
0512R	C520	629		CLHI	AC1,4		OSL06
	0004						
0516R	4280	630		BL	MA		OSL06
	0506R						
051AR	0301	631		BR	RTN		OSL06
051CR	C800	632	MSYM	LHI	PAR,X'20'	VALUE AND SYMBOL, MOVE SYMBOL (SIX	OSL06
	0020						
0520R	D203	633		STB	PAR,BUF(AC2)	BYTES) TO BUFFER FROM (NDX). NDX GETS	OSL06
	1244R						
0524R	0A3D	634		AHR	AC2,ONE	BUMPED FOR EACH BYTE, AC1 COUNTS DOWN	OSL06
0526R	082F	635		LHR	AC1,SIX	FROM 6 AND AC2 IS INCREMENTED TO THE	OSL06
0528R	D305	636	MA2	LB	PAR,0(NDX)	NEXT CHARACTER-POSITION OF BUF.	OSL06
	0000						
052CR	D203	637		STB	PAR,BUF(AC2)		OSL06
	1244R						
0530R	0A3D	638		AHR	AC2,ONE		OSL06
0532R	0A5D	639		AHR	NDX,ONE		OSL06
0534R	082D	640		SHR	AC1,ONE		OSL06
0536R	4230	641		BNZ	MA2	WHEN SYMBOL HAS BEEN MOVED, RESET NDX	OSL06
	0528R						
053AR	0B5F	642		SHR	NDX,SIX	TO BEGINNING OF TABLE-ENTRY. TEST AC2	OSL06
053CR	40D0	643		STH	ONE,MADA		OSL06
	1012R						
0540R	C530	644		CLHI	AC2,55	TO SEE IF A FULL LINE IS IN BUF, IF	OSL06
	0037						
0544R	4280	645		BL	NEWSYM	NOT, FETCH ANOTHER SYMBOL. WHEN A	OSL06
	04D6R						
0548R	C800	646		LHI	PAR,NEWLIN	LINE IS FILLED, RESET BUF INDEX (AC2)	OSL06
	04D4R						
054CR	4820	647	EMAP1	LH	AC1,MADA	TEST IF ANY HAVE BEEN LOGGED...	OSL06
	1012R						
0550R	4230	648		BNZ	EMAP15	IF NOT, PRINT "NONE"	OSL06
	0560R						
0554R	C820	649		LHI	AC1,NMS		OSL06
	1182R						
0558R	C830	650		LHI	AC2,NMS+4		OSL06
	1186R						
055CR	4300	651		B	EMAP16		OSL06
	0568R						
0560R	CA30	652	EMAP15	AHI	AC2,BUF-1	AT NEWLIN AND OUTPUT THE CURRENT LINE	OSL06
	1243R						
0564R	C820	653		LHI	AC1,BUF	THRU MLOG.	OSL06
	1244R						
0568R	4110	654	EMAP16	BAL	RTN,MLOG		OSL06
	0C76R						
056CR	0300	655		BR	PAR		OSL06
056ER	0866	656	EMAP2	LHR	KEY,KEY	WHEN TABLE IS EXHAUSTED, TEST KEY:	OSL06
0570R	4330	657		BZ	PPTP	IF KEY=0, OUTPUT PTOP TO SHOW HIGH	OSL06
	057ER						
0574R	0833	658	EMAP3	LHR	AC2,AC2	END OF LAST COMMON-BLOCK. OTHERWISE,	OSL06
0576R	0337	659		BZR	RTX	OUTPUT THE LAST LINE (IF A PARTIAL	OSL06
0578R	0807	660		LHR	PAR,RTX	LINE REMAINS) AND RETURN TO MAPPER	OSL06
057AR	4300	661		B	EMAP1	ROUTINE.	OSL06
	054CR						



7ER 4800	662	PPTP	LHI	PAR,PTOP		OSL06580
0FF4R						
82R 4110	663	PTP	BAL	RTN,WRTV		OSL06590
0500R						
86R 4300	664		B	EMAP3		OSL06600
0574R						
	665	*				OSL06610
8AR 4820	666	LOAD	LH	AC1,TOC	ON LOAD-COMMAND, PURGE LABEL-TABLE	OSL06620
0FEAR						
8ER 4820	667		SH	AC1,LABLEN	ADJUST SYMTAB TOP FOR LAB COM	OSL06630
102AR						
92R 4020	668		STH	AC1,LCMBOT	BELOW TOP OF CORE	OSL06640
1026R						
96R 4020	669		STH	AC1,LCMTOP	SET TOP OF LAB COM TO BOTTOM	OSL06650
1028R						
9AR 4020	670		STH	AC1,SBOT		OSL06660
0FF6R						
9ER 4820	671		SH	AC1,BLKLEN	ADJUST BOTTOM OF BLANK-COMMON	OSL06670
102CR						
9A2R 4020	672		STH	AC1,BCMBOT	TO PUT IT BELOW LABELED COMMON.	OSL06680
1022R						
9A6R 4020	673		STH	AC1,BCMTOP	SET TOP OF BLK COM	OSL06690
1024R						
9AAR 40C0	674		STH	ZRO,XFER	CLEAR TRANSFER-ADDRESS, ON OTHER	OSL06700
0FFCR						
9AER 40C0	675		STH	ZRO,LDFLG		OSL06710
123ER						
9B2R 40C0	676		STH	ZRO,RLFLG		OSL06720
1010R						
9B6R 40C0	677		STH	ZRO,CMFLG		OSL06730
1016R						
9BAR 40C0	678		STH	ZRO,OLOC		OSL06740
101AR						
9BER 40C0	679		STH	ZRO,PTOP	RESET POINTER TO TOP OF USER PROG	OSL06750
0FF4R						
9C2R 40C0	680		STH	ZRO,FWFLG		OSL06760
1014R						
9C6R 40C0	681		STH	ZRO,BCFLG		OSL06770
1018R						
	682	*		MODIFICATION TO SET ENTRY POINT TO OVERLAY CODE.		OSL06780
	683		IF	DOS		OSL06790
9CAR C810	684		LHI	RTN,OVLY	GET EP	OSL06800
0658R						
9CER 4010	685		STH	RTN,OVTAB+2	FOR FIRST TIME	OSL06810
1098R						
	686		IF	1		OSL06820
9D2R C810	687		LHI	RTN,SBOT		OSL06830
0FF6R						
9D6R 4010	688		STH	RTN,RESBOT+2		OSL06840
046CR						
9DAR 40C0	689	LINK	STH	ZRO,SKPFLG		OSL06850
1006R						
9DER 40C0	690		STH	ZRO,EDTFLG		OSL06860
0FF8R						
9E2R 4110	691		BAL	RTN,FINDER	POSITION TO NAMED PROGRAM, IF ANY	OSL06870
01F2R						

05E6R	4170	692		BAL	RTX,PRGLOD	CALL THE PROGRAM-LOADER...	OSL068
	0790R						
05EAR	4300	693		B	LOADER	ON COMPLETION, FETCH NEXT COMMAND.	OSL068
	0014R						
		694	*				
05EER	D200	695	EDIT	STB	PAR,BINDV+1	SET UP EDIT INPUT DEVICE	OSL069
	11FDR						OSL069
05F2R	D200	696		STB	PAR,RWD+1		OSL069
	121BR						
05F6R	E110	697		SVC	1,RWD		OSL069
	121AR						
05FAR	4810	698		LH	RTN,RWD+2		OSL069
	121CR						
05FER	4210	699		BM	INER2+4		OSL069
	0FAAR						
0602R	4110	700		BAL	RTN,FINDER		OSL069
	01F2R						
0606R	40F0	701		STH	SIX,EDTFLG	SET EDIT-FLAG	OSL069
	0FF8R						
060AR	086E	702	EDIT1	LHR	KEY,TWO	SET KEY AND SEARCH TABLE	OSL069
		708		IF	1		OSL070
060CR	4110	709		BAL	RTN,TSRCH	FOR ANY REFS. IF NO REFS THERE.	OSL070
	0CA0R						
0610R	4300	710		B	LOADER	END EDIT OPERATION NOW.	OSL070
	0014R						
0614R	4170	711		BAL	RTX,PRGLOD	IF ANY REFS, KEEP LOOKING.	OSL070
	0790R						
0618R	4300	712		B	EDIT1	IF NON-ZERO, SCAN SOME MORE.	OSL070
	060AR						

		714	*				OSL07100
		715	*				OSL07110
		716	*	SET TOP-OF-CORE FOR OOUT MODULE			OSL07120
		717	*				OSL07130
		718	TOPCOR	CLHI	PAR,8190	ADDR < BK ?	OSL07140
01CR	C500						
	1FFE						
020R	4280	719		HL	CMDERR	YES - ERROR	OSL07150
	010AR						
024R	4840	720		LH	ARG,OUTFLG	OUT MODE ?	OSL07160
	OFFAR						
028R	4330	721		BZ	CMDERR	NO - ERROR	OSL07170
	010AR						
02CR	4840	722		LH	ARG,OLOC	LOADING STARTED ?	OSL07180
	101AR						
030R	4230	723		BNZ	CMDERR	YES - ERROR	OSL07190
	010AR						
034R	4000	724		STH	PAR,TOC	SET TOP-OF-CORE	OSL07200
	OFFEAR						
038R	4300	725		B	LOADER	B	OSL07210
	0014R						
03CR	E130	726	RETURN	SVC	3,0	RETURN TO OS BY SVC END-OF-JOB.	OSL07220
	0000						



59CR	4020	763		STH	AC1,SBOT	TO REMOVE ANY SIGN OF OVERLAY	OSL07590
	OFF6R						
5A0R	4820	764		LH	AC1,BIASV		OSL07600
	06F4R						
5A4R	4020	765		STH	AC1,PTOP		OSL07610
	OFF4R						
5A8R	4850	766		LH	NDX,STOP		OSL07620
	1026R						
5ACR	086D	767	D1	LHR	KEY,ONE		OSL07630
5AER	4110	768	D2	BAL	RTN,CSRCH		OSL07640
	0CA4R						
5B2R	4300	769		B	L1		OSL07650
	06CER						
5B6R	4805	770		LH	PAR,6(NDX)		OSL07660
	0006						
5BAR	4500	771		CLH	PAR,BIASV		OSL07670
	06F4R						
5BER	4280	772		BL	D2		OSL07680
	06AER						
5C2R	086D	773		LHI	KEY,7		OSL07690
	0007						
5C6R	4110	774		BAL	RTN,TINSRT		OSL07700
	0D50R						
5CAR	4300	775		B	D1		OSL07710
	06ACR						
5CER	4850	776	L1	LH	NDX,STOP		OSL07720
	1026R						
5D2R	0766	777	L2	XHR	KEY,KEY		OSL07730
5D4R	4110	778	L3	BAL	RTN,CSRCH		OSL07740
	0CA4R						
5D8R	4300	779		B	LOADER		OSL07750
	0014R						
5DCR	4805	780		LH	PAR,6(NDX)		OSL07760
	0006						
5E0R	4500	781		CLH	PAR,BIASV		OSL07770
	06F4R						
5E4R	4280	782		BL	L3		OSL07780
	06D4R						
5E8R	086D	783		LHI	KEY,7		OSL07790
	0007						
5ECR	4110	784		BAL	RTN,TINSRT		OSL07800
	0D50R						
5FOR	4300	785		B	L2		OSL07810
	06D2R						
5F4R		786	BIASV	DS	1		OSL07820
5F6R		787	SROTSV	DS	1		OSL07830
168		788	DEFAULT	EQU	360		OSL07840
		1226		IF	1		OSL12220
		1227	*				OSL12230
		1228	* BUBBLE SORT SUBROUTINE				OSL12240
		1229	*				OSL12250
		1230	* SORTS THE SYMBOL TABLE FROM (SBOT) TO NSTOP)				OSL12260
		1231	*				OSL12270
		1232	* IF NAFLAG) = 0 SORTS BASED ON ADDRESS				OSL12280
		1233	* OTHERWISE SORTS ALPHABETICALLY				OSL12290
		1234	*				OSL12300

		1235	* LINKAGE :	BAL RTN, BUBBLE		OSL123
		1236	*			OSL123
		1237	* NOTE REGISTER REDIFINITIONS :			OSL123
0002		1238	ACJ	EQU AC1		OSL123
0003		1239	ACK	EQU AC2		OSL123
000C		1240	X1	EQU 12		OSL123
000D		1241	X2	EQU 13		OSL123
000E		1242	X3	EQU 14		OSL123
000F		1243	X4	EQU 15		OSL123
		1244	*			OSL123
06F8R	4830	1245	BUBBLE	LHI ACK,SBOT	INITIALIZE K	OSL124
	0FF6R					OSL124
06FCR	C860	1246		LHI KEY,X'7FFF'		OSL124
	7FFF					
0700R	CA30	1247		AHI ACK,8		OSL124
	0008					
0704R	0823	1248	BUBBL2	LHR ACJ,ACK	J = K	OSL124
0706R	D1C2	1249	BUBBL3	LM X1,0(ACJ)	GET ENTRY	OSL124
	0000					
070AR	4840	1250		LH ARG,AFLAG	AFLAG = 0 ?	OSL124
	078ER					
070ER	4230	1251		BNZ CMPALF		OSL124
	071ER					
0712R	45F2	1252		CLH X4,-2(ACJ)	YES - X(J) = ADDRESS	OSL124
	FFFE					
0716R	4280	1253		BL BUBBL4		OSL124
	0750R					
071AR	4300	1254		B BUBBL7		OSL125
	075ER					
071ER	04C6	1255	CMPALF	NHR X1,KEY	NO - X(J) = SYMBOL (ANDED W KEYS)	OSL125
0720R	4842	1256		LH ARG,-8(ACJ)		OSL125
	FFFA					
0724R	0446	1257		NHR ARG,KEY		OSL125
0726R	05C4	1258		CLHR X1,ARG		OSL125
0728R	4280	1259		BL BUBBL4		OSL125
	0750R					
072CR	4230	1260		BNE BUBBL7		OSL125
	075ER					
0730R	04D6	1261		NHR X2,KEY		OSL125
0732R	4842	1262		LH ARG,-6(ACJ)		OSL125
	FFFA					
0736R	0446	1263		NHR ARG,KEY		OSL125
0738R	05D4	1264		CLHR X2,ARG		OSL126
073AR	4280	1265		BL BUBBL4		OSL126
	0750R					
073ER	4230	1266		BNE BUBBL7		OSL126
	075ER					
0742R	04E6	1267		NHR X3,KEY	X(J) > X(J-,) ?	OSL126
0744R	4842	1268		LH ARG,-4(ACJ)		OSL126
	FFFC					
0748R	0446	1269		NHR ARG,KEY		OSL126
074AR	05E4	1270		CLHR X3,ARG		OSL126
074CR	4380	1271		BNL BUBBL7		OSL126
	075ER					
0750R	CA30	1272	BUBBL4	AHI ACK,8		OSL126
	0008					

754R	4530	1273		CLH	ACK,STOP	K = (STOP) ?	OSL12690
	1026R						
758R	4280	1274		BL	BUBBL2		OSL12700
	0704R						
75CR	0301	1275		BR	RTN	YES - RETURN	OSL12710
75ER	D1C2	1276	BUBBL7	LM	X1,0(ACJ)	EXCHANGE FULL ENTRY (J,J-8)	OSL12720
	0000						
762R	D0C0	1277		STM	X1,SAVE		OSL12730
	0786R						
766R	D1C2	1278		LM	X1,-8(ACJ)		OSL12740
	FFF8						
76AR	D0C2	1279		STM	X1,0(ACJ)		OSL12750
	0000						
76ER	D1C0	1280		LM	X1,SAVE		OSL12760
	0786R						
772R	D0C2	1281		STM	X1,-8(ACJ)		OSL12770
	FFF8						
776R	CB20	1282		SHI	ACJ,8		OSL12780
	0008						
77AR	4520	1283		CLH	ACJ,SBOT		OSL12790
	0FF6R						
77ER	4320	1284		BNP	BUBBL4		OSL12800
	0750R						
782R	4300	1285		B	BUBBL3	B	OSL12810
	0706R						
786R		1286	SAVE	DS	8		OSL12820
78ER	0000	1287	AFLAG	DC	0		OSL12830

	1289	*					OSL128	
	1290	*	PROGRAM-LOADER				OSL128	
	1291	*					OSL128	
	1292	*					OSL128	
	1293	*	THIS ROUTINE IS CALLED BY COMMAND-IMPLEMENTATION ROUTINES TO LOAD					OSL128
	1294	*	PROGRAMS, OR OUTPUT CORE-IMAGE TAPES.					OSL128
	1295	*					OSL128	
	1296	*	INPUT DEVICE IS SET BY CALLER AT BINDV BEFORE CALLING THIS MODULE.					OSL128
	1297	*					OSL128	
0790R	089D		PRGLOD	LHR	SEQ,ONE	INITIALIZE SEQ NUMBER TO 1	OSL128	
				IF	1		OSL128	
0792R	4820		LH	AC1,BIAS		FORCE INITIAL VALUE OF BIAS	OSL128	
	0FF2R						OSL128	
0796R	4020		STH	AC1,LOC		INTO LOC.	OSL130	
	1008R						OSL130	
079AR	40D0		STH	ONE,RELFLG		SET RELATIVE MODE	OSL130	
	101CR						OSL130	
079ER	4110		CRNR	BAL	RTN,LREAD		OSL130	
	0F30R						OSL130	
07A2R	4300		B	INER		IF INPUT ERROR, GO TO INER	OSL130	
	0F98R						OSL130	
07A6R	0B9D		BSEQ	SHR	SEQ,ONE	THIS STARTS AT -1 ON TAPE. WHEN SEQ	OSL130	
07A8R	4230			BNZ	CSEQ	IS ZERO, TEST FLAG, IF THIS IS	OSL130	
	0812R						OSL130	
07ACR	4820		LH	AC1,EDTFLG		NOT AN EDIT OPERATION, OR IF IT IS	OSL130	
	0FF8R						OSL130	
07B0R	4330		BZ	NSKP		BUT NO LABEL APPEARS AS THE FIRST	OSL130	
	07C6R						OSL130	
07B4R	0788		XHR	ITM,ITM		ITEM OF THE RECORD, THEN LOAD THE	OSL130	
07B6R	4110		BAL	RTN,FETCH		RECORD. RESET SKPFLG AND AND	OSL130	
	08EER						OSL130	
07BAR	C500		CLHI	PAR,x'0F'		GO BACK TO BSEQ TO DECREMENT DEQ TO	OSL131	
	000F						OSL131	
07BER	4330		BE	LABTST		-1 FOR TESTING FIRST RECORD.	OSL131	
	07CEK						OSL131	
07C2R	4300		B	SKPPRG		UNLABELED PROGRAMS ARE SKIPPED	OSL131	
	080AR						OSL131	
07C6R	40C0		NSKP	STH	ZRO,SKPFLG		OSL131	
	1006R						OSL131	
07CAR	4300		B	BSEQ		TO SCAN AND TEST LABEL (FOR EDIT CMD)	OSL131	
	07A6R						OSL131	
07CER	086E		LABTST	LHR	KEY,TWO	SET SEARCH-KEY TO 2 TO LOOK FOR REFS	OSL131	
				IF	1		OSL132	
07D0R	4110		BAL	RTN,SFETCH		MOVE SYMBOL FORM TAPE TO SYMBUF	OSL132	
	0C1AR						OSL132	
07D4R	4820		LH	AC1,SYMBUF		TEST FOR LABEL "ENDVOL"	OSL132	
	11B0R						OSL132	
07D8R	C520		CLHI	AC1,C'EN'		IF NOT, GO ON	OSL132	
	454E						OSL132	
07DCR	4230		BNE	EDTSCN		IF SO, STOP EDITING.	OSL132	
	0800R						OSL132	
07E0R	4820		LH	AC1,SYMBUF+2			OSL132	
	11B2R						OSL132	
07E4R	C520		CLHI	AC1,C'DV'			OSL132	
	4456						OSL132	
07E8R	4230		BNE	EDTSCN			OSL132	



7ECR	0800R 4820	1333	LH	AC1,SYMBUF+4		OSL13290
7FOR	11B4R C520	1334	CLHI	AC1,C*OL'		OSL13300
7F4R	4F4C 4230	1335	UNE	EDTSCN		OSL13310
7F8R	0800R E120	1336	SVC	2,MEOFS	EOF MESSAGE	OSL13320
7FCR	1124R 4300 0014R	1337	B	LOADER		OSL13330
800R	4110	1338	*			OSL13340
804R	GCDAR 0855	1339	EDTSCN	BAL	RTN,SFIND	FIND A MATCHING REG
806R	4230	1340	LHR	NDX,NDX	IF NONE, SKIP THIS PROG	OSL13360
80AR	07C6R 40F0	1341	UNZ	NSKP	ELSE LOAD	OSL13370
80ER	1006R 4300 07A6R	1342	SKPPRG	STH	SIX,SKPFLG	OSL13380
812R	4590	1343	B	BSEQ		OSL13390
816R	1244R 4330	1344	*			OSL13400
81AR	0822R E120	1345	CSEQ	CLH	SEQ,BUF	TEST SEQUENCE NUMBER
81ER	1196R 4300 0F80R	1346	BE	DECODE	OKAY? THEN START PROCESSING RECORD	OSL13420
822R	0788	1347	SVC	2,MSSEQ	NO? THEN OUTPUT MESSAGE AND EXIT	OSL13430
824R	4110	1348	B	CRN2	WITH PAUSE FOR RE-TRY	OSL13440
828R	0BEER CU00	1349	DECODE	XHR	ITM,ITM	START NEW RECORD HERE
82CR	0002 0820	1350	NEWITM	BAL	RTN,FETCH	FETCH NEXT CONTROL-ITEM
82ER	4802	1351		SLHL	PAR,2	OSL13450
832R	109AR C810	1352	DEC2	LH	PAR,CITTAB(AC1)	OSL13470
836R	0824R 4830	1353	LH	PAR,CITTAB(AC1)	FETCH ADRS OF ITEM-SERVICE ROUTINE	OSL13480
83AR	1006R 0330	1354	LHI	RTN,NEWITM	SET RETURN ADDRESS	OSL13490
83CR	4832	1355	LH	AC2,SKPFLG	TEST SKIP-FLAG IF RESET,	OSL13500
840R	109CR 0210	1356	BZR	PAR	EXIT TO SERVICE-ROUTINE.	OSL13510
842R	0A83	1357	LH	AC2,CITTAB+2(AC1)	IF SET, FETCH DATA-COUNT, IF DATA-	OSL13520
844R	4300	1358	B	NEWITM	COUNT NEGATIVE, EXIT TO ROUTINE.	OSL13530
848R	0824R 4110	1359	BMR	PAR	OTHERWISE, BUMP ITEM.	OSL13540
850R	0HEER C500	1360	AHR	ITM,AC2		OSL13550
	0005 4280	1361	B	NEWITM		OSL13560
	0860R	1362	* CCOM	USED TO DECODE SECOND DIGIT IF FIRST IS E		OSL13570
		1363	CCUM	BAL	RTN,FETCH	OSL13580
		1364	CLHI	PAR,5	TEST FOR LEGAL CONTROL ITEM	OSL13590
		1365	BL	CCOM2	IF ALL IS WELL, VECTOR IT	OSL13600

0854R	E120 11D4R	1365	INVLOD	SVC	2,ICM	ABORT LOAD	OSL130	
0858R	E120 11E2R	1366		SVC	2,AHLM		OSL130	
085CR	4300 0FCER	1367		B	CKDEV		OSL130	
0860R	CD00 0002	1368	CCOM2	SLHL	PAR,2		OSL130	
0864R	C820 10DAR	1369		LHI	AC1,COMTAB	SET UP AC1 TO VECTOR DECODE-ROUTINE	OSL130	
0868R	C820 109AR	1370		SHI	AC1,CITTAB	TO COMTAB INSTEAD OF CITAB BY ADDING	OSL130	
086CR	0A20	1371		AHR	AC1,PAR	TO IT THE LENGTH OF CITTAB.	OSL130	
086ER	4300 082ER	1372		B	DEC2		OSL130	
		1373	* END OF PROGRAM					OSL130
0872R	4820 0FF4R	1374	CEND	LH	AC1,PTOP	FETCH LAST LOC USED	OSL130	
0876R	4330 087ER	1375		BZ	GOFET	BRANCH IF 0	OSL130	
		1381		IF	1		OSL130	
087AR		1382	GOEND	EQU	*		OSL130	
087AR	4020 0FF2R	1383		STH	AC1,BIAS	AND UPDATE BIAS FOR LOADING	OSL130	
087ER		1384	GOFET	EQU	*		OSL130	
087ER	4110 0C54R	1385		BAL	RTN,DFETCH	GET PTOP IF ITS ON TAPE	OSL130	
0882R	0844	1386		LHR	ARG,ARG	IF ITS ZERO, THEN SKIP IT	OSL130	
0884R	4330 0894R	1387		BZ	USET	AND USE VALUES COMPUTED IN LOADER	OSL130	
		1391		IF	1		OSL130	
0888R	4040 0FF4R	1392		STH	ARG,PTOP	IF NON-ZERO, THIS IS A LOAD-MODULE	OSL130	
088CR		1393	GOFET2	EQU	*		OSL130	
088CR	4110 0C54R	1394		BAL	RTN,DFETCH	SO FETCH COMBOT FROM INPT, TOO	OSL130	
0890R	4040 1022R	1395		STH	ARG,COMBOT	NOW PTOP AND COMBOT ARE SETUP.	OSL130	
0894R	4820 0FFAR	1396	USET	LH	AC1,OUTFLG	UNLESS THIS WAS AN OUTPUT JOB.	OSL130	
0898R	0237	1397		BNZR	RTX	SET UP THE PARAMETERS IN BOSS.	OSL130	
		1401		IF	1		OSL130	
089AR	E120 0FEER	1402		SVC	2,FETCHP	GET TABLE-ADRS INTO AC1	OSL130	
089ER	4830 0FF4R	1403		LH	AC2,PTOP	MOVE PTOP INTO TABLE	OSL130	
08A2R	4032 0004	1404		STH	AC2,4(AC1)		OSL140	
08A6R	4830 1022R	1405		LH	AC2,COMBOT	MOVE COMBOT INTO TABLE	OSL140	
08AAR	4032 0002	1406		STH	AC2,2(AC1)		OSL140	
08AER	0307	1407		BK	RTX	AND EXIT LOADER TO COM-PROCESSOR	OSL140	
		1408	* DEFINE CHAIN					OSL140
08BOR	4820 0FFAR	1409	CDCN	LH	AC1,OUTFLG	ON CHAIN ITEM, TEST OUT FLAG.	OSL140	

0B4R	1410	CDCN0	EQU	*		OSL14060
0B4R 4230	1411		BNZ	CDCN2	IF OUT, GO OUTPUT SOMETHING	OSL14070
08D2R						
0B8R 482B	1412	CDCN1	LH	AC1,0(REF)	OTHERWISE, DEF THE CHAIN: FETCH END	OSL14080
0000						
0BCR 40AB	1413		STH	DEF,0(REF)	LOCATION. THEN FOLLOW REF	OSL14090
0000						
0C0R 08B2	1414		LHR	REF,AC1	CHAIN UNLESS NEXT REF-LOC	OSL14100
0C2R 0331	1415		HZR	RTN	IS ZERO.	OSL14110
0C4R 05BA	1416		CLHR	REF,DEF	IF REF = DEF, CHAIN IS LOOPEO	OSL14120
0C6R 4230	1417		BNE	CDCN1	ELSE, CONTINUE	OSL14130
08B8R						
0CAR E120	1418	CDCN7	SVC	2,MSSLOP	IF CHAIN-LOOP IS DETECTED, CANCEL	OSL14140
11A2R						
0CER 4300	1419		B	CMDERR	THE OPERATION AND TYPE A MESSAGE.	OSL14150
010AR						
	1420			* OUTPUT REF, DEF, CHAIN		OSL14160
0D2R 4010	1421	CDCN2	STH	RTN,CDCN3	SAVE RETURN ADDRESS	OSL14170
08F8R						
0D6R 082B	1422		LHR	AC1,REF		OSL14180
0D8R 080F	1423		LHR	PAR,SIX		OSL14190
0DAR 4110	1424		BAL	RTN,OHW	PACK OUT SIX, REF	OSL14200
0E94R						
0DER 082A	1425		LHR	AC1,DEF		OSL14210
0E0R C800	1426		LHI	PAR,7		OSL14220
0007						
0E4R 4110	1427		BAL	RTN,OHW	PACK OUT 7, DEF	OSL14230
0E94R						
0E8R C840	1428		LHI	ARG,206		OSL14240
00CE						
0ECR 4110	1429		BAL	RTN,OTST		OSL14250
0EAA R						
0FOR 080E	1430		LHR	PAR,TWO		OSL14260
0F2R 4110	1431		BAL	RTN,OPAK	PACK OUT 2 (DEF-CHAIN)	OSL14270
0E1CR						
0F6R 4300	1432		B	*		OSL14280
08F6R						
0F8R	1433	CDCN3	EQU	*-2		OSL14290
	1434			* TOGGLE ABS/REL MODE		OSL14300
0FAR 4820	1435	CTOG	LH	AC1,LOC+2	ON MODE-TOGGLE ITEM, SWAP LOCATION-	OSL14310
100AR						
	1439		IF	1		OSL14350
0FER 4830	1440		LH	AC2,LOC	COUNTERS AND FLIP RELFLG	OSL14360
1008R						
002R 4020	1441		STH	AC1,LOC		OSL14370
1008R						
006R 4030	1442		STH	AC2,LOC+2		OSL14380
100AR						
00AR 4820	1443		LH	AC1,RELFLG		OSL14390
101CR						
00ER 4330	1444		BZ	CTOG2		OSL14400
091AR						
012R 082C	1445		LHR	AC1,ZRO		OSL14410
014R 4020	1446	CTOG1	STH	AC1,RELFLG		OSL14420
101CR						
018R 0301	1447		BR	RTN		OSL14430

091AR	C820	1448	CTOG2	LHI	AC1,7		OSL144
	0007						
091ER	4300	1449		B	CTOG1		OSL144
	0914R						
		1450	* LOAD		TRANSFER ADDRESS, LOAD-ADDRESS, REF/DEF ADDRESS		OSL144
0922R	4110	1451	CTRA	BAL	RTN,RFETCH		OSL144
	0C44R						
0926R	4040	1452		STH	ARG,XFER	MOVE TRANSFER ADRS TO XFER	OSL144
	OFFCR						
092AR	4810	1453		LH	RTN,OUTFLG	IF OUTFLAG IS SET, PACK XFER ADRS.	OSL144
	OFFAR						
092ER	4330	1454		BZ	NEWITM	ON OUTPUT FILE	OSL145
	0824R						
0932R	C800	1455		LHI	PAR,4		OSL145
	0004						
0936R	0824	1456		LHR	AC1,ARG		OSL145
0938R	4110	1457		BAL	RTN,OHW		OSL145
	0E94R						
093CR	4300	1458		B	NEWITM		OSL145
	0824R						
0940R	4110	1459	CLDA	BAL	RTN,RFETCH		OSL145
	0C44R						
		1466		IF	1		OSL146
0944R	4040	1467		STH	ARG,LOC	MOVE LOAD-ADRS TO LOC-COUNTER	OSL146
	1008R						
0948R	4540	1468		CLH	ARG,PTOP	SEE IF THIS WILL BUMP PTOP	OSL146
	OFF4R						
094CR	4280	1469		BL	NEWITM	IF NOT, GO BACK FOR MORE	OSL146
	0824R						
0950R	4040	1470		STH	ARG,PTOP	OTHERWISE, UPDATE PTOP	OSL146
	OFF4R						
0954R	4300	1471		B	NEWITM		OSL146
	0824R						
0958R	4110	1472	CREF	BAL	RTN,RFETCH		OSL146
	0C44R						
095CR	08B4	1473		LHR	REF,ARG	FETCH REF ADDRESS	OSL146
		1491		IF	1-OS16MT		OSL148
095ER	4300	1492		B	NEWITM		OSL148
	0824R						
		1493		IF	1		OSL148
0962R	4110	1494	CDEF	BAL	RTN,RFETCH		OSL149
	0C44R						
0966R	08A4	1495		LHR	DEF,ARG	FETCH DEF ADDRESS	OSL149
0968R	4300	1496		B	NEWITM		OSL149
	0824R						
		1497	* LOAD	TEXT			OSL149
096CR	4110	1498	C1AB	BAL	RTN,DFETCH	FETCH A HALFWORD OF DATA	OSL149
	0C54R						
0970R	4850	1499	C1AB1	LH	NDX,LOC	AND GET LOAD-LOCATION FROM LOC.	OSL149
	1008R						
0974R	4550	1500		CLH	NDX,COMBOT	TEST FOR OVERLAP WITH COMMON	OSL149
	1022R						
0978R	4380	1501		BNL	MFULL		OSL149
	0EF8R						
097CR	0824	1502		LHR	AC1,ARG	CALL MHW TO MOVE THE DATA INTO CORE	OSL149
097ER	4110	1503		BAL	RTN,MHW	OR PACK IT INTO THE OUTPUT BUFFER	OSL149

0084R							
82R 4300	1504		B	C2AB2			OSL15000
09C8R							
86R 4110	1505	C1RL	BAL	RTN,DFETCH	IF DATA IS RELOCATABLE, RELOCATE IT,		OSL15010
0C54R							
8AR 4A40	1506		AH	ARG,BIAS	BY ADDING BIAS		OSL15020
0FF2R							
8ER 4300	1507		B	C1AB1	BUT OUTPUT IS ONLY ABSOLUTE.		OSL15030
0970R							
	1508	*					OSL15040
92R 4000	1509	C2RL	STH	ONE,RLFLG			OSL15050
1010R							
96R 4110	1510	C2AB	BAL	RTN,DFETCH	FOR FULLWORD,SAME AS ABOVE.		OSL15060
0C54R							
9AR 4040	1511		STH	ARG,TMP1			OSL15070
1002R							
9ER 4110	1512		BAL	RTN,DFETCH	FETCH SECOND HW OF FW		OSL15080
0C54R							
9A2R 4820	1513		LH	AC1,RLFLG	TEST RELOCATION FLAG		OSL15090
1010R							
9A6R 4330	1514		BZ	**8	IF ABSOLUTE, SKIP		OSL15100
09AER							
9AAR 4A40	1515		AH	ARG,BIAS	ADDING THE BIAS		OSL15110
0FF2R							
9AER 40C0	1516		STH	ZRO,RLFLG	RESET RL FLAG		OSL15120
1010R							
9B2R 0834	1517		LHR	AC2,ARG			OSL15130
9B4R 4820	1518		LH	AC1,TMP1			OSL15140
1002R							
9B8R 4850	1519		LH	NDX,LOC			OSL15150
1008R							
9BCR 4550	1520		CLH	NDX,COMBOT			OSL15160
1022R							
9C0R 4380	1521		BNL	MFULL			OSL15170
0LF8R							
9C4R 4110	1522		BAL	RTN,MFW			OSL15180
0D80R							
9C8R 4050	1523	C2AB2	STH	NDX,LOC	UPDATE LOC AND PTOP		OSL15190
1008R							
9CCR 4550	1524		CLH	NDX,PTOP	AFTER MOVING DATA		OSL15200
0FF4R							
9D0R 4280	1525		BL	NEWITM			OSL15210
0824R							
9D4R 4050	1526		STH	NDX,PTOP			OSL15220
0FF4R							
9D8R 4300	1527		B	NEWITM			OSL15230
0824R							
	1528	* LOAD REF SYMBOL					OSL15240
9DCR 4110	1529	CRFS	BAL	RTN,SFETCH	MOVE SYMBOL TO SYMBUF		OSL15250
UC1AR							
	1533		IF	1			OSL15290
9E0R 0860	1534		LHR	KEY,ONE			OSL15300
9E2R 4110	1535		BAL	RTN,SFIND	FIND A MATCHING DEF (IF ANY)		OSL15310
0CDAR							
9E6R 0855	1536		LHR	NDX,NDX	IF MATCHING DEF EXISTS, GO DEFINE IT.		OSL15320
9E8R 4230	1537		BNZ	CRFS1			OSL15330

09E2R	0A2CR							
09E2R	086E	1538		LHR	KEY,TWO			
09E2R	482D	1539		LH	AC1,OUTFLG			OSL15
	UFFAK							OSL15
09F2R	4230	1540		BNZ	CRFS0			OSL15
	0A22R							
09F6R		1541	CRFLNK	EQU	*			OSL15
09F6R	4110	1542		BAL	RTN,SFIND			OSL15
	0CDAR							
09FAR	0855	1543		LHR	NDX,NDX			OSL15
09FCR	4330	1544		BZ	CRFS0			OSL15
	0A22R							
0A00R	0711	1545		XHR	RTN,RTN			OSL15
0A02R	4825	1546		LH	AC1,6(NDX)			OSL15
	0006							OSL15
0A06R	4832	1547	CP1	LH	AC2,0(AC1)			OSL15
	0000							
0A0AR	4330	1548		BZ	CR2			OSL15
	0A1AR							
0A0ER	0823	1549		LHR	AC1,AC2			OSL15
0A10R	0A1D	1550		AHR	RTN,ONE			OSL15
0A12R	4310	1551		BNM	CR1			OSL15
	0A06R							
0A16R	4300	1552		B	CUCN7			OSL15
	08CAR							
0A1AR	40B2	1553	CR2	STH	REF,0(AC1)			OSL15
	0000							
0A1LR	4300	1554		B	NEWITM			OSL15
	0824R							
0A22R	080B	1555	CRFS0	LHR	PAR,REF			OSL15
0A24R	4110	1556		BAL	RTN,SINSRT			OSL15
	0D18R							
0A28R	4300	1557		B	NEWITM			OSL15
	0824R							
0A2CR	46A5	1558	CRFS1	LH	DEF,6(NDX)			OSL15
	0006							
0A30R	4110	1559		BAL	RTN,CDCN			OSL15
	08B0R							
0A34R	4300	1560		B	NEWITM			OSL15
	0824R							
		1576		IF	1			
		1577	* LOAD	DEF	SYMBOL			OSL15
0A38R	4110	1578	CDFS	BAL	RTN,SFATCH			OSL15
	0C1AR							OSL15
0A3CR	086D	1579		LHR	KEY,ONE			OSL15
0A3ER	4110	1580		BAL	RTN,SFIND			OSL15
	0CDAR							
		1584		IF	1			OSL15
0A42R	0855	1585		LHR	NDX,NDX			OSL15
0A44R	4330	1586		BZ	CDFS2			OSL15
	0A6AR							
0A48R	0830	1587		LHI	AC2,4			OSL15
	0004							
0A4CR	4823	1588	CDFS1	LH	AC1,SYMBUF(AC2)			OSL15
	11B0R							
0A50R	4023	1589		STH	AC1,MSYMB(AC2)			OSL15

IF NO MATCH, INSERT THIS REF IN TABLE AT FIRST AVAILABLE SLOT.

IF MATCH AND ITS A REF ALSO.

LINK TWO THREADS TOGETHER.

FETCH REF-VALUE

GET LINK OF CHAIN TO FOLLOW THREAD.

IF ZERO, WE'RE AT THE END - GO STORE

NEW REF-VALUE. OTHERWISE, BUMP CHAIN-COUNTER, AND LOOP AROUND AGAIN. IF COUNTER OVERFLOWS, THREAD

MUST BE AN ENDLESS LOOP STO SKIP IT.

IF MATCH, FETCH DEF-VALUE AND GO

DEFINE THE CHAIN

MOVE DEF SYMBOL TO SYMBUF

LOOK FOR ANOTHER DEF SAME SYMBOL THIS IS A MULTIPLY-DEFINED ENTRY-

POINT.

IF ANOTHER DEF FOUND, MOVE SYMBOL

TO MESSAGE-AREA AND LOG M MESSAGE.

TYPE IT OUT, BUT

11C2R						
154R 0B3E	1590		SHR	AC2,TWO	IGNORE THIS CONTROL-ITEM COMPLETELY.	OSL15860
156R 4310	1591		BNM	CDFS1		OSL15870
0A4CP						
15AR E110	1592		SVC	1,MSMDF		OSL15880
11B6R						
15ER 4800	1593		LH	PAR,MSMDF+2		OSL15890
11B8R						
162R 4210	1594		BM	INER2+4		OSL15900
0FAAR						
166R 4300	1595		B	NEWITM		OSL15910
0824R						
16AR 080A	1596	CDFS2	LHR	PAR,DEF	IF NOT MULTI-DEFINED, ADD IT TO TABLE	OSL15920
16CR 4110	1597		BAL	RTN,SINSRT	AS A DEF AND THEN LOOK FOR MATCHING	OSL15930
0D18R						
170R 0766	1598		XHR	KEY,KEY	NAME. IF WE FIND ONE	OSL15940
172R 4110	1599		BAL	RTN,SFIND		OSL15950
0CDAR						
176R 0855	1600		LHR	NDX,NDX		OSL15960
178R 4330	1601		BZ	CDFGO	SET ITS ADDRESS TO REF	OSL15970
0A88R						
17CR 45A5	1602		CLH	DEF,6(NDX)		OSL15980
0006						
180R 4380	1603		BNL	CDFGO		OSL15990
0A88R						
184R 40A5	1604		STH	DEF,6(NDX)	AND GO LOOK FOR MATCHING	OSL16000
0006						
188R 4850	1605	CDFGO	LH	NDX,STOP	REFS.	OSL16010
1026R						
18CR 086E	1606	CDFS3	LHR	KEY,TWO		OSL16020
18ER 4110	1607		BAL	RTN,CFIND		OSL16030
0CDER						
192R 0855	1608		LHR	NDX,NDX		OSL16040
194R 4330	1609		BZ	NEWITM	NO MORE REFS, WE'RE DONE]	OSL16050
0824R						
198R 48B5	1610		LH	REF,6(NDX)	REF FOUND, FETCH VALUE	OSL16060
0006						
19CR C860	1611		LHI	KEY,7	CHANGE TABLE-ENTRY TO NULL (KEY=7)	OSL16070
0007						
1A0R 4110	1612		BAL	RTN,TINSRT	TO BE REPLACED BY ANOTHER SINSRT...	OSL16080
0D50R						
1A4R 4110	1613		BAL	RTN,CDCN	DEFINE THIS REF'S CHAIN	OSL16090
08B0R						
1A8R 4300	1614		B	CDFS3		OSL16100
0A8CR						
	1653		IF	1		OSL16490
	1654	* LOG	PROGRAM-LABEL			OSL16500
1ACR 4110	1655	CLAB	BAL	RTN,SFETCH	MOVE PROGRAM-LABEL TO SYMBUS	OSL16510
0C1AR						
1B0R 4800	1656		LH	PAR,BIAS	VALUE IS CURRENT BIAS	OSL16520
0FF2R						
1B4R 0766	1657		XHR	KEY,KEY	KEY IS ZERO	OSL16530
1B6R 4110	1658		BAL	RTN,SINSRT	INSERT PROGRAM LABEL	OSL16540
0D18R						
1BAR 4300	1659		B	NEWITM		OSL16550
0824R						

Address	Block	Code	Op	Op2	Description	OSL168
	1660		*	DEFINE COMMON-BLOCK		
OABER 4110 OC1AR	1661	CMDEF	BAL	RTN,SFETCH	FETCH SYMBOL (NAME OF BLOCK)	OSL168 OSL168
OAC2R 4110 OC54R	1662		BAL	RTN,DFETCH	FETCH BLOCK-LENGTH	OSL168
OAC6R 4040 1004R	1663		STH	ARG,TMP2		OSL168
OACAR C860 0003	1664		LHI	KEY,3		OSL168
OACER 4110 OCDAR	1665		BAL	RTN,SFIND	IF THIS BLOCK IS ALREADY DEFINED.	OSL168
OAD2R 0855	1666		LHR	NDX,NDX	IGNORE THIS DEFINITION AND LET PREV-	OSL168
OAD4R 4230 0824R	1667		BNZ	NEWITM	IOUSLY-DEFINED BLOCK STAND.	OSL168
OAOBR C820 2F2F	1668		LHI	AC1,C'///'	IF NAME OF BLOCK IS	OSL168
OADCR 4520 1180R	1669		CLH	AC1,SYMBUF	/// " SET BCFLG AND FETCH	OSL168
OAEOR 4230 0B04R	1670		BNE	CMD1	TOC RATHER THAN COMTOP TO DEFINE	OSL168
OAE4R C820 2020	1671		LHI	AC1,C' '	THE BLOCK. THIS PUTS BLANK COMMON	OSL168
OAE8R 4520 1182R	1672		CLH	AC1,SYMBUF+2	AT THE TOP OF CORE WITH ROOM TO	OSL168
OAECR 4230 0B04R	1673		BNE	CMD1	GROW LATER IN THE LOAD.	OSL168
OAFOR 4520 1184R	1674		CLH	AC1,SYMBUF+4		OSL167
OAF4R 4230 0B04R	1675		BNE	CMD1		OSL167
OAF8R 40E0 1018R	1676		STH	TWO,BCFLG		OSL167
OAFCR 4800 1022R	1677		LH	PAR,BCMBOT		OSL167
OB00R 4300 0B08R	1678		B	CMD2		OSL167
OB04R 4800 1028R	1679	CMD1	LH	PAR,LCMTOP		OSL167
OB08R 4000 1002R	1680	CMD2	STH	PAR,TMP1		OSL167
OB0CR 4110 0D18R	1681		BAL	RTN,SINSRT	OTHERWISE, INSERT SYMBOL	OSL167
OB10R 4800 1002R	1682		LH	PAR,TMP1		OSL167
OB14R 4A00 1004R	1683		AH	PAR,TMP2	AND BUMP TOP OF COMMON	OSL167
OB18R 4280 0EF8R	1684		BC	MFULL		OSL168
OB1CR 4500 0FEAR	1685		CLH	PAR,TOC	TEST FOR MEMORY FULL	OSL168
OB20R 4330 0B28R	1686		BE	*+8		OSL168
OB24R 4380 0EF8R	1687		BNL	MFULL		OSL168
OB28R 4820 1018R	1688		LH	AC1,BCFLG		OSL168



12CR	4230	1689		BNZ	CMD3		OSL16850
	0B38R						
130R	4000	1690		STH	PAR,LCMTOP		OSL16860
	1028R						
134R	4300	1691		B	NEWITM		OSL16870
	0824R						
138R	40C0	1692	CMD3	STH	ZRO,BCFLG		OSL16880
	1018R						
13CR	4500	1693		CLH	PAR,LCMBOT		OSL16890
	1026R						
140R	4330	1694		BE	**+8		OSL16900
	0B48R						
144R	4380	1695		BNL	MFULL		OSL16910
	0EF8R						
148R	4500	1696		CLH	PAR,BCMTOP		OSL16920
	1024R						
14CR	4280	1697		BL	NLWITM		OSL16930
	0824R						
150R	4000	1698		STH	PAR,BCMTOP		OSL16940
	1024R						
154R	4300	1699		B	NEWITM		OSL16950
	0824R						
		1700	* REFERENCE		COMMON-BLOCK		OSL16960
158R	4110	1701	CMREF	BAL	RTN,SFETCH	MOVE SYMBOL TO SYMBUF	OSL16970
	0C1AR						
15CR	C860	1702		LHI	KEY,3		OSL16980
	0003						
160R	4110	1703		BAL	RTN,SFIND	FIND COMMON-DEF MATCHING NAME.	OSL16990
	0CDAR						
164R	4110	1704		BAL	RTN,DFETCH	FETCH DISPLACEMENT VALUE	OSL17000
	0C54R						
168R	0855	1705		LHR	NDX,NDX	IF NO SUCH DEFINITION, INSERT	OSL17010
16AR	4330	1706		BZ	CMREF1	SYMBOL AS A REF. THIS CAUSES IT TO	OSL17020
	0B78R						
16ER	48A5	1707		LH	DEF,6(NDX)	APPEAR AS UNDEFINED IN THE MAP.	OSL17030
	0006						
172R	0AA4	1708		AHR	DEF,ARG	ELSE,LOAD EFFECTIVE DEFINITION-ADRS	OSL17040
174R	4300	1709		B	NEWITM	ADDING THE DISPLACEMENT TO THE	OSL17050
	0824R						
178R	086E	1710	CMREF1	LHR	KEY,TWO	BOTTOM-ADDRESS OF THE BLOCK.	OSL17060
17AR	4110	1711		BAL	RTN,SINSRT		OSL17070
	0D18R						
17ER	4300	1712		B	NEWITM		OSL17080
	0824R						
		1713	* BLOCK-DATA		(LOAD COMMON)		OSL17090
182R	40D0	1714	CM2DA	STH	ONE,CMFLG		OSL17100
	1016R						
186R		1715	CM1DA	EQU	*		OSL17110
		1719		IF	1		OSL17150
186R	4110	1720		BAL	RTN,SFETCH	MOVE NAME OF BLOCK TO SYMBUF	OSL17160
	0C1AR						
18AR	C860	1721		LHI	KEY,3		OSL17170
	0003						
18ER	4110	1722		BAL	RTN,SFIND	FIND MATCHING DEF-COM IN TABLE.	OSL17180
	0CDAR						
192R	4110	1723		BAL	RTN,DFETCH	FETCH DISPLACEMENT	OSL17190

0B96R	0C54R 0855	1724	LHR	NDX,NDX	IF NO SUCH BLOCK FOUND, INSERT NAME	OSL172
0B98R	4330	1725	BZ	CM1DA2	AS A REF TO SHOW AS UNDEFINED SYMBOL	OSL172
0B9CR	0BD2R 4855	1726	LH	NDX,6(NDX)	IN MEMORY-MAP.	OSL172
	0006					
0BA0R	0A54	1727	AHR	NDX,ARG	BUMP BLOCK-LOCATION BY DISPLACEMENT	OSL172
0BA2R	4110	1728	BAL	RTN,DFETCH	FETCH DATA	OSL172
	0C54R					
0BA6R	4830	1729	LH	AC2,CMFLG	CMFLG INDICATES FULLWORD/HALFWORD	OSL172
	1016R					
0BAAr	4330	1730	BZ	CM1DA1		OSL172
	0BC8R					
0BAER	4040	1731	STH	ARG,TMP1	CALL TEXT-MOVER TO PUT TEXT INTO	OSL172
	1002R					
0BB2R	4110	1732	BAL	RTN,DFETCH	CORE OR OUTPUT BUFFER	OSL172
	0C54R					
0BB6R	0834	1733	LHR	AC2,ARG		OSL172
0BH8R	4820	1734	LH	AC1,TMP1		OSL173
	1002R					
0B8CR	4110	1735	BAL	RTN,MFW		OSL173
	0D80R					
0BC0R	40C0	1736	STH	ZRO,CMFLG		OSL173
	1016R					
0BC4R	4300	1737	B	NEWITM		OSL173
	0824R					
0BC8R	0824	1738	CM1DA1 LHR	AC1,ARG		OSL173
0BCAR	4110	1739	BAL	RTN,MHW		OSL173
	0D84R					
0RCER	4300	1740	B	NEWITM		OSL173
	0824R					
0BD2R	4820	1741	CM1DA2 LH	AC1,CMFLG		OSL173
	1016R					
0BD6R	4330	1742	BZ	CM1DA3		OSL173
	0BE2R					
0BDAR	40C0	1743	STH	ZRO,CMFLG		OSL173
	1016R					
0BDER	4110	1744	BAL	RTN,DFETCH		OSL174
	0C54R					
0BE2R	4110	1745	CM1DA3 BAL	RTN,DFETCH		OSL174
	0C54R					
0BE6R	4300	1746	B	CMREF1		OSL174
	0B78R					
	1747	*				OSL174
0BEAR	0799	1748	CMSEG XHR	SEQ,SEQ	RESET SEQUENCE-NR	OSL174
0RECR	0301	1749	BR	RTN	AND FETCH NEW DIRECTIVE	OSL174

	1751	* LOADER SERVICE-ROUTI ES			OSL17470
	1752	*			OSL17480
	1753	* THIS ROUTINE CALLED TO FETCH NEXT 4-BIT ITEM FROM BUFFER AND BUMP ITM			OSL17490
BEER C580	1754	FECH CLHI ITM,208		IF ITM AT OR PAST END OF BUFFER.	OSL17500
0000					
BF2R 4280	1755	BL FT1		RETURN ZERO IN PAR. THIS CAUSES	OSL17510
0BFAR					
BF6R 0700	1756	XHR PAR,PAR		DECODER TO READ-NEXT-RECORD	OSL17520
BF8R 0301	1757	BR RTN			OSL17530
BFAK CC80	1758	FT1 SRHL ITM,1		DIVIDE ITM TO INDEX BYTE ADDRESS.	OSL17540
0001					
BFER D308	1759	LB PAR,BUF+4(ITM)		FETCH BYTE (TWO ITEMS) ITEM POINTER	OSL17550
1248R					
C02R 4380	1760	BFC 8,FT2		(ITM BIT 15) IS IN C FLAG.	OSL17560
0C10R					
C06R C400	1761	NHI PAR,15		IF 1, MASK FIRST ITEM OUT.	OSL17570
000F					
C0AR 0A8D	1762	AHR ITM,ONE		AND BUMP ITM	OSL17580
C0CR 0A88	1763	AHR ITM,ITM			OSL17590
C0ER 0301	1764	BR RTN			OSL17600
C10R CC00	1765	FT2 SRHL PAR,4		IF 0, SHIFT SECOND ITEM OUT	OSL17610
0004					
C14R 0A88	1766	AHR ITM,ITM		AND BUMP ITM	OSL17620
C16R 0A8D	1767	AHR ITM,ONE			OSL17630
C18R 0301	1768	BR RTN			OSL17640
	1769	*			OSL17650
C1AR 4010	1770	SFETCH STH RTN,SFE2		MOVE SIX-BYTE SYMBOL TO SYMBUF	OSL17660
0C42R					
C1ER 0722	1771	XHR AC1,AC1		AC1 TO BE USED TO COUNT BYTES MOVED	OSL17670
C20R 4110	1772	SFE1 BAL RTN,FETCH		FETCH FIRST HALF-BYTE	OSL17680
0BEER					
C24R 0830	1773	LHR AC2,PAR		SAVE IT IN AC2	OSL17690
C26R 4110	1774	BAL RTN,FETCH		FETCH SECOND HALF-BYTE	OSL17700
0BEER					
C2AR CD30	1775	SLHL AC2,4		PUT TOGETHER A FULL BYTE IN AC2	OSL17710
0004					
C2ER 0630	1776	OHR AC2,PAR		AND MOVE IT TO SYMBUF	OSL17720
C30R C430	1777	NHI AC2,X'7F'		MASK TO 7-BIT ASCII	OSL17730
007F					
C34R D232	1778	STB AC2,SYMBUF(AC1)		INDEXED BY AC1	OSL17740
11B0R					
C38R 0A2D	1779	AHR AC1,ONE		BUMP AC1 AND TEST IF SYMBUF IS	OSL17750
C3AR 052F	1780	CLHR AC1,SIX		NOT FILLED, GO AND GET ANOTHER	OSL17760
C3CR 4280	1781	BL SFE1		BYTE. WHEN SYMBUF IS FILLED,	OSL17770
0C20R					
C40R 4300	1782	B 0		RETURN TO CALLER	OSL17780
0000					
C42R	1783	SFE2 EQU *-2			OSL17790
	1784	*			OSL17800
C44R 4830	1785	RFETCH LH AC2,RELFLG		MOVE A HW TO ARG, ADDING BIAS IF	OSL17810
101CR					
C48R 4330	1786	BZ DFE1		RELFLG SET, IF RELFLG IS NOT SET,	OSL17820
0C56R					
C4CR 4830	1787	LH AC2,BIAS		DON'T ADD BIAS. BIAS OR ZERO IS	OSL17830
0FF2R					
C50R 4300	1788	B DFE1		LEFT IN AC2	OSL17840

GC56R		1789	*							
OC54R	0733	1790	DFETCH	XHR	AC2,AC2					OSL178
OC56R	4010	1791	DFE1	STH	RTN,DFE3			MOVE A HW TO ARG. ADD CONTENTS OF		OSL178
	OC74R							AC2. AC2 ZEROED BY CALLING DFETCH		OSL178
OC5AR	0744	1792		XHR	ARG,ARG					OSL178
OC5CR	C820	1793		LHI	AC1,4			OR BY CALLING RFETCH WHEN RELFLG IS		OSL178
	0004							NOT SET. AC1 COUNTS FOUR HALF-BYTES		OSL178
OC60R	4110	1794	DFE2	BAL	RTN,FETCH			MOVED. FETCH HALF-BYTE		OSL179
	0BEER									OSL179
OC64R	CD40	1795		SLHL	ARG,4			MOVE IT INTO ARG		OSL179
	0004									OSL179
OC68R	0640	1796		OHR	ARG,PAR			AND LOOP BACK UNTIL A HALF-WORD		OSL179
OC6AR	0B2D	1797		SHR	AC1,ONE			HAS BEEN MOVED.		OSL179
OC6CR	4230	1798		BNZ	DFE2					OSL179
	OC60R									OSL179
OC70R	0A43	1799		AHR	ARG,AC2			ADD ZERO OR BIAS (SEE ABOVE)		OSL179
OC72R	4300	1800		B	0			AND RETURN TO CALLER.		OSL179
	0000									OSL179
OC74R		1801	DFE3	EQU	*-2					OSL179
		1802	*							OSL179
OC76R	4020	1803	MLOG	STH	AC1,MAPLO			LOGS MAP-MESSAGES... FBA IN AC1, LBA		OSL179
	1138R									OSL179
OC7AR	4030	1804		STH	AC2,MAPHI			IN AC2, MAP-DEVICE (LU) IN MAPDEV+1.		OSL180
	113AR									OSL180
OC7ER	E110	1805	MLOG1	SVC	1,MAPDEV					OSL180
	1134R									OSL180
OC82R	4820	1806		LH	AC1,MAPDEV+2			FETCH OUTPUT CALL-STATUS		OSL180
	1136R									OSL180
OC86R	0311	1807		BNMR	RTN			IF OKAY, RETURN		OSL180
OC88R	E120	1808		SVC	2,**+B					OSL180
	OC90R									OSL180
OC8CR	4300	1809		B	CKDEV					OSL180
	0FCER									OSL180
OC90R		1810	MLOG2	EQU	*					OSL180
OC90R	0007	1811		DC	7,11,C' I/O DEV ERR'					OSL180
	000B									OSL180
	2049									OSL180
	2F4F									OSL180
	2044									OSL180
	4556									OSL180
	2045									OSL180
	5252									OSL180
		1812	* SEARCHES TABLE FOR ENTRY MATCHING KEY. FOR EACH ONE FOUND, RETURN+B							OSL180
OCA0R	4850	1813	TSRCH	LH	NDX,STOP			IF NONE FOUND, RETURN+4. TO FIND MORE		OSL180
	1026R									OSL180
OCA4R		1814	CSRCH	EQU	*					OSL181
OCA4R	0700	1815	CSO	XHR	PAR,PAR			CALL CSRCH. ON EXITS, NDX POINTS AT		OSL181
OCA6R	CB50	1816		SHI	NDX,8			ANOTHER SYMBOL AT CSO		OSL181
	000B									OSL181
OCAA	4550	1817		CLH	NDX,SBOT			HOT OF FOUND ENTRY. IF EXIT+4, NDX		OSL181
	0FF6R									OSL181
OCAER	0281	1818		BLR	RTN			EQUALS SBOT		OSL181
OCB0R	4825	1819		LH	AC1,0(NDX)			FETCH SYMBOL AND ASSEMBLE KEY IN PAR		OSL181
	0000									OSL181
OCB4R	4310	1820		BNM	CS1			ADD 1 TO PAR FOR FIRST HW,		OSL181

0CBAR	0A0D	1821		AHR	PAR,ONE	ADD 2 FOR SECOND, AND	OSL18170
CBAR	4825	1822	CS1	LH	AC1,2(NDX)	ADD 4 FOR THIRD IF RESPECTIVE KEY-	OSL18180
	0002						
CBER	4310	1823		BNM	CS2	BITS ARE SET. THUS BY CS3, AC1	OSL18190
	0CC4R						
CC2R	0A0E	1824		AHR	PAR,TWO	CONTAINS BINARY VALUE OF KEY (0-7)	OSL18200
CC4R	4825	1825	CS2	LH	AC1,4(NDX)		OSL18210
	0004						
CC8R	4310	1826		BNM	CS3		OSL18220
	0CD0R						
CCCR	CA00	1827		AHI	PAR,4		OSL18230
	0004						
CD0R	0506	1828	CS3	CLHR	PAR,KEY	DOES KEY MATCH? EXIT TO CALL+8	OSL18240
CD2R	4331	1829		BE	4(RTN)	NO, THEN BUMP NDX AND RETURN TO FIND	OSL18250
	0004						
CD6R	4300	1830		B	CS0		OSL18260
	0CA4R						
		1831	*				OSL18270
CDAR	4850	1832	SFIND	LH	NDX,STOP	CALL SFIND TO LOOK FOR A SYMBOL WITH	OSL18280
	1026R						
CDER	4010	1833	CFIND	STH	RTN,SFI4	KEY MATCHING KEY AND SYMBOL MATCHING	OSL18290
	0D0AR						
CE2R	4110	1834	SFI1	BAL	RTN,CSRCH	SYMBUF, ON RETURN NDX POINTS AT THE	OSL18300
	0CA4R						
CE6R	4300	1835		B	SFI6	FOUND MATCH. IF NO MATCH IS FOUND,	OSL18310
	0D12R						
CEAR	0722	1836		XHR	AC1,AC1	RETURN IS MADE WITH NDX ZERO.	OSL18320
CECR	4835	1837	SFI2	LH	AC2,0(NDX)		OSL18330
	0000						
CF0R	C430	1838		NHI	AC2,X'7FFF'	CALL CFIND TO CONTINUE LOOKING	OSL18340
	7FFF						
CF4R	4532	1839		CLH	AC2,SYMBUF(AC1)		OSL18350
	1180R						
CF8R	4230	1840		BNE	SFI5	FETCH HW AT A TIME FROM TABLE. MASK	OSL18360
	0D0CR						
CFCR	0A2E	1841		AHR	AC1,TWO	OFF KEY (MSB) AND COMPARE WITH NEXT	OSL18370
CFER	0A5E	1842		AHR	NDX,TWO	HW OF SYMBUF. IF NO MATCH, GET	OSL18380
DO0R	052F	1843		CLHR	AC1,SIX	ANOTHER SYMBOL. IF THEY MATCH, RESET	OSL18390
DO2R	4280	1844		BL	SFI2	NDX TO SYMBOL AND EXIT TO CALLER.	OSL18400
	0CECR						
DO6R	0B5F	1845		SHR	NDX,SIX		OSL18410
DO8R	4300	1846	SFI3	B	0		OSL18420
	0000						
DOAR		1847	SFI4	EQU	*-2		OSL18430
DOCR	0B52	1848	SFI5	SHR	NDX,AC1		OSL18440
DOER	4300	1849		B	SFI1		OSL18450
	0CE2R						
DI2R	0755	1850	SFI6	XHR	NDX,NDX	IF NO MATCH FOUND, ZERO NDX	OSL18460
DI4R	4300	1851		B	SFI3		OSL18470
	0D08R						
		1852	* INSERT A SYMBOL IN TABLE, WITH KEY				OSL18480
DI8R	4010	1853	SINSRT	STH	RTN,TIN4	CALL SINSRT, OR TINSRT	OSL18490
	0D7ER						
DICR	0846	1854		LHR	ARG,KEY	KEY IS IN KEY	OSL18500
DIER	C860	1855		LHI	KEY,7	VALUE IS IN PAR	OSL18510

0007						
0022R	0830	1856	LHR	AC2,PAR	SAVE (PAR)	OSL188
0024R	4110	1857	BAL	RTN,TSRCH	SYMBOL IS IN SYMBUF	OSL188
	UCA0R					
0028R	4050	1858	STH	NDX,SBOT	IF NEW ENTRY, RESET SBOT	OSL188
	OFF6R					
002CR	0803	1859	LHR	PAR,AC2	RESTORE (PAR)	OSL188
002ER	0864	1860	LHR	KEY,ARG	TINSRT MOVES SYMBOL TO TABLE WHERE	OSL188
0030R	4820	1861	LH	AC1,OUTFLG	IF OUTFLG, KEEP STAB FROM OVER-	OSL188
	OFFAR					
0034R	4230	1862	BNZ	SIN1	LAPPING THE OUTPUT BUFFER. ON LOAD	OSL188
	0D44R					
0038R	4550	1863	CLH	NDX,PTOP	KEEP IT FROM BELOW PTOP.	OSL188
	OFF4R					
003CR	4380	1864	BNL	TIN1		OSL186
	0D54R					
0040R	4300	1865	B	MFULL		OSL186
	0EF8R					
0044R	C550	1866	SIN1	CLHI	NDX,0BUF+107	OSL186
	1318R					
0048R	4380	1867	BNL	TIN1		OSL186
	0D54R					
004CR	4300	1868	B	MFULL		OSL186
	0EF8R					
0050R	4010	1869	TINSRT	STH	RTN,TIN4	AVAILABLE SLOT
	0D7ER					
0054R	4035	1870	TIN1	STH	AC2,6(NDX)	SAVED (PAR) IS SYMBOL-VALUE IN TABLE
	0006					
0058R	0733	1871	XHR	AC2,AC2		OSL186
005AR	4823	1872	TIN2	LH	AC1,SYMBUF(AC2)	OSL186
	1180R					
005ER	CC60	1873	SRHL	KEY.1		OSL186
	0001					
0062R	4380	1874	BFC	B,TIN3		OSL187
	0D6AR					
0066R	C620	1875	OHI	AC1,X*8000'		OSL187
	8000					
006AR	4025	1876	TIN3	STH	AC1.0(NDX)	OSL187
	0000					
006ER	0A3E	1877	AHR	AC2,TWO		OSL187
0070R	0A5E	1878	AHR	NDX,TWO		OSL187
0072R	053F	1879	CLHR	AC2,SIX		OSL187
0074R	4280	1880	BL	TIN2		OSL187
	0D5AR					
0078R	0B5F	1881	SHR	NDX,SIX		OSL187
007AR	0B64	1882	LHR	KEY,ARG		OSL187
007CR	4300	1883	B	0		OSL187
	0000					
007ER		1884	TIN4	EQU	*-2	
		1885			* TEXT-MOVER ROUTINES	
0080R	40E0	1886	MFW	STH	TWO,FWFLG	CALL MFW TO MOVE FULLWORD, MHW FOR A
	1014R					HALFWORD. IF OUTFLG IS SET, TEXT IS
0084R	4010	1887	MHW	STH	RTN,MHWR	PACKED INTO OUTPUT BUFFER. AC1 AND
	0DD6R					
0088R	4800	1888	LH	PAR,OUTFLG		OSL1884
	OFFAR					

D8CR	4230	1889		BNZ	MHW2	AC2 HOLD FIRST (AND SECOND) HW OF	OSL18850
	0DD8R						
J90R	4550	1890		CLH	NDX,SBOT	TEXT. IF LOADING, CHECK FOR OVERLAP	OSL18860
	OFF6R						
J94R	4280	1891		BL	MWT2	WITH SYMBOL-TABLE, AND MOVE TEXT	OSL18870
	0DA0R						
D98R	4550	1892		CLH	NDX,STOP	BUT FIRST,	OSL18880
	1026R						
D9CR	4280	1893		BL	MFULL		OSL18890
	0EF8R						
DA0R	C550	1894	MWT2	CLHI	NDX,PBOT	TEST FOR OVERLAP WITH LOADER	OSL18900
	131CR						
DA4R	4280	1895		BL	ADER	AND IF SO, STOP LOAD.	OSL18910
	0F00R						
DA8R	4025	1896		STH	AC1,0(NDX)	CORE-LOCATION SPECIFIED IN NDX.	OSL18920
	0000						
DACK	4820	1897		LH	AC1,FWFLG		OSL18930
	1014R						
DB0R	4330	1898		BZ	MHW1		OSL18940
	0DCAR						
DB4R	0A5E	1899		AHR	NDX,TWO		OSL18950
DB6R	4550	1900		CLH	NDX,SBOT	MAKE SURE SECOND HW DOWNNT	OSL18960
	OFF6R						
DBAR	4280	1901		BL	MWT25	OVERLAP SYMBOL-TABLE	OSL18970
	0DC6R						
DBER	4550	1902		CLH	NDX,STOP		OSL18980
	1026R						
DC2R	4280	1903		BL	MFULL		OSL18990
	0EF8R						
DC6R	4035	1904	MWT25	STH	AC2,0(NDX)		OSL19000
	0000						
DCAR	0A5E	1905	MHW1	AHR	NDX,TWO	BUMP NDX AND UPDATE OLOC.	OSL19010
DCCR	40C0	1906		STH	ZRO,FWFLG		OSL19020
	1014R						
DD0R	4050	1907		STH	NDX,OLOC		OSL19030
	101AR						
DD4R	4300	1908		B	*		OSL19040
	0DD4R						
DD6R		1909	MHWR	EQU	*-2		OSL19050
DD8R	4550	1910	MHW2	CLH	NDX,OLOC	IF OUTPUTTING, SEE IF LOCATION-COUNT	OSL19060
	101AR						
DDCR	4330	1911		BE	MHW5	HAS BEEN MODIFIED.	OSL19070
	0DFAR						
DE0R	4020	1912		STH	AC1,MHW3	IF SO, SAVE TEXT (IN REGS AC1 AND	OSL19080
	0DF4R						
DE4R	4030	1913		STH	AC2,MHW4	AC2) AND PACK OUT A LDA (CITM 5)	OSL19090
	0DF8R						
DE8R	0825	1914		LHR	AC1,NDX	AND NEW LOC VALUE.	OSL19100
DEAR	C800	1915		LHI	PAR,5		OSL19110
	0005						
DEER	4110	1916		BAL	RTN,OHW		OSL19120
	0E94R						
DF2R	C820	1917		LHI	AC1,*	RESTORE AC1, AND AC2	OSL19130
	0DF2R						
DF4R		1918	MHW3	EQU	*-2		OSL19140
DF6R	C830	1919		LHI	AC2,*		OSL19150

Address	Year	Model	Op	Op	Op	Op	Op	Op
0DF6R								
0DF8R	1920	MHW4	EQU	*-2				
0DFAR	1921	MHW5	LH	PAR,FWFLG		PACK OUT EITHER:		OSL192
								OSL192
0DFER	1922		BZ	MHW6		. CITM 8 AND A HALFWORD		OSL192
0E02R	1923		LHI	PAR,10		OR		OSL192
0E06R	1924		BAL	RTN,OFW		. CITM A AND A FULLWORD		OSL192
0E0AR	1925		AHR	NDX,TWO				OSL192
0E0CR	1926		B	MHW1				OSL192
0E10R	1927	MHW6	LHI	PAR,8				OSL192
0E14R	1928		BAL	RTN,OHW				OSL192
0E18R	1929		B	MHW1				OSL192
0E1CR	1930	OPAK	NHI	PAR,15		CALL OPAK TO PACK A 4-BIT ITEM TO		OSL192
0E20R	1931		STH	NDX,OPAK3				OSL192
0E24R	1932		LH	NDX,OITM		OBUF. FETCH OUTPUT HALF-BYTE POINTER		OSL192
0E28R	1933		SRHL	NDX,1		SHIFT TO GET BYTE-INTEX, C-FLAG		OSL192
0E2CR	1934		BTC	8,OPAK1		IS HALF-BYTE POINTER		OSL193
0E30R	1935		SLHL	PAR,4		IF EVEN, MOVE PAR INTO BYTE, UPPER		OSL193
0E34R	1936		STB	PAR,OBUF+4(NDX)		FOUR BITS		OSL193
0E38R	1937		AHR	NDX,NDX		RESTORE NDX AND BUMP.		OSL193
0E3AR	1938		AHR	NDX,ONE				OSL193
0E3CR	1939		B	OPAK2				OSL193
0E40R	1940	OPAK1	LB	ARG,OBUF+4(NDX)		IF ODD, FETCH UPPER HALF-BYTE		OSL193
0E44R	1941		OHR	PAR,ARG		OR THEM TOGETHER		OSL193
0E46R	1942		STB	PAR,OBUF+4(NDX)		AND MOVE FULL-BYTE INTO BUFFER.		OSL193
0E4AR	1943		AHR	NDX,NDX				OSL193
0E4CR	1944		AHR	NDX,TWO		BUMP AND SAVE OITM		OSL194
0E4ER	1945	OPAK2	STH	NDX,OITM				OSL194
0E52R	1946		LHI	NDX,*				OSL194
0E54R	1947	OPAK3	EQU	*-2				OSL194
0E56R	1948		BR	RTN				OSL194
	1949	*						OSL194
	1950	*				CALL OFW TO PACK A CITM AND A FULLWORD TO OUTPUT BUFFER		OSL194
	1951	*				PAR HALDS CONT ITEM, AC1 AND AC2 HOLD FIRST AND SECOND HALFWORDS		OSL194
0E58R	1952	OFW	STH	RTN,OBYTE2		SAVE RETURN-ADDRESS		OSL194
0E5CR	1953		STH	AC2,TMP2		SAVE SECOND HW		OSL194



E60R	1004R 4020	1954	STH	AC1,TMP1	AND FIRST HW	OSL19500
E64R	1002R C840	1955	LHI	ARG,198	TEST IF ROOM FOR A FULLWORD IN OBUF	OSL19510
E68R	00C6 4110	1956	BAL	RTN,OTST	OTST WILL OUTPUT OBUF IF NOT	OSL19520
E6CR	0EAAK 0722	1957	XHR	AC1,AC1		OSL19530
E6EK	1958 * 4110	1959	PITM	BAL	RTN,OPAK	PACK THD CONTROL-ITEM OSL19540 OSL19550
E72R	0E1CR D332	1960	OBYTE	LB	AC2,TMP1(AC1)	FETCH NEXT DATA-BYTE (AC1 = BYTE NOX) OSL19560
E76R	1002R 9303	1961	LBR	PAR,AC2		OSL19570
E78R	CC00	1962	SRHL	PAR,4		OSL19580
E7CR	0004 4110	1963	BAL	RTN,OPAK	PACK FIRST FUR-BITS	OSL19590
E80R	0E1CR 9303	1964	LHR	PAR,AC2		OSL19600
E82R	4110	1965	BAL	RTN,OPAK	AND SECOND FOUR-BITS OF THIS BYTE.	OSL19610
E86R	0E1CR 0A2D	1966	AHR	AC1,ONE	BUMP BYTE-INDEX	OSL19620
E88R	C520	1967	CLHI	AC1,4	AND GET ANOTHER BYTE.	OSL19630
E8CR	0004 4280	1968	BL	OBYTE	WHEN ALL ARE OUTPUT,	OSL19640
E90R	0E72R 4300	1969	B	*	EXIT TO CALLER.	OSL19650
E92R	0E90R	1970	OBYTE2	EQU	*-2	OSL19660
		1971	*			OSL19670
		1972	*	CALL OHW TO PACK A CITM AND A HALFWORD TO OBUF		OSL19680
		1973	*	PAR HALDS CONTROL-ITEM. AC1 HOLDS HALF-WORD		OSL19690
E94R	4010	1974	OHW	STH	RTN,OBYTE2	SAVE RETURN-ADDRESS OSL19700
E98R	0E92R 4020	1975	STH	AC1,TMP2	SAVE DATA TO BE OUTPUT	OSL19710
E9CR	1004R C840	1976	LHI	ARG,202	TEST IF ROOM FOR A HALF-WORD IN OBUF	OSL19720
EA0R	00CA 4110	1977	BAL	RTN,OTST	IF NOT, OTST WRITES OUT THE BUFFER	OSL19730
EA4R	0EAAK 082E	1978	LHR	AC1,TWO	GO AND PACK IT OUT...	OSL19740
EA6R	4300	1979	B	PITM		OSL19750
EAAR	0E6ER	1980	*			OSL19760
	4540	1981	OTST	CLH	ARG,OITM	OSL19770
EALR	100EK 0381	1982		DNLR	RTN	OSL19780
EB0R	1983 * 4840	1984	OUTTXT	LH	ARG,0SEQ	CALL OUTTXT TO WRITE OUTPUT BIFFER. OSL19790 OSL19800
EB4R	100CK 0B4D	1985	SHR	ARG,ONE		OSL19810
EB6R	4040	1986	STH	ARG,0SEQ	DECREMENT SEQUENCE-NUMBER	OSL19820
EBAR	100CK 4040	1987	STH	ARG,OBUF	AND MOVE IT INTO OBUF	OSL19830
EBER	12B0R C830	1988	LHI	AC2,106		OSL19840

006A							
0EC2R C840	1989		LHI	ARG,-1			OSL19
FFFF							
0EC6R 4743	1990	OT1	XH	ARG,OBUF(AC2)	COMPUTE CHECKSUM AND		OSL19
12B0R							
0ECAR 0B3E	1991		SHR	AC2,TWO	MOVE IT INTO OBUF+2		OSL19
0ECCR 4310	1992		BNM	OT1			OSL19
0EC6R							
0ED0R 4040	1993		STH	ARG,OBUF+2			OSL19
12B2R							
0ED4R E110	1994		SVC	1,BOUTDV	WRITE OBUF ON BOUTDV		OSL19
1204R							
0ED8R 0840	1995		LHR	ARG,PAR			OSL19
0EDAR 4800	1996		LH	PAR,BOUTDV+2			OSL19
1206R							
0EDER 4230	1997		BNZ	INER2+4			OSL19
0FAAR							
0EE2R 0804	1998		LHR	PAR,ARG			OSL19
	1999	*					OSL19
0EE4R C830	2000	ZOBUF	LHI	AC2,106	AND THEN ZERO OUT BUFFER		OSL19
006A							OSL19
0EE8R 40C0	2001		STH	ZRO,OITM			OSL19
100ER							
0EECR 40C3	2002	ZOB1	STH	ZRO,OBUF(AC2)			OSL19
12B0R							
0EF0R 0B3E	2003		SHR	AC2,TWO			OSL19
0EF2R 4310	2004		BNM	ZOB1			OSL20
0EECR							
0EF6R 0301	2005		BR	RTN			OSL20
	2006	*					OSL20
	2007	*					OSL20
0EF8R E120	2008	MFULL	SVC	2,MFM	LOG MESSAGE ABOUT MEMORY-FULL		OSL20
10EER							
0EFCR 4300	2009		B	CKDEV			OSL20
0FCER							
0F00R 4100	2010	ADER	BAL	PAR,CKTTY			OSL20
0FDER							
0F04R E120	2011		SVC	2,ADERM	ADDRESS-ERROR MESSAGE		OSL20
120CR							
0F08R 0307	2012		BR	RTX			OSL20
0F0AR E120	2013	MEOF	SVC	2,MEOFS			OSL20
1124R							
0F0ER D320	2014		LB	AC1,BINDV+1	FETCH INPUT LU		OSL20
11FDR							
0F12R D220	2015		STB	AC1,OLU+1	AND BACKSPACE IT OVER		OSL20
1021R							
0F16R 4300	2016		B	BSR	FINAL EOF.		OSL20
015AR							
0F1AR 4100	2017	MDU	BAL	PAR,CKTTY			OSL20
0FDER							
0F1ER E120	2018		SVC	2,MDUS			OSL20
0F24R							
0F22R 0301	2019		BR	RTN			OSL20
0F24R 0007	2020	MDUS	DC	7,8,C DEV END			OSL20
0008							
2044							

	4556						
	2045						
	4L44						
0F30R	E110	2021	LREAD	SVC	1,BINDV	READ NEXT INPUT RECORD	OSL20170
	11FCR						
0F34R	4820	2022		LH	AC1,BINDV+2	FETCH IO STATUS AFTER READING	OSL20180
	11FER						
0F38R	4310	2023		BNM	CRN1	IF NOT MINUS, READ WAS OKAY	OSL20190
	0F64R						
0F3CR	0A22	2024		AHR	AC1,AC1		OSL20200
0F3ER	4210	2025		BM	CMDERR		OSL20210
	010AR						
0F42R	0A22	2026		AHR	AC1,AC1		OSL20220
0F44R	4210	2027		BM	MDU		OSL20230
	0F1AR						
0F48R	CD20	2028		SLHL	AC1,2		OSL20240
	0002						
0F4CR	4280	2029		BC	MEOM	EOM	OSL20250
	0F58R						
0F50R	4210	2030		BM	MEOF	EOF	OSL20260
	0F0AR						
0F54R	4300	2031		B	INER2		OSL20270
	0FA6R						
0F58R	E120	2032	MEOM	SVC	2,MEOMS		OSL20280
	112CR						
0F5CR	E120	2033		SVC	2,PAUSE		OSL20290
	101ER						
0F60R	4300	2034		B	LREAD	RE TRY	OSL20300
	0F30R						
0F64R	C880	2035	CRN1	LHI	ITM,106	SET UP INDEX	OSL20310
	006A						
0F68R	0722	2036		XHR	AC1,AC1	AND COMPUTE CHECKSUM.	OSL20320
0F6AR	4728	2037	CKSM	XH	AC1,BUF(ITM)	EXCLUSIVE-OR OF ZERO WITH	OSL20330
	1244R						
0F6ER	0B8E	2038		SHR	ITM,TWO	EACH HALF-WORD OF BUFFER (INCLUDING	OSL20340
0F70R	4310	2039		BNM	CKSM	THE HALF-WORD CONTAINING THE CHECK-	OSL20350
	0F6AR						
0F74R	C720	2040		XHI	AC1,-1	SUM) AND -1 SHOULD PRODUCE ZERO.	OSL20360
	FFFF						
0F78R	4331	2041		BZ	4(RTN)		OSL20370
	0004						
0F7CR	E120	2042		SVC	2,MSCKSM	AND PAUSE...	OSL20380
	1188R						
0F80R	4100	2043	CRN2	BAL	PAR,CKTTY		OSL20390
	0FDER						
0F84R	E120	2044		SVC	2,PAUSE	ON RESTART, READ AGAIN	OSL20400
	101ER						
0F88R	0831	2045		LHR	AC2,RTN		OSL20410
0F8AR	4820	2046		LH	AC1,BINDV		OSL20420
	11FCR						
0F8ER	4110	2047		BAL	RTN,BSR+4	BACKSPACE INPUT IF MAG TAPE	OSL20430
	015ER						
0F92R	0813	2048		LHR	RTN,AC2		OSL20440
0F94R	4300	2049		B	LREAD	IF CKSUM IS OK, BUMP SEQUENCE-NUMBER	OSL20450
	0F30R						
0F98R	0899	2050	INER	LHR	SEQ,SEQ	CHECK IF LOAD IN PROGRESS	OSL20460

0F9AR	4310	2051	BNM	CKDEV		OSL204
	0FCER					
0F9ER	E120	2052	SVC	2,ABLM		OSL204
	11E2R					
0FA2R	4300	2053	B	CKDEV		OSL204
	0FCER					
0FA6R	4800	2054	INER2	LH	PAR,BINDV+2	IF ERR STATUS, LOG
	11FER					OSL205
0FAAR	E120	2055	SVC	2,UNPAK		"I/O DEV ERR" AND HEX STATUS,
	10FCR					OSL205
0FAER	E120	2056	SVC	2,MLOG2		EXIT VIA INER, TO TYPE
	0C90R					OSL205
0FB2R	E120	2057	SVC	2,IoEM		"LOAD ABORTED" IF LOADING.
	0FFER					OSL205
0FB6R	4300	2058	B	INER		EXIT TO LOADER.
	0F98R					OSL205
0FBAR	E120	2059	RPMSG	SVC	2,UNPAK	HEX STATUS
	10FCR					OSL205
0FBER	E120	2060	SVC	2,MLOG2		"I/O ERROR" MSG
	0C90R					OSL205
0FC2R	E120	2061	SVC	2,IOEM		HEX STAT OUT
	0FFER					OSL205
0FC6R		2062	PGO	LQU	*	
0FC6R	E120	2063	SVC	2,PAUSE		WAIT FOR DEVICE OK
	101ER					OSL205
0FCAR	4300	2064	B	LOADER		TRY AGAIN
	0014R					OSL206
0FCER		2065	CKDEV	EQU	*	
		2066	*			IF INPUT LU IS TTY THEN GO TO LOADER
		2067	*			OTHERWISE PAUSE BEFORE RETURNING TO LOADER
0FCER	4800	2068	LH	PAR,TTYSWT		OSL206
	0656R					OSL206
0FD2R	4330	2069	BZ	LOADER		OSL206
	0014R					
0FD6R	E120	2070	SVC	2,ECHO		OSL206
	1240R					
0FDAR	4300	2071	B	PGO		OSL206
	0FC6R					
0FDER	4840	2072	CKTTY	LH	ARG,TTYSWT	OSL206
	0656R					
0FE2R	0330	2073	BZR	PAR		OSL206
0FE4R	E120	2074	SVC	2,ECHO		OSL207
	1240R					
0FE8R	0300	2075	BR	PAR		OSL207

	2077	* LOADER CONSTANTS			OSL20730
	2078	*			OSL20740
FEAR 0000	2079	TOC	DC	0	OSL20750
FECR 0005	2080	FETCHP	DC	5.AC1	OSL20760
0002					
FF0R 0000	2081	PTR	DC	0	OSL20770
FF2R 131CR	2082	BIAS	DC	PHOT	OSL20780
FF4R 0000	2083	PTOP	DC	0	OSL20790
FF6R 0000	2084	SBOT	DC	0	OSL20800
FF8R 0000	2085	EDTFLG	DC	0	OSL20810
FFAR 0000	2086	OUTFLG	DC	0	OSL20820
FFCR 0000	2087	XFER	DC	0	OSL20830
FFER 0007	2088	IOEM	DC	7,4	OSL20840
0004					
002R 0000	2089	TMP1	DC	0	OSL20850
004R 0000	2090	TMP2	DC	0	OSL20860
006R 0000	2091	SKPFLG	DC	0	OSL20870
008R 0000	2092	LOC	DC	0,0	OSL20880
0000					
000CR 0000	2093	OSEQ	DC	0	OSL20890
000ER 0000	2094	OITM	DC	0	OSL20900
0010R 0000	2095	RLFLG	DC	0	OSL20910
0012R 0000	2096	MADA	DC	0	OSL20920
0014R 0000	2097	FWFLG	DC	0	OSL20930
0016R 0000	2098	CMFLG	DC	0	OSL20940
0018R 0000	2099	BCFLG	DC	0	OSL20950
001AR 0000	2100	OLOC	DC	0	OSL20960
001CR 0000	2101	RELFLG	DC	0	OSL20970
001ER 0001	2102	PAUSE	DC	1	OSL20980
0020R 0000	2103	OLU	DC	0	OSL20990
0022R 0000	2104	BCMBOT	DC	0	OSL21000
0024R 0000	2105	BCMTOP	DC	0	OSL21010
0026R 0000	2106	LCMBOT	DC	0	OSL21020
0028R 0000	2107	LCMTOP	DC	0	OSL21030
0026R	2108	STOP	EQU	LCMBOT	OSL2104
002AR 0000	2109	LABLEN	DC	0	OSL21050
002CR 0000	2110	BLKLEN	DC	0	OSL21060
0022R	2111	COMBOT	EQU	BCMBOT	OSL21070
002ER 0000	2112	SYMFLG	DC	0	OSL21080
0030R 0000	2113	SEQCON	DC	0	OSL21090
0032R 0000	2114	ZROCON	DC	0	OSL21100
0034R 0001	2115	ONECON	DC	1	OSL21110
0036R 0002	2116	TWOCON	DC	2	OSL21120
0038R 0006	2117	SIXCON	DC	6	OSL21130
	2118	* OPERATOR-COMMAND DECODER TABLE			OSL21140
103AR 4F52	2119	CMDTAB	DC	C*OR*,ORG SET BIAS (ORIGIN)	OSL21150
0118R					
103ER 4249	2120		DC	C*BI*,ORG	OSL21160
0118R					
1042R 4455	2121		DC	C*DU*,DUPER DUPLICATOR	OSL21170
0306R					
1046R 4C43	2122		DC	C*LC*,LABCOM SET LENGTH OF LAB COMMON	OSL21180
0136R					
104AR 4243	2123		DC	C*BC*,BLKCOM SET LENGTH OF BLANK COMMON	OSL21190
013ER					
104ER 4F55	2124		DC	C*OU*,OUT OUTPUT TAPE	OSL21200

1052R	03D2R 584F 03F6R	2125	DC	C'XO',XOUT	DON'T OUTPUT	OSL211
1056R	474F 0438R	2126	DC	C'GO',GO	TRANSFER TO PROGRAM	OSL211
105AR	4D41 0452R	2127	DC	C'MA',MAP	LIST MEMORY-MAP	OSL211
105ER	4C49 05DAR	2128	DC	C'LI',LINK	LINK-LOAD	OSL211
1062R	4C4F 058AR	2129	DC	C'LO',LOAD	LOAD WITH PURGED TABLE	OSL211
1066R	4544 05EER	2130	DC	C'ED',EDIT	CONDITIONAL LOAD	OSL211
106AR	5245 0424R	2131	DC	C'RE',REWIND	REWIND FILE	OSL211
106ER	4C41 0178R	2132	DC	C'LA',LABOPT		OSL211
1072R	4649 01E6R	2133	DC	C'FI',FIND	FIND A LABEL	OSL211
1076R	434F 0280R	2134	DC	C'CO',COPY	COPY A PROGRAM	OSL211
107AR	454F 0146R	2135	DC	C'EO',CWFM	WRITE A FILE-MARK	OSL211
107ER	5441 0382R	2136	DC	C'TA',TABCON	TABLE-OF-CONTENTS	OSL211
1082R	454E 063CR	2137	DC	C'EN',RETURN	RETURN TO BOSS	OSL211
1086R	4C47 0640R	2138	DC	C'LG',LOG	LOG COMMAND	OSL211
108AR	5041 0FC6R	2139	DC	C'PA',PGO	PAUSE COMMAND	OSL211
108ER	414D 044AR	2140	DC	C'AM',AMAP	ALPHABETIZED MEMORY-MAP	OSL211
1092R	544F 061CR	2141	DC	C'TO',TOPCOR	SET TOP-OF-CORE	OSL211
		2142	IF	DOS		OSL211
1096R	4F56 0658R	2143	OVTAB DC	C'OV',OVLY	OVERLAY COMMAND	OSL211
		2151	IF	1		OSL211
109AR		2152	ECMD EQU	*		OSL211
		2153	* LOADER	CONTROL-ITEM DECODER	TABLE	OSL211
109AR	079ER FFFF	2154	CITTAB DC	CRNR,-1	0 READ NEXT RECORD	OSL211
109ER	0872R FFFF	2155	DC	CEND,-1	1 END OF PROGRAM	OSL211
10A2R	08B0R 0000	2156	DC	COCN,0	2 DEFINE CHAIN	OSL211
10A6R	08FAR 0000	2157	DC	CTOG,0	3 TOGGLE MODE	OSL211
10AAR	0922R 0004	2158	DC	CTRA,4	4 LOAD XFER	OSL211
10AER	0940R 0004	2159	DC	CLDA,4	5 SET LOAD-ADDRESS	OSL211
10B2R	0958R 0004	2160	DC	CREF,4	6 LOAD REF-ADDRESS	OSL211
10B6R	0962R	2161	DC	CDEF,4	7 LOAD DEF-ADDRESS	OSL211

0004							
0BAR	096CR	2162	DC	C1AB,4	8 1 HW ABSOLUTE TEXT		OSL21580
0004							
0BER	0986R	2163	DC	C1RL,4	9 1 HW RELATIVE TEXT		OSL21590
0004							
0C2R	0996R	2164	DC	C2AB,8	A 2 HW ABSOLUTE TEXT		OSL21600
0008							
0C6R	0992R	2165	DC	C2RL,8	B 2 HW RELATIVE TEXT		OSL21610
0008							
0CAR	09DCR	2166	DC	CRFS,12	C REF-SYMBOL		OSL21620
000C							
0CER	0A3BR	2167	DC	CDFS,12	D DEF-SYMBOL		OSL21630
000C							
0D2R	084BR	2168	DC	CCOM,-1	E COMMON-DECODE FURTHER		OSL21640
FFFF							
0D6R	0AACR	2169	DC	CLAB,12	F LABEL-SYMBOL		OSL21650
000C							
	2170	* COMMON DECODE-TABLE FOR SECOND DIGIT FOLLOWING E ITEMS ABOVE					OSL21660
0DAR	0ABER	2171	COMTAB DC	CMDEF,16	E0 DEFINE COMMON BLOCK		OSL21670
0010							
0DER	0H5BR	2172	DC	CMREF,16	E1 REF-COMMON BLOCK		OSL21680
0010							
0E2R	0B86R	2173	DC	CM1DA,20	E2 LOAD COMMON 1 HW		OSL21690
0014							
0E6R	0B82R	2174	DC	CM2DA,24	E3 LOAD COMMON 2 HW		OSL21700
0018							
0EAR	0BEAR	2175	DC	CMSEQ,-1	E4 RESET SEQUENCE-NR		OSL21710
FFFF							
0EER		2176	COMEND EQU	*			OSL21720
		2177	* PARAMETER-BLOCKS FOR ALL SUPERVISOR-CALLS:				OSL21730
0EER	0007	2178	MFM DC	7,9,C' MEM-FULL'			OSL21740
0009							
2040							
4540							
2046							
554C							
4C20							
0FCR	0006	2179	UNPAK DC	6,TMP1			OSL21750
1002R							
100R	0007	2180	CMERR DC	7,8,C' CMD-ERR'			OSL21760
0008							
2043							
4044							
2045							
5252							
100CR	0007	2181	UNPRG DC	7,19,C' * * NO PROGRAMS * *'			OSL21770
0013							
2A20							
2A20							
4E4F							
2050							
524F							
4752							
4140							
5320							
2A20							

1124R	2A20 0007 0004 2045 4F46	2182	MEOFS	DC	7.4.C' EOF'	END OF FILE MESSAGE	OSL217
112CR	0007 0004 2045 4F4D	2183	MEOMS	DC	7.4.C' EOM'		OSL217
1134R	2800 0000	2184	MAPDEV	DC	WRITE+WAIT,0		OSL218
1138R	0000	2185	MAPLO	DC	0		OSL218
113AR	0000	2186	MAPHI	DC	0		OSL218
113CR	0C00 0000 0000 0000 0000 2050 524F 4752 414D 533A	2187	PRMS	DC	X'0C00',0,0,0,0,0,C' PROGRAMS:'		OSL218
1152R	0A0A 2045 4E54 5259 2D50 4F49 4E54 533A	2188	EPMS	DC	X'0A0A',C' ENTRY-POINTS:'		OSL218
1162R	0A0A 2043 4F4D 4D4F 4E2D 424C 4F43 4B53 3A20	2189	CRMS	DC	X'0A0A',C' COMMON-BLOCKS:'		OSL218
1174R	0A0A 2055 4E44 4546 494E 4544 3A20	2190	USMS	DC	X'0A0A',C' UNDEFINED:'		OSL218
1182R	204E 4F4E 4520	2191	NMS	DC	C' NONE'		OSL218
1188R	0007 0009 2043 4B53 4D20	2192	MSCKSM	DC	7.9.C' CKSM ERR'		OSL218



4552						
5220						
196R 0007	2193	MSSEQ	DC	7,8,C' SEQ ERR'		OSL21890
0008						
2053						
4551						
2045						
5252						
1A2R 0007	2194	MSSLOP	DC	7,16,C' REF-LOOP'		OSL21900
0010						
2052						
4546						
204C						
4F4F						
5020						
1B0R	2195	SYMBUF	DS	6		OSL21910
1B6R 2800	2196	MSMDF	DC	WRITE+WAIT,0,**+4,MSND		OSL21920
0000						
11BER						
11C7R						
1BER 2040	2197		DC	C' M SSSSSS'		OSL21930
2020						
5353						
5353						
5353						
1C7R	2198	MSND	EQU	*-1		OSL21940
1C2R	2199	MSYMB	EQU	MSND-5		OSL21950
1C8R 0007	2200	LDMS	DC	7,7,C' LOADER'		OSL21960
0007						
204C						
4F41						
4445						
5220						
1D4R 0007	2201	ICM	DC	7,9		OSL21970
0009						
1D8R 204C	2202		DC	C' LOAD-ERR'		OSL21980
4F41						
4420						
4552						
5220						
1E2R 0007	2203	ABLM	DC	7,13,C' LOAD ABORTED'		OSL21990
0000						
204C						
4F41						
4420						
4142						
4F52						
5445						
4420						
1F4R 4805	2204	LDCMD	DC	READ+WAIT+5,0,BUF,BUF+79		OSL22000
0000						
1244R						
1293R						
1F4R 5800	2205	BINOV	DC	READ+BINARY+WAIT,0,BUF,BUF+107		OSL22010
0000						
1244R						

1204R	12AFR 3800 0000 12B0R 131BR	2206	BOUTDV	DC	WRITE+WAIT+BINARY,0,0BUF,0BUF+107	OSL221
120CR	0007 0009 2041 4452 532D 4552 5220	2207	ADERM	DC	7,9,C' ADRS-ERR'	OSL221
121AR	C000 0000	2208	RWD	DC	X'C000',0	OSL221
121ER	0800 0000	2209	COWT	DC	WAIT,0	OSL221
1222R	A000 0000	2210	RSB	DC	X'A000',0	OSL221
1226R	8800 0000	2211	FMB	DC	X'8800',0	OSL221
122AR	8400 0000	2212	SFB	DC	X'8400',0	OSL221
122ER	3000 0000 1244R	2213	COOUT	DC	WRITE+BINARY,0,BUF,BUF+107	OSL221
1236R	12AFR 2800 0000 1244R 1293R	2214	LOGBL	DC	X'2800',0,BUF,BUF+79	OSL221
0FBAR		2215	LOGGER	EQU	RPMSG	OSL221
123ER	FFFF	2216	LDFLG	DC	-1	OSL221
1240R	0007 0000	2217	ECHO	DC	7,0	OSL221
1244R	454E	2218	BUF	DC	C'EN'	OSL221
1246R		2219		DS	106	OSL221
000D		2220	CR	EQU	X'0D'	OSL221
0020		2221	SP	EQU	X'20'	OSL221
2000		2222	WRITE	EQU	X'2000'	OSL221
0800		2223	WAIT	EQU	X'0800'	OSL221
4000		2224	READ	EQU	X'4000'	OSL221
1000		2225	BINARY	EQU	X'1000'	OSL221
002C		2226	COMMA	EQU	X'2C'	OSL221
12B0R		2227	0BUF	DS	108	OSL221
		2294		IF	1	OSL221
131CR		2295	PROT	END	GOLOAD	OSL221



















```

* OS HEX DEBUG (OS CLUB), PROGRAM NUMBER 03-032, IS AN
* ON-LINE, INTERACTIVE HEXADECIMAL DEBUG PROGRAM
* THAT ASSISTS THE USER IN DEBUGGING HIS PROGRAM.
*
* THE USER DIRECTS OS CLUB THROUGH KEYBOARD COMMANDS
* AND OBTAINS MOST OS CLUB RESPONSES ON THE KEYBOARD/PRINTER
* DEVICE ASSIGNED TO LOGICAL UNIT 05.
* THE PRINT(P) AND DISASSEMBLE(N) DIRECTIVES OUTPUT TO
* LOGICAL UNIT NUMBER 03.
* THE OUTPUT (O) ABSOLUTE BINARY OBJECT TAPE AND THE
* (Q) OUTPUT BINARY CORE IMAGE TAPE DIRECTIVES WRITE TO
* LOGICAL UNIT NUMBER 02.
*
* OS CLUB RUNS ON ANY INTERDATA MODEL 3,4, OR 5 PROCESSOR
* THAT SUPPORTS AN OPERATING SYSTEM.
*
* OS CLUB REQUIRES APPROXIMATELY 3KB CORE MEMORY.
*
* PRIOR TO EXECUTION,
* THREE LOGICAL UNITS MUST BE ASSIGNED TO PHYSICAL DEVICES:
* LOGICAL UNIT NUMBER 02 = BINARY OUTPUT DEVICE
* LOGICAL UNIT NUMBER 03 = LIST OUTPUT DEVICE
* LOGICAL UNIT NUMBER 05 = KEYBOARD/PRINTER DEVICE
*
* STARTING LOCATION FOR EXECUTION IS AT ORIGIN(X'0000'R).
*
* OBJECT PROGRAM HAS ENTRY AND LABEL=OSCLUB

```

0000030  
0000040  
0000050  
0000060  
0000070  
0000080  
0000090  
0000100  
0000110  
0000120  
0000130  
0000140  
0000150  
0000160  
0000170  
0000180  
0000190  
0000200  
0000210  
0000220  
0000230  
0000240  
0000250  
0000260  
0000270  
0000280  
0000290  
0000300  
0000310  
0000320  
0000330  
0000340  
0000350  
0000360  
0000370  
0000380  
0000390  
0000400  
0000410  
0000420  
0000430  
0000440  
0000450  
0000460  
0000470  
0000480  
0000490  
0000490  
0000490  
0000490  
0000490  
0000490

0000R		ENTRY OSCLUB	
0000	R0	EQU	0
0000	INPUT	EQU	0
0001	OPEN	EQU	1
0002	P	EQU	2
0003	ONE	EQU	3
0004	CHAR	EQU	4
0005	TWO	EQU	5
0006	DEV	EQU	6
0007	STAT	EQU	7
0008	OUTPUT	EQU	8
0009	SHOW	EQU	9
000A	A	EQU	10
000B	B	EQU	11
000C	C	EQU	12
000D	D	EQU	13
000E	E	EQU	14
000F	F	EQU	15

0000R	4000	OSCLUB	STH	R0,SAVE0	SAVE ALL REGISTERS &
	016ER				
0004R	4100		BAL	R0,SAVER	INITIALIZE CONSTANTS
	00COR				
0008R	48A0		LH	A,OTRPSW	
	0096				
000CR	C4A0		NHI	A,X'7EFF'	
	7EFF				
0010R	40A0		STH	A,SAVOLD	
	0196R				

0014R	48A0	LH	A,TRPLOC	SAVE SVC15	553A*8/73	00000501
	00BA					
0018R	40A0	STH	A,SVC15	LOC CNTR	553A*8/73	00000502
	01A2R					
001CR	C8A0	LHI	A,BRKENT	SET ILLEGAL INSTRUCTION	553A*8/73	00000503
	043ER					
0020R	40A0	STH	A,TRPLOC	INTERUPT NEW PSW=BRKENT	553A*8/73	00000504
	00BA					
0024R	4010	STH	OPEN,BIAS	CLEAR BIAS		00000510
	0114R					00000520
						00000530
0028R	4120	* CLRTBL	BAL	P,BSERCH	CLEAR BREAKPOINT ENTRIES	
	042AR					
002CR	4010	STH	OPEN,BRKTBL+80(D)	ZERO OUT BRKTBL ADRS CELL		00000540
	016ER					
0030R	010C	BALR	RO,C	CONTINUE UNTIL DONE		00000550
						00000560
						00000570
0032R	C860	* TYCLUB	LHI	DEV,5	RESET DEVICE NO = TTY	
	0005					
0036R	01E8	BALR	E,OUTPUT	OUTPUT CR/LF		00000580
	01E2R	DC	A(CRLF)			00000590
0038R	01E8	BALR	E,OUTPUT	TYPEOUT CLUB ON TTY		00000600
	019CR	DC	A(MSCLUB)			00000610
003CR	0B11	SHR	OPEN,OPEN	CLOSE OPEN CELL IF ANY		00000620
						00000630
						00000640
0040R	C8F0	* READIN	LHI	F,0	INITIALIZE INDEX	
	0000					
0044R	C800	LHI	INPUT,0			00000650
	0000					
0048R	E110	SVC	1,SVCBL1	READ IN COMMAND		00000660
	00ACR					
004CR	48D0	LH	D,SVCBL1+2			00000670
	00AER					
0050R	4210	BM	SCANNER	READ ERROR		00000680
	00B4R					
0054R	D34F	READMX	LB	CHAR,BUFFER(F)	GET 8-BIT CHARACTER	00000690
	08EER					
0058R	CAF0	AHI	F,1	INCREMENT INDEX		00000700
	0001					
005CR	C540	CLHI	CHAR,C'0'	IS CHAR ALPHANUMERIC?		00000710
	0030					
0060R	4280	BL	SCAN	NO, SCAN FURTHER		00000720
	007CR					
0064R	C540	CLHI	CHAR,C'9'+1	IS CHAR NUMERIC?		00000730
	003A					
0068R	4280	BL	HEXNUM	YES, GOTO HEX ROUTINE		00000740
	009ER					
006CR	C540	CLHI	CHAR,C'A'	CHAR=HEXIDECIMAL ALPHA?		00000750
	0041					
0070R	4280	BL	SCAN	NO,SCAN FURTHER		00000760
	007CR					
0074R	C540	CLHI	CHAR,C'F'+1	CHAR=HEXIDECIMAL ALPHA?		00000770
	0047					
0078R	4280	BL	HEXALF	YES, GOTO HEXALF ROUTINE		00000780
	009AR					00000790

\*



BE0R	017ER 4090	STH	9,SAVE0+18	00001180
DE4R	0180R 40A0	STH	10,SAVE0+20	00001190
DE8R	0182R 40B0	STH	11,SAVE0+22	00001200
OECR	0184R 40C0	STH	12,SAVE0+24	00001210
OF0R	0186R 40D0	STH	13,SAVE0+26	00001220
OF4R	0188R 40E0	STH	14,SAVE0+28	00001230
OF8R	018AR 40F0	STH	15,SAVE0+30	00001240
	018CR			00001250
OFCR	C830	LHI	ONE,1 SET CONSTANTS	00001260
	0001			
100R	C850	LHI	TWO,2	00001270
	0002			
104R	C860	LHI	DEV,5 INITIALIZED DEVICE NO = TTY	00001280
	0005			
				00001290
				00001300
				00001310
				00001320
				00001330
				00001340
				00001350
				00001360
108R	C880	LHI	OUTPUT,OUTPR ALLOWS RR BALR TO OUTPUT	
	029AR			
010CR	C890	LHI	SHOW,SHOWA ALLOWS RR BRANCH TO SHOW	00001370
	0312R			
0110R	0B11	SHR	OPEN,OPEN INDICATES CLOSED CELL	00001380
0112R	0300	BR	R0 RETURN	00001410
				00001420
				00001430
				00001440
0114R	0000	BIAS	DC 0	00001450
0116R	0000	VALUE	DC 0	00001460
0118R	0000	MASK	DC 0	00001470
011AR	0000	LOWLIM	DC 0	00001480
011CR	0000	HGHLIM	DC 0	00001490
011ER		BRKTBL	DS 80	00001500
016ER		SAVE0	DS 32	00001510
018ER		HEX	DS 4	00001520
0192R	0000	SAVE	DC 0	00001530
0194R	0000		DC 0	00001540
0196R	0000	SAVOLD	DC 0	00001550
0198R	0000		DC 0	00001560
019AR		FLAG	DS 2	00001570
				00001580
				00001590
019CR	434C	MSCLUB	DC C*CLUB*,X*0D2E*	
	5542			
	0D2E			
01A2R		SVC15	DS 2 553A*8/73	00001595

01A4R	494C	MILL6L	DC	C'ILLEGAL INSTRUCTION.'	00001600
	4C45				
	4741				
	4C20				
	494E				
	5354				
	5255				
	4354				
	494F				
	4E2E				
01B8R	4E4F	MSGNO	DC	C'NO BREAKPOINT.'	00001610
	2042				
	5245				
	4148				
	504F				
	494E				
	542E				
01C6R	4558	MxCESS	DC	C'EXCESS '	00001620
	4345				
	5353				
	2020				
01CER	4252	MBKPNT	DC	C'BREAKPOINT.'	00001630
	4541				
	4B50				
	4F49				
	4E54				
	2E20				
01DAR	5245	REGNUM	DC	X'5245',X'4700',X'2E52'	00001640
	4700				
	2E52				
01E0R	202E		DC	X'202E'	00001650
01DFR		RSPACE	EQU	*-3	00001660
01E2R	0D2E	CRLF	DC	X'0D2E'	00001670
01E4R	2020	SPACE8	DC	X'2020'	00001680
01E6R	2020	SPACE6	DC	X'2020'	00001690
01E8R	2020	SPACE4	DC	X'2020'	00001700
01EAR	2020	SPACE2	DC	X'2020'	00001710
01ECR	2E2E		DC	X'2E2E'	00001720
01EER	282E	LPAREN	DC	C'(.'	00001730
01FOR	292E	RPAREN	DC	C').'	00001740
01F2R	2C2E	COMMA	DC	C',.'	00001750
01F4R	3F2E	QUEST	DC	X'3F2E'	00001760
					00001770
					00001780
					00001790
01F6R	A498	TTYIN	DC	X'A498'	00001800
01F7R		TTYOUT	EQU	TTYIN+1	00001810
					00001820
0096		OTRPSW	EQU	X'96'	00001830
0098		OTRLOC	EQU	X'98'	00001840
009A		TRPPSW	EQU	X'9A'	00001850
00BA		TRPLOC	EQU	X'BA'	00001860
					00001870
					00001880
007A		BOUADV	EQU	X'7A'	00001890
007E		LISTDV	EQU	X'7E'	00001900

\* TABLE OF DIRECTIVES

DTABLE DC X'0A0D',X'207F'

01F8R 0A0D  
 207F  
 01FCR 212B  
 2D2A  
 2F2E  
 3A47  
 4849  
 4A4B  
 4C4D  
 4E4F  
 5051  
 5253  
 5455  
 5657  
 5859  
 5A20

00001910  
 00001920  
 00001930

\* VECTOR TABLE OF SUBROUTINE ADDRESSES

VTABLE DC A(LINFED) LINE FEED  
 DC A(CARRET) CARRIAGE RETURN  
 DC A(SPACE) SPACE BAR  
 DC A(DELETE) RUBOUT  
 DC A(PAUSE) J  
 DC A(PLUS) +  
 DC A(MINUS) -  
 DC A(STAR) \*  
 DC A(SLASH) /  
 DC A(PERIOD) .  
 DC A(COLON) :  
 DC A(GO) G  
 DC A(SETHGH) H  
 DC A(SETBIS) I  
 DC A(TND) J  
 DC A(REMOVE) K  
 DC A(SETLOW) L  
 DC A(SETMSK) M  
 DC A(DISASM) N  
 DC A(OUTPTT) O  
 DC A(PRINT) P  
 DC A(OUTPTT) Q  
 DC A(REGSTR) R  
 DC A(SEARCH) S  
 DC A(TRANSF) T  
 DC A(ND) U  
 DC A(SETVAL) V  
 DC A(GOWAIT) W  
 DC A(SETBRK) X  
 DC A(SHOWTB) Y  
 DC A(ZAPBRK) Z  
 DC A(ND) ALL OTHER UNDEFINES  
 ND EQU READHX

00001940  
 00001950  
 00001960  
 00001970  
 00001980  
 00001990  
 00002000  
 00002010  
 00002020  
 00002030  
 00002040  
 00002050  
 00002060  
 00002070  
 00002080  
 00002090  
 00002100  
 00002110  
 00002120  
 00002130  
 00002140  
 00002150  
 00002160  
 00002170  
 00002180  
 00002190  
 00002200  
 00002210  
 00002220  
 00002230  
 00002240  
 00002250  
 00002260  
 00002270  
 00002280  
 00002290  
 00002300  
 00002310  
 00002320

\*  
 \* 'TEARP' CONVERTS THE 4-HEXIDECIMAL DIGITS CONTAINED IN THE  
 \* OPEN CELL TO ASCII CODE AND STORES IT IN LOC SPECIFIED BY  
 \* REG C REG C IS INCREMENTED  
 \*



0258R	48A1 0000	TEARP	LH	A,0(OPEN)	GET CONTENTS OF OPENCELL	00002330
		*				00002340
		*			* 'TEAR' CONVERTS THE 4-HEXIDECIMAL DIGITS CONTAINED IN	00002350
		*			* REGISTER A, ASSUMED TO BE PRESET BY THE CALLING ROUTINE.	00002360
		*				00002370
025CR	C8B0 FFFC	TEAR	LHI	B,-4	INITIALIZE TEAR(P) INDEX	00002380
0260R	084A	TEAR1	LHR	CHAR,A	LOAD CHAR W/CONTENTS OF A	00002390
0262R	CC40 000C		SRHL	CHAR,12	ALIGN MOST SIGNIFICANT 4BITS	00002400
0266R	CA40 0030		AHI	CHAR,C'0'	COPY IN ASCII DIGIT 3	00002410
026AR	C540 003A		CLHI	CHAR,C'9'+1	DIGIT=HEXADECIMAL ALPHA?	00002420
026ER	4280 0276R		BL	TEAR2	NO, SKIP CONVERTING TOALFA	00002430
0272R	CA40 0007		AHI	CHAR,7	CONVERT TO ASCII ALPHA	00002440
0276R	D24B 0192R	TEAR2	STB	CHAR,HEX+4(B)	STR CONVERTED DIGIT IN HEX	00002450
027AR	CDA0 0004		SLHL	A,4	ALIGN NEXT DIGIT MS 4BITS	00002460
027ER	0AB3		AHR	B,ONE	INCREMENT NEG. INDEX	00002470
0280R	4210 0260R		BTC	1,TEAR1	LOOP TIL 4 DIGITS DONE	00002480
0284R	48D0 018ER		LH	D,HEX		00002490
0288R	40DC 0000		STH	D,0(C)	STORE IN BUFFER	00002500
028CR	48D0 0190R		LH	D,HEX+2		00002510
0290R	40DC 0002		STH	D,2(C)		00002520
0294R	CAC0 0004		AHI	C,4		00002530
0298R	030E		BR	E	RETURN	00002540
		*				00002550
		*			* THE CALLING SEQUENCE FOR 'OUTPR' WHOSE ADDRESS IS IN	00002560
		*			* REGISTER OUTPUT IS: BALR E, OUTPUT FOLLOWED BY A DATA	00002570
		*			* CONSTANT ENTRY CONTAINING THE ADDRESS OF ANY MESSAGE	00002580
		*			* ENDING WITH 2E(PERIOD) OUTPR OUTPUTS TO THE DEVICE IN	00002590
		*			* REGISTER DEV ASSUMED PRESET BY THE CALLING ROUTINE.	00002600
		*				00002610
029AR	48BE 0000	OUTPR	LH	B,0(E)	GET ADDRESS OF MESSAGE	00002620
029ER	D260 02B5R		STB	DEV,SVCBL2+1	STORE LOGICAL UNIT	00002630
02A2R	D348 0000	OUTPR1	LB	CHAR,0(B)	GET 1BYTE MESSAGE CONTENTS	00002640
02A6R	C540 002E		CLHI	CHAR,C'.'	END OF MESSAGE?	00002650
02AAR	4330 02BCR		BE	OUTPR2		00002660
02AER	0AB3		AHR	B,ONE	GET ADDRESS OF NEXT BYTE	00002670
02BOR	4300		B	OUTPR1	LOOP	00002680

02B4R	02A2R 2800	SVCBL2	DC	X+2800*		00002690
02B6R	0000		DC	0		00002700
02B8R	0000		DC	0		00002710
02BAR	0000		DC	0		00002720
02BCR	C8B0	OUTPR2	SHI	B,1		00002730
02C0R	0001 40B0		STH	B,SVCBL2+6	FINAL ADDRESS	00002740
02C4R	02BAR 48BE		LH	B,0(E)		00002750
02C8R	0000 40B0		STH	B,SVCBL2+4	START ADDRESS	00002760
02CCR	02B8R E110		SVC	1,SVCBL2		00002770
02D0R	02B4R 48D0		LH	D,SVCBL2+2		00002780
02D4R	02B6R 4210		BM	SCANNER	WRITE ERROR	00002790
02D8R	00B4R 430E		B	2(E)		00002800
	0002					00002810
						00002820
						00002830
02DCR	C820		SLASH	LHI	P,READIN	PREPARE SHOCNT EXIT
02E0R	0040R C8C0		LHI	C,BUFFER		00002840
02E4R	08EER 48D0		LH	D,BIAS	BIAS ENTERED? NO,	00002850
02E8R	0114R 4330		BZ	SHOCNT	RESHOW CONTENTS ABSOLUTE	00002860
02ECR	0368R 48A1		LH	A,0(OPEN)	YES, GET OPENCELL CONTENTS	00002870
02FOR	0000 08A0		SHR	A,D	SUBTRACT BIAS	00002880
02F2R	41E0		BAL	E,TEAR	SHOW RELATIVE CONTENTS	00002890
02F6R	025CR 48C0		LH	C,RSPACE		00002900
02FAR	01DFR 40C0		STH	C,BUFFER+4		00002910
02FER	08F2R 48C0		LH	C,RSPACE+2		00002920
0302R	01E1R 40C0		STH	C,BUFFER+6		00002930
0306R	08F4R 01E8		BALR	E,OUTPUT	SHOW RSPACE	00002940
0308R	08EER		DC	A(BUFFER)		00002950
030AR	0302		BR	P	RETURN FOR NEW INPUT	00002960
030CR	0810		SPACE	LHR	OPEN,INPUT	OPEN CELL IN INPUT
030ER	4A10		AH	OPEN,BIAS	MAKE IT ABSOLUTE	00002980
	0114R					00002990
						00003000
						00003010
						00003020
						00003030
						00003040

\* THE 'SHOW' ROUTINE OUTPUTS THE ADDRESS AND CONTENTS  
 \* OF THE OPEN/SPECIFIED CELL IN REGISTER OPEN.  
 \* ALL ADDRESSES GREATER THAN OR EQUAL TO THE BIAS  
 \* (IF A BIAS IS ENTERED) ARE OUTPUT RELATIVE TO THAT

```

* BIAS WITH AN R APPENDED.
* IF NO BIAS IS ENTERED, THE ADDRESS IS SHOWN ABSOLUTE.
* IF THE OPENCELL IS WITHIN CLUB'S REGISTER SAVE AREA,
* 'SHOW' OUTPUTS REGN XXXX WHERE N=USER'S SAVED
* REGISTER NUMBER AND XXXX=ITS SAVED CONTENTS.
0312R C820      * SHOWA LHI P,READIN      SET EXIT=RETURN FOR INPUT      00003050
      0040R
0316R C8C0      LHI C,BUFFER      00003060
      08EER
* PRINT/SEARCH/DISASM ROUTINES CALL SHOWC
031AR 08A1      SHOWC LMR A,OPEN      SAVE ABS ADRS OF OPEN CELL      00003130
031CR C510      CLHI OPEN,SAVE0      IS OPEN CELL A REGISTER      00003140
      016ER
0320R 4280      BL SHOADD      NO, GO SHOW ADDRESS      00003160
      032CR
0324R C510      CLHI OPEN,SAVE0+32      IS OPEN CELL A REGISTER?      00003170
      018ER
0328R 4280      BL SHOREG      YES, GO SHOW REGNUM&CONT      00003180
      0390R
032CR 4510      SHOADD CLM OPEN,BIAS      IF ADRS OF OPENCELL<BIAS      00003190
      0114R
0330R 4280      BL SHOABS      DON'T SUBTRACT&GO SHOABS      00003200
      0352R
0334R 48D0      LH D,BIAS      IS A BIAS ENTERED?      00003210
      0114R
0338R 4330      BZ SHOABS      NO, GO HANDLE ABS ADDRESS      00003220
      0352R
033CR 0BAD      SHR A,D      GET RELATIVE ADDRESS      00003230
033ER 41E0      BAL E,TEAR      SEND OUT 4 HEX DIGITS      00003240
      025CR
0342R C8D0      LHI D,C,R      00003250
      5220
0346R 40DC      STH D,0(C)      00003260
      0000
034AR CAC0      AHI C,2      00003270
      0002
034ER 4300      B PQUEST      ASK IF PRINT CALLED SHOW      00003280
      0362R
0352R 41E0      SHOABS BAL E,TEAR      SEND OUT 4 HEX DIGITS      00003290
      025CR
0356R 48D0      SHOSP2 LH D,SPACE2      00003300
      01EAR
035AR 40DC      STH D,0(C)      00003310
      0000
035ER CAC0      AHI C,2      00003320
      0002
0362R C520      PQUEST CLHI P,PRETRN      IF PRETRN=P,ONLY SHOW ADRS      00003330
      062CR
0366R 0332      BFCR 3,P      TAKE SPECIAL EXIT TO PRNT      00003340
* THE SLASH ROUTINE BRANCHES TO SHOCNT
0368R 41E0      SHOCNT BAL E,TEARP      STORE CONTENTS      00003350
      0258R
036CR 48D0      LH D,SPACE2      00003370
      01EAR
0370R 40DC      STH D,0(C)      00003380

```

0000				00003390
0374R	CAC0	AHI	C,2	
0002				00003400
0378R	C520	CLHI	P,READIN IF P=READIN, TYPE MESSAGE	
0040R				00003410
037CR	0232	BTCR	S,P	00003420
037ER	48D0	LH	D,SPACE2+2 STORE.	
01ECR				00003430
0382R	40DC	STH	D,0(C)	
0000				00003440
0386R	CAC0	AHI	C,2	
0002				00003450
038AR	01E8	BALR	E,OUTPUT OUTPUT	00003460
038CR	08EER	DC	A(BUFFER)	00003470
038ER	0302	BR	P	00003480
0390R	CBA0	SHOREG SHI	A,SAVE0 GET REGISTER NUMBER	
016ER				00003490
0394R	CCA0	SRHL	A,1	
0001				00003500
0398R	C4A0	NHI	A,X'F' MASK OUT HEX DIGIT	
000F				00003510
039CR	CAA0	AHI	A,X'30' ADD IN NUMERIC ASCII CODE	
0030				00003520
03A0R	C5A0	CLMI	A,X'3A' DIGIT=HEXADECIMAL ALPHA7	
003A				00003530
03A4R	4280	BL	SHONUM NO,NUMERIC SKIP	
03ACR	03ACR			00003540
03A8R	CAA0	AHI	A,7 CONVERT TO ASCII ALPHA	
0007				00003550
03ACR	D2A0	SHONUM STB	A,REGNUM+3 STR CODE IN REG MESSAGE	
01DDR				00003560
03B0R	48D0	LH	D,REGNUM STORE RE	
01DAR				00003570
03B4R	40DC	STH	D,0(C)	
0000				00003580
03B8R	48D0	LH	D,REGNUM+2 STORE GR	
01DCR				00003590
03BCR	40DC	STH	D,2(C)	
0002				00003600
03C0R	CAC0	AHI	C,4	
0004				00003610
03C4R	4300	B	SHOSP2	
0356R				00003620
03C8R	0B11	* [RUBOUT] DIRECTIVE "CLOSE OPEN CELL"		00003630
		DELETE SHR OPEN,OPEN	CLOSE THE OPEN CELL	00003640
		* THE DIRECTIVE ROUTINES THAT BRANCH TO NOSHOW ARE		00003650
		* THE RUBOUT, L, H, I, M, AND V DIRECTIVES.		00003660
		* 'NOSHOW' OUTPUTS A CARRIAGE RETURN AND LINEFEED AND		00003670
		* RETURNS TO READIN TO CLEAR THE INPUT BUFFER REGISTER		00003680
		* AND WAIT FOR FURTHER INPUT		00003690
		* NOSHOW		00003700
03CAR	01E8	BALR	E,OUTPUT OUTPUT CR AND LF	00003710
03CCR	01E2R	DC	A(CRLF)	00003720
03CER	4300	B	READIN RETURN FOR FURTHER INPUT	00003730
0040R				

03D2R	4000	* [H] DIRECTIVE "HIGHLIMIT"			00003740
	011CR	SETHGH	STH	INPUT,HGHLIM	00003750
03D6R	4300	B		NOSHOW	00003760
	03CAR				
03DAR	4000	* [L] DIRECTIVE "LOWLIMIT"			00003770
	011AR	SETLOW	STH	INPUT,LOWLIM	00003780
03DER	4300	B		NOSHOW	00003790
	03CAR				
03E2R	4000	* [I] DIRECTIVE "BIAS"			00003800
	0114R	SETBIS	STH	INPUT,BIAS	00003810
03E6R	4300	B		NOSHOW	00003820
	03CAR				
03EAR	4000	* [M] DIRECTIVE "MASK"			00003830
	0118R	SETMSK	STH	INPUT,MASK	00003840
03EER	4300	B		NOSHOW	00003850
	03CAR				
03F2R	4000	* [V] DIRECTIVE "VALUE"			00003860
	0116R	SETVAL	STH	INPUT,VALUE	00003870
03F6R	4300	B		NOSHOW	00003880
	03CAR				
03FAR	4811	* [T] DIRECTIVE "TRANSFER TO CONTENTS"			00003890
	0000	TRANSF	LH	OPEN,0(OPEN)	00003900
03FER	0309	BR		SHOW	00003910
0400R	C400	* [R] DIRECTIVE "REGISTER"			00003920
	000F	REGSTR	NHI	INPUT,X'F'	00003930
0404R	CD00	SLHL		INPUT,1	00003940
	0001	PRECEDING 'R' DIRECTIVE			
0408R	CA00	AHI		INPUT,SAVE0	00003950
	016ER	GET HW ADRS OF SAVED REG			
040CR	0810	LHR		OPEN,INPUT	00003960
040ER	0309	BR		SHOW	00003970
		SHOW 'REGN & CONTENTS'			
0410R	0B15	* [CARRIAGE RETURN] DIRECTIVE "OPEN PRECEDING CELL"			00003980
0412R	0309	CARRET	SHR	OPEN,TWO	00003990
		BR		SHOW	00004000
0414R	0A15	* [LINEFEED] DIRECTIVE "OPEN NEXT SEQUENTIAL CELL"			00004010
0416R	0309	LINFED	AHR	OPEN,TWO	00004020
		BR		SHOW	00004030
0418R	4A00	* [:] DIRECTIVE "MODIFY CONTENTS RELATIVE"			00004040
	0114R	COLON	AH	INPUT,BIAS	00004050
		ADD REL BIAS+INPUT			
041CR	4001	* [.] DIRECTIVE "MODIFY CONTENTS ABSOLUTE"			00004060
	0000	PERIOD	STH	INPUT,0(OPEN)	00004070
0420R	0309	BR		SHOW	00004080
0422R	0A10	* [+] DIRECTIVE "ADD"			00004090
0424R	0309	PLUS	AHR	OPEN,INPUT	00004100
		BR		SHOW	00004110
0426R	0B10	* [-] DIRECTIVE "SUBTRACT"			00004120
		MINUS	SHR	OPEN,INPUT	00004130
		OPEN CELL @A(OPEN-INPUT)			

0428R	0309	BR	SHOW		00004140	
		*	* 'BSERCH' PERFORMS A LOOP OBTAINING EVERY OTHER CELL IN			00004150
		*	* BRKTBL. AS EACH CELL IS OBTAINED,BSERCH RETURNS TO THE			00004160
		*	* CALLING ROUTINE FOR A DECISION DEPENDING ON THE CELL'S			00004170
		*	* CONTENTS. THAT ROUTINE WILL RETURN TO BSERCH EITHER THRU			00004180
		*	* R0 OR HAVE R0 PRESET FOR AN EXIT.			00004190
		*				00004200
042AR	C8D0	BSERCH	LHI	D,-80	INITIALIZE BRKTBL INDEX	00004210
	FFB0					00004220
042ER	48AD	BLOOP	LH	A,BRKTBL+80(D)	PICKUP ADRS CELL IN BRKTBL	00004230
	016ER					
0432R	01C2	BALR	C,P		RETURN TO ROUTINE QUESTION	00004240
0434R	CAD0	AHI	D,10		INCREMENT BRKTBL INDEX	00004250
	000A					
0438R	4230	BNZ	BLOOP		DONE LOOKING THRU BRKTBL?	00004260
	042ER					
043CR	0300	BR	R0		YES.EXIT	00004270
		*	* THE ADDRESS OF 'BRKENT' IS STORED IN THE PROCESSOR'S			00004280
		*	* ILLEGAL INSTRUCTION TRAP, LOCATION X'BA' WHEN CLUB			00004290
		*	* IS STARTED AT ORIGIN. 'BRKENT' IS ENTERED ANY TIME THE			00004300
		*	* PROCESSOR EXECUTES A BREAKPOINT OR OTHER ILLEGAL			00004310
		*	* INSTRUCTION. THEREAFTER.			00004320
		*	* THE ROUTINES THAT CALL ON 'BSERCH' ARE 'CLR7BL',			00004330
		*	* 'BRKENT', AND THE DIRECTIVE ROUTINES FOR X,Z,K, AND Y.			00004340
		*	* SETBRK, ZAPBRK, REMOVE, SHOWTB.			00004350
		*				00004360
043ER	4000	BRKENT	STH	R0,SAVE0	SAVE ALL REGISTERS	00004370
	016ER					00004380
0442R	4100	BAL	R0,SAVER		& INITIALIZE CONSTANTS	00004390
	00C0R					
0446R	4810	LH	1,SVCFLG			00004400
	07BCR					
044AR	4310	BNM	BRKEN1			00004410
	0458R					
044ER	0711	XHR	1,1			00004420
0450R	4010	STH	1,SVCFLG			00004430
	07BCR					
0454R	4300	B	SVCEXT			00004440
	075ER					
0458R	48C0	BRKEN1	LH	C,OTRPSW		00004450
	0096					
045CR	40C0	STH	C,SAVOLD			00004460
	0196R					
0460R	C4C0	NHI	C,X'7EFF'			00004461
	7EFF					
0464R	40C0	STH	C,ENRET	SAVE STATUS & CC	555A*8/73	00004465
	0478R					
0468R	48C0	LH	C,OTRLOC			00004470
	0098					
046CR	C8C0	SHI	C,4			00004480
	0004					
0470R	40C0	STH	C,SAVOLD+2			00004490
	0198R					
0474R	C200	LPSW	**4			00004500

04E6R	04F6R C580 9000	CLHI	B,X'9000'		00004860
04EAR	4280 0512R	BL	STSCND	RX,RS OP=40-6D	00004870
04EER	C580 C000	CLHI	B,X'C000'		00004880
04F2R	4380 0512R	BNL	STSCND	RX,RS OP=C0-DF	00004890
04F6R	08B1	RTNST	LHR	B,OPEN	RR OP=01-2D,90-9F
04F8R	CAB0 0002		AHI	B,2	RETURN ADD+2
04FCR	40BD 0174R		STH	B,BRKTBL+86(D)	STORE
0500R	C880		LHI	B,X'4300'	00004930
0504R	4300 40BD		STH	B,BRKTBL+84(D)	STORE BRANCH INSTRUCTION
0508R	0172R C880	SETFLG	LHI	B,X'E1F0'	GET BREAKPOINT INSTR
050CR	E1F0 40B1		STH	B,0(OPEN)	SET BRKPNT INTO OPENCCELL
0510R	0000 0309		BR	SHOW	SHOW IT
0512R	48B1	STSCND	LH	B,2(OPEN)	GET 2ND HALF RX,RS INSTR
0516R	0002 40BD		STH	B,BRKTBL+84(D)	STORE
051AR	0172R 08B1		LHR	B,OPEN	
051CR	CAB0 0004		AHI	B,4	RETURN ADD +4
0520R	40BD		STH	B,BRKTBL+86(D)	STORE
0524R	0176R C880		LHI	B,X'4300'	
0528R	4300 40BD		STH	B,BRKTBL+86(D)	STORE BRANCH INSTR
052CR	0174R 4300 0508R		B	SETFLG	
0530R	41E0	* [Z] DIRECTIVE *ZAP BREAKPOINT*			00005060
0534R	04A0R 0811	ZAPBRK	BAL	E,TSTINP	TEST INPUT BUFFER/OPENCCELL
0536R	4330		LHR	OPEN,OPEN	DONT TEST CELL ZERO
053AR	0546R 4120		BZ	NOBRK	FOR BREAKPOINT ENTRY
053ER	042AR 051A		BAL	P,BSERCH	SEARCH BRKTBL FOR ENTRY
0540R	4330		CLHR	OPEN,A	DOES OPENCCELL MATCH ENTRY
0544R	054CR 010C		BE	ZCLEAR	IF SO,RESTORE IT&ZAP ENTRY
0546R	01E8	NOBRK	BALR	RO,C	RETURN TO BSERCH&SETUPEXIT
0548R	0188R		BALR	E,OUTPUT	TYPE 'NO'
054AR	0309		DC	A(MSGNO)	FINISH TYPING
054CR	08E9	ZCLEAR	BR	SHOW	RESHOW OPEN CELL
054ER	48BD 0170R	KCLEAR	LHR	E,SHOW	SETUP ZCLEAR EXIT TO SHOW
			LH	B,BRKTBL+82(D)	GET ORIGINAL CONTENTS

0552R	40B1	STH	B,0(OPEN)	RESTORE THEM INTO OPENCELL		00005190
	0000					
0556R	08BB	SHR	B,B	GET A ZERO TO		00005200
0558R	40B0	STH	B,BRKTBL+80(D)	CLEAR BRKTBL ENTRY ADDRS		00005210
	016ER					
055CR	030E	BR	E	EXIT TO SHOW/BACKTO REMOVE		00005220
		* [K] DIRECTIVE "KILL ALL BREAKPOINTS"				00005230
055ER	C800	REMOVE LHI	R0,TYCLUB	SETUP BSERCH EXIT R0		00005240
	0032R					
0562R	4120	BAL	P,BSERCH	SEARCH BRKTBL FOR ENTRY		00005250
	042AR					
0566R	033C	BFCR	3,C	IF NONE,RETURN TO BSERCH		00005260
0568R	081A	LHR	OPEN,A	OPEN ENTRY		00005270
056AR	41E0	BAL	E,KCLEAR	RESTORE CELL & ZAP ENTRY		00005280
	054ER					
056ER	030C	BR	C	RETURN TO BSERCH&CONTINUE		00005290
		* END DIRECTIVE *				00005300
0570R	E120	STAR	SVC	2,SVCBL8		00005310
	0580R					
0574R	48F0	LH	F,SVC15	RESTORE SVC 15	553A*8/73	00005320
	01A2R					
0578R	40F0	STH	F,TRPLOC	LOC CNTR	553A*8/73	00005330
	00BA					
057CR	E130	SVC	3,0			00005340
	0000					
0580R	0005	SVCBL8 DC	5			00005350
0582R	000F	DC	X'F'			00005360
		* PAUSE DIRECTIVE				00005370
0584R	E120	PAUSE SVC	2,#+8			00005380
	058CR					
0588R	4300	B	READIN			00005390
	0040R					
058CR	0001	DC	I			00005400
		* [Y] DIRECTIVE "SHOW BREAKPOINT LOCATIONS"				00005410
058ER	C800	SHOWTB LHI	R0,TYCLUB	SETUP BSERCH EXIT R0		00005420
	0032R					
0592R	4120	BAL	P,BSERCH	SEARCH BRKTBL FOR ENTRY		00005430
	042AR					
0596R	033C	BFCR	3,C	IF NONE,RETURN TO BSERCH		00005440
0598R	40C0	STH	C,SAVE			00005450
	0192R					
059CR	40D0	STH	D,SAVE+2			00005460
	0194R					
05A0R	C8C0	LHI	C,BUFFER			00005470
	08EER					
05A4R	41E0	BAL	E,TEAR	SEND ABS ADRS OF BRKPT		00005480
	025CR					
05A8R	48B0	LH	B,SPACE2+2			00005490
	01EER					
05ACR	40B0	STH	B,BUFFER+4			00005500
	08F2R					
05B0R	01E8	BALR	E,OUTPUT			00005510
05B2R	08EER	DC	A(BUFFER)			00005520
05B4R	48C0	LH	C,SAVE			00005530
	0192R					
05B8R	48D0	LH	D,SAVE+2			00005540



Address	Label	Op	Op2	Op3	Description	Address
0194R						00005550
05BCR 030C	BR	C			RETURN TO BSEARCH&CONTINUE	00005560
	*				* 'LIMITS' SETS UP THE LOW & HIGH LIMITS RELATIVE TO THE	00005570
	*				* BIAS AND FORCES HALFWORD ALIGNMENT, FOR S (SEARCH), P (PRINT)	00005580
	*				* N (DISASM), AND THE G AND O (OUTPUT TAPE) ROUTINES	00005590
	*					00005600
05BER 4810	LIMITS	LH	OPEN,LOWLIM		PREPARE LOW & HIGH LIMITS	00005610
011AR						
05C2R 4800	LIMITS1	LH	RO,BIAS		GET BIAS	00005620
0114R						
05C6R 0A10	AHR		OPEN,RO		OPEN = LOWLIM + BIAS	00005630
05C8R 4A00	AH		RO,HGHLIM		RO = HGHLIM + BIAS	00005640
011CR						
05CCR C410	NHI		OPEN,X'FFFE'		FORCE HALFWORD ALIGNMENT	00005650
FFFE						
05D0R C400	NHI		RO,X'FFFE'		ON LOWLIMIT & HIGHLIMIT	00005660
FFFE						
05D4R 030E	BR	E			RETURN	00005670
	*	[S]	DIRECTIVE "SEARCH FOR MASKED VALUE"			00005680
05D6R 41E0	SEARCH	BAL	E,LIMITS		GET LOW & HIGH LIMITS	00005690
05DAR 48C0	SHERE	LH	C,VALUE		PICK UP VALUE	00005700
0116R						
05DER 4820		LH	P,MASK		PICK UP MASK	00005710
0118R						
05E2R 04C2	NHR		C,P		MASK VALUE	00005720
05E4R 48A1	LH		A,0(OPEN)		GET CONTENTS	00005730
0000						
05E8R 04A2	NHR		A,P		MASK CONTENTS	00005740
05EAR 07AC	XHR		A,C		DOES MASKED CONTENT EQUAL	00005750
05ECR 4230	BNZ		SRETRN		MASKED VALUE? NO,SKIP	00005760
0604R						
05FOR C8C0	LHI		C,BUFFER			00005770
08EER						
05F4R 4120	BAL		P,SHOWC		FOUND A MATCH,SHOW IT	00005780
031AR						
05F8R 48A0	LH		A,SPACE2+2			00005790
01ECR						
05FCR 40AC	STH		A,0(C)			00005800
0000						
0600R 01E8	BALR		E,OUTPUT			00005810
0602R 08EER	DC		BUFFER			00005820
0604R 0A15	SRETRN	AHR	OPEN,TWO		OPEN NEXT CELL	00005830
0606R 0510	CLHR		OPEN,RO		DONE SEARCHING LIMITS	00005840
0608R 4330	BE		**8			00005850
0610R						
060CR 4380	BNL		TYCLUB			00005860
0032R						
0610R 4300	B		SHERE		NO, CONTINUE SEARCH	00005870
05DAR						
	*	[P]	DIRECTIVE "PRINT"			00005880
0614R 4810	PRINT	LH	OPEN,LOWLIM		GET LOWLIMIT ENTERED	00005890
011AR						
0618R C8C0	LHI		C,BUFFER			00005900
08EER						

061CR	C410 FFF0	NHI	OPEN,X'FFF0'	MASK OFF HEX LSD OF LOWLIM	00005910	
0620R	41E0 05C2R	BAL	E,LIMITS1	FINISH SETTING UP LIMITS	00005920	
0624R	C860 0003	LHI	DEV,3		00005930	
0628R	4120 031AR	PRINT1	BAL	P,SHOWC	SHOW LINE'S 1ST CELL ADRS	00005940
062CR	48D0 01EAR	PRETRM	LH	D,SPACE2		00005950
0630R	40DC 0000	STH	D,0(C)		00005960	
0634R	CAC0 0002	AHI	C,2		00005970	
0638R	C8F0 0008	LHI	F,8		00005980	
063CR	4120 0368R	PRINT8	BAL	P,SHOCNT	SHOW CONTENTS OF NEXT 8	00005990
0640R	0A15	AHR	OPEN,TWO	CONSECUTIVE CELLS	00006000	
0642R	0BF3	SHR	F,ONE		00006010	
0644R	4230	BNZ	PRINT8	DONE PRINTING ONE LINE?	00006020	
0648R	063CR 48D0 01ECR	LH	D,SPACE2+2		00006030	
064CR	40DC 0000	STH	D,0(C)		00006040	
0650R	01E8	BALR	E,OUTPUT		00006050	
0652R	08EER	DC	A(BUFFER)		00006060	
0654R	0510	CLHR	OPEN,R0	PASSED UPPER LIMIT?	00006070	
0656R	4330 065ER	BE	*+8		00006080	
065AR	4380 0032R	BNL	TYCLUB		00006090	
065ER	C8C0 08EER	LHI	C,BUFFER		00006100	
0662R	4300 0628R	B	PRINT1	NO, CONTINUE ON NEXT LINE	00006110	
0666R	4820	* [W] DIRECTIVE "GOWAIT"			00006120	
	0196R	GOWAIT	LH	P,SAVOLD	00006130	
066AR	C620 8000	OHI	P,X'8000'		00006140	
066ER	4020 0196R	STH	P,SAVOLD		00006150	
0672R	4300 0682R	B	STRPSW		00006160	
0676R	4820	* [G] DIRECTIVE "GO EXECUTE"			00006170	
	0196R	GO	LH	P,SAVOLD	00006180	
067AR	C420 7FFF	NHI	P,X'7FFF'		00006190	
067ER	4020 0196R	STH	P,SAVOLD	555A*8/73	00006195	
0682R	41E0 04A0R	STRPSW	BAL	E,TSTINP	TEST INPUT FOR ENTRY	00006200
0686R	4010	STH	OPEN,SAVOLD+2	SAVE ADRS FOR LPSW	00006210	

068AR	0198R 01E8	BALR	E,OUTPUT	TELL USER G/W ACCEPTED &		00006220
068CR	01E2R	DC	A(CRLF)	CLUB IS BEING EXITED		00006230
068ER	4120	BAL	P,BSERCH	SEARCH BRKTBL		00006240
0692R	042AR 051A	CLHR	OPEN,A			00006250
0694R	4330	BE	GFIX	BREAKPOINT FOUND		00006260
0698R	069ER 010C	BALR	R,C	CONTINUE SEARCH		00006270
069AR	4300	B	GFIX1			00006280
069ER	0750R C8AD	GFIX	LHI	A,BRKTBL+82(D)	LOAD A WITH LOCAT OF WHERE	00006290
	0170R	*		PRGRM IS TO RTN TO EXECUTE BKPT		00006300
06A2R	40A0	STH	A,SAVOLD+2	STORE		00006310
06A6R	0198R 48AD	LH	A,BRKTBL+82(D)	CHECK FOR RR INSTR.		00006320
06AAR	0170R C4A0	NHI	A,X'FF00'			00006330
06AER	FF00 C5A0	CLHI	A,X'4100'	IS IT A BAL?	556*8/73	00006331
06B2R	4100 4330	BE	BALRTN	YES,PROCESS IT	556*8/73	00006332
06B6R	0766R C5A0	CLHI	A,X'0100'	IS IT A BALR?	556*8/73	00006333
06BAR	0100 4330	BE	BALRTN	YES,PROCESS IT	556*8/73	00006334
06BER	0766R C5A0	GFIX0	CLHI	A,X'2400'	POSSIBLE SHORT BRANCH?	534A*8/73
06C2R	2400 4380	BNL	GFIX1	NO	534A*8/73	00006342
06C6R	0750R C5A0	CLHI	A,X'2000'	DEFINITE SHORT BRANCH?	534A*8/73	00006343
06CAR	2000 4280	BL	GFIX1	NO	534A*8/73	00006344
06CER	0750R 48CD	LH	C,BRKTBL+80(D)	OPEN CELL ADDRESS	534A*8/73	00006345
06D2R	016ER C8E0	LHI	E,X'4200'		534A*8/73	00006346
06D6R	4200 C4A0	NHI	A,X'0200'	BRANCH FALSE?	534A*8/73	00006347
06DAR	0200 4330	BZ	GFIX0A	NO	534A*8/73	00006348
06DER	06E2R C6E0	OMI	E,X'0100'	YES-SET OPCODE	534A*8/73	00006349
06E2R	0100 D3AD	GFIX0A	LB	A,BRKTBL+82(D)	OP CODE	534A*8/73
06E6R	0170R D3BD	LB	B,BRKTBL+83(D)	MASK	534A*8/73	00006351
06EAR	0171R CCB0	SRHL	B,4	SHIFT OFF INCR.	534A*8/73	00006352
06EER	0004 CDB0	SLHL	B,4	POSITION MASK	534A*8/73	00006353
06F2R	0004 06EB	OHR	E,B	INSER INTO LONG OPCODE	534A*8/73	00006354
06F4R	D3BD	LB	B,BRKTBL+83(D)	ADR INCREMENT	534A*8/73	00006355

06F8R	0171R C4B0		NHI	B,X'000F'	STRIP MASK OFF	534A*8/73	00006356
06FCR	000F C0B0		SLHL	B,1	INDEXIZE INCR	534A*8/73	00006357
0700R	0001 C4A0		NHI	A,X'0001'	FWD OR BK7L	534A*8/73	00006358
0704R	0001 4230		BNZ	GFIX0B	FWD	534A*8/73	00006359
0708R	070ER 0BCB		SHR	C,B	BK, DECR ADR	534A*8/73	00006360
070AR	4300		B	GFIX0C	534A*8/73		00006361
070ER	0710R 0ACB	GFIX0B	AHR	C,B	INCREMENT ADR	534A*8/73	00006362
0710R	40E0	GFIX0C	STH	E,BRKSHT	PRIME OPCODE & MASK	534A*8/73	00006363
0714R	072CR 40C0		STH	C,BRKSHT+2	PRIME BRANCH ADR	534A*8/73	00006364
0718R	072ER 40C0		LH	C,BRKTBL+86(D)	534A*8/73		00006365
071CR	0174R 40C0		STH	C,BRKSHT+6	PRIME FALL-THRU	534A*8/73	00006366
0720R	0732R C8A0		LHI	A,BRKSHT	EXECUTION ADR	534A*8/73	00006367
0724R	072CR 40A0		STH	A,SAVOLD+2	PRIME PSW	534A*8/73	00006368
0728R	0198R 4300		B	GFIX1	534A*8/73		00006369
072CR	0750R 0000	BRKSHT	DC	0	BRANCH CONDITIONALLY	534A*8/73	00006370
072ER	0000		DC	0	BRANCH ADR	534A*8/73	00006371
0730R	4300		DC	X'4300'	UNCONDITIONAL BRANCH	534A*8/73	00006372
0732R	0000		DC	0	INLINE ADR	534A*8/73	00006373
0734R	C5A0		CLHI	A,X'9000'			00006374
0738R	9000 4280		BL	GFIX2			00006375
073CR	0744R C5A0		CLHI	A,X'C000'			00006380
0740R	C000 4280		BL	GFIX1	RR		00006390
0744R	0750R 48AD	GFIX2	LH	A,BRKTBL+80(D)	RX,RS		00006400
0748R	016ER 48AA		LH	A,2(A)			00006410
074CR	0002 40AD		STH	A,BRKTBL+84(D)			00006420
0750R	0172R C810	GFIX1	LHI	1,X'8000'			00006430
0754R	8000 4010		STH	1,SVCFL6			00006440
0758R	07BCR D100		LM	0,SAVE0			00006450
075CR	016ER E1F0		DC	X'E1F0'			00006460
075ER	D100	SVCEXT	LM	0,SAVE0			00006465
0762R	016ER C200		LPSW	SAVOLD	EXIT CLUB		00006470
	0196R						

0766R	48AD	BALRTN	LH	A,BRKTBL+82(D)	INSTR HW	556*8/73	00006471
	0170R						
076AR	C4A0		NHI	A,X'00F0'	R1	556*8/73	00006472
	00F0						
076ER	CCA0		SRHL	A,3	DOUBLE FOR INDEX	556*8/73	00006473
	0003						
0772R	48CD		LH	C,BRKTBL+80(D)	ADR OF BRK PT	556*8/73	00006474
	016ER						
0776R	48ED		LH	E,BRKTBL+82(D)	INSTR HW	556*8/73	00006475
	0170R						
077AR	CAC0		AHI	C,2	556*8/73		00006476
	0002						
077ER	C4E0		NHI	E,X'4000'	BAL OR BALR?	556*8/73	00006477
	4000						
0782R	4330		BZ	**8	BALR	556*8/73	00006478
	078AR						
0786R	CAC0		AHI	C,2	BAL	556*8/73	00006479
	0002						
078AR	48ED		LH	E,BRKTBL+82(D)	GET LINK REG	556*8/73	00006480
	0170R						
078ER	40CA		STH	C,SAVE0(A)	SAVE REG	556*8/73	00006481
	016ER						
0792R	07CC		XHR	C,C	556*8/73		00006482
0794R	C4E0		NHI	E,X'000F'	GET INDEX REG	556*8/73	00006483
	000F						
0798R	4330		BZ	NOINDX	NO INDEX	556*8/73	00006484
	07A4R						
079CR	CDE0		SLHL	E,1	DOUBLE IT	556*8/73	00006485
	0001						
07A0R	48CE		LH	C,SAVE0(E)	GET REG CONTENTS	556*8/73	00006486
	016ER						
07A4R	48AD	NOINDX	LH	A,BRKTBL+82(D)	INSTR HW	556A*8/73	00006487
	0170R						
07A8R	C4A0		NHI	A,X'4000'	BAL OR BALR?	556A*8/73	00006488
	4000						
07ACR	4330		BZ	**8	BALR - NO BASE	556A*8/73	00006489
	07B4R						
07B0R	4ACD		AH	C,BRKTBL+84(D)	ADD BASE ADR	556A*8/73	00006490
	0172R						
07B4R	40C0		STH	C,SAVOLD+2	SET FOR LPSW	556A*8/73	00006491
	0198R						
07B8R	4300		B	GFIX1	DO IT ALL	556A*8/73	00006492
	0750R						
07BCR	0000	SVCFLG	DC	0			00006495

\* REDEFINE REGISTERS FOR OUTPUT TAPE ROUTINES

0000	HILIM	EQU	R0	00006500
0002	BUFFX	EQU	2	00006510
0004	INDEX2	EQU	CHAR	00006520
0006	COUNT	EQU	DEV	00006530
000A	SEQNUM	EQU	A	00006540
000B	PACK	EQU	B	00006550
000C	BITX	EQU	C	00006560
000D	DATA	EQU	D	00006570
000E	LINK	EQU	E	00006580
000F	LINK1	EQU	F	00006590
				00006600
				00006610

		* [O] DIRECTIVE "DUMP TAPE IN STANDARD LOADER FORMAT"	00006620
		* [Q] DIRECTIVE "DUMP TAPE IN 8-BIT FORMAT"	00006630
07BER	41E0	OUTPTT BAL E,LIMITS	00006640
07C2R	05BER C540	CLHI CHAR,X'0051'	00006650
07C6R	0051 4330	BE Q8BIT	00006660
07CAR	095AR C8B0	LHI PACK,PACKA	00006670
07CER	0856R 0BAA	SHR SEQNUM,SEQNUM	00006680
07D0R	41E0	BAL LINK,CLRBUF	00006690
07D4R	08D2R C8D0	LHI DATA,X'0035'	00006700
07D8R	0035 D2D0	STB DATA,BUFFER+4	00006710
07DCR	08F2R 08D1	LHR DATA,OPEN	00006720
07DER	CCD0	SRHL DATA,8	00006730
07E2R	0008 D2D0	STB DATA,BUFFER+8	00006740
07E6R	08F3R D210	STB OPEN,BUFFER+6	00006750
07EAR	08F4R C820	LHI BUFFX,14	00006760
07EER	000E C501	QUEST1 CLHI HILIM,2(OPEN)	00006770
07F2R	0002 4280	BL QUEST2	00006780
07F6R	081CR C520	CLHI BUFFX,207	00006790
07FAR	00CF 4280	BL PACFUL	00006800
07FER	0802R 41E0	BAL LINK,PUNCH	00006810
0802R	08AAR C840	PACFUL LHI CHAR,X'A'	00006820
0806R	000A 01EB	BALR LINK,PACK	00006830
0808R	41F0	BAL LINK1,HALFW	00006840
080CR	087ER 41F0	BAL LINK1,HALFW	00006850
0810R	087ER 4300	B QUEST1	00006860
0814R	07EER 3802	SVCBL3 DC X'3802'	00006870
0816R	0002	DC X'2'	00006880
0818R	08EER	DC A(BUFFER)	00006890
081AR	0959R	DC A(BUFFER+107)	00006900
081CR	0510	QUEST2 CLHR OPEN,HILIM	00006910
081ER	4230	BNE QUEST3	00006920
0822R	0838R C520	CLHI BUFFX,211	00006930
0826R	00D3 4280	BL PACHAF	00006940
	082ER		

082AR	41E0		BAL	LINK,PUNCH	NO, OUTPUT FILLED BUFFER	00006950
	08AAR					
082ER	C840	PACHAF	LHI	CHAR,X'8'		00006960
	0008					
0832R	01EB		BALR	LINK,PACK	PACK CONTROL ITEM '8'	00006970
0834R	41F0		BAL	LINK1,HALFW	PACK HALFWORD	00006980
	087ER					
0838R	C520	QUEST3	CLHI	BUFFX,215	BUFFER HAVE ROOM ENDCODE?	00006990
	00D7					
083CR	4280		BL	ENDCOD	YES, GO PACK IT	00007000
	0844R					
0840R	41E0		BAL	LINK,PUNCH	NO, OUTPUT FILLED BUFFER	00007010
	08AAR					
0844R	C840	ENDCOD	LHI	CHAR,3	PACK ABSOLUTE*RELATIVE	00007020
	0003					
0848R	01EB		BALR	LINK,PACK	TOGGLE SWITCH = 3	00007030
084AR	0843		LHR	CHAR,ONE	PACK END CODE = 1	00007040
084CR	01EB		BALR	LINK,PACK		00007050
084ER	41E0		BAL	LINK,PUNCH	OUTPUT LAST BUFFER	00007060
	08AAR					
0852R	4300	CUTOFF	B	TYCLUB		00007070
	0032R					
0856R	08C2	PACKA	LHR	BITX,BUFFX	THIS SUBROUTINE PACKS	00007080
0858R	CCC0		SRHL	BITX,1	THE RIGHTMOST 4-BITS	00007090
	0001					
085CR	4280		BC	PACK1	IN CHAR INTO THE NEXT	00007100
	0864R					
0860R	CD40		SLHL	CHAR,4	AVAILABLE 4-BIT BYTE	00007110
	0004					
0864R	CCC0	PACK1	SRHL	BITX,1	IN THE BUFFER	00007120
	0001					
0868R	4280		BC	PACK2		00007130
	0870R					
086CR	CD40		SLHL	CHAR,8		00007140
	0008					
0870R	0ACC	PACK2	AHR	BITX,BITX		00007150
0872R	464C		OH	CHAR,BUFFER(BITX)		00007160
	08EER					
0876R	404C		STH	CHAR,BUFFER(BITX)		00007170
	08EER					
087AR	0A23		AHR	BUFFX,ONE		00007180
087CR	030E		BR	LINK		00007190
087ER	C860	HALFW	LHI	COUNT,4	THIS SUBROUTINE PACKS	00007200
	0004					
0882R	48D1		LH	DATA,0(OPEN)	A HALFWORD FROM CORE,	00007210
	0000					
0886R	084D	LOOPHW	LHR	CHAR,DATA	THE OPEN CELL, INTO THE	00007220
0888R	CC40		SRHL	CHAR,12	NEXT AVAILABLE 2 BYTES OR	00007230
	000C					
088CR	01EB		BALR	LINK,PACK	16 BITS IN THE BUFFER	00007240
088ER	CDD0		SLHL	DATA,4		00007250
	0004					
0892R	0B63		SHR	COUNT,ONE		00007260
0894R	4230		BNZ	LOOPHW		00007270
	0886R					
0898R	0A15		AHR	OPEN,TWO	GET NEXT CELL OPEN	00007280

089AR	030F		BR	LINK1		00007290
089CR	E110	WRITE	SVC	1,SVCBL3		00007300
	0814R					
08A0R	48D0		LH	D,SVCBL3+2		00007310
	0816R					
08A4R	4210		BM	SCANNER		00007320
	00B4R					
08A8R	030F		BR	LINK1	RETURN	00007330
08AAR	C8D0	PUNCH	LHI	DATA,-1	SETUP A WORD OF ALL ONES	00007340
	FFFF					
08AER	C840		LHI	INDEX2,4		00007350
	0004					
08B2R	C860		LHI	COUNT,107	SETUP ENDOFBUFFER COUNT	00007360
	006B					
08B6R	47D0		XH	DATA,BUFFER	COMPUTE CHECKSUM BY XH	00007370
	08EER					
08BAR	47D4	CHKSUM	XH	DATA,BUFFER(INDEX2)	ALL WORDS IN BUFFER	00007380
	08EER					
08BER	C140		BXLE	INDEX2,CHKSUM	EXCEPT ITSELF PLUS	00007390
	08BAR					
		*			A WORD OF ALL ONES	00007400
08C2R	40D0		STH	DATA,BUFFER+2	STORE CHECKSUM IN BUFFER	00007410
	08FOR					
08C6R	E110		SVC	1,SVCBL3		00007420
	0814R					
08CAR	48D0		LH	D,SVCBL3+2		00007430
	0816R					
08CER	4210		BM	SCANNER		00007440
	00B4R					
08D2R	0BDD	CLRBUF	SHR	DATA,DATA	CLEAR BUFFER ROUTINE	00007450
	08D4R		SHR	INDEX2,INDEX2		00007460
08D6R	C860		LHI	COUNT,107		00007470
	006B					
08DAR	40D4	CLEAR	STH	DATA,BUFFER(INDEX2)	ZERO OUT HALFWORD	00007480
	08EER					
08DER	C140		BXLE	INDEX2,CLEAR	DONE CLEARING BUFFER?	00007490
	08DAR					
08E2R	0BA3		SHR	SEQNUM,ONE	DECREMENT RECORD SEQNUM	00007500
08E4R	40A0		STH	SEQNUM,BUFFER	STORE IT	00007510
	08EER					
08E8R	C820		LHI	BUFFX,8	8 = INDICATES FIRST BUFFER	00007520
	000B					
		*			4-BITS AVAILABLE FOR DATA	00007530
08ECR	030E		BR	LINK		00007540
		*				00007550
08EER		BUFFER	DS	108		00007560
		*				00007570
095AR	0821	Q8BIT	LHR	BUFFX,OPEN	BUFFX=EVEN LOWLIMIT	00007580
095CR	0603		OHR	HILIM,ONE	FORCE HIGHLIMIT ODD	00007590
095ER	0840		LHR	INDEX2,HILIM	INDEX2=ODD HIGHLIMIT	00007600
0960R	4020		STH	BUFFX,SVCBL4+4		00007610
	097CR					
0964R	4040		STH	INDEX2,SVCBL4+6		00007620
	097ER					
0968R	E110		SVC	1,SVCBL4		00007630
	0978R					





09D6R	CAC0	AHI	C,6		00007980
	0006				
09DAR	082A	LHR	P,A	SAVE MNEMONIC ADRS&RXFLAG	00007990
09DCR	C4A0	NHI	A,X,FFFE	MASK OFF RX FLAG	00008000
	FFFF				
09E0R	480A	LH	D,0(A)		00008010
	0800				
09E4R	400C	STH	D,0(C)		00008020
	0000				
09E8R	400A	LH	D,2(A)		00008030
	0002				
09EER	400C	STH	D,2(C)		00008040
	0002				
09FER	CAC0	AHI	C,4		00008050
	0004				
09F0R	48E0	LH	E,SPACE2		00008060
	01E0				
09F4R	40EC	STH	E,0(C)		00008070
	0000				
09F8R	CAC0	AHI	C,2		00008080
	0002				
0A00R	D340	LB	CHAR,HEX+2	GET R1/M1 FIELD	00008090
	0190				
0A04R	D24C	STB	CHAR,0(C)		00008100
	0000				
0A08R	D340	LB	CHAR,COMMA		00008110
	01F2				
0A0CR	D24C	STB	CHAR,1(C)		00008120
	0001				
0A10R	CAC0	AHI	C,2		00008130
	0002				
0A24R	D340	LB	CHAR,HEX+3	GET R2/X2 FIELD	00008140
	0191				
0A18R	0423	NHR	P,ONE	RXFLAG PRESENT?	00008150
0A1AR	4230	BNZ	RX	YES, GO HANDLE RX,RS INST	00008160
	0A42				
0A1ER	D24C	STB	CHAR,0(C)		00008170
	0000				
0A22R	D340	LB	CHAR,SPACE2+2		00008180
	01E0				
0A26R	D24C	STB	CHAR,1(C)		00008190
	0001				
0A2AR	01E8	BALR	E,OUTPUT		00008200
0A2CR	08EE	DC	A(BUFFER)		00008210
0A2ER	4300	B	NEXT	RETURN FOR NEXT INSTR	00008220
	098A				
0A32R	4840	NEXT1	LH	CHAR,SPACE2+2	00008230
	01E0				
0A36R	404C	STH	CHAR,0(C)		00008240
	0000				
0A3AR	01F8	BALR	E,OUTPUT		00008250
0A3CR	08EE	DC	A(BUFFER)		00008260
0A3ER	4300	B	NEXT		00008270
	098A				
0A42R	0824	RX	LHR	P,CHAR	00008280
0A44R	0A15		AHR	OPEN,TWO	00008290
				SAVE X2 FIELD	
				GET ADRS NEXT HALFWORD	

0A46R	41E0	BAL	E,TEARP	SEND IT OUT AS'A'FIELD	00008300
	0258R				
0A4AR	C520	CLHI	P,C'0'	WAS X2 FIELD ZERO?	00008310
	0030				
0A4ER	4330	BE	RX2		00008320
	0AB0R				
0A52R	C8D0	LHI	D,C'('		00008330
	0028				
0A56R	D2DC	STB	D,0(C)		00008340
	0000				
0A5AR	C8D0	LHI	D,C'('		00008350
	0029				
0A5ER	D2DC	STB	D,2(C)		00008360
	0002				
0A62R	0842	LHR	CHAR,P	GET X2 FIELD AGAIN	00008370
0A64R	D24C	STB	CHAR,1(C)		00008380
	0001				
0A68R	CAC0	AHI	C,3		00008390
	0003				
0A6CR	48D0	LH	D,SPACE2+2		00008400
	01E0R				
0A70R	D2DC	STB	D,0(C)		00008410
	0000				
0A74R	01E8	BALR	E,OUTPUT		00008420
0A76R	08EER	DC	A(BUFFER)		00008430
0A78R	C8C0	LHI	C,BUFFER		00008440
	08EER				
0A7CR	48D0	LH	D,SPACE2		00008450
	01EAR				
0A80R	40DC	STH	D,0(C)		00008460
	0000				
0A84R	40DC	STH	D,2(C)		00008470
	0002				
0A88R	40DC	STH	D,4(C)		00008480
	0004				
0A8CR	CAC0	AHI	C,5		00008490
	0006				
0A90R	48D0	LH	D,HEX		00008500
	018ER				
0A94R	40DC	STH	D,0(C)		00008510
	0000				
0A98R	48D0	LH	D,HEX+2		00008520
	0190R				
0A9CR	40DC	STH	D,2(C)		00008530
	0002				
0AA0R	48D0	LH	D,SPACE2+2		00008540
	01E0R				
0AA4R	40DC	STH	D,4(C)		00008550
	0004				
0AA8R	01E8	BALR	E,OUTPUT		00008560
0AAA	08EER	DC	A(BUFFER)		00008570
0AACR	4300	B	NEXT		00008580
	098AR				
0AB0R	48D0	LH	D,SPACE2+2		00008590
	01E0R				
0AB4R	40DC	STH	D,0(C)		00008600

RX0

RX2

0000				
0AB8R	01E8	BALR	E,OUTPUT	00008610
0ABAR	08EER	DC	A(BUFFER)	00008620
0ABCR	4300	B	RX0	00008630
	0A78R			
		* CLUB'S DISASSEMBLY TABLE OF MNEMONIC OP CODE ADDRESSES		00008640
0001		RXF	EQU 1	00008650
0AC0R	0AE0R	TABLE	DC A(MSD0),0	00008660
	0000			
0AC4R	0B20R		DC A(MSD2),0	00008670
	0000			
0AC8R	0B61R		DC A(MSD4+RXF),0	00008680
	0000			
0ACCR	0BA1R		DC A(MSD6+RXF),0,0	00008690
	0000			
0AD2R	0BE0R		DC A(MSD9),0,0	00008700
	0000			
	0000			
0AD8R	0C21R		DC A(MSDC+RXF),A(MSDD+RXF),A(MSDE+RXF),0	00008710
	0C61R			
	0CA1R			
	0000			
		* MNEMONIC OP CODES LISTED UNDER THEIR RESPECTIVE * MOST SIGNIFICANT DIGIT		00008720
0AE0R	FFFF	MSD0	DC -1,-1	00008730
	FFFF			00008740
0AE4R	4241		DC C*BALR*	00008750
	4C52			
0AE8R	4254		DC C*BTCR*	00008760
	4352			
0AECR	4246		DC C*BFCR*	00008770
	4352			
0AF0R	4E48		DC C*NHR *	00008780
	5220			
0AF4R	434C		DC C*CLHR*	00008790
	4852			
0AF8R	4F48		DC C*OHR *	00008800
	5220			
0AFCR	5848		DC C*XHR *	00008810
	5220			
0B00R	4C48		DC C*LHR *	00008820
	5220			
0B04R	4348		DC C*CHR *	00008830
	5220			
0B08R	4148		DC C*AHR *	00008840
	5220			
0B0CR	5348		DC C*SHR *	00008850
	5220			
0B10R	4D48		DC C*MHR *	00008860
	5220			
0B14R	4448		DC C*DHR *	00008870
	5220			
0B18R	4143		DC C*ACHR*	00008880
	4852			
0B1CR	5343		DC C*SCHR*	00008890

0B20R	4852 4254	MSD2	DC	C'BTBS'	00008900
0B24R	4253 4254		DC	C'BTFS'	00008910
0B28R	4653 4246		DC	C'BFBS'	00008920
0B2CR	4253 4246		DC	C'BFFS'	00008930
0B30R	4653 4C49		DC	C'LIS'	00008940
0B34R	5320 4C43		DC	C'LCS'	00008950
0B38R	5320 4149		DC	C'AIS'	00008960
0B3CR	5320 5349		DC	C'SIS'	00008970
0B40R	5320 4C45		DC	C'LER'	00008980
0B44R	5220 4345		DC	C'CER'	00008990
0B48R	5220 4145		DC	C'AER'	00009000
0B4CR	5220 5345		DC	C'SER'	00009010
0B50R	5220 4D45		DC	C'MER'	00009020
0B54R	5220 4445		DC	C'DER'	00009030
0B58R	5000 FFFF		DC	-1,-1	00009040
0B5CR	FFFF FFFF		DC	-1,-1	00009050
0B60R	5354 4820	MSD4	DC	C'STH'	00009060
0B64R	4241 4C20		DC	C'BAL'	00009070
0B68R	4254 4320		DC	C'BTC'	00009080
0B6CR	4246 4320		DC	C'BFC'	00009090
0B70R	4E48 2020		DC	C'NH'	00009100
0B74R	434C 4820		DC	C'CLH'	00009110
0B78R	4F48 2020		DC	C'OH'	00009120
0B7CR	5848 2020		DC	C'XH'	00009130
0B80R	4C48 2020		DC	C'LNH'	00009140
0B84R	4548 2020		DC	C'CH'	00009150
0B88R	4148 2020		DC	C'AH'	00009160
0B8CR	5348		DC	C'SH'	00009170

0B90R	2020 4048		DC	C'MH °	00009180
0B94R	2020 4448		DC	C'DH °	00009190
0B98R	2020 4143		DC	C'ACH °	00009200
0B9CR	4820 5343		DC	C'SCH °	00009210
0BA0R	4820 5354	MSD6	DC	C'STE °	00009220
0BA4R	4520 4148		DC	C'AHM °	00009230
0BA8R	4020 FFFF		DC	-1,-1	00009240
0BACR	FFFF FFFF		DC	-1,-1	00009250
0BB0R	FFFF 4154		DC	C'ATL °	00009260
0BB4R	4C20 4142		DC	C'ABL °	00009270
0BB8R	4C20 5254		DC	C'RTL °	00009280
0BBCR	4C20 5242		DC	C'RBL °	00009290
0BC0R	4C20 4045		DC	C'LE °	00009300
0BC4R	2020 4345		DC	C'CE °	00009310
0BC8R	2020 4145		DC	C'AE °	00009320
0BCCR	2020 5345		DC	C'SE °	00009330
0BD0R	2020 4045		DC	C'ME °	00009340
0BD4R	2020 4445		DC	C'DE °	00009350
0BD8R	2020 FFFF		DC	-1,-1	00009360
0BDCR	FFFF FFFF		DC	-1,-1	00009370
0BE0R	FFFF 5352	MSD9	DC	C'SRLS°	00009380
0BE4R	4C53 534C		DC	C'SLLS°	00009390
0BE8R	4C53 5354		DC	C'STBR°	00009400
0BECR	4252 4C42		DC	C'LBR °	00009410
0BF0R	5220 4558		DC	C'EXBR°	00009420
0BF4R	4252 4550		DC	C'EPSR°	00009430
0BF8R	5352 5742		DC	C'WBR °	00009440
0BFCR	5220 5242		DC	C'RBR °	00009450

5220				
0C00R 5748		DC	C*WHR *	00009460
5220				
0C04R 5248		DC	C*RHR *	00009470
5220				
0C06R 5744		DC	C*WDR *	00009480
5220				
0C0CR 5244		DC	C*RDR *	00009490
5220				
0C10R 4048		DC	C*MHR *	00009500
5552				
0C14R 5353		DC	C*SSR *	00009510
5220				
0C16R 4F43		DC	C*OCR *	00009520
5220				
0C1CR 4149		DC	C*AIR *	00009530
5220				
0C20R 4258	MSUC	DC	C*BXH *	00009540
4820				
0C24R 4256		DC	C*BXLE *	00009550
4045				
0C28R 4050		DC	C*LPSW *	00009560
5357				
0C2CR 5448		DC	C*THI *	00009570
4920				
0C30R 4E48		DC	C*NHI *	00009580
4920				
0C34R 434C		DC	C*CLHI *	00009590
484E				
0C38R 4F48		DC	C*OHI *	00009600
4920				
0C3CR 5848		DC	C*XHI *	00009610
4920				
0C40R 4C46		DC	C*LHI *	00009620
4920				
0C44R 434E		DC	C*CHI *	00009630
4920				
0C48R 4148		DC	C*AHI *	00009640
4920				
0C4CR 5348		DC	C*SHI *	00009650
4920				
0C50R 5352		DC	C*SRHL *	00009660
484C				
0C54R 534C		DC	C*SLHL *	00009670
484C				
0C58R 5352		DC	C*SRHA *	00009680
4841				
0C5CR 534C		DC	C*SLHA *	00009690
4841				
0C60R 5354	MSDD	DC	C*STM *	00009700
4820				
0C64R 4C48		DC	C*LM *	00009710
2020				
0C68R 5354		DC	C*STB *	00009720
4220				
0C6CR 4C42		DC	C*LB *	00009730

2020				
0C70R 434C		DC	C'CLB *	0000974
4220				
0C74R 414C		DC	C'AL *	0000975
2020				
0C78R 5742		DC	C'WB *	0000976
2020				
0C7CR 5242		DC	C'RB *	0000977
2020				
0C80R 5738		DC	C'WB *	0000978
2020				
0C84R 5248		DC	C'RH *	0000979
2020				
0C88R 5744		DC	C'WD *	0000980
2020				
0C8CR 5244		DC	C'RD *	0000981
2020				
0C90R 4048		DC	C'MHU *	0000982
5520				
0C94R 5353		DC	C'SS *	0000983
2020				
0C98R 4F43		DC	C'OC *	0000984
2020				
0C9CR 4149		DC	C'AI *	0000985
2020				
0CA0R FFFF	MSDE	DC	-1,-1	0000986
FFFF				
0CA4R 5356		DC	C'SVC *	0000987
4320				
0CA8R 5349		DC	C'SINT *	0000988
4E54				
0CACR		DO	7	0000989
0CACR FFFF		DC	-1,-1	0000990
FFFF				
0CB0R FFFF		DC	-1,-1	0000990
FFFF				
0CB4R FFFF		DC	-1,-1	0000990
FFFF				
0CB8R FFFF		DC	-1,-1	0000990
FFFF				
0CBCR FFFF		DC	-1,-1	0000990
FFFF				
0CC0R FFFF		DC	-1,-1	0000990
FFFF				
0CC4R FFFF		DC	-1,-1	0000990
FFFF				
0CC8R 5252		DC	C'RRL *	0000991
4C20				
0CCCR 524C		DC	C'RLL *	0000992
4C20				
0CD0R 5352		DC	C'SRL *	0000993
4C20				
0CD4R 534C		DC	C'SLL *	0000994
4C20				
0CD8R 5352		DC	C'SRA *	0000995
4120				



OCDCR 534C  
4120  
OCEOR

DC C'SLA '  
END OSCLUB

00009960  
00009970

## NO ERRORS

A	000A
B	000B
BALRTN	0766R
BIAS	0114R
BITX	000C
BLOOP	042ER
BOUADV	007A
BRKEN1	0458R
BRKENT	043ER
BRKPNT	0496R
BRKSHT	072CR
BRKTBL	011ER
BSECH	042AR
BUFFER	08EER
BUFFX	0002
C	000C
CARRET	0410R
CHAR	0004
CHKSUM	08BAR
CLEAR	080AR
CLRBUF	08D2R
CLRTBL	0028R
COLON	0418R
COMMA	01F2R
COUNT	0006
CRLF	01E2R
CUTOFF	0852R
D	000D
DATA	000D
DELETE	03C8R
DEV	0006
DISASM	0980R
DTABLE	01F8R
E	000E
ENDCOD	0844R
ENRET	0478R
ERROR	0490R
F	000F
FLAG	019AR
FOUND	0092R
GFIX	069ER
GFIX0	06BER
GFIX0A	06E2R
GFIX0B	070ER
GFIX0C	0710R
GFIX1	0750R
GFIX2	0744R
GO	0676R
GOWAIT	0666R
HALFW	087ER
HEX	018ER
HEXALF	009AR
HEXNUM	009ER
HGHLIM	011CR
HILIM	000D

INDEX2 0004  
INPUT 0000  
KCLEAR 054ER  
LIMITS 058ER  
LIMITS1 05C2R  
LINFED 0414R  
LINK 000E  
LINK1 000F  
LISTDV 007E  
LOOPHW 0886R  
LOWLIM 011AR  
LPAREN 01EEA  
MASK 0118R  
MBKPNT 01CER  
MILLGL 01A4R  
MINUS 0426R  
MSCLUB 019CR  
MSD0 0AE0R  
MSD2 0B20R  
MSD4 0B60R  
MSD6 0BA0R  
MSD9 0BE0R  
MSDC 0C20R  
MSDD 0C60R  
MSDE 0CA0R  
MSGNO 01B8R  
MXCESS 01C6R  
ND 0054R  
NEXT 098AR  
NEXT1 0A32R  
NOBRK 0546R  
NOINDX 07A4R  
NOSHOW 03CAR  
ONE 0003  
OPEN 0001  
\* OSCLUB 0000R  
OTRLOC 0098  
OTRPSW 0096  
OUTPR 029AR  
OUTPR1 02A2R  
OUTPR2 02D0CR  
OUTPTT 078ER  
OUTPUT 0008  
P 0002  
PACFUL 0802R  
PACHAF 082ER  
PACK 0008  
PACK1 0864R  
PACK2 0870R  
PACKA 0856R  
PAUSE 0584R  
PERIOD 041CR  
PLUS 0422R  
PQUEST 0362R  
PRETRN 062CR  
PRINT 0614R

PRINT1 0628R  
PRINT8 063CR  
PUNCH 08AAR  
Q8BIT 095AR  
QUEST 01F4R  
QUEST1 07EER  
QUEST2 081CR  
QUEST3 0838R  
R0 0000  
READHX 0054R  
READIN 0040R  
REGNUM 010AR  
REGSTR 0400R  
REMOVE 055ER  
RESTRT 047CR  
RPAREN 01F0R  
RSPACE 010FR  
RTNST 04F6R  
RX 0A42R  
RX0 0A78R  
RX2 0A80R  
RXF 0001  
SAVE 0192R  
SAVE0 016ER  
SAVER 00C0R  
SAVOLD 0196R  
SCAN 007CR  
SCAN1 007ER  
SCANNER 00B4R  
SEARCH 05D6R  
SEQNUM 000A  
SETBIS 03E2R  
SETBK1 04ACR  
SETBRK 049CR  
SETFLG 0508R  
SETHGH 03D2R  
SETLOW 03DAR  
SETMSK 03EAR  
SETVAL 03F2R  
SHERE 05DAR  
SHOABS 0352R  
SHOADD 032CR  
SHOCNT 0368R  
SHONUM 03ACR  
SHOREG 0390R  
SHOSP2 0356R  
SHOW 0009  
SHOWA 0312R  
SHOWC 031AR  
SHOWTB 058ER  
SLASH 02DCR  
SPACE 030CR  
SPACE2 01EAR  
SPACE4 01E8R  
SPACE6 01E6R  
SPACE8 01E4R

SRETRN	0604R
STAR	0570R
STAT	0007
STRCEL	04CER
STRPSW	0682R
STSCND	0512R
SVC15	01A2R
SVCBL1	00ACR
SVCBL2	02B4R
SVCBL3	0814R
SVCBL4	0978R
SVCBL8	0580R
SVCEXT	075ER
SVCFLG	07BCR
TABLE	0AC0R
TEAR	025CR
TEAR1	0260R
TEAR2	0276R
TEARP	0258R
TRANSF	03FAR
TRPLOC	00BA
TRPPSW	009A
TSTINP	04A0R
TTYIN	01F6R
TTYOUT	01F7R
TWO	0005
TYCLUB	0032R
VALUE	0116R
VTABLE	0218R
WRITE	089CR
ZAPBRK	0530R
ZCLEAR	054CR

0478R	0478R	ENRET	DC	X'0000'	DISABLE INTERRUPTS	555A+8/73	00004510
047AR	047CR		DC	A(RESTRT)			00004520
047CR	4810	RESTRT	LH	OPEN,SAVOLD+2			00004530
	0198R						
0480R	4330		BZ	ERROR	IF CELL ZERO,ILLEGAL		00004540
	0490R						
0484R	4120		BAL	P,BSERCH	SEARCH BRKTBL FOR ENTRY		00004550
	042AR						
0488R	051A		CLHR	OPEN,A	DOES OPENCELL MATCH ENTRY		00004560
048AR	4330		BE	BRKPNT	YES,GO SHOW 'BREAKPOINT'		00004570
	0496R						
048ER	010C		BALR	R0,C	RETURN TO BSERCH&SETUPEXIT		00004580
0490R	01E8	ERROR	BALR	E,OUTPUT	TYPE 'ILLEGAL INSTRUCTION'		00004590
0492R	01A4R		DC	A(MILLGL)			00004600
0494R	0309		BR	SHOW	SHOW CELL		00004610
0496R	01E8	BRKPNT	BALR	E,OUTPUT	TYPE 'BREAKPOINT'		00004620
0498R	01CER		DC	A(MBKPNT)			00004630
049AR	0309		BR	SHOW	SHOW CELL		00004640
							00004650
049CR	C8E0	* [X] DIRECTIVE "ENTER BREAKPOINT"	SETBRK	LHI	E,SETBK1	SETUP EXIT FOR TSTINP	00004660
	04ACK						
04A0R	0800	TSTINP	LHR	INPUT,INPUT	INPUT BUFFER REGISTER=ZERO		00004670
04A2R	033E		BPCR	3,E	YES,EXIT		00004680
04A4R	0810		LHR	OPEN,INPUT	OPEN INPUT CELL		00004690
04A6R	4A10		AH	OPEN,BIAS	GET ABSOLUTE ADRS		00004700
	0114R						
04AAR	030E		BR	E	EXIT		00004710
04ACR	0811	SETBK1	LHR	OPEN,OPEN	DONT ALLOW USER TO ENTER		00004720
04AER	4330		BZ	NOBRK	BREAKPOINT IN CELL ZERO		00004730
	0546R						
04B2R	4801		LH	R0,0(OPEN)			00004731
	0000						
04B6R	C500		CLHI	R0,X'E1F0'			00004732
	E1F0						
04BAR	4330		BE	READIN			00004733
	0040R						
04BER	4120		BAL	P,BSERCH	SEARCH BRKTBL FOR OPENING		00004740
	042AR						
04C2R	4330		BZ	STRCEL	IF NO ENTRY,SETBRK&STRCELL		00004750
	04CER						
04C6R	010C		BALR	R0,C	RETURN TO BSERCH&SETUPEXIT		00004760
04C8R	01E8		BALR	E,OUTPUT	TYPE 'EXCESS BREAKPOINT'		00004770
04CAR	01C6R		DC	A(MXCESS)			00004780
04CCR	0309		BR	SHOW	RESHOW CELL		00004790
04CER	401D	STRCEL	STH	OPEN,BRKTBL+80(D)	SAVE OPEN CELL ADRS		00004800
	016ER						
04D2R	48B1		LH	B,0(OPEN)	GET CONTENTS OF OPENCELL		00004810
	0000						
04D6R	40BD		STH	B,BRKTBL+82(D)	SAVE ORIGINAL CONTENTS		00004820
	0170R						
04DAR	C4B0		NHI	B,X'FF00'	SET OP FIELD		00004830
	FF00						
04DER	C5B0		CLHI	B,X'2E00'	SEE IF RR INSTRUCTION		00004840
	2E00						
04E2R	4280		BL	RTNST	RR OP=01-2D		00004850

```
* COPYRIGHT INTERDATA INC., AUGUST 1, 1972
* OBJECT PROGRAM LABEL=BOTLDR
* AUTHOR: IOM IRACY
* PROGRAM USES MODEL 70 INSTRUCTION SET
*
* THIS PROGRAM IS A LOADER THAT READS A CORE IMAGE FROM THE
* DISC OR THE DRUM INTO MEMORY. IT ASSUMES THE STARTING
* POINT IS THE FIRST SECTOR OF THE DISC OR THE DRUM.
* THE BOOT LOADER RELOCATES ITSELF TO THE TOP OF MEMORY
* AND DOES NOT DESTROY ITSELF WHEN LOADING
* LOCATION X'78' MUST CONTAIN THE DEVICE NUMBER
* OF THE PAPER TAPE INPUT DEVICE
* LOCATION X'7A' MUST CONTAIN THE DEVICE NUMBER OF
* THE DISC OR THE DRUM.
```

```
* SYS GEN PARAMETERS: NONE
```

```
* ENTRY/EXTRN DECLARATIONS: NONE
```

```
* REGISTER ALLOCATION
```

```
0000 AC1 EQU 0
0001 AC2 EQU 1
0002 STAT EQU 2
0003 BASE EQU 3
0004 START EQU 4
0005 END EQU 5
0006 CYLADR EQU 6
0007 SECTOR EQU 7
0008 LNK EQU 8
0000 DISC EQU 13
0000 DEV EQU DISC
0000 DRUM EQU DISC
000E CTRL EQU 14
000F SELCH EQU 15
00F0 SELNUM EQU X'F0'
00B6 CONTRL EQU X'B6'
0078 BINDV EQU X'78'
007A MSDEV EQU X'7A'
2000 MS8K EQU X'2000'
00B5 DRMDEV EQU X'B5'
02D0 INIT1 EQU X'2D0'
```

```
*
```

```
BOT00030
BOT00040
BOT00050
BOT00060
BOT00070
BOT00080
BOT00090
BOT00100
BOT00110
BOT00120
BOT00130
BOT00140
BOT00150
BOT00160
BOT00170
BOT00180
BOT00190
BOT00200
BOT00210
BOT00220
BOT00230
BOT00240
BOT00250
BOT00260
BOT00270
BOT00280
BOT00290
BOT00300
BOT00310
BOT00320
BOT00330
BOT00340
BOT00350
BOT00360
BOT00370
BOT00380
BOT00390
BOT00400
BOT00420
BOT00430
BOT00440
BOT00450
```

0080		ORG	X*80*		B0T00470
	*				B0T00480
0080	D3D0	LB	DEV,BINDV	DEV NUM OF PT INPUT	B0T00490
	0078				
0084	C830	LHI	BASE,MS8K	START AT 8 K	B0T00500
	2000				
0088	4813	SERCH LH	AC2,0(BASE)	SERCH FOR TOP OF CORE	B0T00510
	0000				
008C	2138	BNZS	INCR		B0T00520
008E	4033	STH	BASE,0(BASE)		B0T00530
	0000				
0092	4803	LH	AC1,0(BASE)		B0T00540
	0000				
0096	2336	BZS	FOUND		B0T00550
0098	4013	STH	AC2,0(BASE)	RESTORE ORIG CONT	B0T00560
	0000				
009C	CA30	INCR AHI	BASE,MS8K	BUMP BY 8K	B0T00570
	2000				
00A0	203C	BNZS	SERCH	KEEP LOOKING	B0T00580
00A2	C800	FOUND LHI	AC1,LENGTH	COMPUTE LOAD BIAS	B0T00590
	009E				
00A6	0830	SHR	BASE,AC1		B0T00600
00A8	0813	LHR	AC2,BASE		B0T00610
00AA	9DD2	SS SSR	DEV,STAT	PT BSY?	B0T00620
00AC	20F1	BTBS	15,1		B0T00630
00AE	DBD1	RD	DEV,0(AC2)	READ 1 BYTE	B0T00640
	0000				
00B2	2611	AIS	AC2,1		B0T00650
00B4	2701	SIS	AC1,1		B0T00660
00B6	2216	BNMS	SS		B0T00670
00B8	C8F0	LHI	SELCH,SELNUM		B0T00680
	00F0				
00BC	0744	XHR	START,START		B0T00690
00BE	C853	LHI	END,-1(BASE)		B0T00700
	FFFF				
00C2	D3D0	LB	DEV,MSDEV	DRUM OR DISC?	B0T00710
	007A				
00C6	C5D0	CLHI	DEV,DRMDEV		B0T00720
	00B5				
00CA	0223	BPR	BASE	DISC	B0T00730
00CC	4303	B	URMADR(BASE)	DRUM	B0T00740
	0064				







NO ERRORS

AC1	0000
AC2	0001
BASE	0003
BINDV	0078
CONT	0018
CONT1	0076
CONTRL	0086
CTRL	000E
CYLADR	0006
DEV	0000
DISC	0000
DRMADR	0064
DRMDEV	0085
DRUM	0000
END	0005
ERRHLT	0014
ERRHT1	0072
FOUND	00A2
INCR	009C
INIT1	0200
LAST	009E
LENGTH	009E
LNK	0008
LOAD	0000
LOAD1	0064
MS8K	2000
MSDEV	007A
PAUSE	0096
PAUSE1	009A
RDDISC	0095
RDSLCH	0093
READ	0008
READ1	0068
SECTOR	0007
SEEK	0094
SELCH	000F
SELNUM	00F0
SERCH	0088
SS	00AA
START	0004
STAT	0002
STOP	0092

786  
784