

IDENTIFICATION

PRODUCT CODE:

DEC-9U-SR3A-LA
PART 7

PRODUCT NAME:

Listing of PDP-9 Advanced
Software System V4E FOCAL

DATE:

February 20, 1970

```

        .TITLE FOCAL
        EDIT 1# 1--21--70
        /SAME AS EDIT 8 EXCEPT VERSION NO. CHANGE TO 6A
    /
    /FOCAL FOR THE POP-9 ADVANCED SOFTWARE SYSTEM
    /
    /COPYRIGHT 1969
    /DIGITAL EQUIPMENT CORP.
    /MAYNARD, MASS.
    /
    /DAVE LENEY
    /2-7-69
    /
    /FOCAL IS A REGISTERED TRADEMARK OF
    /DIGITAL EQUIPMENT CORPORATION
    /
    /DEFINE MULTI=N IF MULTI-USER VERSION WHERE N=2 OR 4 USERS
    /DEFINE BF=0 IF BACKGROUND-FOREGROUND MONITOR
        .IFZER MULTI-4
    USR4=0
        .ENDC
    /
    /GLOBAL CALLS TO F4 ARITHMETIC PACKAGE
    /
        .GLOBL .AA          /EXPONENT
        .GLOBL .AB          /HIGH ORDER MANTISSA
        .GLOBL .AC          /LOW ORDER MANTISSA
        .GLOBL .AO          /LOAD (3 WORDS)
        .GLOBL .AP          /STORE (3 WORDS)
        .GLOBL .AQ          /ADD
        .GLOBL .AR          /SUBTRACT
        .GLOBL .AS          /MULTIPLY
        .GLOBL .AT          /DIVIDE
        .GLOBL .AX          /FIX
        .GLOBL .BA          /NEGATE
        .GLOBL .AW          /FLOAT
        .GLOBL .CD          /NORMALIZE
        .GLOBL .BH          /A**B-POWER
        .GLOBL DSIN         /SINE
        .GLOBL DCOS         /COSINE
        .GLOBL DATAN        /ARCTANGMENT
        .GLOBL DLOG         /LOGARITHM
        .GLOBL DEXP         /EXPONENTIAL
        .GLOBL DSQRT        /SQUARE ROOT
        .GLOBL .ER          /.OTS ERROR (?36)
    /
    /GLOBAL REFERENCES FOR EXTERNAL FUNCTIONS
    /
        .GLOBL .NEWF        /FUNCTION TABLE
        .GLOBL XPUSHJ       /PUSH JUMP
        .GLOBL XPUSHA       /PUSH AC
        .GLOBL PD2          /PUSH FLOATING
        .GLOBL PD3          /POP FLOATING
        .GLOBL UTRA         /UNPACK
    /

```

```

.GLOBL XSPNOR /IGNORE SPACES
.GLOBL FUNERR /ERROR IN EXTERNAL FUNCTION
.GLOBL EFUN3 /FUNCTION RETURN
.GLOBL FINT /FLOATING INTERPRETER
.GLOBL CHAR /CHRACTER STORAGE
.GLOBL EVAL /EVALUATION ROUTINE
.GLOBL LASTV /END OF TEXT/VARIABLES
.GLOBL BOTTOM /START OF PUSH-DOWN LIST
.GLOBL LINENO /CURRENT LINE
.GLOBL FLAG /FLOATING ARGUMENT
.GLOBL BUFSTX /POINTER TO 3 WD UNPACK AREA
.GLOBL XGETLN /FETCH LINE NUMBER
.GLOBL XPOPJ /POP JUMP
.GLOBL FETVAR /GET 3,6,OR 9 .SIXBT CHARS

```

```

000000 A FPOM=000000 /PSEUDO-FLOATING POINT INSTRUCTIONS,
100000 A FADD=100000
200000 A FSUB=200000
300000 A FMPY=300000
300000 A FMUL=300000
400000 A FDIV=400000
500000 A FGET=500000
600000 A FPUT=600000
700000 A FNOR=700000
000000 A FEXT=0
000000 A FXIT=0
000003 A WORDS=3
000011 A DIGITS=11
000100 A .SCOM=100
000000 A XX=0

```

.EJECT

```

                                .IFUND  MULTI
                                .IODEV  -3,-2,3,5
000776 A      TTI=776
000775 A      TIO=775
000003 A      BKI=3
000005 A      BK0=5

                                .ENDC
                                .IFDEF  MULTI
                                .IODEV  1,2,3,4
                                .IFDEF  USR4
                                .IODEV  5,6,7,10
                                .ENDC

TTI=0
TIO=0
BKI=0
BK0=0
TTI1=1
TIO1=1
BKI1=2
BK01=2
TTI2=3
TIO2=3
BKI2=4
BK02=4

                                .IFDEF  USR4

TTI3=5
TIO3=5
BKI3=6
BK03=6
TTI4=7
TIO4=7
BKI4=10
BK04=10

                                .ENDC

COMEIN=0
COMOUT=0
IMBUF=0
INBUF=0
OUTBUF=0

                                .ENDC

/AUTO-INDEX REGISTERS
000010 A      AXIN=10      /STORAGE INDEX
000011 A      XRT=11      /EXTRA XR
000012 A      XRT2=12     /EXTRA XR
000013 A      PDLXR=13    /PUSHDOWN LIST INDEX REGISTER.
000014 A      FLTXR=14    /IOBUF-1 XR14 FOR FLOATING POINT
000015 A      X15=15      /FOR COMMON RESTORE
000016 A      X16=16      /FOR COMMON RESTORE
/
/IN THE MULTI USER SYSTEM 15 AND 16 ARE ALSO USED BY THE SWAP ROUTINES
/

                                .EJECT

```

```

/
/ THE FOLLOWING BLOCK IS THE ENTIRE IMPURE
/ AREA FOR EACH FOCAL JOB
/
SWPSZE .IFDEF MULTI
SWPBGN= SWPBGN-SWPEND /BLOCK SIZE
/
RESTAR XSREGN /RESTART ADDRESS FOR THIS JOB
CTLP XX /ADDRESS OF *P SWITCH FOR THIS USER
BWAIT XX
/
FLAC 0 /.AA SAVE
0 /.AB SAVE
0 /.AC SAVE
/
FRSTSV 0 /FRST SAVE
LIST31 0 /LIST3+1 SAVE
/
AUTOXR 0 /X10 SAVE
0 /X11 SAVE
0 /X12 SAVE
0 /X13 SAVE
0 /X14 SAVE
/
IMBFSV XX /BUFFER HEADER POINTERS
IMBF2S XX
INBFSV XX
OTBFSV XX
.FLINP XX /SUBROUTINE ENTRY POINTERS
.XI33 XX
.XOUTL XX
.INPUT XX
.DECON XX
.DECNV XX
.IMAGR XX
.IMAGW XX
/
XX /RCAL01
XX /RCAL03
XX /WCAL01
XX /WCAL03
XX /WCAL04
XX /LBIN01
XX /LBIN1A
XX /LBIN02
XX /LBIN03
XX /LBOUT1
XX /LBOUT2
XX /LBOUT3
XX /LBOUT4
XX /FILE01
XX /FILE02
XX /FILE03
.ENDC

```

```

/REENTRANT VARIABLES
/
00000 R 000000 A BOTTOM XX /TOP OF PUSH-DOWN LIST
00001 R 000000 A BUFSTX XX /3 REG AREA BELOW TEXT AND VARIABLES
                                /USED TO CONSTRUCT VARIABLES AND FILE
                                /NAMES(FILE01 AND FILE02 CONTAIN SAME ADDR)
00002 R 000000 A ENDT XX /START OF TEXT
00003 R 000000 A STARTV XX /LAST LOCATION OF TEXT
00004 R 000000 A BUFR XX /NEXT LOCATION IN BUFFER (VARIABLES)
                                /ADDRESS OF LAST COMMON VARIABLE
00005 R 000000 A LASTCV=STARTV FRSTCV XX /ADDRESS OF FIRST COMMON VARIABLE
00006 R 000000 A LASTV XX /ADDRESS OF LAST VARIABLE
00007 R 003756 R COMBUF COMEIN /COMMAND BUFFER START
00010 R 004066 R COMBOT COMOUT /AND END
00011 R 003125 R IMBUFP IMBUff+2 /BUFFER DATA POINTERS
00012 R 003131 R INBUFP INBUff+2
00013 R 003213 R OTBUFP OUTBUff+2
                                /TEXT POINTERS
00014 R 000000 A AXOUT XX /OUTPUT INDEX
00015 R 000000 A XCTX 0 /UNPACK SWITCH
00016 R 000000 A GTEM 0 /UNPACK STORAGE
00017 R 000000 A MODBUF 0 /POINTER FOR MODIFY
00020 R 000215 A ENDCR 215 /LAST CHAR FOR GETC
00021 R 000000 A GETVCT 0 /VARIABLE COUNT
00022 R 000000 A SAVEOT 0 /OUTPUT CHAR
00023 R 777777 A PUTCNT -1 /OUTPUT COUNTER FOR HEADER PAIR
00024 R 000000 A TEMPK 0 /TEMP FOR PACK
00025 R 000000 A INSUB 0 /0= GETC; #0 = READC
00026 R 000776 A TTIN TTI
00027 R 000775 A TTOUT TTO
00030 R 000003 A BLKIN BKI
00031 R 000005 A BLKOUT BKO
00032 R 000000 A LIBRSW 0 /IN LIBRARY MODE
00033 R 000000 A EX1 0
00034 R 000000 A AC1H 0
00035 R 000000 A AC1L 0
00036 R 000000 A OVER1 0
00037 R 000000 A OVER2 0
                                OTEMP=OVER1
                                LTEMP=AC1L
                                HTEMP=AC1H
00040 R 000010 A FISW 10
00041 R 000000 A GETP 0 /ASCII STRING POINTER
00042 R 000000 A GETCX 0 /CHAR COUNTER (2'S COMP)
00043 R 000000 A GET1X 0 /TEMP
00044 R 000000 A GET2 0 /TEMP
00045 R 000000 A GET3 0 /TEMP
00046 R 000000 A PUTP 0 /ASCII STRING POINTER
00047 R 000000 A PUTC 0 /CHAR COUNTER
00050 R 000000 A PUT6 0 /TEMP
00051 R 000000 A SORTCN 0 /NUMBER IN TABLE FROM SORTC
00052 R 000000 A LASTOP 0 /LAST OPERATION FOR EVAL
                                EFOP=.
                                ATSW 0
                                /FUNCTION CODE.
                                /ASK-TYPE CODE.

```

```

00054 R 777760 A CNTR -20 /DELETE AND ERROR COUNTER(USED BY F.P. ALSO).
00055 R 000004 A DECP 4 /NUMBER OF DECIMAL POINTS
00056 R 000000 A ADD XX /CHAR. BUF. IN. (DEBUG AIDS.SEE BELOW.)
00057 R 000000 A XCTIN XX /PACK SWITCH
00060 R 000001 A NAGSW 0001 /NOT ALL AND/OR GROUP SWITCH (4000=ONE;1=ALL;0=GROUP)
00061 R 000215 A CHAR 215 /THE MOST IMPORTANT REGISTER
00062 R 000000 A LINENO 0000 /LINE NUMBER READ BY GETLN
00063 R 000114 R PC FRST+1 /PROGRAM COUNTER
00064 R 000000 A THISLN 0 /LINE POINTER FROM 'FINDLN'
00065 R 000000 A THISOP 0 /CURRENT 'EVAL' OPERATION
00066 R 000000 A LASTLN 0 /BACK POINTER FROM 'FINDLN'
00067 R 000001 A DEBGSW 1 /DEBUG SWITCH ; NON-ZERO FOR LITERAL.
00070 R 000001 A DWRPW 1 /#0 FOR TRACE ON.
00071 R 000000 A PACKST 0 /RUBOUT PROTECTION
00072 R 000000 A PT1 0 /VARIABLE POINTER
00073 R 000000 A T1 0 /TEMPORARY REGISTER - MAIN
00074 R 000000 A T2 0 /TEMP REGISTER - FOR NEW INST. ROUTINES.
00075 R 000000 A SACH 0 /SEARCH CHAR STORAGE
00076 R 000000 A FLARG 0 /DATA TEMPORARY STORAGE
00077 R 000000 A 0
00100 R 000000 A 0
00101 R 000000 A FLARG2 0
00102 R 000000 A 0
00103 R 000000 A 0

```

.IFDEF MULTI

SWPEND=.

.ENDC

/NON-REENTRANT VARIABLES

```

00104 R 000000 A BOX 0 /FOR DIGIT PRINT
00105 R 000000 A ER2T 0 /ERROR TEMP
00106 R 000000 A ERR2CT 0 /ERROR COUNT
00107 R 000107 R OP . /
00110 R 000000 A XX /VARIABLE NAME (.SIXBT)
00111 R 505175 A .SIXBT /()/=
00112 R 000000 A RANPT 0 /PUSED0 RANDOM POINTER
00113 R 000000 A FRST 0 /TEXT POINTER
00114 R 000000 A 0 /DUMMY LINE NUMBER

```

.IFDEF PDP15
.SIXBT /C FOCAL15 V6A/<77><15>

.ENDC

.IFUND PDP15
.SIXBT /C FOCAL9 V6A/<77><15>

```

00115 R 034006 A
00116 R 170301 A
00117 R 147140 A
00120 R 266601 A
00121 R 771500 A

```

.ENDC

```

00122 R 000000 A SIGN2 0 /TEMP SIGN
00123 R 000000 A SCOUNT 0
PLCE=. 000124 R
FCOUNT 00124 R 000000 A 0
TEMPO 00125 R 000000 A 0
REMAIN 00126 R 000000 A 0

```

00127	R	000000	A	DIGIT	0		/DIGIT STORAGE (CURRENT)
00130	R	000000	A	ISIGN	0		/0=MINUS,-1=PLUS
00131	R	000000	A	DNUMBER	0		/NUMBER OF DIGITS
00132	R	000000	A	BEXP	0		
00133	R	000000	A	SEXP	0		/DECIMAL EXPONENT
		000134	R	MODBF1=.			
00134	R	000000	A	JUMP	0		
		000135	R	MODBF2=.			
00135	R	000000	A	JUMP2	0		
00136	R	000000	A	ADDR	0		
		000137	R	XY=.			/TEMP FLOATING POINT
00137	R	000000	A	FUNAME	0		/FUNCTION NAME
00140	R	000000	A	FUNCTR	0		/FUNCTION COUNTER
00141	R	000000	A	FUNPTR	0		/FUNCTION POINTER
00142	R	000000	A	ARRAYN	0		/ARRAY NAME
				/			
				/CONSTANTS			
				/			
00143	R	000013	A	P13	13		
00144	R	000017	A	P17	17		
00145	R	000277	A	C277	277		
00146	R	000003	A	P3	3		
00147	R	000002	A	P2	2		
00150	R	000100	A	C100	100		
00151	R	000077	A	C77	77		
00152	R	000260	A	C260	260		
00153	R	777700	A	M100	-100		
00154	R	000200	A	C200	200		
00155	R	000177	A	P177	177		
00156	R	000005	A	GINC	WORDS+2		
00157	R	000113	R	CFRS	FRST		/DUMMY LINE ADDRESS
00160	R	000076	R	FLARGP	FLARG		/DATA ADDRESS
00161	R	000314	A	FILEXT	SIXBT	/FCL/	
00162	R	000171	R	CFRSX	FLTZER		/FLOATING 0 ADDRESS
00163	R	000306	A	C306	306		
00164	R	000314	A	C314	314		
00165	R	777641	A	M137	-137		
00166	R	000337	A	P337	337		
		000167	R	C1=.			
00167	R	000001	A	FLTONE	000001		/FLOATING 1.0
00170	R	200000	A		200000		
00171	R	000000	A	FLTZER	000000		/FLOATING 0.0
00172	R	000000	A		000000		
00173	R	000000	A		000000		
00174	R	000040	A	P40	40		
00175	R	000140	A	C140	140		
00176	R	777640	A	M140	-140		
00177	R	004002	A	FOCAL9	4002		
00200	R	000000	A		0		
				.IFDEF	PDP15		
				.ASCII	/FOCAL15 V6A/<15>		
				.ENDC			
				.IFUND	PDP15		
				.ASCII	/FOCAL9 V6A/<15>		
00201	R	432370	A				
00202	R	340630	A				

00203 R 345012 A
00204 R 633202 A
00205 R 064000 A
00206 R 000000 A

00207 R 000032 R
00210 R 000012 A
00211 R 004045 R
00212 R 000144 A
00213 R 777634 A
00214 R 000004 A
00215 R 240000 A
00216 R 000000 A
00217 R 000043 A
00220 R 020000 A
00221 R 017777 A
00222 R 000007 A
00223 R 620404 R
00224 R 000376 R

.ENDC
EX1-1
RND2 DIGITS+1
BUFST BUFFER-1
C144 144
M144 -144
TEN 000004
P43 43
INDRCT 20000
MASK7 17777
C7 7
TABLE JMP* ITABLE
OPTABL OPTABS
.EJECT

/FLOATING 10.0

```
/
/SUBROUTINE CONVENTIONS
/
/1)USE AC OR 'CHAR' ON ENTRY
/   SORTJ
/   PRINTC
/2)USE 'CHAR' ONLY ON ENTRY
/   PACKC
/   SORTC
/   SPNOR
/   TESTN
/   TESTC
/3)RETURN WITH 'CHAR' IN AC
/   READC
/   GETC
/   PACKC
/   SPNOR
/   SORTC
/   PRINTC
/   TESTC
/   INPUT
/4)USE AC ONLY ON ENTRY
/   DECON
/
      .EJECT
```

/NEW INSTRUCTIONS:

```

.DEFIN PUSHJ,A
JMS      XPUSHJ      /RECURSIVE SUBROUTINE CALL
      A

.ENDM
.DEFIN POPA
LAC*    PDLXR      /RESTORE AC
.ENDM
.DEFIN POPJ
JMP     XPOPJ      /SUBROUTINE RETURN
.ENDM
.DEFIN PUSHA
JMS     XPUSHA     /SAVE AC
.ENDM
.DEFIN PUSHF,A
JMS     P02        /SAVE GROUP OF DATA
      A
.ENDM
.DEFIN POPF,A
JMS     P03        /RESTORE GROUP
      A
.ENDM
.DEFIN GETC
JMS     UTRA       /UNPACK A CHARACTER
.ENDM
.DEFIN PACKC
JMS     PACBUF     /PACK A CHARACTER
.ENDM
.DEFIN SORTJ,A,B
JMS     SORTB      /SORT AND BRANCH ON AC OR CHAR
      A-1
      B-A
.ENDM
.DEFIN SORTJX,A
JMS     XSORTX     /SORT + BRANCH ON COMMAND
      A-1
.ENDM
.DEFIN SORTC,A
JMS     XSORTC     /SORT CHAR
      A-1
.ENDM
.DEFIN PRINTC
JMS     XOUTL      /PRINT AC OR CHAR
.ENDM
.DEFIN READC
JMS     XI33       /READ KSR-33/35 INTO CHAR
.ENDM
.DEFIN PRNTLN
JMS     XPRNT      /PRINT C(LINENO)
.ENDM
.DEFIN GETLN
JMS     XGETLN     /UNPACK AND FORM A LINENUMBER
.ENDM
.DEFIN FINDLN
JMS     XFIND      /SEARCH FOR A GIVEN LINE

```

```

.ENDM
.DEFIN ENDLN
JMS      XENDLN      /INSERT LINE POINTERS
.ENDM
.DEFIN RTL6
JMS      XRTL6       /ROTATE LEFT SIX
.ENDM
.DEFIN SPNOR
JMS      XSPNOR      /IGNORE SPACES
.ENDM
.DEFIN TESTN
JMS      XTESTN      /PERIOD; OTHER; NUMBER
.ENDM
.DEFIN TSTLPR
JMS      LPRTST      /SKIP IF 5<SORTCN<= 11 (I.E. AN L-PAR)
.ENDM
.DEFIN TSTGRP
JMS      GRPTST      /SKIP IF G(AC) = G(LINENO)
.ENDM
.DEFIN TESTC
JMS      XTESTC      /TERM; NUMBER; FUNCTION; LETTER
.ENDM
.DEFIN ERROR,A
.DEC
JMP      ERR2-A      /ERROR MSG
.ENDM
.DEFIN GETSGN
LAC*     .AB
.ENDM
.DEFIN RETURN
JMP      EFUN3
.ENDM
.IFDEF BF
.DEFIN .RLXIT,A
CAL      A
20
.ENDM
.ENDC
.IFUND BF
.DEFIN .RLXIT,A
OBR
JMP*     A
.ENDM
.ENDC
.EJECT

```

/ FOCAL COMMAND TABLES

```

/
COMLST -22
        .SIXBT /IFe/
        JMP IF
        .SIXBT /Doe/
        JMP DO
        .SIXBT /GOe/
        JMP GOTO
        .SIXBT /GOTOe/

        JMP GOTO
        .SIXBT /SETe/

        JMP SET
        .SIXBT /FORe/

        JMP FOR
        .SIXBT /COMMENTe/

        JMP COMMEN
        .SIXBT /CONTINUEe/

        JMP COMMEN
        .SIXBT /ERASEe/

        JMP ERASE
        .SIXBT /WRITEe/

        JMP WRITE
        .SIXBT /MODIFYe/

        JMP MODIFY
        .SIXBT /QUITe/

        JMP START
        .SIXBT /RETURNe/

        JMP RETURX
        .SIXBT /*e/
        JMP HSPX
        .SIXBT /ASKe/

        JMP ASK
        .SIXBT /TYPEe/

        JMP TYPE
        .SIXBT /LIBRARYe/
00225 R 777756 A
00226 R 110600 A
00227 R 601356 R
00230 R 041700 A
00231 R 600776 R
00232 R 071700 A
00233 R 601176 R
00234 R 071724 A
00235 R 170000 A
00236 R 601176 R
00237 R 230524 A
00240 R 000000 A
00241 R 601407 R
00242 R 061722 A
00243 R 000000 A
00244 R 601407 R
00245 R 031715 A
00246 R 150516 A
00247 R 240000 A
00250 R 601210 R
00251 R 031716 A
00252 R 241116 A
00253 R 250500 A
00254 R 601210 R
00255 R 052201 A
00256 R 230500 A
00257 R 602633 R
00260 R 272211 A
00261 R 240500 A
00262 R 601234 R
00263 R 151704 A
00264 R 110631 A
00265 R 000000 A
00266 R 601705 R
00267 R 212511 A
00270 R 240000 A
00271 R 600512 R
00272 R 220524 A
00273 R 252216 A
00274 R 000000 A
00275 R 602262 R
00276 R 520000 A
00277 R 604155 R
00300 R 012313 A
00301 R 000000 A
00302 R 601603 R
00303 R 243120 A
00304 R 050000 A
00305 R 601604 R
00306 R 141102 A
00307 R 220122 A
00310 R 310000 A

```

00311	R	604067	R	JMP	LIBRAR
00312	R	031715	A	.SIXBT	/COMMONe/
00313	R	151716	A		
00314	R	000000	A		
00315	R	601513	R	JMP	COMMON
/					
00316	R	777777	A	ALLCM1	-1
00317	R	011414	A	.SIXBT	/ALLe/
00320	R	000000	A		
00321	R	600755	R	JMP	GEXIT
/					
00322	R	777776	A	ALLCM2	-2
00323	R	011414	A	.SIXBT	/ALLe/
00324	R	000000	A		
00325	R	602644	R	JMP	XSBEGR
00326	R	031715	A	.SIXBT	/COMMONe/
00327	R	151716	A		
00330	R	000000	A		
00331	R	600543	R	JMP	STARTQ
/					
00332	R	777773	A	LIBCMD	-5
00333	R	172524	A	.SIXBT	/OUTe/
00334	R	000000	A		
00335	R	604157	R	JMP	LBOUT
00336	R	111600	A	.SIXBT	/INe/
00337	R	604105	R	JMP	LBIN
00340	R	031417	A	.SIXBT	/CLOSEe/
00341	R	230500	A		
00342	R	604177	R	JMP	LBCLOS
00343	R	131114	A	.SIXBT	/KILLe/
00344	R	140000	A		
00345	R	604205	R	JMP	LBKILL
00346	R	272211	A	.SIXBT	/WRITEe/
00347	R	240500	A		
00350	R	604220	R	JMP	LBWRIT

/TABLES FOR FOCAL FUNCTIONS (INTERNAL)

/					
00351	R	777766	A	FNTABF	FNTABE--1/2\777777+1
00352	R	231116	A	.SIXBT	/SIN/
00353	R	604340	R	JMP	FSIN
00354	R	031723	A	.SIXBT	/COS/
00355	R	604334	R	JMP	FCOS
00356	R	012416	A	.SIXBT	/ATN/
00357	R	604315	R	JMP	ARTN
00360	R	053020	A	.SIXBT	/EXP/
00361	R	604272	R	JMP	FEXP
00362	R	141707	A	.SIXBT	/LOG/
00363	R	604321	R	JMP	FLOG
00364	R	232124	A	.SIXBT	/SQT/
00365	R	605363	R	JMP	XSQRT
00366	R	010223	A	.SIXBT	/ABS/
00367	R	602514	R	JMP	XABS
00370	R	230716	A	.SIXBT	/SGN/
00371	R	602510	R	JMP	XSGN

00372 R 112422 A
 00373 R 602506 R
 00374 R 220116 A
 00375 R 603734 R
 000376 R

.SIXRT /ITR/
 JMP XINT
 .SIXRT /RAN/
 JMP XRAM

FNTARE=.

/ OPERATION TABLES FOR FLOATING POINT INTERPRETER

00376 R 520072 R
 00377 R 120072 R
 00400 R 220072 R
 00401 R 420072 R
 00402 R 320072 R
 00403 R 020072 R

OPTABS FGET* PT1
 FADD* PT1
 FSUB* PT1
 FDIV* PT1
 FMUL* PT1
 FPOW* PT1

00404 R 005234 R
 00405 R 005226 R
 00406 R 005222 R
 00407 R 005337 R
 00410 R 005352 R
 00411 R 005216 R
 00412 R 005212 R
 00413 R 005232 R

ITABLE EXITF /EXIT OR POWER
 FLAQ /ADD
 FLSU /SUBTRACT
 FLMY /MULTIPLY
 FLOV /DIVIDE
 FLGT /GET FLOATING POINT
 FLPT /PUT FLOATING POINT
 NORF /NORMALIZE

/ TABLE OF TERMINATORS (FOR EVAL AND GETVAR)

00414 R 000305 A
 00415 R 000256 A
 000416 R
 00416 R 000240 A
 00417 R 000253 A
 000420 R
 00420 R 000255 A
 00421 R 000257 A
 00422 R 000252 A
 00423 R 000336 A
 00424 R 000250 A
 00425 R 000333 A
 00426 R 000274 A
 00427 R 000251 A
 00430 R 000335 A
 00431 R 000276 A
 00432 R 000254 A
 00433 R 000273 A
 00434 R 000215 A
 00435 R 000275 A

C305 305 /E - FOR INPUT NUMBERS
 PER 256 /. - FOR INPUT NUMBERS
 TERMS=.
 C240 240 /SPACE 0
 253 /+ 1
 C255=.
 SMIN 255 /- 2
 257 /1 3
 C252 252 /* 4
 336 /* 5
 C250 250 / (6 L - PARENS
 333 / [7
 274 / < 10
 C251 251 /) 11 R - PARENS
 335 /] 12
 276 / > 13
 C254 254 / . 14
 C273 273 / ; 15
 215 / CR 16
 C275 275 / = 17

/ CONTROL TABLE FOR ASK/TYPE OPERATIONS

00436 R 601670 R
 00437 R 601645 R
 00440 R 601655 R
 00441 R 601661 R
 00442 R 603657 R
 00443 R 601657 R

ATLIST JMP TINTR
 JMP TQUOT
 JMP TCRLF
 JMP TCRLF2
 JMP TDUMP
 JMP TASK4

```

00444 R 601657 R      JMP      TASK4
00445 R 601203 R      JMP      PROCES
00446 R 601210 R      JMP      PC1

00447 R 000245 A      /
00450 R 000242 A      ALIST    245      /% - FLOATING FORMAT
00451 R 000241 A      C242    242      /" - LITERAL
00452 R 000243 A      241      /! - CR AND LF
00453 R 000244 A      243      /# - CR ONLY
00454 R 000240 A      244      /$ - SYMBOL DUMP
00455 R 000254 A      GLIST   240      /SPACE - END NAMES
00456 R 000273 A      TLIST   254      /, - END EXPRESSIONS
00457 R 000215 A      TLISTX  273      /; - END COMMANDS
                                215      /C.R. - END STRINGS

/
/ DISPATCH TABLES FOR IF AND COMMON STATEMENTS
/
00460 R 601405 R      ILIST   JMP      IF1      /,
00461 R 601203 R      JMP      PROCES     /;
00462 R 601210 R      JMP      PC1        /CR

/
00463 R 601447 R      FLIST2  JMP      FLIMIT     /,
00464 R 601507 R      JMP      FINFIN     /;
                                ERROR  11      /CR
                                .DEC

00465 R 603462 R      GEN*   JMP      ERR2-11

/
00466 R 601433 R      FLIST1  JMP      FINCR      /,
00467 R 601203 R      JMP      PROCES     /;
00470 R 601210 R      JMP      PC1        /CR

/
00471 R 601512 R      CLISTX  JMP      COMMON-1   /,
00472 R 601203 R      JMP      PROCES     /;
00473 R 601210 R      JMP      PC1        /CR

/
/ CONTROL TABLE FOR MODIFY OPERATION
/
00474 R 000225 A      LIST6   225      /*U - KILL LINE
00475 R 000375 A      C375    375      /ALTMODE - NEXT OCCURANCE OF SEARCH CHAR.
00476 R 000207 A      207      /BELL - NEW SEARCH CHAR
00477 R 000212 A      C212    212      /L.F. - END LINE SAVING REST
00500 R 000377 A      C377    377      /RUBOUT - DELETE LAST CHAR
                                000501 R      LIST3=.
                                000501 R      CCR=.
00501 R 000215 A      C215    215      /C.R. - END LINE DELETING REST
00502 R 000000 A      000     /SEARCH CHAR

/
00503 R 601744 R      SRNLST  JMP      SBAR       /*U
00504 R 601731 R      JMP      SCHAR      /F.F.
00505 R 601724 R      JMP      SCONT      /BELL
00506 R 601726 R      JMP      SCONTX     /L.F.
00507 R 602003 R      JMP      SCRUB      /RUBOUT
00510 R 601762 R      LISTGO  JMP      SRETN      /CR
00511 R 601750 R      JMP      SFOUND     /SEARCH CHAR

/
.EJECT

```



```

        .IFDEF MULTI
/THIS CODE CONTROLS THE MULTI-USER PROCESSING
/ OF TWO OR FOUR CONCURRENT FOCAL USERS.
/
BUFFER=.
MSTART  LAC*      (.SCOM+2
        DAC      T1
        JMS      TWOS
        TAD*     (.SCOM+3 /GET SIZE
        CLL:RAR  /DIVIDE BY TWO OR FOUR
        .IFDEF  USR4
        CLL:RAR
        .ENDC
        DAC      ENDT /AMT FOR EACH
/DETERMINE IF BG OR FG IN BF ENVIRONMENT
/.SCOM+26 = 0 IF FG, = 1 IF BG
        LAC*     (.SCOM+26
        SNA
        JMP FGRY
        LAC BG1
        .IFDEF  USR4
        DAC SCANQ /SET PROCESSING LOOP TO BYPASS IDLEC
        JMP FGBY
        .ENDC
        DAC WAIT3
FGBY    LAC      T1 /GO INITIALIZE REGS
        DAC     BUF1
        DAC     FILA1
        DAC     FILB1
        DAC     FILC1
        TAD     (3
        DAC     ENDT1
        DAC     ENDT1+1
        DAC     ENDT1+3
        DAC     ENDT1+4
        LAW     -1
        TAD     T1
        TAD     ENDT
        DAC     BOT1
        TAD     C1
        DAC     T1
        DAC     BUF2
        DAC     FILA2
        DAC     FILB2
        DAC     FILC2
        TAD     (3
        DAC     ENDT2
        DAC     ENDT2+1
        DAC     ENDT2+3
        DAC     ENDT2+4
        LAW     -1
        TAD     T1
        TAD     ENDT
        DAC     BOT2
    
```

```

        .IFDEF      USR4
        TAD         C1
        DAC         T1
        DAC         BUF3
        DAC         FILA3
        DAC         FILB3
        DAC         FILC3
        TAD         (3
        DAC         ENDT3
        DAC         ENDT3+1
        DAC         ENDT3+3
        DAC         ENDT3+4
        LAW         -1
        TAD         T1
        TAD         ENDT
        DAC         BOT3
        TAD         C1
        DAC         T1
        DAC         BUF4
        DAC         FILA4
        DAC         FILB4
        DAC         FILC4
        TAD         (3
        DAC         ENDT4
        DAC         ENDT4+1
        DAC         ENDT4+3
        DAC         ENDT4+4
        LAW         -1
        TAD         T1
        TAD         ENDT
        DAC         BOT4
        .ENDC
        .INIT       TT01,1,CP1+400000
        .INIT       TT02,1,CP2+400000
        .IFDEF      USR4
        .INIT       TT03,1,CP3+400000
        .INIT       TT04,1,CP4+400000
        .ENDC
        .WRITE      TT01,2,FOCAL9,40
        .WRITE      TT02,2,FOCAL9,40
        .IFDEF      USR4
        .WRITE      TT03,2,FOCAL9,40
        .WRITE      TT04,2,FOCAL9,40
        .ENDC
        JMP         WAIT1
CTLP1   0
CTLP2   0
        .IFDEF      USR4
CTLP3   0
CTLP4   0
        .ENDC
CP1     0
        ISZ CTLP1.
        .RLXIT CP1.
CP2     0
    
```

```

ISZ CTLP2
.RLXIT CP2
.IFDEF   USR4
CP3      0
ISZ CTLP3
.RLXIT CP3
CP4      0
ISZ CTLP4
.RLXIT CP4
.ENDC
.EJECT
/MAIN PROCESSING LOOP
WAIT1    .WAITR   TTI1,WAIT2
WAITB1   .WAITR   TTI1,WAIT2
          LAC      (AREA1-1
          JMS      RUN
WAIT2    .WAITR   TTI2,WAIT3
WAITB2   .WAITR   TTI2,WAIT3
          LAC      (AREA2-1
          JMS      RUN
          .IFUND   USR4
WAIT3    NOP      /WILL BE JMP WAIT1 IN 2 USER CASE
          .ENDC
          .IFDEF   USR4
WAIT3    .WAITR   TTI3,WAIT4
WAITB3   .WAITR   TTI3,WAIT4
          LAC      (AREA3-1
          JMS      RUN
WAIT4    .WAITR   TTI4,SCANQ
WAITB4   .WAITR   TTI4,SCANQ
          LAC      (AREA4-1
          JMS      RUN
          .ENDC
SCANQ    CAL      /N0 - GIVE BGD SOME TIME
/SCANQ WILL CONT. JMP WAIT1 IN CASE OF 4USER
          14
          IDLE+700000
          -12
          CAL+1000      /.IDLEC ROUTINE WILL BYPASS THIS ON
          17              /TIMER OVERFLOW
          CAL+1000
          14
          IDLE+700000
          0
BG1      JMP      WAIT1      /GO SEE IF ANYTHING DONE NOW
/
          .EJECT
/RUN INITIALIZATION
RUN      0
          DAC      NEWUSR      /CHECK FOR SAME USER
          SAD      CURUSR      /?
          JMP      RESTAX      /YES - NO SWAP
          LAC      (MVSZE      /SET TO SAVE INLINE
          DAC+     (16          /TEMPORARIES
          LAC      (BWAIT

```

```

DAC*      (17
LAC       MVSZE
DAC       RUNCT
DAC*     16 /SAVE REGS LOOP
DAC       RUNTP
LAC*     17
DAC*     17
ISZ      RUNCT
JMP      .-5
LAC      CURUSR /SWAP USERS INITIALIZATION
DAC*    (16
LAC     NEWUSR
DAC*    (17
DAC     CURUSR
LAC     (SWPBG
DAC     RUNTP
LAC     SWPSZE
DAC     RUNCT
LAC*   RUNTP /NOW DO SWAP
DAC*   16
LAC*   17
DAC*   RUNTP
ISZ    RUNTP
ISZ    RUNCT
JMP    .-6
LAC    (MVSZE /SET TO RESTORE INLINE
DAC*   (16 /TEMPORARIES
LAC    (BWAIT
DAC*   (17
LAC    MVSZE
DAC    RUNCT
LAC*   16 /RESTOR REGS LOOP
DAC    RUNTP
LAC*   17
DAC*   RUNTP
ISZ    RUNCT
JMP    .-5
RESTAX
CAL    14
TIME+700000
-12
DZM DELAY
LAC* CTLP
SZA
JMP RECOVR
/I/O BUSY OR OUT OF TIME RETURNS HERE /GO START UP USER
IOBUSY 0
LAC    .-1 /GET RETURN PC
DAC    RESTAR
.IFDEF BF
CAL+1000 /CLEAR OUT CALL FOR TIME
14
TIME+700000
0

```

```

.ENDC
JMP*      RUN
/
RUNTP    0
RUNCT    0
DELAY    0
/
TIME      0                               /SET DELAY ON OVERFLOW
        ISZ      DELAY
        .RLXIT TIME
/
IDLE     0                               /FORCE RETURN TO FGD
        .RLXIT IDLE
/
        .EJECT
/
/COMMUNICATION BLOCK
/
CURUSR   SWP8GN-1                       /CURRENT USER AREA-1
NEWUSR   0                               /NEW USER AREA-1
/
/SPECIAL POINTERS FOR SAVE/RESTORE
/
MVSZE    .+1-MVEND
FLAC14   XX                               /.AA
FLAC15   XX                               /.AB
FLAC16   XX                               /.AC
FRST
LIST3+1
10
11
12
13
14
IMBF01
IMBF02
INBF01
OTBF01
FLINTP
XI33
XOUTL
INPUT
DECON
DECONV
IMAGER
IMAGEW
RCAL01
RCAL03
WCAL01
WCAL03
WCAL04
LBIN01
LBIN1A
LBIN02
LBIN03

```


XX	
XX	
XX	
XX	
CMR1	/COMBUF=COMEIN
CMR1+77	/COMBOT=COMOUT
IMR1+2	/IMBUF=IMBUF+2
INR1+2	/INBUF=INBUF+2
OTR1+2	/OTBUF=OUTBUF+2
0	/AXOUT (TEXTP)
0	/XCTX
0	/GTEM
0	/MODBUF
215	/ENDCR
0	/GETVCT
0	/SAVEOT
-1	/PUTCNT
0	/TEMPK
0	/INSUB
TTI1	/TTIN
TT01	/TTOUT
BKI1	/BLKIN
BK01	/BLKOUT
0	/LIBRSW
0	/EX1
0	/AC1H
0	/AC1L
0	/OVER1
0	/OVER2
10	/FISW
0	/GETP
0	/GETCX
0	/GET1X
0	/GET2
0	/GET3
OTB1+2	/PUTP
0	/PUTC
0	/PUT6
0	/SORTCN
0	/LASTOP
0	/ATSW (EFOP)
-20	/CNTR
4	/DECP
0	/ADD
0	/XCTIN
1	/NAGSW
215	/CHAR
0	/LINENO
FRST+1	/PC
0	/THISLN
0	/THISOP
0	/LASTLN
1	/DEBCSW
1	/DMPSW
0	/PACKST

	0	
	0	
	0	
FILA2	0	/FILE POINTERS - 3 REGS
FILB2	0	
FILC2	0	
BOT2	XX	/FREE CORE POINTERS - 6 REGS
RUF2	XX	
ENDT2	XX	
	XX	
	XX	
	XX	
	XX	
	XX	
CMB2		/COMBUF=COMEIN
CMB2+77		/COMBOT=COMOUT
IMB2+2		/INBUF=IMBUFF+2
INB2+2		/INBUF=INBUF+2
OTB2+2		/OTBUF=OUTBUF+2
0		/AXOUT(TEXTP)
0		/XCTX
0		/GTEM
0		/M0DBUF
215		/ENDCR
0		/GETVCT
0		/SAVEOT
-1		/PUTCNT
0		/TEMPK
0		/INSUB
TTI2		/TTIN
TT02		/TTOUT
BKI2		/BLKIN
BK02		/BLKOUT
0		/LIBRSW
0		/EX1
0		/AC1H
0		/AC1L
0		/OVER1
0		/OVER2
10		/FISW
0		/GETP
0		/GETCX
0		/GET1X
0		/GET2
0		/GET3
OTB2+2		/PUTP
0		/PUTC
0		/PUT6
0		/SORTCN
0		/LASTOP
0		/ATSW(EFOP)
-20		/CNTR
4		/DECP
0		/ADD
0		/XCTIN
1		/NAGSW

	2000+TTI3	/CAL SAVES - 14(8) REGS
	3000+TTI3	
	2000+TTO3	
	3000+TTO3	
	2000+TTO3	
	0	
	0	
	0	
	0	
	0	
	0	
	0	
FILA3	0	/FILE POINTERS - 3 REGS
FILB3	0	
FILC3	0	
BOT3	XX	/FREE CORE POINTERS - 6 REGS
BUF3	XX	
ENDT3	XX	
	XX	
	XX	
	XX	
	XX	
	XX	
	XX	
	CMB3	/COMBUF=COMEIN
	CMB3+77	/COMBOT=COMOUT
	IMR3+2	/IMBUF=IMBUFF+2
	INR3+2	/INBUF=INBUF+2
	OTR3+2	/OTBUF=OUTBUF+2
	0	/AXOUT(TEXTP)
	0	/XCTX
	0	/GTEM
	0	/MODBUF
	215	/ENDCR
	0	/GETVCT
	0	/SAVEOT
	-1	/PUTCNT
	0	/TEMPK
	0	/INSUB
	TTI3	/TTIN
	TT03	/TTOUT
	BKI3	/BLKIN
	BK03	/BLKOUT
	0	/LIBRSW
	0	/EX1
	0	/AC1H
	0	/AC1L
	0	/OVER1
	0	/OVER2
	10	/FISH
	0	/GETP
	0	/GETCX
	0	/GET1X
	0	/GET2
	0	/GET3
	OTB3+2	/PUTP

```

0 /PUTC
0 /PUT6
0 /SORTCN
0 /LASTOP
0 /ATSW(EFOP)
-20 /CNTR
4 /DECP
0 /ADD
0 /XCTIN
1 /NAGSW
215 /CHAR
0 /LINEND
FRST+1 /PC
0 /THISLN
0 /THISOP
0 /LASTLN
1 /DEBGSW
1 /DMPSW
0 /PACKST
0 /PT1
0 /T1
0 /T2
0 /SACH
0 /FLARG - 3 REGS
0
0 /FLARG2 - 3 REGS
0
0
AREA3S=-AREA3
IMB3 2003; 0; 0; -1
INB3 .BLOCK 60
.ASCII <15>
OTB3 1000
.BLOCK 57
.ASCII <15>
CMB3 .BLOCK 100
215
/
AREA4 XSBEGN /RESTART - 1 REG
CTLP4
WAITB4
0 /FLAC - 3 REGS
0
0 /FRST
0 /LIST3+1
0 /AUTOXR - 5 REGS
0
0
IMB4 /BUFFER HEADER POINTERS - 4 REG
IMB4
INB4

```

	OTR4	
	0	/SUBROUTINE ENTRY POINTS - 10 (8) REG
	0	
	0	
	0	
	0	
	0	
	2000+TTI4	/CAL SAVES - 14(8) REGS
	3000+TTI4	
	2000+TT04	
	3000+TT04	
	2000+TT04	
	0	
	0	
	0	
	0	
	0	
	0	
	0	/FILE POINTERS - 3 REGS
FILA4	0	
FILB4	0	
FILC4	0	
BOT4	XX	/FREE CORE POINTERS - 6 REGS
BUF4	XX	
ENDT4	XX	
	XX	
	XX	
	XX	
	XX	
	CMB4	/COMBUF=COMEIN
	CMB4+77	/COMBOT=COMOUT
	IMB4+2	/IMBUFP=IMBUFF+2
	INB4+2	/INBUFP=INBUF+2
	OTB4+2	/OTBUFP=OUTBUF+2
	0	/AXOUT(TEXPT)
	0	/XCTX
	0	/GTEM
	0	/MOOBUF
	215	/ENDCR
	0	/GETVCT
	0	/SAVEOT
	-1	/PUTCNT
	0	/TENPK
	0	/INSUB
	TTI4	/TTIN
	TT04	/TTOUT
	BKI4	/BLKIN
	BK04	/BKOUT
	0	/LIBRSW
	0	/EX1
	0	/AC1H
	0	/AC1L

```

0 /OVER1
0 /OVER2
10 /FISW
0 /GETP
0 /GETCX
0 /GET1X
0 /GET2
0 /GET3
OTB4+2 /PUTP
0 /PUTC
0 /PUT6
0 /SORTCN
0 /LASTOP
0 /ATSW(EFOP)
-20 /CNTR
4 /DECP
0 /ADD
0 /XCTIN
1 /NAGSW
215 /CHAR
0 /LINENO
FRST+1 /PC
0 /THISLN
0 /THISOP
0 /LASTLN
1 /DEBGSW
1 /DMPSW
0 /PACKST
0 /PT1
0 /T1
0 /T2
0 /SACH
0 /FLARG - 3 REGS
0
0 /FLARG2 - 3 REGS
0
0

```

```

AREA4S=-AREA4
INB4 2003; 0; 0; -1
INB4 .BLOCK 60
      .ASCII <15>
OTB4 1000
      .BLOCK 57
      .ASCII <15>
CMB4 .BLOCK 100
      215
/

```

```

.ENDC
.ENDC
.EJECT

```

00512	R	200005	R	START	LAC	FRSTCV	
00513	R	540003	R		SAD	LASTCV	/ANY COMMON?
00514	R	600543	R		JMP	STARTQ	/NO - GO RESET POINTERS
00515	R	201146	R		LAC	MOVCOM	
00516	R	750200	A		SZA:CLA		
00517	R	200150	R		LAC	C100	
00520	R	040073	R		DAC	T1	
00521	R	200005	R		LAC	FRSTCV	
00522	R	340614	R		TAD	M1	
00523	R	340073	R		TAD	T1	
00524	R	065540	R		DAC*	(X15	
00525	R	200004	R		LAC	BUFR	/SETUP NEW COMMON START
00526	R	040005	R		DAC	FRSTCV	
00527	R	340614	R		TAD	M1	
00530	R	065541	R		DAC*	(X16	
00531	R	200073	R	STARTL	LAC	T1	
00532	R	103752	R		JMS	TWOS	
00533	R	365540	R		TAD*	(X15	
00534	R	540003	R		SAD	LASTCV	/ANY MORE COMMON?
00535	R	600541	R		JMP	STARTC	/NO
00536	R	220015	A		LAC*	X15	/YES - MOVE REG
00537	R	060016	A		DAC*	X16	
00540	R	600531	R		JMP	STARTL	
00541	R	225541	R	STARTC	LAC*	(X16	/SET NEW LAST ADDR
00542	R	600545	R		JMP	STARTB	
00543	R	200004	R	STARTQ	LAC	BUFR	
00544	R	040005	R		DAC	FRSTCV	
00545	R	040003	R	STARTB	DAC	LASTCV	
00546	R	040006	R		DAC	LASTV	/AND VARIABLE POINTERS
00547	R	440067	R	STARTZ	ISZ	DEBSW	/DISABLE TRACE FOR INPUT
00550	R	141146	R		DZM	MOVCOM	
00551	R	200010	R		LAC	COMBOT	/PROTECT COMMAND BUFFER
00552	R	065542	R		DAC*	(PDLXR	
00553	R	440070	R		ISZ	OMPSW	/INIT UNPACK AND TRACE SWITCH
00554	R	140502	R		DZM	LIST3+1	/CLEAR SEARCH CHARACTER FOR INPUT
00555	R	203327	R		LAC	RCAL01	/IS INPUT TTY IN?
00556	R	505543	R		AND	(777	
00557	R	540026	R		SAD	TTIN	
00560	R	741000	A		SKP		
00561	R	600575	R		JMP	IBAR	/DON'T PRINT *
00562	R	203412	R		LAC	WCAL01	
00563	R	505543	R		AND	(777	
00564	R	540027	R		SAD	TTOUT	/MAKE SURE TTY OUT
00565	R	600571	R		JMP	IBARX	
00566	R	200422	R		LAC	C252	/IF NOT TTY OUT USE IMAGE MODE
00567	R	103273	R		JMS	IMAGEN	
00570	R	600575	R		JMP	IBAR	
00571	R	200422	R	IBARX	LAC	C252	/ANNOUNCE PRESENCE
					PRINTC		
00572	R	103362	R	GEN*	JMS	XOUTL	
00573	R	200475	R		LAC	C375	
					PRINTC		
00574	R	103362	R	GEN*	JMS	XOUTL	
00575	R	200007	R	IBAR	LAC	COMBUF	/INITIALIZE COMMAND BUFFER.
00576	R	065544	R		DAC*	(AXIN	

00577	R	140057	R		DZM	XCTIN	
00600	R	200162	R		LAC	CFRSX	
00601	R	040063	R		DAC	PC	
00602	R			IGNOR	READC		/READ COMMAND STRING
00602	R	103322	R GEN*		JMS	XI33	
00603	R	540501	R		SAD	C215	
00604	R	600607	R		JMP	IRETN	
					PACKC		/SAVE STRING CHARACTER.
00605	R	103562	R GEN*		JMS	PACBUF	
00606	R	600602	R		JMP	IGNOR	
				/////			
00607	R			IRETN	PACKC		/ PACK C.R.
00607	R	103562	R GEN*		JMS	PACBUF	
00610	R	440063	R		ISZ	PC	
00611	R	200007	R		LAC	COMBUF	/INITIALIZE "TEXTP"
00612	R	340167	R	GONE	TAD	C1	
00613	R	040014	R		DAC	AXOUT	/SETUP CURRENT LINE
00614	R	777777	A	M1	LAW	-1	
00615	R	040015	R		DAC	XCTX	
					GETC		/READ FIRST CHARACTER.
00616	R	102752	R GEN*		JMS	UTRA	
00617	R	200000	R		LAC	BOTTOM	/INIT PUSH-DOWN-LIST
00620	R	065542	R		DAC*	(PDLXR	
					SPNOR		
00621	R	102233	R GEN*		JMS	XSPNOR	
00622	R	102242	R GEN*		TESTN		/DOES THE LINE BEGIN WITH 1-9?
					JMS	XTESTN	
					ERROR	1	
				GEN*	.DEC		
00623	R	603474	R GEN*		JMP	ERR2-1	
00624	R	741000	A		SKP		
00625	R	600640	R		JMP	INPUTX	/YES
00626	R	140067	R		DZM	DEBGSW	/ENABLE TRACE
00627	R	140062	R		DZM	LINENO	
					PUSHJ	PROC	/PROCESS IMMEDIATE COMMAND.
00630	R	101100	R GEN*		JMS	XPUSHJ	
00631	R	001204	R GEN*			PROC	
00632	R	220063	R		LAC*	PC	/CHECK NEXT LINE (X-MEM)
00633	R	741200	A		SNA		/END OF PROGRAM?
00634	R	600547	R		JMP	STARTZ	/YES
00635	R	040063	R		DAC	PC	/SAVE NEW LINE NO.
00636	R	340167	R		TAD	C1	/START NEW LINE
00637	R	600612	R		JMP	GONE	/PROCESS OTHER COMMANDS
				/////			
00640	R			INPUTX	GETLN		/READ THIS LINE NUMBER
00640	R	100664	R GEN*		JMS	XGETLN	
00641	R	200060	R		LAC	NAGSW	
00642	R	750100	A		SMA:CLA		/TEST FOR SINGLE LINE
					ERROR	2	
				GEN*	.DEC		
00643	R	603473	R GEN*		JMP	ERR2-2	
00644	R	101146	R		JMS	MOVCOM	/OFFSET COMMON
00645	R	200004	R		LAC	BUFR	/SET POINTERS
00646	R	065544	R		DAC*	(AXIN	
00647	R	140057	R		DZM	XCTIN	


```

00650 R 200062 R
00651 R 060010 A
00652 R 751000 A

00653 R 102752 R GEN*

00654 R 103562 R GEN*
00655 R 540501 R
00656 R 741000 A
00657 R 600653 R

00660 R 101100 R GEN*
00661 R 002550 R GEN*

00662 R 103051 R GEN*
00663 R 600512 R

```

```

LAC          LINENO  /SAVE LINE #
DAC*        AXIN    /(X-MEM)
SKP:CLA
GETC
JMS         /READ 1ST AFTER LINENO TERMINATOR.
PACKC      UTRA
JMS        /SAVE SPACE AND OTHERS - RESTORE DATA FIELD
SAD        PACBUF
C215      /TEST FOR END
JMP
PUSHJ     .-4
JMS       DELETE  /REMOVE OLD LINE, IF ANY.
          XPUSHJ
          DELETE
ENDLN     /INSERT NEW LINE
JMS
JMP      XENDLN
          START
.EJECT

```

/TEXT LINE BUFFER FORMAT*
 /#1 : POINTER OR ZERO IN LAST
 /#2 : LINENO
 /#3 - #N+1 : TEXT
 /#N : C.R.

00664	R	000000	A	XGETLN	0		/DEVELOP I.D. - "GETLN"
				SPNOR			/IGNORE LEADING ZEROS AND SPACES.
00665	R	102233	R GEN*	JMS		XSPNOR	
				TESTN			
00666	R	102242	R GEN*	JMS		XTESTN	
00667	R	740000	A	NOP			
00670	R	600736	R	JMP		TESTA	
00671	R	140025	R	DZM		INSUB	/CALL 'GETC' FROM 'INPUT' FROM 'DECON'
00672	R	165522	E	DZM* .AB			
00673	R	165523	E	DZM* .AC			
00674	R	140037	R	DZM		OVER2	
00675	R	104550	R	JMS		DECON	
00676	R	200037	R	LAC OVER2			
				RTL6			
00677	R	100770	R GEN*	JMS		XRTL6	
00700	R	740010	A	RAL			
00701	R	040062	R	DAC		LINENO	
00702	R	500155	R	AND		P177	/GROUP TOO LARGE
00703	R	750200	A	SZA!CLA			
				ERROR		3	/YES
			GEN*	.DEC			
00704	R	603472	R GEN*	JMP		ERR2-3	
00705	R	225523	E	LAC* .AC			
00706	R	740200	A	SZA			
				ERROR		3	/GROUP TOO LARGE
			GEN*	.DEC			
00707	R	603472	R GEN*	JMP		ERR2-3	
				TESTN			/TEST3
00710	R	102242	R GEN*	JMS		XTESTN	
00711	R	102752	R GEN*	GETC		UTRA	/READ STEP NUMBER.
				JMS			
			/OTHER	TESTN			/TEST4
00712	R	102242	R GEN*	JMS		XTESTN	
				ERROR		5	/DOUBLE PERIODS
			GEN*	.DEC			
00713	R	603470	R GEN*	JMP		ERR2-5	
00714	R	600755	R	JMP		GEXIT	/OTHER
00715	R	744000	A	CLL			/NUMBER *12
00716	R	742010	A	RTL			
00717	R	340051	R	TAD		SORTCN	
00720	R	740010	A	RAL			
00721	R	340062	R	TAD		LINENO	
00722	R	040062	R	DAC		LINENO	
				GETC			/GET FINAL DIGIT
00723	R	102752	R GEN*	JMS		UTRA	
				TESTN			/TEST5
00724	R	102242	R GEN*	JMS		XTESTN	
				ERROR		5	/MULTIPLE PERIODS
			GEN*	.DEC			

00725	R	603470	R	GEN*	JMP	ERR2-5	
00726	R	600755	R		JMP	GEXIT	/OTHER
00727	R	340062	R		TAD	LINENO	
00730	R	040062	R		DAC	LINENO	
00731	R	102752	R	GEN*	GETC		/TEST FOR CORRECT TERMINATOR
					JMS	UTRA	
					TESTN		/TEST6 - I.E. NOT A NUMBER OR "."
00732	R	102242	R	GEN*	JMS	XTESTN	
00733	R	741000	A		SKP		
00734	R	600755	R		JMP	GEXIT	
					ERROR	6	/TOO LARGE A LINE NUMBER.
				GEN*	.DEC		
00735	R	603467	R	GEN*	JMP	ERR2-6	
00736	R	200061	R		LAC	CHAR	
00737	R	540450	R	TESTA	SAD	C242	
00740	R	604224	R		JMP	LBTEXT	
00741	R	140062	R		DZM	LINENO	
					SORTC	GLIST	
00742	R	101321	R	GEN*	JMS	XSORTC	
00743	R	000453	R	GEN*		GLIST-1	
00744	R	600755	R		JMP	GEXIT	
00745	R	777767	A		LAW	-11	
00746	R	102110	R		JMS	FETVAR	
00747	R	220001	R		LAC*	BUFSTX	
00750	R	741200	A		SNA		
00751	R	600755	R		JMP	GEXIT	
					SORTJX	ALLCM1	
00752	R	102040	R	GEN*	JMS	XSORTX	
00753	R	000315	R	GEN*		ALLCM1-1	
					ERROR	23	
				GEN*	.DEC		
00754	R	603446	R	GEN*	JMP	ERR2-23	
00755	R	200062	R		LAC	LINENO	/TEST FOR GROUP NUMBER.
00756	R	500155	R	GEXIT	AND	P177	
00757	R	754200	A		SZA:CLA:CLL		
00760	R	740002	A		CML		
00761	R	340062	R		TAD	LINENO	
00762	R	501347	R		AND	P7600	
00763	R	751200	A		SNA:CLA		
00764	R	340147	R		TAD	P2	
00765	R	740020	A		RAR		
00766	R	040060	R		DAC	NAGSW	
00767	R	620664	R		JMP*	XGETLN	
					/RANGE OF ACCEPTIBLE LINE NUMBERS = 1.01 TO 99.99		
					/NAGSW:		
					/GROUP=000000		
					/LINE=400000		
					/ALL=000001		
00770	R	000000	A		XRTL6	0	/ROTATE AC LEFT SIX - "RTL6"
00771	R	744000	A		CLL		
00772	R	742010	A		RTL		
00773	R	742010	A		RTL		
00774	R	742010	A		RTL		
00775	R	620770	R		JMP*	XRTL6	
					.EJECT		

```

/RECURSIVE OPERATE, EXECUTE, OR CALL
00776 R          DO          GETLN          /EXECUTE ONE LINE, A GROUP, OR ALL
00776 R 100664 R GEN*      JMS          XGETLN
00777 R 200063 R          LAC          PC          /SAVE ADDRESS
                                PUSHA         /OF CURRENT LINE
01000 R 101055 R GEN*      JMS          XPUSHA
                                PUSHF        TEXTP        /SAVE REST OF THIS LINE
01001 R 101113 R GEN*      JMS          PD2
01002 R 000014 R GEN*      JMS          TEXTP
01003 R          DGRP      PUSHF        NAGSW        /SAVE NAGSW; CHAR; AND LINENO.
01003 R 101113 R GEN*      JMS          PD2
01004 R 000060 R GEN*      LAC          NAGSW
01005 R 200060 R          LAC          NAGSW
01006 R 751100 A          SPA:CLA        /CHECK DATA FROM GETLN.
01007 R 601034 R          JMP          ONE        /SKIP IF GROUP OR ALL
                                FINDLN       /DO ONE LINE
                                JMS          /INIT FOR GROUP AND SET THISLN
01010 R 102675 R GEN*      JMS          XFIND
01011 R 601047 R          JMP          TGRP2
01012 R          DGRP1    PUSHJ        PROCES-2 /EXECUTE OBJECT LINE AND SET PC.
01012 R 101100 R GEN*      JMS          XPUSHJ
01013 R 001201 R GEN*      POPF        PROCES-2
                                JMS          NAGSW        /RESTORE THE DATA
                                JMS          PD3
                                JMS          NAGSW
01014 R 101132 R GEN*      LAC*        PC          /CHECK FOR END OF TEXT (X-MEM)
01015 R 000060 R GEN*      SNA
01016 R 220063 R          JMP          DCONT
01017 R 741200 A          JMP          /ALL DONE
01020 R 601042 R          TAD          C1
01021 R 340167 R          DAC          PT1        /SAVE POINTER TO LINENO
01022 R 040072 R          LAC          NAGSW
01023 R 200060 R          SMA:SZ:CLA /CHECK FOR GROUP
01024 R 750300 A          JMP          .+4        /DO ALL
01025 R 601031 R          TAD*        PT1        /TEST GROUP (X-MEM)
01026 R 360072 R          TSTGRP
                                JMS          GRPTST
01027 R 101343 R GEN*      JMP          DCONT
01030 R 601042 R          LAC*        PT1        /NOT IN GROUP
01031 R 220072 R          DAC          LINENO /READ NEXT LINE NO. (X-MEM)
01032 R 040062 R          JMP          DGRP      /CONTINUE THE SUBROUTINE
01033 R 601003 R          //
01034 R          ONE      FINDLN
01034 R 102675 R GEN*      JMS          XFIND
                                ERROR          7
                                .DEC
01035 R 603466 R GEN*      JMP          ERR2-7
                                PUSHJ        PROCES
01036 R 101100 R GEN*      JMS          XPUSHJ /EXECUTE IT
01037 R 001203 R GEN*      LPROCS=-1
                                JMS          PROCES
01040 R 101132 R GEN*      POPF        /RESTORE CHAR
01041 R 000060 R GEN*      JMS          PD3
01042 R          DCONT    POPF        NAGSW
01042 R 101132 R GEN*      JMS          TEXTP /RESTORE TEXT POINTERS
01043 R 000014 R GEN*      JMS          PD3
                                TEXTP

```

01044 R 220013 A GEN*
01045 R 040063 R
01046 R 601204 R

01047 R 200064 R
01050 R 065545 R
01051 R 220011 A

01052 R 101343 R GEN*

01053 R 603465 R GEN*
01054 R 601012 R

//////
TGRP2

POPA
LAC*
DAC
JMP

LAC
DAC*
LAC*
TSTGRP
JMS
ERROR
.DEC
JMP
JMP
.EJECT

/RESTORE ADDRESS OF CURRENT LINE.

/CONTINUE PROCESSING THIS LINE.

/TEST FOR GOOD GROUP NUMBER.

POLXR
PC
PROC

THISLN
(XRT
XRT

GRPTST
8

ERR2-8
DGRP1

			/PUSHDOWN LIST CONTROLS			
01055	R	000000	A	XPUSHA	0	/PUSHDOWN THE AC - "PUSHA"
01056	R	040074	R	DAC	T2	/BACKUP POINTER
01057	R	750001	A	CLA:CMA		/AND THEN
01060	R	101067	R	JMS	PCHK	/CHECK CORE USAGE
01061	R	200074	R	LAC	T2	/OK
01062	R	060013	A	DAC*	PDLXR	/PUSH DOWN LIST POINTER
01063	R	750001	A	CLA:CMA		/BACKUP AGAIN
01064	R	101067	R	JMS	PCHK	
01065	R	200074	R	LAC	T2	
01066	R	621055	R	JMP*	XPUSHA	
01067	R	000000	A	0		
01070	R	365542	R	TAD*	(PDLXR	/INC IN AC
01071	R	065542	R	DAC*	(PDLXR	
01072	R	103752	R	JMS	TWOS	
01073	R	744000	R	CLL		
01074	R	340006	R	TAD	LASTV	
01075	R	751400	A	SZL:CLA		
				ERROR	9	/STORAGE FILLED BY PUSH-DOWN LIST
				.DEC		
01076	R	603464	R	JMP	ERR2-9	
01077	R	621067	R	JMP*	PCHK	
01100	R	000000	A	0		/RECURSIVE SUBROUTINE CALL - "PUSHJ"
01101	R	221100	R	LAC*	XPUSHJ	
01102	R	040074	R	DAC	T2	/SAVE SUBR. ADDR.
01103	R	750001	A	CLA:CMA		
01104	R	101067	R	JMS	PCHK	
01105	R	341100	R	TAD	XPUSHJ	
01106	R	340167	R	TAD	C1	
01107	R	060013	A	DAC*	PDLXR	/SAVE RETURN
01110	R	750001	A	CLA:CMA		
01111	R	101067	R	JMS	PCHK	
01112	R	620074	R	JMP*	T2	/TRANSFER CONTROL
01113	R	000000	A	0		/SAVE A FLOATING POINT NUMBER - "PUSHF"
01114	R	750001	A	CLA:CMA		/COMPUTE VARIABLE ADDR
01115	R	361113	R	TAD*	PD2	
01116	R	065545	R	DAC*	(XRT	
01117	R	441113	R	ISZ	PD2	/FIX RETURN
01120	R	777775	A	LAW	-WORDS	/COMPUTE PUSH. POINTER
01121	R	040074	R	DAC	T2	
01122	R	101067	R	JMS	PCHK	
01123	R	220011	A	LAC*	XRT	/ (X-MEM)
01124	R	060013	A	DAC*	PDLXR	
01125	R	440074	R	ISZ	T2	
01126	R	601123	R	JMP	-3	
01127	R	201120	R	LAC	MFLT	/RESET POINTER
01130	R	101067	R	JMS	PCHK	
01131	R	621113	R	JMP*	PD2	
01132	R	000000	A	0		/RESTORE A FLOATING POINT NUMBER - "POPF"
01133	R	750001	A	CLA:CMA		/GET VAR. ADDR.
01134	R	361132	R	TAD*	PD3	
01135	R	441132	R	ISZ	PD3	
01136	R	065545	R	DAC*	(XRT	
01137	R	201120	R	LAC	MFLT	
01140	R	040074	R	DAC	T2	

01141 R 220013 A
 01142 R 060011 A
 01143 R 440074 R
 01144 R 601141 R
 01145 R 621132 R

////
 MOVCOM

01146 R 000000 A
 01147 R 200005 R
 01150 R 540003 R
 01151 R 621146 R
 01152 R 040017 R
 01153 R 200003 R
 01154 R 040074 R
 01155 R 340150 R
 01156 R 040073 R
 01157 R 103752 R
 01160 R 365542 R
 01161 R 741100 A

GEN*
 GEN*

MOVUPX

01162 R 603455 R
 01163 R 220074 R
 01164 R 060073 R
 01165 R 200074 R
 01166 R 540017 R
 01167 R 621146 R
 01170 R 340614 R
 01171 R 040074 R
 01172 R 200073 R
 01173 R 340614 R
 01174 R 040073 R
 01175 R 601163 R

LAC* PDLXR /MOVE
 DAC* XRT
 ISZ T2
 JMP .-3
 JMP* PD3 /EXIT

Ø /MOVE COMMON AREA
 LAC FRSTCV
 SAD LASTCV /ANY COMMON?
 JMP* MOVCOM /NO
 DAC MODBUF
 LAC LASTCV
 DAC T2 /CURRENT END
 TAD C100
 DAC T1 /NEW END
 JMS TWOS
 TAD* (PDLXR
 SPA /OVERFLOW

FRROR 16
 .DEC
 JMP ERR2-16
 LAC* T2
 DAC* T1
 LAC T2 /IS IT AT END?
 SAD MODBUF
 JMP* MOVCOM /YES - EXIT
 TAD M1 /NO - BACKUP POINTERS
 DAC T2
 LAC T1
 TAD M1
 DAC T1
 JMP MOVUPX

.EJECT

			/PRIMARY CONTROL AND TRANSFER		
01176	R		GOTO	GETLN	/READ THE LINE NUMBER REQUESTED
01176	R	100664 R GEN*	JMS	XGETLN	
01177	R	102675 R GEN*	FINDLN		/LOCATE IT AND RESET TEXTP
			JMS	XFIND	
			ERROR	7	/NOT THERE
			.DEC		
01200	R	603466 R GEN*	JMP	ERR2-7	
01201	R	200064 R	LAC	THISLN	/SET PC
01202	R	040063 R	DAC	PC	
01203	R		PROCES	GETC	/TEST FOR END OF LINE
01203	R	102752 R GEN*	JMS	UTRA	
		001204 R	PROC=.		
			.IFDEF MULTI		
			LAC	DELAY	
			SZA		
			JMS	IOBUSY	
			.ENDC		
01204	R	200433 R	LAC	C273	
01205	R	040456 R	DAC	TLISTX	/RESET IN CASE ENTRY FROM COMMON STMT
01206	R	200061 R	LAC	CHAR	/FIRST CHARACTER READY = USE PROC
01207	R	540501 R	SAD	C215	
01210	R		PC1	POPJ	/EXIT "PROCESS"
01210	R	602265 R GEN*	JMP	XPOPJ	
			SORTC	GLIST	/IGNORE "SPACE", ",", AND ":",
			JMS	XSORTC	
				GLIST-1	
01211	R	101321 R GEN*	JMP PROCES		
01212	R	000453 R GEN*	LAC	LIBRSW	
01213	R	601203 R	RAL		
01214	R	200032 R	LAC	CHAR	
01215	R	740010 A	SAD	C314	
01216	R	200061 R	JMP	.*3	
01217	R	540164 R	SZL		
01220	R	601223 R	ERROR	31	
01221	R	741400 A	.DEC		
			JMP	ERR2-31	
01222	R	603436 R GEN*	LAW	-11	
01223	R	777767 A	JMS	FETVAR	/GET COMMAND (3 WORDS)
01224	R	102110 R	SORTC	GLIST	
			JMS	XSORTC	
01225	R	101321 R GEN*		GLIST-1	
01226	R	000453 R GEN*	SKP		
01227	R	741000 A	ERROR	10	
			.DEC		
01230	R	603463 R GEN*	JMP	ERR2-10	
			SORTJX	COMLST	/GO DO COMMAND
01231	R	102040 R GEN*	JMS	XSORTX	
01232	R	000224 R GEN*		COMLST-1	
			ERROR	10	/ILLEGAL COMMAND
			.DEC		
01233	R	603463 R GEN*	JMP	ERR2-10	
			.EJECT		


```

001210 R      //
COMMEN=PC1      /IS CONTINUE OR COMMENT
/OUTPUT COMMAND TEXT
WRITE          GETLN          /SET LINENO
01234 R      JMS          XGETLN
01234 R 100664 R GEN*      JSZ          DEBGSW          /DISABLE TRACE (ALWAYS DURING WRITE)
01235 R 440067 R          FINDLN          /SEARCH FOR LINE NUMBER
01236 R 102675 R GEN*      JMS          XFIND
01237 R 601271 R          JMP          WTESTG          /NOT THERE OR GROUP
01240 R 200062 R          LAC          LINENO
01241 R 750200 A          SZA:CLA
01242 R 601246 R          JMP .+4
01243 R 200501 R          LAC C215
01244 R 103362 R GEN*      PRINTC          XOUTL
01245 R 741000 A          JMS          SKP          /PRINT LINE NUMBER
01246 R 103065 R GEN*      JMS          XPRNT
01247 R 102752 R GEN*      GETC          UTRA
01250 R 103362 R GEN*      PRINTC          /PRINT TEXT OF A LINE.
01251 R 540501 R          JMS          XOUTL
01252 R 741000 A          SAD          C215
01253 R 601247 R          SKP          /SKIP IF END OF LINE
01254 R 220064 R          JMP          .-4
01255 R 741200 A          LAC*          THISLN          /TEST FOR END OF TEXT (X-MEM)
01256 R 601276 R          WTEST2      SNA          /EXIT:DO NEXT INDIRECT LINC.
01257 R 340167 R          JMP          WRITED
01260 R 040072 R          TAD          C1
01261 R 200060 R          DAC          PT1          /SAVE POINTER TO LINENO OF NEXT (X-MEM)
01262 R 750100 A          LAC          NAGSW
01263 R 360072 R          SMA:CLA
01264 R 101343 R GEN*      TAD*          PT1          /(X-MEM)
01265 R 601273 R          TSTGRP          /TRY NEXT LINENO FOR GROUP.
01266 R 220072 R          JMS          GRPTST
01267 R 040062 R          WALL          WX
01270 R 601236 R          LAC*          PT1          /SET LINENO (X-MEM)
01271 R 200064 R          DAC          LINENO
01272 R 601255 R          JMP          WRITE+2
01273 R 200060 R          //          LAC          NAGSW
01274 R 750300 A          WX          SMA: SZA:CLA          /SKIP IF NOT ALL
01275 R 601266 R          WALL
01276 R 200501 R          WRITED      LAC C215
01277 R 103362 R GEN*      PRINTC
01300 R 602265 R GEN*      JMS          XOUTL
001300 R          POPJ
LPOPJ=-1        JMP          XPOPJ
.EJECT

```

01301	R	000000	A	//// XTESTC	0	/TEST THE NATURE OF THE NEXT ALPHANUMERIC - "TESTC"
01302	R	102233	R GEN*		SPNOR	/IGNORE SPACES AND ZEROS HERE
01303	R	101321	R GEN*		JMS	XSPNOR
01304	R	000415	R GEN*		XSORTC	TERMS
01305	R	601317	R		JMS	/TEST THE VARIABLE TERMINATOR FOR EVAL
01306	R	441301	R		JMP	XTESTX
01307	R	540163	R		ISZ	XTESTC
01310	R	601316	R		SAD	C306
					JMP	XTESTC
					TESTN	/TEST FOR . OR 0-9
01311	R	102242	R GEN*		JMS	XTESTN
01312	R	601317	R		JMP	XTESTX
01313	R	741000	A		SKP	
01314	R	601317	R		JMP	XTESTX
01315	R	441301	R		ISZ	XTESTC
01316	R	441301	R		ISZ	XTESTC
01317	R	200061	R	XT3	LAC	CHAR
01320	R	621301	R		JMP*	XTESTC
01321	R	000000	A	XSORTC	0	/SORT CHAR AGAINST TABLE - "SORTC"
01322	R	221321	R		LAC*	XSORTC
01323	R	065546	R		DAC*	(XRT2
01324	R	220012	A		LAC*	XRT2
01325	R	741100	A		SPA	
01326	R	601337	R		JMP	SEXC
01327	R	540061	R		SAD	CHAR
01330	R	741000	A		SKP	/COMPARE
01331	R	601324	R		JMP	-5
01332	R	221321	R		LAC*	XSORTC
01333	R	740001	A		CMA	/COMPUTE INCREMENT : 0 - N
01334	R	365546	R		TAD*	(XRT2
01335	R	040051	R		DAC	SORTCN
01336	R	741000	A		SKP	/1ST EXIT = YES
01337	R	441321	R	SEXC	ISZ	XSORTC
01340	R	441321	R		ISZ	XSORTC
01341	R	200061	R		LAC	CHAR
01342	R	621321	R		JMP*	XSORTC
01343	R	000000	A	GRPTST	0	/AC VS LINENO - "TSTGRP"
01344	R	501347	R		AND	P7600
01345	R	103752	R		JMS	TWOS
01346	R	040074	R		DAC	T2
01347	R	777600	A	P7600	LAW	17600
01350	R	500062	R		AND	LINENO
01351	R	340074	R		TAD	T2
01352	R	751200	A		SNA:CLA	
01353	R	441343	R		ISZ	GRPTST
01354	R	621343	R		JMP*	GRPTST
					.EJECT	

/CONDITIONAL TRANSFER PROCESS.

				GETC		/IF (EXP) A,B,C;
01355	R	102752	R GEN*	JMS	UTRA	
01356	R	200061	R	LAC	CHAR	
01357	R	540416	R	SAD	C240	
01360	R	601355	R	JMP	IF-1	
01361	R	540424	R	SAD	C250	
01362	R	741000	A	SKP		
				ERROR	11	/NO SPACE AFTER IF OR ILLEGAL FORMAT.
			GEN*	.DEC		
01363	R	603462	R GEN*	JMP	ERR2-11	
				PUSHJ	EVAL-1	/EVALUATE EXPRESSION
01364	R	101100	R GEN*	JMS	XPUSHJ	
01365	R	002271	R GEN*		EVAL-1	
				GETC		/MOVE PAST ")"
01366	R	102752	R GEN*	JMS	UTRA	
01367	R	777776	A	LAW	-2	
01370	R	040073	R	DAC	T1	
01371	R	200077	R	LAC	FLARG+1	/TEST -.0.+
01372	R	741100	A	SPA		
01373	R	440073	R	ISZ	T1	/TO -1,-2,-3
01374	R	751300	A	SPA: SNA: CLA		
01375	R	440073	R	ISZ	T1	/COUNT COMMAS
01376	R	741000	A	SKP		
01377	R	601176	R	JMP	GOTO	/TRANSFER
				SORTJ	TLIST, ILIST	/SEARCH TEXT UNTILL ;;C.R.
01400	R	102013	R GEN*	JMS	SORTB	
01401	R	000454	R GEN*		TLIST-1	
01402	R	000003	A GEN*		ILIST-TLIST	
				GETC		
01403	R	102752	R GEN*	JMS	UTRA	
01404	R	601400	R	JMP	.-4	
01405	R	102752	R GEN*	GETC		/MOVE PAST
01406	R	601375	R	JMS	UTRA	
				JMP	IF3	
				.EJECT		

```

//////
/LOOP CONTROL STATEMENT
01407 R 001407 R SET=.
01407 R 101100 R GEN* FOR PUSHJ GETARG /SUBSET OF "FOR".
01410 R 002135 R GEN* JMS XPUSHJ /LOOPS, ETC.
                                GETARG
                                SPNOR
                                JMS XSPNOR /IGNORE SPACES
                                SAD C275
                                SKP
                                ERROR 12 /LEFT OF "=" IN ERROR: 'FOR' OR 'SET'
                                .DEC
                                JMP ERR2-12
                                LAC PT1
                                PUSHA /SAVE POINTER TO VARIABLE
                                JMS XPUSHA
                                PUSHJ EVAL-1 /GET INITIAL VALUE EXPRESSION
                                JMS XPUSHJ
                                EVAL-1
                                POPA
                                LAC* PDLXR
                                DAC PT1
                                JMS FINT /INITIALIZE NOW.
                                FGET FLARG
                                FPUT* PT1
                                FXIT
                                SORTJ TLIST,FLIST1 /TEST LAST CHAR FROM "EVAL"
                                JMS SORTB
                                TLIST-1
                                FLIST1-TLIST
                                ERROR 13 /EXCESS R-PAR
                                .DEC
                                JMP ERR2-13
//////
FINCR LAC LPROCS /SET OPERATION
01433 R 201037 R DAC FPUSHJ
01434 R 041460 R LAC LPOPJ /SET EXIT
01435 R 201300 R DAC FPOPJ
01436 R 041501 R LAC PT1 /SAVE VARIABLE ADDRESS
01437 R 200072 R PUSHA
                                JMS XPUSHA
                                PUSHJ EVAL-1 /EVALUATE THE INCREMENT, IF ANY.
                                JMS XPUSHJ
                                EVAL-1
                                SORTJ TLIST,FLIST2 /TEST TERMINATORS
                                JMS SORTB
                                TLIST-1
                                FLIST2-TLIST
                                ERROR 14
                                .DEC
                                JMP ERR2-14
//////
FLIMIT PUSHF FLARG /SAVE THE INCRE.
01447 R 101113 R GEN* JMS PD2
01450 R 000076 R GEN* FLARG

```

01451	R	101100	R GEN*	PUSHJ	EVAL-1	/GET THE LIMIT
01452	R	002271	R GEN*	JMS	XPUSHJ	
01453	R			FCONT	EVAL-1	
01453	R	101113	R GEN*	PUSHF	FLARG	/SAVE THE LIMIT *
01454	R	000076	R GEN*	JMS	PD2	
					FLARG	
01455	R	101113	R GEN*	PUSHF	TEXTP	/SAVE TEXT OF OBJECT STATEMENTS
01456	R	000014	R GEN*	JMS	PD2	
					TEXTP	
01457	R	101100	R GEN*	PUSHJ	PROCES	/DO THE OBJECT STATEMENTS
01460	R	001203	R GEN*	JMS	XPUSHJ	
		001460	R		PROCES	
				FPUSHJ=-1		
01461	R	101132	R GEN*	POPF	TEXTP	/RESTORE REMAINING TEXT.
01462	R	000014	R GEN*	JMS	PD3	
					TEXTP	
01463	R	101132	R GEN*	POPF	FLARG	/GET LIMIT
01464	R	000076	R GEN*	JMS	PD3	
					FLARG	
01465	R	101132	R GEN*	POPF	FLARG2	/GET INCREMENT
01466	R	000101	R GEN*	JMS	PD3	
					FLARG2	
01467	R	220013	A GEN*	POPA		/GET VARIABLE ADDRESS
01470	R	040072	R	LAC*	PDLXR	
01471	R	105140	R	DAC	PT1	
01472	R	520072	R	JMS	FINT	/INCREMENT AND TEST
01473	R	100101	R	FGET* PT1	/LOAD THE VARIABLE	
01474	R	620072	R	FADD FLARG2	/INCREMENT IT	
01475	R	200076	R	FPUT* PT1	/CHANGE IT	
01476	R	000000	A	FSUB FLARG	/TEST IT	
				FXIT		
				GETSGN		
01477	R	225522	E GEN*	LAC*	.AB	
01500	R	750300	A	SMA!SZA!CLA		
				POPJ		/END OF LOOP
01501	R	602265	R GEN*	JMP	XPOPJ	
		001501	R			
				FPOPJ=-1		
01502	R	200072	R	LAC	PT1	
				PUSHA		/SAVE ADDRESS *
01503	R	101055	R GEN*	JMS	XPUSHA	
				PUSHF	FLARG2	/SAVE INCREMENT AGAIN *
01504	R	101113	R GEN*	JMS	PD2	
01505	R	000101	R GEN*		FLARG2	
01506	R	601453	R	JMP	FCONT	
				////		
01507	R			FINFIN		
01507	R	101113	R GEN*	PUSHF	FLTONE	/SET INCREMENT TO ONE.
01510	R	000167	R GEN*	JMS	PD2	
01511	R	601453	R	JMP	FLTONE	
				.EJECT	FCONT	

01512	R	102752	R	GEN*		GETC			
01513	R	200003	R		COMMON	JMS	UTRA		
01514	R	540006	R			LAC	STARTV		/CHECK FOR LEGALITY
01515	R	741000	A			SAD	LASTV		/OF COMMON STATEMENT
01516	R	601210	R			SKP			
						JMP	COMMEN		/NOT LEGAL - COMMENT
						SPNOR			/OK
01517	R	102233	R	GEN*		JMS	XSPNOR		
01520	R	540424	R			SAD	C250		/IS IT LEFT PAREN
01521	R	601534	R			JMP	COMARY		/YES - PROCESS ARRAY
						PUSHJ	GETARG		/NO - NORMAL VARIABLE
						JMS	XPUSHJ		
							GETARG		
01522	R	101100	R	GEN*		SPNOR			
01523	R	002135	R	GEN*	COMMX	JMS	XSPNOR		
01524	R					LAC	LASTV		/SET END OF COMMON
01524	R	102233	R	GEN*		DAC	LASTCV		
01525	R	200006	R			LAC	CHAR		
01526	R	040003	R			SORTJ	TLIST,CLISTX		/CHECK FOR TERMINATOR
01527	R	200061	R			JMS	SORTB		
							TLIST-1		
01530	R	102013	R	GEN*			CLISTX-TLIST		
01531	R	000454	R	GEN*			37		/FORMAT ERROR
01532	R	000014	A	GEN*		ERROR			
				GEN*		.DEC			
01533	R	603430	R	GEN*		JMP	ERR2-37		
01534	R				COMARY	GETC			
01534	R	102752	R	GEN*		JMS	UTRA		/CHECK FIRST CHAR
						TESTC			
01535	R	101301	R	GEN*		JMS	XTESTC		
01536	R	740000	A			NOP			
01537	R	740000	A			NOP			
						ERROR	15		/FORMAT ERROR
				GEN*		.DEC			
01540	R	603456	R	GEN*		JMP	ERR2-15		
01541	R	777775	A			LAW	-3		/GET WHOLE VARIABLE
01542	R	102110	R			JMS	FETVAR		/NAME
01543	R	200061	R			LAC	CHAR		
01544	R	540432	R			SAD	C254		/MUST BE A COMMA
01545	R	741000	A			SKP			
						ERROR	37		/FORMAT ERROR
				GEN*		.DEC			
01546	R	603430	R	GEN*		JMP	ERR2-37		
01547	R	220001	R			LAC	BUFSTX		/GET VARIABLE NAME
01550	R	040142	R			DAC	ARRAYN		
						PUSHJ	EVAL-1		/SKIP COMMA AND EVALUATE
						JMS	XPUSHJ		
01551	R	101100	R	GEN*			EVAL-1		
01552	R	002271	R	GEN*		JMS	FINT		
01553	R	105140	R			FGET	FLARG		
01554	R	500076	R			FPUT	XY		
01555	R	600137	R			FXIT			
01556	R	000000	A			LAC	LITX		/USE X AS COUNTER
01557	R	201573	R			DAC	PT1		
01560	R	040072	R			LAC	C251		/SET TERMINATOR
01561	R	200427	R			DAC	TLISTX		
01562	R	040456	R						

01563 R 205547 R
 01564 R 041460 R
 01565 R 205550 R
 01566 R 041501 R
 01567 R 601437 R
 01570 R 200142 R
 01571 R 060001 R
 01572 R 125524 E
 01573 R 000137 R
 01574 R 602160 R
 01575 R 200006 R
 01576 R 040003 R
 01577 R 200061 R
 01600 R 540427 R
 01601 R 601524 R

01602 R 603430 R GEN*
 GEN*

COMDEC

LITX

COMEND

LAC (COMDEC
 DAC FPUSHJ
 LAC (JMP COMEND
 DAC FPOPJ
 JMP FINCRX
 LAC ARRAYN
 DAC* BUFSTX
 JMS* .AO
 XY GS1A
 JMP LASTV
 LAC LASTCV
 DAC CHAR
 LAC C251
 SAD COMMX
 JMP 37
 ERROR
 .DEC
 JMP ERR2-37

/SET OPERATION
 /SET EXIT
 /GO PROCESS ARRAY DEF
 /GET NAME
 /GET COUNTER
 /LAST PAREN?
 /YES
 /NO - FORMAT ERROR

.EJECT

			/INPUT-OUTPUT STATEMENTS		
01603	R	751001	A	ASK	SKP:CLA:CMA /REMEMBER WHICH CALL. (-1) FOR ASK
01604	R	750000	A	TYPE	CLA /0 FOR TYPE
01605	R	040053	R		DAC ATSW
01606	R	140067	R	TASK	DZM DEBGSW /RE-ENABLE THE TRACE
01607	R	750000	A		CLA
					SORTJ ALIST,ATLIST /SPECIAL CHARACTER?
01610	R	102013	R GEN*	JMS	SORTB
01611	R	000446	R GEN*		ALIST-1
01612	R	777767	A GEN*		ATLIST-ALIST
01613	R	440053	R	ISZ	ATSW /TEST QUOTE SWITCH
01614	R	601636	R	JMP	TYPE2
				PUSHJ	GETARG /DO ASK; SETUP PT1
				JMS	XPUSHJ
01615	R	101100	R GEN*		GETARG
01616	R	002135	R GEN*		272 /TYPE COLON
01617	R	760272	A	LAW	
				PRINTC	
01620	R	103362	R GEN*	JMS	XOUTL
01621	R	200475	R	LAC C375	
				PRINTC	
01622	R	103362	R GEN*	JMS	XOUTL
				PUSHA	/SAVE IN-LINE CHARACTER
01623	R	101055	R GEN*	JMS	XPUSHA
01624	R	440025	R	ISZ	INSUB /INDICATE 'READC'
01625	R	200501	R	LAC	C215
01626	R	040020	R	DAC	ENDCR
01627	R	200167	R	LAC	C1 /POINT PAST CHAR
01630	R	105042	R	JMS	FLINTP /READ DATA AND SAVE
01631	R	200501	R	LAC	C215
01632	R	040020	R	DAC	ENDCR
				POPA	/RE-TEST LAST TERMINATOR
01633	R	220013	A GEN*	LAC*	PDLXR
01634	R	040061	R	DAC	CHAR
01635	R	601603	R	JMP	ASK /CONTINUE PROCESSING
				////	
01636	R			TYPE2	
01636	R	101100	R GEN*	PUSHJ	EVAL /DO TYPE
01637	R	002272	R GEN*	JMS	XPUSHJ
01640	R	104672	R	JMS	EVAL /PRINT
				SORTC	FLOUTP
01641	R	101321	R GEN*	JMS	GLIST
01642	R	000453	R GEN*		XSORTC
01643	R	601604	R	JMP	GLIST-1
				ERROR	TYPE
				.DEC	4
01644	R	603471	R GEN*	JMP	ERR2-4
				////	
01645	R	440067	R	TQUOT	ISZ DEBGSW /DISABLE TRACE
				GETC	UTRA /TYPE LITERALS
01646	R	102752	R GEN*	JMS	UTRA
01647	R	540450	R	SAD	C242 /"
01650	R	601657	R	JMP	TASK4
01651	R	540501	R	SAD	C215 /CR
01652	R	601210	R	JMP	PC1
				PRINTC	


```

01653 R 103362 R GEN* JMS XOUTL
01654 R 601646 R JMP TQUOT+1

01655 R 200501 R TCRLF LAC CCR /SLASH=CR,LF.
PRINTC
JMS XOUTL
01656 R 103362 R GEN* TASK4 GETC /MOVE TO NEXT CHARACTER
01657 R 102752 R GEN* JMS UTRA
01660 R 601606 R JMP TASK

01661 R 200475 R TCRLF2 LAC C375
PRINTC
JMS XOUTL
01662 R 103362 R GEN* LAC C215
01663 R 200501 R JMS IMAGEW
01664 R 103273 R LAC C200
01665 R 200154 R JMS IMAGEW
01666 R 103273 R JMP TASK4
01667 R 601657 R

/IF DEBGSW=0 : ENABLE FLIP-FLOP "DMPSW"
/ #0: DISABLE AND RETURN ALL"?" ' S.
/IF DMPSW = 0: TRACE ON, IF ENABLED
/ #0: TRACE OFF
/IF BOTH = 0 : PRINT TRACE.

01670 R TINTR GETC /PASS PERCENT SIGN
01670 R 102752 R GEN* JMS UTRA
GETLN /READ FORMAT CONTROL: "x7.03"
01671 R 100664 R GEN* JMS XGETLN
01672 R 200062 R LAC LINENO
01673 R 500151 R AND C77
01674 R 040055 R DAC DECP
01675 R 200062 R LAC LINENO
01676 R 740020 A RAR
01677 R 742020 A RTR
01700 R 742020 A RTR
01701 R 742020 A RTR
01702 R 500151 R AND C77
01703 R 040040 R DAC FISW
01704 R 601606 R JMP TASK
.EJECT

```

```

////
/SEARCH ROUTINES
01705 R
01705 R 100664 R GEN*
01706 R 102675 R GEN*
01707 R 603466 R GEN*
01710 R 101146 R
01711 R 200004 R
01712 R 065544 R
01713 R 140057 R
01714 R 200062 R
01715 R 741200 A

01716 R 603466 R GEN*
01717 R 060010 A
01720 R 225544 R
01721 R 040071 R
01722 R 200007 R
01723 R 040017 R
01724 R 103307 R SCONT
01725 R 741000 A
01726 R 750000 A SCONTX
01727 R 040075 R
01730 R 440067 R
01731 R 200075 R SCHAR
01732 R 040502 R

01733 R 102752 R GEN*
01734 R 040061 R
01735 R 060017 R
01736 R 440017 R
01737 R 103273 R

01740 R 102013 R GEN*
01741 R 000500 R GEN*
01742 R 000007 A GEN*
01743 R 601731 R

////
SBAR
01744 R 760300 A
01745 R 103273 R
01746 R 200007 R
01747 R 040017 R
01750 R 140502 R SFOUND
01751 R 103307 R
01752 R 040061 R

01753 R 102013 R GEN*
01754 R 000473 R GEN*
01755 R 000007 A GEN*
01756 R 200061 R
01757 R 060017 R

MODIFY GETLN /READ LINE NO.
JMS XGETLN
FINDLN /LOOK IT UP NOW.
JMS XFIND
ERROR 7 /NOT THERE = BAD COMMAND UNLESS ZERO.
.DEC
JMP ERR2-7
JMS MOVCOM /DISPLACE COMMON
LAC BUFR /SET POINTERS
DAC* (AXIN /FOR INPUT
DZM XCTIN
LAC LINENO /COPY THE SAME LINE NUMBER.
SNA /CHECK FOR ALL
ERROR 7
.DEC
JMP ERR2-7
DAC* AXIN /(X-MEM)
LAC* (AXIN /SAVE START OF NEW LINE
DAC PACKST
LAC COMBUF
DAC MODBUF /SET MODIFY COMMAND BUFFER
JMS IMAGER /GET SEARCH CHAR
SKP
CLA /CLEAR SEARCH CHAR
DAC SACH /SAVE SEARCH CHARACTER
ISZ DEBGSW /NO BREAKS.
LAC SACH
DAC LIST3+1 /PUT IN "SORTJ" LIST
GETC
JMS UTRA
DAC CHAR /SAVE FOR SORTJ
DAC* MODBUF
ISZ MODBUF
JMS IMAGEW
SORTJ LIST3,LISTGO /LOOK FOR MATCH
JMS SORTB
LIST3-1
LISTGO-LIST3
JMP SCHAR

LAW 300 /ECHO * FOR +U
JMS IMAGEW
LAC COMBUF /RESET TO BEGINNING OF BUFFER
DAC MODBUF
DZM LIST3+1
JMS IMAGER /READ FROM KEYBOARD
DAC CHAR
SORTJ LIST6,SRNLST /CHECK FOR ACTION CHANGE
JMS SORTB
LIST6-1
SRNLST-LIST6
LAC CHAR
DAC* MODBUF /PACK CHAR

```

01760	R	440017	R		ISZ	MODBUF	
01761	R	601750	R		JMP	SFOUND	/GO GET MORE
01762	R	200501	R	SRETN	LAC	C215	
01763	R	060017	R		DAC*	MODBUF	/SAVE CR
01764	R	200007	R		LAC	COMBUF	
01765	R	040017	R		DAC	MODBUF	
01766	R	220017	R		LAC*	MODBUF	/FINISH LINE AND SAVE IT
01767	R	040061	R		DAC	CHAR	
01770	R	103562	R	GEN*	PACKC		
01771	R	440017	R		JMS	PACBUF	
01772	R	540501	R		ISZ	MODBUF	
01773	R	741000	A		SAD	C215	/END LINE?
01774	R	601766	R		SKP		/YES
01775	R	200477	R		JMP	.-6	
01776	R	103273	R		LAC C212		
					JMS IMAGEW		
01777	R	101100	R	GEN*	PUSHJ	DELETE	/REPLACE WITH NEW LINE
02000	R	002550	R	GEN*	JMS	XPUSHJ	
						DELETE	
02001	R	103051	R	GEN*	ENDLN		
02002	R	600512	R		JMS	XENDLN	
02003	R	200017	R	SCRUB	JMP	START	/RESET POINTERS
02004	R	540007	R		LAC	MODBUF	
02005	R	601750	R		SAD	COMBUF	/AT BEGINNING?
02006	R	340614	R		JMP	SFOUND	/YES
02007	R	040017	R		TAD	M1	/NO
02010	R	760334	A		DAC	MODBUF	
02011	R	103273	R		LAW	334	/ECHO BACK SLASH
02012	R	601750	R		JMS	IMAGEW	
					JMP	SFOUND	/GO PROCESS NEXT
					.EJECT		

```

02013 R 000000 A
02014 R 741200 A
02015 R 200061 R
02016 R 040074 R
02017 R 222013 R
02020 R 442013 R
02021 R 065546 R
02022 R 220012 A
02023 R 741100 A
02024 R 602035 R
02025 R 540074 R
02026 R 741000 A
02027 R 602022 R
02030 R 225546 R
02031 R 362013 R
02032 R 040074 R
02033 R 750000 A
02034 R 620074 R
02035 R 442013 R
02036 R 754000 A
02037 R 622013 R

02040 R 000000 A
02041 R 222040 R
02042 R 065546 R
02043 R 220012 A
02044 R 040074 R
02045 R 200001 R
02046 R 040134 R
02047 R 140135 R
02050 R 220012 A
02051 R 560134 R
02052 R 602060 R
02053 R 042013 R
02054 R 505551 R
02055 R 560001 R
02056 R 602070 R
02057 R 602077 R
02060 R 500151 R
02061 R 741200 A
02062 R 620012 A
02063 R 440134 R
02064 R 440135 R
02065 R 602050 R
02066 R 220012 A
02067 R 741000 A
02070 R 202013 R
02071 R 500151 R
02072 R 740200 A
02073 R 602066 R
02074 R 200135 R
02075 R 741200 A
02076 R 620012 A

```

```

SORTB 0 /SORT AND BRANCH ROUTINE. - "SORTJ"
      SNA
      DAC LAC CHAR /ASSUME CHAR IF AC=0
      T2 /SAVE SORT ITEM
      LAC* SORTB /FIRST ARG IS LIST LESS ONE
      ISZ SORTB /2AND IS INTRA-LIST LENGTH
      DAC* (XRT2
      LAC* XRT2
      SPA /**LIST ENDED BY NEGATIVE NUMBER**
      JMP SEX
      SNO T2 /FIND ADDRESS
      SKP
      JMP .-5
      LAC* (XRT2 /MATCH FOUND.
      TAD* SORTB
      DAC T2
      CLA
      JMP* T2
SEX ISZ SORTB /MATCH NOT FOUND.
      CLA:CLL
      JMP* SORTB /RETURN TO CALLING SEQUENCE.

/SORT AND BRANCH ON COMMAND
XSORTX 0 /"SORTJX"
      LAC* XSORTX /GET TABLE START
      DAC* (XRT2
      LAC* XRT2 /SET SIZE
      DAC T2
      ANYMAT LAC BUFSTX /GET COMMAND POINTER
      DAC MODBF1
      DZM MODBF2
      MORMAT LAC* XRT2 /GET COMMAND TABLE ENTRY
      SAD* MODBF1
      JMP ENDMAT /FULL WORD MATCH
      DAC SORTB /SAVE FOR END TEST
      AND (770000
      SAD* BUFSTX
      JMP YESMAT /ONE LETTER MATCH
      JMP NOTMAT
      ENDMAT AND C77 /IS IT END OF COMMAND
      SNA
      JMP* XRT2 /DISPATCH
      ISZ MODBF1
      ISZ MODBF2 /DISABLE ONE LETTER MATCH
      JMP MORMAT /TEST REST OF COMMAND
      LAC* XRT2
      YESMAT SKP
      LAC SORTB /TEST FOR COMMAND END
      AND C77
      SZA
      JMP YESMAT-2
      LAC MODBF2 /ONE LETTER OK
      SNA
      JMP* XRT2 /DISPATCH

```

02077 R 220012 A
02100 R 505552 R
02101 R 545553 R
02102 R 741000 A
02103 R 602077 R
02104 R 440074 R
02105 R 602045 R
02106 R 442040 R
02107 R 622040 R

NOTMAT

LAC*
AND
SAD
SKP
JMP
ISZ
JMP
ISZ
JMP*

XRT2
(700000
(600000
.-4
T2
ANYMAT
XSORTX
XSORTX

/SKIP REST OF COMMAND
/ENDS WITH JMP
/ANY MORE IN TABLE
/YES
/NO - ERROR RETURN

.EJECT

```

                                /FETCH VARIABLE FROM INPUT
02110 R 000000 A FETVAR 0
02111 R 040021 R DAC GETVCT /-3 OR -6 OR -9
02112 R 777777 A LAW -1 / (BUFFER-1
02113 R 340001 R TAD BUFSTX
02114 R 065544 R DAC* (AXIN
02115 R 140057 R DZM XCTIN /BEGIN PACK OF VARIABLE NAME
02116 R GETVAP PACKC /PACK CHAR
02116 R 103562 R GEN* JMS PACBUF
                                GETC
02117 R 102752 R GEN* JMS UTRA
                                SORTC
                                TERMS /CHECK FOR TERMINATORS
                                JMS
02120 R 101321 R GEN* XSORTC
02121 R 000415 R GEN* TERMS-1
                                JMP GETVAX
02122 R 602130 R ISZ GETVCT /HAVE THREE CHARS BEEN USED
02123 R 440021 R JMP GETVAP /NO-GO PACK THIS ONE
02124 R 602116 R LAW -1 /IGNORE REST
02125 R 777777 A DAC GETVCT
02126 R 040021 R JMP GETVAP+1
02127 R 602117 R GETVAX ISZ GETVCT
02130 R 440021 R SKP!CLA
02131 R 751000 A JMP* FETVAR
02132 R 622110 R JMS PCK1 /USE NULLS
02133 R 103623 R JMP GETVAX
02134 R 602130 R
                                /FIND OR ENTER A VARIABLE IN THE LIST.
02135 R GETARG TESTC /FIRST LETTER OF ARG
02135 R 101301 R GEN* JMS XTESTC
02136 R 740000 A NOP
02137 R 740000 A NOP /FUNCTION OR NUMBER IS NOT AN ARG.
                                ERROR 15 /BAD ARGUMENT IN 'FOR', 'SET', OR 'ASK'
                                .DEC
02140 R 603456 R GEN* JMP ERR2-15
02141 R 777775 A GETVAR LAW -3
02142 R 102110 R JMS FETVAR
                                TSTLPR /LOOK FOR SUBSCRIPT VIA SORTCN
02143 R 102536 R GEN* JMS LPRTST
02144 R 602161 R JMP GS1 /NOT SUBSCRIBED BY L-PAR.
02145 R 200052 R LAC LASTOP /SAVE LAST OPERATION
                                PUSH
02146 R 101055 R GEN* JMS XPUSHA
02147 R 220001 R LAC* BUFSTX /SAVE NAME
                                PUSH
02150 R 101055 R GEN* JMS XPUSHA
                                PUSHJ EVAL-1 /MOVE PAST L-PAR AND EVALUATE SUBSCRIPT
02151 R 101100 R GEN* JMS XPUSHJ
02152 R 002271 R GEN* EVAL-1
                                POPA
02153 R 220013 A GEN* LAC* PDLXR
02154 R 060001 R DAC* BUFSTX /RESTORE NAME
                                GETC
02155 R 102752 R GEN* JMS UTRA
                                POPA
02156 R 220013 A GEN* LAC* PDLXR
02157 R 040052 R DAC LASTOP /RECALL LAST OPERATION

```

02160	R	105343	R	GS1A	JMS	FIX	
02161	R	042233	R	GS1	DAC	SUBS	/SAVE SUBSCRIPT
02162	R	200005	R		LAC	FRSTCV	/SEARCH FOR VARIABLE
02163	R	040072	R	GS3	DAC	PT1	
02164	R	540006	R		SAD	LASTV	/TEST FOR END OF LIST
02165	R	602174	R		JMP	GS2	/END SEARCH
02166	R	220072	R		LAC*	PT1	/GET TABLE ENTRY
02167	R	560001	R		SAD*	BUFSTX	
02170	R	602221	R		JMP	GFND1	/FOUND XX
02171	R	200072	R	GS4	LAC	PT1	/TRY NEXT ONE
02172	R	340156	R		TAD	GINC	
02173	R	602163	R		JMP	GS3	
02174	R	200006	R	GS2	LAC	LASTV	/ADD THE VARIABLE
02175	R	340143	R		TAD	P13	/TEST RAN LIMITS
02176	R	744000	A		CLL		
02177	R	103752	R		JMS	TWOS	
02200	R	365542	R		TAD*	(PDLXR	
02201	R	750400	A		SNL!CLA		
					ERROR	16	
				GEN*	.DEC		
02202	R	603455	R	GEN*	JMP	ERR2-16	
02203	R	200006	R		LAC	LASTV	
02204	R	340156	R		TAD	GINC	
02205	R	040006	R		DAC	LASTV	
02206	R	220001	R		LAC*	BUFSTX	/SAVE NAME
02207	R	060072	R		DAC*	PT1	
02210	R	440072	R		ISZ	PT1	/SAVE SUBSCRIPT
02211	R	202233	R		LAC	SUBS	
02212	R	060072	R		DAC*	PT1	
02213	R	440072	R		ISZ	PT1	/SET PT1
02214	R	105140	R		JMS	FINT	
02215	R	500171	R		FGET	FLTZER	
02216	R	620072	R		FPUT* PT1		
02217	R	000000	A		FXIT		
02220	R	602265	R	GEN*	POPJ		/EXIT
					JMP	XPOPJ	
				////			
02221	R	200072	R	GFND1	LAC	PT1	/FOUND SAME
02222	R	065545	R		DAC*	(XRT	/TEST SUBSCRIPTS
02223	R	220011	A		LAC*	XRT	
02224	R	103752	R		JMS	TWOS	
02225	R	342233	R		TAD	SUBS	
02226	R	750200	A		SZA!CLA		
02227	R	602171	R		JMP	GS4	/WRONG SUBSCRIPT
02230	R	440072	R		ISZ	PT1	/SET POINTER TO DATA
02231	R	440072	R		ISZ	PT1	
					POPJ		
02232	R	602265	R	GEN*	JMP	XPOPJ	
					.EJECT		

```

////
////IGNORE LEADING SPACES - "SPNOR"
SURS=.
XSPNOR
02233 R 002233 R
02234 R 000000 A
02235 R 200061 R
02236 R 540416 R
02237 R 741000 A
02237 R 622233 R
02240 R 102752 R GEN*
02241 R 602235 R
02242 R 000000 A XTESTN
02243 R 777522 A MPER
02244 R 340061 R
02245 R 750200 A
02246 R 442242 R
02247 R 777520 A
02250 R 340061 R
02251 R 040051 R
02252 R 751100 A
02253 R 602260 R
02254 R 777507 A
02255 R 340061 R
02256 R 751300 A
02257 R 442242 R
02260 R 200051 R
02261 R 622242 R
02262 R 200157 R
02263 R 340167 R
02264 R 040063 R
02265 R 220013 A
02266 R 040074 R
02267 R 750000 A
02270 R 620074 R

XSPNOR
LAC
SAD
SKP
JMP*
GETC
JMS
JMP
0
LAW
TAD
SZA:CLA
ISZ
LAW
TAD
DAC
SPA:CLA
JMP
LAW
TAD
SPA:SNA:CLA
ISZ
LAC
JMP*
LAC
TAD
DAC
LAC*
DAC
CLA
JMP*
.EJECT

CHAR
C240
XSPNOR
UTRA
XSPNOR+2
/RETURNS: ; OTHER; NUMBER - "TESTN"
-256
CHAR
XTESTN
-260
CHAR
SORTCN
/SAVE VALUE
ZTESTN
-271
CHAR
/IF A NUMBER
SORTCN
XTESTN
CFRS
C1
PC
PDLXR
T2
/RECURSIVE EXIT - "POPJ"
/((PC) => 0
/TO PRETEND END OF TEXT

```



```

/EVALUATE AN EXPRESSION WHICH
/TERMINATES WITH AN R-PAR,; OR C.R. AND
/LEAVE THE RESULT IN FLAC AND IN FLARG.
02271 R 102752 R GEN* GETC UTRA
02272 R 140052 R EVAL DZM LASTOP /EVALUATION CONTROLLER
/TEST CHARACTER AND IGNORE SPACES
02273 R 101301 R GEN* JMS XTESTC
02274 R 602306 R JMP ETERM1 /TERMINATION
02275 R 602417 R JMP ENUM /NUMBER
02276 R 602431 R JMP EFUN /FUNCTION
PUSHJ GETVAR /FIND OR CREATE VARIABLE; ALSO SET PT1.
JMS XPUSHJ
02277 R 101100 R GEN* JMS GETVAR
02300 R 002141 R GEN* OPNEXT TESTC /PT1=>ARG
02301 R 101301 R GEN* JMS XTESTC
02302 R 602332 R JMP ETERM1 /T
02303 R 740000 A NOP /N-ERROR IN FORMAT
02304 R 740000 A NOP /F
ERROR 17 /L - MISSING OPERATOR
.DEC
02305 R 603454 R GEN* JMP ERR2-17
02306 R 200061 R ETERM1 LAC CHAR
02307 R 540435 R SAD C275
ERROR 17
.DEC
02310 R 603454 R GEN* JMP ERR2-17
PUSHF FLTZER /INITIALIZE RESULT TO ZERO.
JMS PD2
02311 R 101113 R GEN* FLTZER
02312 R 000171 R GEN* FLARG
POPF FLARG
JMS PD3
02313 R 101132 R GEN* FLARG
02314 R 000076 R GEN* FLARGP /SET PT1.
02315 R 200160 R LAC
02316 R 040072 R DAC PT1
02317 R 201367 R LAC M2 /TEST FOR UNARY OPERATIONS
02320 R 340051 R TAD SORTCN
02321 R 741200 A SNA
02322 R 602335 R JMP ETERM /CREATE DUMMY FOR UNARY MINUS
02323 R 340167 R TAD C1
02324 R 751200 A SNA:CLA
02325 R 602411 R JMP ARGNXT /IGNORE UNARY PLUS
02326 R 340051 R TAD SORTCN /TEST FOR NULL PARENS.
02327 R 342537 R TAD M11
02330 R 751100 A SPA:CLA
02331 R 602475 R JMP ELPAR /MIGHT BE AN L-PAR.
02332 R ETERM1 TSTLPR
02332 R 102536 R GEN* JMS LPRST
02333 R 741000 A SKP
ERROR 18 /OPERATOR MISSING BEFORE PAREN
.DEC
02334 R 603453 R GEN* JMP ERR2-18
02335 R 200051 R ETERM LAC SORTCN /SET FROM "TESTC"- "SORTC"
02336 R 040065 R DAC THISOP

```

02337	R	342537	R		TAD	M11	
02340	R	750100	A		SMA:CLA		/END?
02341	R	040065	R		DAC	THISOP	/"THISOP" EQUIV. TO END OF EXP.
02342	R	200065	R	ETERM2	LAC	THISOP	/COMPARE PRIORITIES
02343	R	103752	R		JMS	TWOS	
02344	R	340052	R		TAD	LASTOP	
02345	R	751100	A		SPA:CLA		
02346	R	602376	R		JMP	EPAR	/CONTINUE
02347	R	340052	R		TAD	LASTOP	/FIND OPERATION FROM TABLE
02350	R	340224	R		TAD	OPTABL	
02351	R	042362	R		DAC	FLOP	
02352	R	222362	R		LAC*	FLOP	
02353	R	042362	R		DAC	FLOP	
02354	R	200052	R		LAC	LASTOP	
02355	R	751200	A		SNA:CLA		/TEST FOR END OF DATA INTO FLOATING AC.
02356	R	602361	R		JMP .+3		
					POPF	XX	/GET LAST DATA
02357	R	101132	R GEN*		JMS	PD3	
02360	R	000000	A GEN*			XX	
				FLAC1=-.1			/.AA
02361	R	105140	R		JMS	FINT	
02362	R	000000	A	FLOP	00		/(FLOPR I PT1)+-*/
02363	R	600076	R		FPUT	FLARG	/SAVE RESULT
02364	R	000000	A		FXIT		
02365	R	200160	R		LAC	FLARGP	
02366	R	040072	R		DAC	PT1	
02367	R	200065	R		LAC	THISOP	
02370	R	340052	R		TAD	LASTOP	/=0?
02371	R	751200	A		SNA:CLA		
					POPJ		/EXIT "EVAL"
02372	R	602265	R GEN*		JMP	XPOPJ	
					POPA		/GET PRIOR OP
02373	R	220013	A GEN*		LAC*	PDLXR	
02374	R	040052	R		DAC	LASTOP	
02375	R	602342	R		JMP	ETERM2	/COMPARE THIS OP
				/////			
02376	R			EPAR	TSTLPR		/TEST FOR SUB-EXPRESSION
02376	R	102536	R GEN*		JMS	LPRTST	
02377	R	741000	A		SKP		
02400	R	602477	R		JMP	EPAR2	/GO EVALUATE EXPRESSION
02401	R	200052	R		LAC	LASTOP	/CONTINUE READING THE EXPRESSION
					PUSHA		/SAVE "LASTOP".
02402	R	101055	R GEN*		JMS	XPUSHA	
02403	R	200072	R		LAC	PT1	
02404	R	042406	R		DAC	+.2	
					PUSHF	XX	/SAVE LAST ARGUMENT
02405	R	101113	R GEN*		JMS	PD2	
02406	R	000000	A GEN*			XX	
02407	R	200065	R		LAC	THISOP	/MORE TO COME
02410	R	040052	R		DAC	LASTOP	
02411	R			ARGNXT	GETC		/READ 1ST CHAR OF AN ARG.
02411	R	102752	R GEN*		JMS	UTRA	
					TESTC		/DO SPECIAL CHECK
02412	R	101301	R GEN*		JMS	XTESTC	
02413	R	602475	R		JMP	ELPAR	/COULD BE LEFT PAREN

```

02414 R 602417 R JMP ENUM /N
02415 R 602431 R JMP EFUN /F
02416 R 602277 R JMP OPNEXT-2 /L

02417 R // // ENUM PUSHF XX /TO PROCESS A NUMBER,SAVE AC
02417 R 101113 R GEN* JMS PD2
02420 R 000000 A GEN* FLAC2=-1 XX
02420 R 002420 R LAC FLARGP /.AA
02421 R 200160 R DAC /SET POINTER AS FOR VARIABLE.
02422 R 040072 R PT1
02423 R 140025 R DZM INSUB /POINT TO 'GETC' AND USE CHAR
02424 R 750000 A CLA /READ NEXT
02425 R 105042 R JMS FLINTP /READ TEXT NUMBER => (PT1)
02426 R 101132 R GEN* POPF XX /RESTORE THE AC
02427 R 000000 A GEN* JMS PD3
02427 R 002427 R FLAC3=-1 XX
02430 R 602301 R JMP OPNEXT /.AA
// // /CONTINUE
02431 R EFUN GETC
02431 R 102752 R GEN* JMS UTRA
02432 R 777775 A -3
02433 R 102110 R JMS /GET FUNCTION NAME
02434 R 200051 R LAC SORTCN /SAVE 'SORTCN','LASTOP',AND 'EFOP'
PUSHA
02435 R 101055 R GEN* JMS XPUSHA
02436 R 200052 R LAC LASTOP
PUSHA
02437 R 101055 R GEN* JMS XPUSHA
02440 R 220001 R LAC* BUFSTX /SAVE FUNCTION NAME
PUSHA
02441 R 101055 R GEN* JMS XPUSHA
TSTLPR
02442 R 102536 R GEN* JMS LPRTST
ERROR 19 /MUST BE FOLLOWED BY PARENS TO SET ARGUMENT
GEN*
02443 R 603452 R GEN* .DEC
JMP ERR2-19
PUSHJ EVAL-1 /YES
JMS XPUSHJ
EVAL-1
02446 R 220013 A GEN* POPA
LAC* PDLXR
02447 R 040137 R DAC FUNAME /SAVE FUNCTION NAME
02450 R 205554 R LAC (FNTABF
02451 R 102455 R JMS FUNCHK /IS IT INTERNAL FUNCTION
02452 R 205537 E .NEWF
02453 R 102455 R JMS FUNCHK /IS IT EXTERNAL FUNCTION
ERROR /ILLEGAL FUNCTION NAME
.DEC
02454 R 603451 R GEN* JMP ERR2-20
02455 R 000000 A GEN* FUNCHK 0 /DISPATCH ON FUNCTION NAME
02456 R 040141 R DAC FUNPTR /FUNCTION TABLE START
02457 R 220141 R LAC* FUNPTR
02460 R 740100 A SMA
02461 R 622455 R JMP* FUNCHK

```

02462	R	040140	R		DAC	FUNCTR	/FUNCTION TABLE COUNT
02463	R	440141	R	FUNLOP	ISZ	FUNPTR	/POINTS TO NEXT NAME
02464	R	200137	R		LAC	FUNAME	
02465	R	560141	R		SAD*	FUNPTR	/RIGHT FUNCTION?
02466	R	602473	R		JMP	FUNFND	/YES - GO DISPATCH
02467	R	440141	R		ISZ	FUNPTR	/NO - TRY NEXT
02470	R	440140	R		ISZ	FUNCTR	/ANY MORE IN TABLE?
02471	R	602463	R		JMP	FUNLOP	/YES
02472	R	622455	R		JMP*	FUNCHK	/NO - RETURN
02473	R	440141	R	FUNFND	ISZ	FUNPTR	/TO FUNCTION ADDRESS
02474	R	620141	R		JMP*	FUNPTR	/DISPATCH
				/////			
02475	R			ELPAR	TSTLPR		
02475	R	102536	R	GEN*	JMS	LPRST	
				GEN*	ERROR	21	/DOUBLE OPERATORS
				GEN*	.DEC		
02476	R	603450	R	GEN*	JMP	ERR2-21	
02477	R	200051	R	EPAR2	LAC	SORTCN	/LEFT PARENS FOUND.
					PUSHA		
02500	R	101055	R	GEN*	JMS	XPUSHA	
02501	R	200052	R		LAC	LASTOP	/SAVE DATA
					PUSHA		
02502	R	101055	R	GEN*	JMS	XPUSHA	
					PUSHJ	EVAL-1	/EVALUATE THE EXPRESSION
02503	R	101100	R	GEN*	JMS	XPUSHJ	
02504	R	002271	R	GEN*		EVAL-1	
					RETURN		
02505	R	602517	R	GEN*	JMP	EFUN3	
					.EJECT		

```

////////
/SOME MINOR FUNCTIONS
02506 R 105343 R XINT JMS FIX /INTFGER PART
          RETURN
02507 R 602517 R GEN* JMP EFUN3
02510 R 125524 E XSGN JMS* .AO /TAKE SIGN*1 OF FLARG
02511 R 000167 R FLTONE
02512 R 200077 R LAC FLARG+1
02513 R 741000 A SKP
02514 R XARs GETSGN /TAKE ABSOLUTE VALUE OF FLAC
02514 R 225522 E GEN* LAC* .AB
02515 R 751100 A SPA:CLA /SKIP TO CONTINUE
02516 R 125534 E JMS* .BA /NEGATE THE FLOATING AC

/CONTINUATION OF FUNCTION CALLS
02517 R EFUN3 POPA /RESTORE LAST OPERATION
02517 R 220013 A GEN* LAC* PDLXR
02520 R 040052 R DAC LASTOP
02521 R 125536 E JMS* .CD /NORMALIZE FUNCTION RETURN
02522 R 125525 E JMS* .AP /SAVE FUNCTION VALUE
02523 R 000076 R FLARG
02524 R 200160 R LAC FLARGP /SET POINTER
02525 R 040072 R DAC PT1
          POPA /GET LAST PAREN CODE.
02526 R 220013 A GEN* LAC* PDLXR
02527 R 340146 R TAD P3
02530 R 103752 R JMS TWOS /CHECK FOR PAREN MATCH.
02531 R 340051 R TAD SORTCN /(STILL SET FROM THE LAST "EVAL")
02532 R 750200 A SZA:CLA /SKIP IF MATCH
          ERROR 22 /PAREN ERROR
          .DEC
02533 R 603447 R GEN* JMP ERR2-22
          GEN* GETC
02534 R 102752 R GEN* JMS UTRA /MOVE PAST R-PAR, AND RETURN TO OPNEX.
02535 R 602301 R JMP OPNEXT /FUNCTION RETURN IS OK
02536 R 000000 A LPRTST 0 /SKIP IF LEFT PAREN. - 'TSTLPR'
02537 R 777767 A M11 LAH -11
02540 R 340051 R TAD SORTCN
02541 R 750100 A SMA:CLA
02542 R 622536 R JMP* LPRTST
02543 R 777773 A LAH -5
02544 R 340051 R TAD SORTCN
02545 R 750300 A SMA:SZA:CLA
02546 R 442536 R ISZ LPRTST
02547 R 622536 R JMP* LPRTST
          .EJECT

```

```

/ THE DELETE A LINE ROUTINE
02550 R          DELETE FINDLN /SETS "THISLN" AND "LASTLN".
02550 R 102675 R GEN* JMS      XFINO
                                POPJ      /ALREADY GONE
02551 R 602265 R GEN* JMP      XPOPJ
02552 R 440067 R      ISZ      DEBGSW  /DISABLE TRACE
                                GETC      /MEASURE LENGTH
02553 R 102752 R GEN* JMS      UTRA
02554 R 540501 R      SAD      C215
02555 R 751001 A      SKP:CLA!CMA
02556 R 602553 R      JMP      .-3
02557 R 340014 R      TAD      AXOUT   /SAVE LAST ADDRESS
02560 R 740001 A      CMA
02561 R 340064 R      TAD      THISLN
02562 R 040054 R      DAC      CNTR   /LENGTH < 0
02563 R 220064 R      LAC*     THISLN /DISCONNECT
02564 R 060066 R      DAC*     LASTLN
02565 R 200157 R      LAC      CFRS   /START LIST AT TOP
02566 R 040074 R      DAC      T2     /EXAMINATION ADDRESS
                                DOK
02567 R 220074 R      LAC*     T2     /GET THE NEXT ADDR.
02570 R 741200 A      SNA
02571 R 602605 R      JMP      DONE  /TEST FOR END
02572 R 040073 R      DAC      T1     /YES-WRAP UP ALL.
02573 R 200064 R      LAC      THISLN /SAVE NEXT ADDRESS.
02574 R 744000 A      CLL
02575 R 103752 R      JMS      TWOS
02576 R 340073 R      TAD      T1
02577 R 751400 A      SZL:CLA
02600 R 340054 R      TAD      CNTR  /SKIP IF THISLN > X
02601 R 340073 R      TAD      T1    /CHANGE (X) TO ACCOUNT FOR
02602 R 060074 R      DAC*     T2    /GARBAGE COLLECTION.
02603 R 200073 R      LAC      T1
02604 R 602566 R      JMP      DOK   /GET NEXT
                                .EJECT

```

```

02605 R 750001 A
02606 R 340064 R
02607 R 065545 R
02610 R 200054 R
02611 R 740001 A
02612 R 340064 R
02613 R 065546 R
02614 R 200054 R
02615 R 340004 R
02616 R 040004 R
02617 R 225544 R
02620 R 740001 A
02621 R 365546 R
02622 R 040073 R
02623 R 225544 R
02624 R 340054 R
02625 R 065544 R
02626 R 220012 A
02627 R 060011 A
02630 R 440073 R
02631 R 602626 R
02632 R 602550 R

```

```

/////
/GARBAGE COLLECTION
DONE

```

```

CLA:CMA
TAD THISLN
DAC* (XRT
LAC CNTR
CMA
TAD THISLN
DAC* (XRT2
LAC CNTR
TAD BUFR
DAC BUFR
LAC* (AXIN
CMA
TAD* (XRT2
DAC T1
LAC* (AXIN
TAD CNTR
DAC* (AXIN
LAC* XRT2
DAC* XRT
ISZ T1
JMP .-3
JMP DELETE

```

```

/BACKUP L FOR XR
/SETUP END OF HOSE
/CORRECT END OF BUFFER POINTER.
/COMPUTE COUNT
/SIPHON LOWER PART.
/RESET 'LASTLN', 'THISLN', AND DATA FIELD.

```

/////

.EJECT

02633	R		ERASE	TESTC		/TEST THE SECOND WORD, IF ANY.
02633	R	101301	R GEN*	JMS	XTESTC	
02634	R	602672	R	JMP	ERVX	/ERASE VARIABLES
02635	R	602650	R	JMP	ERL	/LINES OR GROUPS
				ERROR	23	
			GEN*	.DEC		
02636	R	603446	R GEN*	JMP	ERR2-23	
02637	R	777767	A	LAW	-11	
02640	R	102110	R	JMS	FETVAR	
				SORTJX	ALLCM2	
02641	R	102040	R GEN*	JMS	XSORTX	
02642	R	000321	R GEN*		ALLCM2-1	
				ERROR	23	/BAD ARG FOR ERASE.
			GEN*	.DEC		
02643	R	603446	R GEN*	JMP	ERR2-23	
02644	R	200002	R	LAC	ENDT	/ERASE ALL TEXT **
02645	R	040004	R	DAC	BUFR	
02646	R	160157	R	DZM*	CFRS	/POINTERS MAY BE DIFFERENT NOW.
02647	R	600512	R	JMP	START	
			/////			
02650	R		ERL	GETLN		/ERASE LINES.
02650	R	100664	R GEN*	JMS	XGETLN	
02651	R	200062	R	LAC	LINENO	
02652	R	501347	R	AND	P7600	
02653	R	741200	A	SNA		
			GEN*	ERROR	7	
			GEN*	.DEC		
02654	R	603466	R GEN*	JMP	ERR2-7	
02655	R	200004	R	LAC	BUFR	/PROTECT REST OF TEXT.
02656	R	065544	R	DAC*	(AXIN	
02657	R		ERG	PUSHJ	DELETE	/EXTRACT ONE LINE
02657	R	101100	R GEN*	JMS	XPUSHJ	
02660	R	002550	R GEN*		DELETE	
02661	R	440064	R	ISZ	THISLN	
02662	R	200060	R	LAC	NAGSW	
02663	R	750100	A	SMA:CLA		
02664	R	360064	R	TAD*	THISLN	/(X-MEM)
				TSTGRP		/SKIP IF G(AC) = G(LINENO)
02665	R	101343	R GEN*	JMS	GRPTST	
02666	R	600512	R	JMP	START	
02667	R	220064	R	LAC*	THISLN	/(X-MEM)
02670	R	040062	R	DAC	LINENO	
02671	R	602657	R	JMP	ERG	
			/////			
02672	R	200003	R	ERVX	LAC	/INIT VARIABLES MAY BE IN THE TEXT
02673	R	040006	R		DAC	
				POPJ	STARTV	
02674	R	602265	R GEN*	JMP	LASTV	
				JMP	XPOPJ	
				.EJECT		


```

/ROUTINE CALLED VIA "FINDLN":
/SEARCH FOR A GIVEN LINE I.D. =["LINENO" ]
/1ST RETURN IF NOT FOUND,
/2ND IF FOUND.
/"THISLN" = FOUND LINE OR NEXT LARGER.
/"LASTLN" = LESSER AND/OR LAST.
/"TEXTP" IS SET
XFIND 0
LAC CFRS /INITIALIZE POINTERS TO FIRST LINE
DAC LASTLN
DAC THISLN /SAVE THIS ONE
DAC* (XRT2
LAC LINENO
SPA /MAX 99.99
ERROR 24
.DEC
JMP ERR2-24
JMS TWOS
TAD* XRT2 /LINENO=0 WILL ALSO BE FOUND
SNA
JMP FEND2 /FOUND IT.
SMA:CLA
JMP FEND3 /PAST IT.
TAD THISLN /MOVE POINTERS
DAC LASTLN
LAC* THISLN
SZA
JMP FINDN
SKP
ISZ XFIND /2ND EXIT = FOUND
LAC THISLN
TAD P2
DAC AXOUT /SET "TEXTP".
LAW -1
DAC XCTX
.IFUND MULTI
LAC DEBGSW
TAD DMPSW
SZA:CLA
JMP* XFIND
LAC SAVEOT
SAD C215
SKP
SAD C375
JMP .+3
LAC C375
PRINTC
JMS XOUTL
LAC IMRUFF+2
SAD C215
JMP .+3
LAC C215
JMS IMAGEW
LAC C212
JMS IMAGEW

```

02675	R	000000	A			
02676	R	200157	R			
02677	R	040066	R			
02700	R	040064	R	FINDN		
02701	R	065546	R			
02702	R	200062	R			
02703	R	741100	A			
02704	R	603445	R	GEN*		
02705	R	103752	R	GEN*		
02706	R	360012	A			
02707	R	741200	A			
02710	R	602721	R			
02711	R	750100	A			
02712	R	602722	R			
02713	R	340064	R			
02714	R	040066	R			
02715	R	220064	R			
02716	R	740200	A			
02717	R	602700	R			
02720	R	741000	A			
02721	R	442675	R	FEND2		
02722	R	200064	R	FEND3		
02723	R	340147	R			
02724	R	040014	R			
02725	R	777777	A			
02726	R	040015	R			
02727	R	200067	R			
02730	R	340070	R			
02731	R	750200	A			
02732	R	622675	R			
02733	R	200022	R			
02734	R	540501	R			
02735	R	741000	A			
02736	R	540475	R			
02737	R	602742	R			
02740	R	200475	R			
02741	R	103362	R	GEN*		
02742	R	203125	R			
02743	R	540501	R			
02744	R	602747	R			
02745	R	200501	R			
02746	R	103273	R			
02747	R	200477	R			
02750	R	103273	R			

02751	R	622675	R		.ENDC		
02752	R	000000	A	UTRA	JMP*	XFIND	
02753	R	103026	R		0		/UNPACK CHARACTER. - "GETC"
02754	R	751100	A	UTE	JMS	GET1	
02755	R	340150	R		SPA:CLA		/NORM & EXTEND
02756	R	340165	R		TAD	C100	/300-337 & 340-376
02757	R	340061	R		TAD	M137	/240-276 & 200-236
02760	R	741200	A		TAD	CHAR	
02761	R	603014	R		SNA		
02762	R	340166	R		JMP	UTX	/"?" FOUND...
02763	R	040061	R	UTO	TAD	P337	
					DAC	CHAR	
02764	R	200067	R		.IFUND	MULTI	
02765	R	340070	R		LAC	DEBGSW	
02766	R	750200	A		TAD	DMPSW	
02767	R	603007	R		SZA:CLA		/PRINT ONLY IF BOTH ARE ZERO.
02770	R	200022	R		JMP	UTRAX	
02771	R	540501	R		LAC	SAVEOT	
02772	R	741000	A		SAD	C215	
02773	R	540475	R		SKP		
02774	R	602777	R		SAD	C375	
02775	R	200475	R		JMP	..+3	
					LAC	C375	
					PRINTC		
02776	R	103362	R	GEN*	JMS	XOUTL	
02777	R	200061	R		LAC CHAR		
03000	R	103273	R		JMS	IMAGEW	
03001	R	200061	R		LAC CHAR		/FIX FOR TRACE FEATURE...WAD...
03002	R	540501	R		SAD C215		/IF CR OUTPUT LF
03003	R	741000	A		SKP		
03004	R	603007	R		JMP UTRAX	/NOT A CR.	NO LF
03005	R	200477	R		LAC C212		/OUTPUT LF
03006	R	103273	R		JMS IMAGEW		/...END OF FIX...JUNE 69
03007	R	200061	R	UTRAX	LAC	CHAR	
					.ENDC		
03010	R	622752	R		JMP*	UTRA	
				/////			
03011	R	103026	R	EXTR	JMS	GET1	
03012	R	740001	A		CMA		
03013	R	602754	R		JMP	UTE	
				///			
03014	R	200067	R	UTX	LAC	DEBGSW	/TEST FOR TRACE-ENABLED
03015	R	750200	A		SZA:CLA		
03016	R	603024	R		JMP	UTXP6	
03017	R	340070	R		TAD	DMPSW	/FLIP THE TRACE FLOP
03020	R	751200	A		SNA:CLA		
03021	R	340167	R		TAD	C1	
03022	R	040070	R		DAC	DMPSW	
03023	R	602753	R		JMP	UTRA+1	/GET NEXT CHARACTER INSTEAD.
03024	R	340145	R	UTXP6	TAD	C277	/TRACE DISABLED = RETURN "?"
03025	R	602763	R		JMP	UTO	
03026	R	000000	A	GET1	0		/UNPACK 6-BITS
03027	R	440015	R		ISZ	XCTX	/STARTS=-1
03030	R	603036	R		JMP	GENDX	
03031	R	777775	A		LAW	-3	

03032 R 040015 R
03033 R 220014 R
03034 R 440014 R
03035 R 040016 R
03036 R 200016 R

03037 R 100770 R
03040 R 040016 R
03041 R 740010 A
03042 R 500151 R
03043 R 040061 R
03044 R 540151 R
03045 R 603011 R
03046 R 777740 A
03047 R 340061 R
03050 R 623026 R

GENDX

GEN*

DAC
LAC*
ISZ
DAC
LAC
RTL6
JMS
DAC
RAL
AND
DAC
SAD
JMP
LAW
TAD
JMP*
.EJECT

XCTX
AXOUT
AXOUT
GTEM
GTEM

XRTL6
GTEM

C77
CHAR
C77
EXTR
-40
CHAR
GET1

/NEXT WORD

/SAVE

/EXTENDED

03051	R	000000	A	XENDLN	0	/TERMINATE THE BUFFERED LINE - "ENDLN"
03052	R	220066	R		LAC*	LASTLN /SAVE OLD POINTER
03053	R	060004	R		DAC*	BUFR
03054	R	200004	R		LAC	BUFR
03055	R	060066	R		DAC*	LASTLN
03056	R	200056	R		LAC	ADD
03057	R	740200	A		SZA	
03060	R	060010	A		DAC*	AXIN
03061	R	225544	R		LAC*	(AXIN
03062	R	340167	R		TAD	C1
03063	R	040004	R		DAC	BUFR
03064	R	623051	R		JMP*	XENDLN
03065	R	000000	A	XPRNT	0	/PRINT A LINE NUMBER - "PRNTLN"
03066	R	200062	R		LAC	LINENO
03067	R	742020	A		RTR	
03070	R	742020	A		RTR	
03071	R	742020	A		RTR	
03072	R	740020	A		RAR	
03073	R	103101	R		JMS	PRNT
03074	R	200415	R		LAC	PER
					PRINTC	
03075	R	103362	R GEN*		JMS	XOUTL
03076	R	200062	R		LAC	LINENO
03077	R	103101	R		JMS	PRNT
03100	R	623065	R		JMP*	XPRNT
				/////		
				PRNT	0	/PRINT TWO DIGITS
03101	R	000000	A		AND	P177
03102	R	500155	R		DAC	VAL
03103	R	043322	R		LAC	C260
03104	R	200152	R		DAC	BOX
03105	R	040104	R		LAC	VAL
03106	R	203322	R		JMP	+.3
03107	R	603112	R		ISZ	BOX
03110	R	440104	R		DAC	VAL
03111	R	043322	R	XYZ	TAD	M12
03112	R	344770	R		SMA	
03113	R	740100	A		JMP	XYZ-1
03114	R	603110	R		LAC	BOX
03115	R	200104	R		PRINTC	
					JMS	XOUTL
03116	R	103362	R GEN*		LAC	VAL
03117	R	203322	R		TAD	C260
03120	R	340152	R		PRINTC	
					JMS	XOUTL
03121	R	103362	R GEN*		JMP*	PRNT
03122	R	623101	R		.EJECT	

```

03123 R 002003 A          IMBUFF  .IFUND MULTI
                                2003;  0;      0;      -1
03124 R 000000 A
03125 R 000000 A
03126 R 777777 A
03127 R          A          INBUF  .BLOCK 60
                                .ASCII  <15>
03207 R 064000 A
03210 R 000000 A
03211 R 001000 A          OUTBUF  1000
                                .BLOCK 57
03212 R          A          .ASCII  <15>
03271 R 064000 A
03272 R 000000 A
                                .ENDC
03273 R 000000 A          IMAGEW  0
03274 R 043306 R          DAC IMAC          /STORE AC FOR SORTB
03275 R 060011 R          DAC*      IMBUFF
                                003300 R
                                IMBF01= .+2
                                WCAL03  .WRITE  -3,3,IMBUFF,3
03276 R 003775 A GEN*      CAL+3*1000 -3&777
03276 R          A GEN*      11
03277 R 000011 A GEN*      IMBUFF
03300 R 003123 R GEN*      .DEC
                                GEN*
03301 R 777775 A GEN*      -3
                                .IFUND MULTI
                                .WAIT  -3
03302 R 000775 A GEN*      CAL -3&777
03303 R 000012 A GEN*      12
                                .ENDC
                                .IFDEF MULTI
                                JMS IORUSY
                                .ENDC
03304 R 203306 R          LAC IMAC          /RESTORE AC FOR SORTB
03305 R 623273 R          JMP*      IMAGEW
03306 R 000000 A          IMAC      0          /STORAGE FOR AC
03307 R 000000 A          IMAGER   0
                                003312 R
                                IMBF02= .+2
                                RCAL03  .READ   -2,3,IMBUFF,3
03310 R 003776 A GEN*      CAL+3*1000 -2&777
03310 R 000010 A GEN*      10
03311 R 003123 R GEN*      IMBUFF
03312 R          A GEN*      .DEC
                                GEN*
03313 R 777775 A GEN*      -3
                                .IFUND MULTI
                                .WAIT  -2
03314 R 000776 A GEN*      CAL -2&777
03315 R 000012 A GEN*      12
                                .ENDC
                                .IFDEF MULTI
                                JMS IORUSY
                                .ENDC
03316 R 220011 R          LAC*      IMBUFF
03317 R 500155 R          AND      P177
03320 R 240154 R          XOR      C200
03321 R 623307 R          JMP*      IMAGER
                                .EJECT

```

```

03322 R 003322 R      //
03322 R 000000 A      VAL=.
03323 R 200020 R      X133
03324 R 540501 R      0
03325 R 741000 A      LAC      ENDCR      /*READC*/
03326 R 603345 R      SAD      C215      /WAS LAST INPUT CR
03326 R 003331 R      SKP
03327 R      R      JMP      X133NX      /YES
03327 R      R      INBF01=+.2      /NO - GO GET NEXT FROM BUFFER
03327 R 002776 A GEN* RCAL01 .READ      -2.2,INBUF,48
03330 R 000010 A GEN*      CAL+2*1000 -2&777
03331 R 003127 R GEN*      10
03332 R 777720 A GEN*      INBUF
03333 R      R      .DEC
03333 R 000776 A GEN*      -48
03334 R 000012 A GEN*      .IFUND MULTI
03334 R      R      RCAL02 .WAIT      -2
03335 R 223331 R      CAL -2&777
03336 R 500144 R      12
03337 R 545555 R      .ENDC
03340 R 604152 R      .IFDEF MULTI
03341 R 545556 R      JMS IORUSY
03342 R 604152 R      .ENDC
03343 R 200012 R      LAC*      INBF01
03344 R 105374 R      AND      P17
03345 R 105401 R      SAD      (5      /EOF?
03346 R 240154 R      JMP      RDEOM      /YES
03347 R 540154 R      SAD      (6      /EOM?
03350 R 603345 R      JMP      RDEOM      /YES
03351 R 540500 R      LAC      INBUFP
03352 R 603345 R      JMS      .GET1      /UNPACK INITIALIZATION
03353 R 540477 R      JMS      .GET      /UNPACK CHARACTER
03354 R 603345 R      XOR      C200
03355 R 540475 R      SAD      C200      /IGNORE NULL
03356 R 200501 R      JMP      X133NX
03357 R 040020 R      SAD      C377      /IGNORE RUBOUTS
03360 R 040061 R      JMP      X133NX
03361 R 623322 R      SAD      C212      /IGNORE LINE-FEED
03361 R      R      JMP      X133NX
03361 R      R      SAD      C375      /USE CR FOR ESC
03361 R      R      LAC      C215
03361 R      R      DAC      ENDCR
03361 R      R      DAC      CHAR
03361 R      R      JMP*     X133
03361 R      R      .EJECT

```

```

03362 R 000000 A XOUTL 0 /*PRINTC"
03363 R 741200 A SNA /USE AC OR CHAR
03364 R 200061 R LAC CHAR
03365 R 040022 R DAC SAVEOT /SAVF CHAR
03366 R 440023 R ISZ PUTCNT /READY TO RUMP WPC?
03367 R 603377 R JMP XOUTLQ /NO
03370 R 223414 R LAC* OTBF01 /YES
03371 R 545557 R SAD (31000 /HAS THE BUFFER OVERFLOWED?
03372 R 603404 R JMP XOUTLI /YES - IGNORE CHAR
03373 R 345560 R TAD (1000 /NO - UPDATE WPC
03374 R 063414 R DAC* OTBF01
03375 R 777773 A LAW -5
03376 R 040023 R DAC PUTCNT /RESET COUNT
03377 R 223414 R XOUTLQ LAC* OTBF01
03400 R 545557 R SAD (31000
03401 R 603404 R JMP XOUTLI /BUFFER OVERFLOWED - DONT SAVE CHAR
03402 R 200022 R LAC SAVEOT /GET CHAR
03403 R 105435 R JMS .PUT
03404 R 200022 R XOUTLI LAC SAVEOT
03405 R 540501 R SAD C215 /OUTPUT BUFFER ON CR OR ESC
03406 R 603412 R JMP XOUTLX
03407 R 540475 R SAD C375
03410 R 603412 R JMP XOUTLX
03411 R 603426 R JMP XOUTLX
03412 R 003412 R XOUTLX=.
03413 R 003414 R OTBF01=.*+2
03412 R 002775 A GEN* WCAL01 .WRITE -3,2,OUTBUF,48
03413 R 000011 A GEN* CAL+2*1000 -3&777
03414 R 003211 R GEN* 11
03415 R 777720 A GEN* OUTBUF
03416 R 200013 R GEN* .DEC
03417 R 105431 R LAC -48
03420 R 777777 A JMS .PUTI
03421 R 040023 R LAW -1
03422 R 000775 A GEN* DAC PUTCNT
03423 R 000012 A GEN* .IFUND MULTI
03424 R 205560 R WCAL02 .WAIT -3
03425 R 063414 R CAL -3&777
03426 R 200061 R 12
03427 R 623362 R .ENDC
XOUTLZ LAC (1000
DAC* OTBF01
LAC CHAR
JMP* XOUTL
.EJECT

```

.DEC
 .REPT 37
 ISZ ERR2CT

03430 R 440106 R
 03431 R 440106 R *RPT
 03432 R 440106 R *RPT
 03433 R 440106 R *RPT
 03434 R 440106 R *RPT
 03435 R 440106 R *RPT
 03436 R 440106 R *RPT
 03437 R 440106 R *RPT
 03440 R 440106 R *RPT
 03441 R 440106 R *RPT
 03442 R 440106 R *RPT
 03443 R 440106 R *RPT
 03444 R 440106 R *RPT
 03445 R 440106 R *RPT
 03446 R 440106 R *RPT
 03447 R 440106 R *RPT
 03450 R 440106 R *RPT
 03451 R 440106 R *RPT
 03452 R 440106 R *RPT
 03453 R 440106 R *RPT
 03454 R 440106 R *RPT
 03455 R 440106 R *RPT
 03456 R 440106 R *RPT
 03457 R 440106 R *RPT
 03460 R 440106 R *RPT
 03461 R 440106 R *RPT
 03462 R 440106 R *RPT
 03463 R 440106 R *RPT
 03464 R 440106 R *RPT
 03465 R 440106 R *RPT
 03466 R 440106 R *RPT
 03467 R 440106 R *RPT
 03470 R 440106 R *RPT
 03471 R 440106 R *RPT
 03472 R 440106 R *RPT
 03473 R 440106 R *RPT
 03474 R 440106 R *RPT

003475 R ERR2=.
 03475 R 200145 R ERRPCX
 03476 R 103362 R GEN*
 03477 R 200106 R
 03500 R 103101 R
 03501 R 140106 R
 03502 R 440063 R
 03503 R 220063 R
 03504 R 741200 A
 03505 R 603512 R
 03506 R 040062 R
 03507 R 200416 R
 03510 R 103362 R GEN*

.OCT
 LAC C277
 PRINTC
 JMS XOUTL
 LAC ERR2CT
 JMS PRNT
 DZM ERR2CT
 ISZ PC
 LAC* PC
 SNA
 JMP ERR2T
 DAC LINENO
 LAC C240
 PRINTC
 JMS XOUTL

/CLEAR FOR NEXT ERROR
 /PRINT LINENO IF INDIRECT COMMAND

/SAVE FOR PRINTING

03511	R	103065	R	GEN*		PRNTLN		
03512	R	200501	R		ERR2T	JMS	XPRNT	
						LAC	CCR	
03513	R	103362	R	GEN*		PRINTC		
03514	R	603527	R			JMS	XOUTL	
03515	R	040106	R		FUNERR	JMP	RECVR	
03516	R	200145	R			DAC	ERR2CT	/SAVE TWO DIGIT ERROR CODE
						LAC	C277	
						PRINTC		/PRINT DOUBLE ?
03517	R	103362	R	GEN*		JMS	XOUTL	
03520	R	603475	R			JMP	ERRPCX	
						.IFUND MULTI		
03521	R	200501	R		RECOVR	LAC	CCR	
						PRINTC		
03522	R	103362	R	GEN*		JMS	XOUTL	
						.ENDC		
					RECOVR	.IFDEF MULTI		
						DZM*	CTLP	
						LAC CCR		
						PRINTC		
						JMS	IOBUSY	
						.ENDC		
03523	R				WCAL04	.WRITE	-3,2,FOCAL9,40	
03523	R	002775	A	GEN*		CAL+2*1000	-3&777	
03524	R	000011	A	GEN*		11		
03525	R	000177	R	GEN*		FOCAL9		
				GEN*		.DEC		
03526	R	777730	A	GEN*		-40		
		003527	R		RECVR=.			
						.IFDEF MULTI		
						JMS IOBUSY		
						.ENDC		
03527	R	200013	R			LAC OTBUFP		
03530	R	105431	R			JMS .PUTI		
03531	R	200501	R			LAC C215		
03532	R	040020	R			DAC ENDCR		
03533	R	777777	A			LAW -1		
03534	R	040023	R			DAC PUTCNT		
03535	R	205560	R			LAC (1000		
03536	R	063414	R			DAC* OTBF01		
03537	R	200026	R			LAC	TTIN	/RESET CAL FOR NORMAL INPUT
03540	R	103553	R			JMS	SETRCL	
03541	R	200027	R			LAC	TTOUT	
03542	R	103544	R			JMS	SETWCL	
03543	R	600547	R			JMP	STARTZ	
						.EJECT		

```

/SET WRITE CAL'S SUBROUTINE
03544 R 000000 A SETWCL 0
                                .IFUND MULTI
03545 R 043422 R DAC WCAL02 /.WAIT
                                .ENDC
03546 R 245560 R XOR (1000
                                .IFDEF MULTI
                                DAC* BWAIT /PROPER .WAITR
                                .ENDC
03547 R 044165 R DAC LBOUT1 /.INIT
03550 R 345560 R TAD (1000
03551 R 043412 R DAC WCAL01 /.WRITE
03552 R 623544 R JMP* SETWCL

/SET READ CAL'S SUBROUTINE
03553 R 000000 A SETRCL 0
                                .IFUND MULTI
03554 R 043333 R DAC RCAL02 /.WAIT
03555 R 245561 R XOR (2000
                                .ENDC
                                .IFDEF MULTI
                                XOR (1000
                                DAC* BWAIT /PROPER .WAITR
                                TAD (1000
                                .ENDC
03556 R 043327 R DAC RCAL01 /.READ
03557 R 345560 R TAD (1000
03560 R 044123 R DAC LBIN1A
03561 R 623553 R JMP* SETRCL

/.
.EJECT

```

03562	R	000000	A	PACBUF	0		/PACK A CHARACTER - "PACKC"
03563	R	777501	A		LAW	-277	
03564	R	340061	R		TAD	CHAR	
03565	R	741200	A		SNA		/CHANGE 277 TO 337
03566	R	340174	R		TAD	P40	
03567	R	340153	R		TAD	M100	
03570	R	741200	A		SNA		/TEST FOR RUBOUT.
03571	R	740040	A		HLT		
03572	R	340500	R		TAD	C377	
03573	R	040074	R		DAC	T2	/SAVE INPUT ITEM
							/SO THAT QUESTION DOESN'T MAKE
03574	R	500175	R		AND	C140	/CHAR LOOK LIKE A LEFT-ARROW
03575	R	340176	R		TAD	M140	
03576	R	740200	A		SZA		/DATA WORD.
03577	R	340175	R		TAD	C140	
03600	R	751200	A		SNA:CLA		
03601	R	603620	R		JMP	ESCA	/340-377 AND 200-237
03602	R	200074	R	TR1	LAC	T2	/240-337
03603	R	500151	R		AND	C77	
03604	R	740200	A		SZA		/IGNORE 300
03605	R	103623	R		JMS	PCK1	
03606	R	200074	R		LAC	T2	
03607	R	540501	R		SAD	C215	
03610	R	603613	R		JMP	+.3	
03611	R	200061	R		LAC	CHAR	
03612	R	623562	R	PACBXT	JMP*	PACBUF	
03613	R	200057	R		LAC	XCTIN	
03614	R	751200	A		SNA:CLA		
03615	R	603611	R		JMP	PACBXT	
03616	R	103623	R		JMS	PCK1	
03617	R	603613	R		JMP	.-4	
				/////			
03620	R	200151	R	ESCA	LAC	C77	
03621	R	103623	R		JMS	PCK1	
03622	R	603602	R		JMP	TR1	
					.EJECT		

03623	R	000000	A	PCK1	0	
03624	R	040024	R		DAC	TEMPK
03625	R	200057	R		LAC	XCTIN /:=0 TO START
03626	R	345562	R		TAD	(JMP PCKTB
03627	R	043631	R		DAC	.+2
03630	R	200056	R		LAC	ADD
03631	R	000000	A		XX	
03632	R	603651	R	PCKTB	JMP	ROT-1
03633	R	603652	R		JMP	ROT
					RTL6	
03634	R	100770	R	GEN*	JMS	XRTL6
03635	R	140057	R		DZM	XCTIN
03636	R	340024	R		TAD	TEMPK
03637	R	060010	A		DAC*	AXIN
03640	R	140056	R		DZM	ADD
03641	R	225542	R		LAC*	(PDLXR /CLEAR PACKING WORD
03642	R	744001	A		CMA:CLL	/CHECK FOR OVERFLOW (TAD P7600) TO PROTECT (X-MEM)
03643	R	340167	R		TAD	C1
03644	R	340143	R		TAD	P13 /RESERVATIONS
03645	R	365544	R		TAD*	(AXIN
03646	R	750400	A		SNL:CLA	
03647	R	623623	R		JMP*	PCK1
					ERROR	16 /FULL BUFFER
					.DEC	
03650	R	603455	R	GEN*	JMP	ERR2-16
					GEN*	
					////	
03651	R	750000	A		CLA	
03652	R			ROT	RTL6	
03652	R	100770	R	GEN*	JMS	XRTL6
03653	R	340024	R		TAD	TEMPK
03654	R	040056	R		DAC	ADD
03655	R	440057	R		ISZ	XCTIN
03656	R	623623	R		JMP*	PCK1
					.EJECT	

03657	R	200005	R	TDUMP	LAC	FRSTCV	/INIT POINTER FOR SYMBOL DUMP.
03660	R	440070	R		ISZ	DMPSW	/TURN OFF THE TRACE FOR EXIT
03661	R	040072	R		DAC	PT1	
03662	R	540003	R		SAD	LASTCV	
03663	R	103725	R		JMS	TDUMPC	
03664	R	540006	R		SAD	LASTV	/TEST FOR END OF LIST.
					POPJ		
03665	R	602265	R	GEN*	JMP	XPOPJ	
03666	R	220072	R		LAC*	PT1	
03667	R	040110	R		DAC	OP+1	/(DCA I XOP)-FOR(X-MEM)
03670	R	200107	R		LAC	OP	
03671	R	340167	R		TAD	C1	
03672	R	040014	R		DAC	AXOUT	
03673	R	777777	A		LAW	-1	
03674	R	040015	R		DAC	XCTX	
03675	R	777774	A		LAW	-4	
03676	R	103716	R		JMS	TDUMPC	
03677	R	440072	R		ISZ	PT1	
03700	R	220072	R		LAC*	PT1	/READ SUBSCRIPT TO 99
03701	R	103101	R		JMS	PRNT	
03702	R	777776	A		LAW	-2	
03703	R	103716	R		JMS	TDUMPC	
03704	R	440072	R		ISZ	PT1	
03705	R	125524	E		JMS*	.AO	/PICK UP VALUE
03706	R	400072	R		PT1+400000		
03707	R	104672	R		JMS	FLOUTP	/PRINT VALUE
03710	R	200501	R		LAC	C215	
					PRINTC		
03711	R	103362	R	GEN*	JMS	XOUTL	
03712	R	200156	R		LAC	GINC	
03713	R	341367	R		TAD	M2	
03714	R	340072	R		TAD	PT1	
03715	R	603661	R		JMP	TDUMP+2	
				///			
03716	R	000000	A	TDUMPC	0		
03717	R	040073	R		DAC	T1	
					GETC		
03720	R	102752	R	GEN*	JMS	UTRA	
					PRINTC		
03721	R	103362	R	GEN*	JMS	XOUTL	
03722	R	440073	R		ISZ	T1	
03723	R	603720	R		JMP	.-3	
03724	R	623716	R		JMP*	TDUMPC	
03725	R	000000	A	TDUMPC	0		
03726	R	200420	R		LAC	C255	
					PRINTC		
03727	R	103362	R	GEN*	JMS	XOUTL	
03730	R	200501	R		LAC	C215	
					PRINTC		
03731	R	103362	R	GEN*	JMS	XOUTL	
03732	R	200072	R		LAC	PT1	
03733	R	623725	R		JMP*	TDUMPC	
					.EJECT		

```

///
XLAN          LAC*      RANPT  /RANDOM NUMBER GENERATOR.
03734 R 220112 R          ISZ      RANPT
03735 R 440112 R          RAL
03736 R 740010 A          TAD      RANPT
03737 R 340112 R          TAD*     PT1
03740 R 360072 R          RAL
03741 R 740010 A          DAC* .AB
03742 R 065522 E          TAD*     RANPT
03743 R 360112 R          DAC* .AC
03744 R 065523 E          DZM* .AA
03745 R 165521 E          LAC      RANPT
03746 R 200112 R          SAD*     (.SCOM
03747 R 565563 R          DZM      RANPT
03750 R 140112 R          RETURN
03751 R 602517 R GEN*    JMP      EFUN3
                          /TWOS COMPLEMENT - CIA
                          TWOS
03752 R 000000 A          CMA
03753 R 740001 A          TAD      C1
03754 R 340167 R          JMP*     TWOS
03755 R 623752 R          .EJECT

```

```

003756 R 003756 R
03756 R 225564 R
03757 R 040000 R
03760 R 225565 R
03761 R 040001 R
03762 R 044140 R
03763 R 044173 R
03764 R 044125 R
03765 R 345566 R
03766 R 040002 R
03767 R 040005 R
03770 R 040003 R
03771 R 040006 R
03772 R 204140 R
03773 R 345567 R
03774 R 042360 R
03775 R 200161 R
03776 R 062360 R

03777 R 000775 A GEN*
04000 R 000001 A GEN*
04001 R 003521 R GEN*
04002 R 000000 A GEN*

04003 R 002775 A GEN*
04004 R 000011 A GEN*
04005 R 000177 R GEN*
GEN*
04006 R 777730 A GEN*

COMEIN=.
BEGIN
  .IFUND MULTI
  LAC* (.SCOM+3
  DAC BOTTOM
  LAC* (.SCOM+2
  DAC RUFSTX
  DAC FILE01
  DAC FILE02
  DAC FILE03
  TAD (3
  DAC ENDT
  DAC FRSTCV
  DAC LASTCV
  DAC LASTV
  LAC FILE01
  TAD (2
  DAC FLAC1
  LAC FILEXT
  DAC* FLAC1
  .INIT -3,0,RECOVR
  CAL+0*1000 -38777
  1
  RECOVR+0
  0
  .WRITE -3,2,FOCAL9,40
  CAL+2*1000 -38777
  11
  FOCAL9
  .DEC
  -40
  .ENDC
  .IFDEF MULTI

BEGIN=.
  .ENDC
  LAC .AA
  DAC FLAC1
  DAC FLAC2
  DAC FLAC3
  .IFUND MULTI
  LAC (OUTBUF+2
  JMS .PUTI
  JMP XSBEGN
  BUFFER=COMEIN+70
  COMOUT=COMEIN+110
  .LOC COMOUT
  215 /STOPPER
  .ENDC
  .IFDEF MULTI
  DAC FLAC14
  TAD (1
  DAC FLAC15
  TAD (1
  DAC FLAC16
  JMP MSTART
  .ENDC

```

.EJECT


```

/LIBRARY COMMAND FORMAT:
/
/ LIBRARY IN FILE
/ LIBRARY OUT FILE
/ LIBRARY WRITE "ANY COMMAND
/ LIBRARY WRITE ALL
/ LIBRARY WRITE XX.00
/ LIBRARY WRITE XX.YY
/ LIBRARY KILL
/ LIBRARY CLOSE
/
LIBRAR SPNOR /IGNORE SPACES
04067 R 102233 R GEN* JMS XSPNOR
04070 R 540501 R SAO C215 /IGNORE COMMAND IF CR
POPJ
04071 R 602265 R GEN* JMP XPOPJ
04072 R 540433 R SAO C273 /IGNORE IF;
POPJ
04073 R 602265 R GEN* JMP XPOPJ
04074 R 777767 A LAW -11
04075 R 102110 R JMS FETVAR
SORTC GLIST
04076 R 101321 R GEN* JMS XSORTC
04077 R 000453 R GEN* GLIST-1
04100 R 741000 A SKP
ERROR 32
GEN*
04101 R 603435 R GEN* .DEC
JMP ERR2-32
SORTJX LIBCMD
JMS XSORTX
LIBCMD-1
04102 R 102040 R GEN* ERROR 32 /BAD LIBR CMD ARG
04103 R 000331 R GEN* .DEC
JMP ERR2-32
GEN*
04104 R 603435 R GEN*
/
LBIN JMS LBFILE /GET FILE NAME
04105 R 104234 R LAC BLKIN /SETUP INPUT CAL'S
04106 R 200030 R DAC LIBRSW /SIGN BIT 0
04107 R 040032 R DAC LBIN01
04110 R 044114 R DAC LBIN02
04111 R 044136 R DAC LBIN03
04112 R 044266 R JMS SETRCL
04113 R 103553 R .IFDEF MULTI
JMS IOBUSY
.ENDC
LBIN01 .INIT 0,0,RECOVR
04114 R 000000 A GEN* CAL+0*1000 08777
04114 R 000001 A GEN* 1
04115 R 000001 A GEN* RECOVR+0
04116 R 003521 R GEN* 0
04117 R 000000 A GEN* .IFDEF MULTI
JMS IOBUSY
.ENDC
04120 R 204125 R LAC FILE03
04121 R 505571 R AND (77777
04122 R 044125 R DAC FILE03

```

```

004125 R FILE03=.+2
04123 R LBIN1A .FSTAT 0,XX
04123 R 003000 A GEN* CAL+3000 08777
04124 R 000002 A GEN* 2
04125 R 000000 A GEN* XX
04126 R 044234 R DAC LBFILF
04127 R 204125 R LAC FILE03
04130 R 505552 R AND (700000
04131 R 741200 A SNA
04132 R 604136 R JMP .+4
04133 R 204234 R LAC LBFILF
04134 R 741200 A SNA
ERROR 34
04135 R 603433 R GEN* .DEC
R GEN* JMP ERR2-34
.IFDEF MULTI
JMS IOBUSY
.ENDC
004140 R FILE01=.+2
04136 R LBIN02 .SEEK 0,XX
04136 R 000000 A GEN* CAL 08777
04137 R 000003 A GEN* 3
04140 R 000000 A GEN* XX
.IFDEF MULTI
JMS IOBUSY
.ENDC
04141 R 200162 R LAC CFRSX
04142 R 040063 R DAC PC
04143 R 200061 R LBINLP LAC CHAR
04144 R 540501 R SAD C215
POPJ /ALL DONE
04145 R 602265 R GEN* JMP XPOPJ
04146 R 540433 R SAD C273
04147 R 601203 R JMP PROCES /MORE IN COME IN
GETC
04150 R 102752 R GEN* JMS UTRA
04151 R 604144 R JMP LBINLP+1
04152 R 104260 R RDEOM JMS LBIEND /END LIBR IF OPEN
04153 R 200501 R LAC C215
04154 R 603347 R JMP XI33NX+2
04155 R 104260 R HSPX JMS LBIEND /END LIBR IF OPEN
04156 R 604143 R JMP LBINLP
/
04157 R 104234 R LBOUT JMS LBFILF /GET FILE NAME
04160 R 200031 R LAC BLKOUT /SETUP OUTPUT CAL'S
04161 R 044171 R DAC LBOUT2
04162 R 044202 R DAC LBOUT3
04163 R 044210 R DAC LBOUT4
04164 R 103544 R JMS SETWCL
.IFDEF MULTI
JMS IOBUSY
.ENDC
04165 R LBOUT1 .INIT 0,0,RECOVR
04165 R 000000 A GEN* CAL+0*1000 08777
04166 R 000001 A GEN* 1

```

04167	R	003521	R	GEN*	RECOVR+0		
04170	R	000000	A	GEN*	0		
					.IFDEF	MULTI	
					JMS	IOBUSY	
					.ENDC		
		004173	R		FILE02=.+2		
04171	R				LBOUT2	.ENTER 0,XX	
04171	R	000000	A	GEN*		CAL 0&777	
04172	R	000004	A	GEN*		4	
04173	R	000000	A	GEN*		XX	
					.IFDEF	MULTI	
					JMS	IOBUSY	
					.ENDC		
04174	R	777777	A		LAW -1		
04175	R	040032	R		DAC LIIRSW		
04176	R	604143	R		JMP LBINLP		
					/		
04177	R	200032	R		LBCLOS	LAC	LIBRSW
04200	R	740100	A			SMA	
						ERROR	35
						.DEC	
04201	R	603432	R	GEN*	JMP	ERR2-35	
				GEN*	.IFDEF	MULTI	
					JMS	IOBUSY	
					.ENDC		
					.CLOSE	0	
04202	R				LBOUT3	CAL 0&777	
04202	R	000000	A	GEN*		6	
04203	R	000006	A	GEN*		JMP	LBOUTZ
04204	R	604214	R			LAC	LIBRSW
04205	R	200032	R		LBKILL	SMA	
04206	R	740100	A			ERROR	35
						.DEC	
04207	R	603432	R	GEN*	JMP	ERR2-35	
				GEN*	.IFDEF	MULTI	
					JMS	IOBUSY	
					.ENDC		
					.INIT	0,0,RECOVR	
04210	R				LBOUT4	CAL+0*1000 0&777	
04210	R	000000	A	GEN*		1	
04211	R	000001	A	GEN*		RECOVR+0	
04212	R	003521	R	GEN*		0	
04213	R	000000	A	GEN*		DZM	LIBRSW
04214	R	140032	R		LBOUTZ	LAC	TTOUT
04215	R	200027	R			JMS	SETWCL
04216	R	103544	R			JMP	LBINLP
04217	R	604143	R				/GO FINISH CMD
					/		
04220	R	200032	R		LBWRIT	LAC	LIBRSW
04221	R	740100	A			SMA	
						ERROR	35
						.DEC	
04222	R	603432	R	GEN*	JMP	ERR2-35	
				GEN*		WRITE	
04223	R	601234	R			LAC	LIBRSW
04224	R	200032	R		LBTEXT		
04225	R	740100	A			SMA	

				ERROR	8	
			GEN*	.DEC		
04226	R	603465	R GEN*	JMP	ERR2-8	
04227	R			GETC		/PUT COMMAND INTO
04227	R	102752	R GEN*	JMS	UTRA	
				PRINTC		/OUTPUT BUFFER
04230	R	103362	R GEN*	JMS	XOUTL	
04231	R	540501	R	SAD	C215	/ALL DONE?
				POPJ		/YES
04232	R	602265	R GEN*	JMP	XPOPJ	
04233	R	604227	R	JMP	LBTEXT	/NO-MORE
04234	R	000000	A	LBFILE	0	
04235	R	204125	R	LAC	FILE03	
04236	R	345567	R	TAD	(2	
04237	R	044260	R	DAC	LBIEND	
04240	R	200161	R	LAC	FILEXT	/SETUP 'FCL' AS EXTENSION
04241	R	064260	R	DAC*	LBIEND	
04242	R	200061	R	LAC CHAR		
04243	R	540416	R	SAD C240		
04244	R	741000	A	SKP		
04245	R	604250	R	JMP .+3		
				GETC		
04246	R	102752	R GEN*	JMS	UTRA	
04247	R	604243	R	JMP .-4		
04250	R	777772	A	LAW	-6	
04251	R	102110	R	JMS	FETVAR	/GET FILE NAME
04252	R	200061	R	LAC	CHAR	
04253	R	540501	R	SAD	C215	/CR
04254	R	741000	A	SKP		
04255	R	540433	R	SAD	C273	/;
04256	R	624234	R	JMP*	LBFILE	
				ERROR	33	
				.DEC		
04257	R	603434	R GEN*	JMP	ERR2-33	
04260	R	000000	A	LBIEND	0	
04261	R	200026	R	LAC	TTIN	/RESTORE INPUT
04262	R	103553	R	JMS	SETRCL	
04263	R	200032	R	LAC	LIBRSW	/IF LIBRARY OPEN,
04264	R	741200	A	SNA		/CLOSE IT
04265	R	624260	R	JMP*	LBIEND	
04266	R			LBIN03	0	
04266	R	000000	A GEN*	.CLOSE	0	
04267	R	000006	A GEN*	CAL 0&777		
04270	R	140032	R	6		
04271	R	624260	R	DZM	LIBRSW	/CLEAR SWITCH
				JMP*	LBIEND	
				.EJECT		

```

                                /FLOATING POINT ARITHMETIC INTERPRETER FOR FOCAL
                                /FLOATING POINT PACKAGE - EXPONENTIAL
04272 R                                FEXP GETSGN
04272 R 225522 E GEN*                LAC*      .AB
04273 R 750100 A                      SMA:CLA
04274 R 604277 R                      JMP       .+3
04275 R 125534 E                      JMS*     .BA
04276 R 750001 A                      CLA:CMA

04277 R 040122 R                      DAC      SIGN2    /C(SIGN)=-1 IF X<0
04300 R 125525 E                      JMS*     .AP      /PUT
04301 R 000076 R                      FLARG
04302 R 125515 E                      JMS*     DEXP
04303 R 604305 R                      JMP       .+2
04304 R 000076 R                      FLARG
04305 R 440122 R                      ISZ      SIGN2
                                RETURN
04306 R 602517 R GEN*                JMP      EFUN3
04307 R 105140 R                      JMS     FINT
04310 R 600137 R                      FPUT XY
04311 R 500167 R                      FGET FLTONE
04312 R 400137 R                      FDIV XY
04313 R 000000 A                      FEXT
                                RETURN
04314 R 602517 R GEN*                JMP      EFUN3
                                /FLOATING POINT ARC TANGENT
04315 R 125513 E                      ARTN    JMS*     DATAN
04316 R 604320 R                      JMP     .+2
04317 R 000076 R                      FLARG
                                RETURN
04320 R 602517 R GEN*                JMP      EFUN3
                                /FLOATING LOGARITHM
04321 R                      FLOG    GETSGN
04321 R 225522 E GEN*                LAC*     .AB
04322 R 741200 A                      SNA
                                ERROR     25      /ZERO ARGUEMENT FOR LOG
                                .DEC
                                GEN*
04323 R 603444 R GEN*                JMP     ERR2-25
04324 R 751100 A                      SPA:CLA
04325 R 125534 E                      JMS*    .BA      /NEGATIVE ARGUMENT
04326 R 125525 E                      JMS*    .AP      /PUT
04327 R 000076 R                      FLARG
04330 R 125516 E                      JMS*    DLOG
04331 R 604333 R                      JMP     .+2
04332 R 000076 R                      FLARG
                                RETURN
04333 R 602517 R GEN*                JMP      EFUN3
                                /FLOATING POINT SINE AND COSINE
04334 R 125514 E                      FCOS    JMS*    DCOS
04335 R 604337 R                      JMP     .+2
04336 R 000076 R                      FLARG
                                RETURN
04337 R 602517 R GEN*                JMP      EFUN3
                                FSIN
04340 R 125517 E                      JMS*    DSIN
04341 R 604343 R                      JMP     .+2
04342 R 000076 R                      FLARG

```

FOCAL PAGE 85

04343 R 602517 R GEN*

RETURN
JMP EFUN3
.EJECT

```

/INPUT/OUTPUT ROUTINES FOR THE FOCAL
/FLOATING POINT PACKAGE.
/IN THE COMMENTS BELOW:-
/ F = NUMBER OF DIGITS TO BE OUTPUT      =FISH
/ D = NUMBER OF DECIMAL PLACES           =DECP
/ E = DECIMAL EXPONENT                   =BEXP
/ P = NUMBER OF PLACES REMAINING TO BE
/ PRINTED BEFORE DECIMAL POINT
TGO      0
DAC      SCOUNT      /SAVE NUMBER OF DIGITS AVAILABLE - *SET COUNTS*
LAC      FISH
SNA      /FLOATING OUTPUT?
JMP      R6            /YES, ROUND OFF TO 6 PLACES
JMS TWOS /NO, COMPUTE FIELD SIZES
TAD      DECP
SPA      / F-D > 0 ?
JMP      .+5          /YES
CLA:CLA  /NO,
TAD      FISH
DAC      DECP        /MAKE D = F-1
CLA:CLA  BEXP        /COMPARE DECIMAL EXPONENT
TAD      SMA         / F-D > E ?
CLA      /NO, ROUND OFF TO .F PLACES
TAD      FISH        /YES
SPA      / D+E < 0 ?
JMP      RET1        /YES, NO ROUNDING NEEDED, GO TO PRINT
TAD      MD          /NO, ROUND TO D+E PLACES,
SMA      /TO A MAXIMUM OF D PLACES
CLA
TAD      RND2        / *ROUND UP *
DAC      TEMPO       /SAVE NUMBER+1 OF PLACES TO ROUND TO
TAD      BUFST       /SET UP BUFFER ADDRESS AT WHICH
DAC      PLCE        /ROUNDING OFF SHOULD START
LAC      TEMPO
JMS TWOS /SET UP COUNT OF MAXIMUM NUMBER
DAC      TEMPO       /OF CARRIES ALLOWABLE
LAC      (5          /LITTLE EXTRA ON FIRST DIGIT.
RET      ISZ*        /ADD 1 TO DIGIT AT CURRENT POSITION
TAD*     PLCE
TAD      M12
SPA:CLA  /CARRY REQUIRED?
JMP      FPRNT      /NO, GO TO OUTPUT
DAC*     PLCE       /YES, MAKE CURRENT DIGIT ZERO
ISZ      TEMPO     /BEGINNING OF BUFFER REACHED?
JMP      DECR      /NO, DECREMENT BUFFER ADDRESS AND REPEAT
ISZ*     PLCE       /YES, SET MANTISSA TO 0.1
RET1     LAC BEXP   /COMPENSATE BY INCREM EXPONENT
TAD (1   /FIX FOR OUTPUT OF .1
DAC BEXP /...WAD JUNE 69
/FORMERLY ISZ BEXP REPLACED ABOVE THREE INSTRUCTIONS
FPRNT   LAC FISH   /AUTO-INDEX REGISTER ALREADY SET. - *PRINT*
SNA     / F = 0 ?
JMP     FLOPX    /YES, OUTPUT AS FLOATING NUMBER
JMS TWOS /NO.

```

```

04344 R 000000 A
04345 R 040123 R
04346 R 200040 R
04347 R 741200 A
04350 R 604372 R
04351 R 103752 R
04352 R 340055 R
04353 R 741100 A
04354 R 604361 R
04355 R 750001 A
04356 R 340040 R
04357 R 040055 R
04360 R 750001 A
04361 R 340132 R
04362 R 740100 A
04363 R 750000 A
04364 R 340040 R
04365 R 741100 A
04366 R 604413 R
04367 R 344505 R
04370 R 740100 A
04371 R 750000 A
04372 R 340210 R
04373 R 040125 R
04374 R 340211 R
04375 R 040124 R
04376 R 200125 R
04377 R 103752 R
04400 R 040125 R
04401 R 205555 R
04402 R 460124 R
04403 R 360124 R
04404 R 344770 R
04405 R 751100 A
04406 R 604416 R
04407 R 060124 R
04410 R 440125 R
04411 R 604453 R
04412 R 460124 R
04413 R 200132 R
04414 R 345572 R
04415 R 040132 R
04416 R 200040 R
04417 R 741200 A
04420 R 604502 R
04421 R 103752 R

```

04422	R	040124	R		DAC	FCOUNT	/SET UP COUNT TO PRINT F PLACES
04423	R	340132	R		TAD	BEXP	
04424	R	740300	A		SMA:SZ		/ E > F ?
04425	R	604500	R		JMP	XXX	/YES, PRINT X'S
04426	R	340055	R		TAD	DECP	
04427	R	740100	A		SMA		/ E < F-D ?
04430	R	750000	A		CLA		/NO, TAKE P = E
04431	R	103752	R		JMS TWOS		/YES, TAKE P = F-D
04432	R	340132	R		TAD	BEXP	
04433	R	103752	R		JMS TWOS		
04434	R	040125	R		DAC	TEMPO	/SET UP MINUS P
04435	R	200132	R	BACK	LAC	BEXP	/PRINT DD.OOD
04436	R	340125	R		TAD	TEMPO	
04437	R	751200	A		SNA:CLA		/ P = E ?
04440	R	604465	R		JMP	DIG	/YES, PRINT DIGIT
04441	R	340125	R		TAD	TEMPO	/NO,
04442	R	340167	R		TAD	C1	
04443	R	751100	A		SPA:CLA		/ P > 1 ?
04444	R	777760	A		LAW	240-260	/YES, TAKE SPACE; OTHERWISE ZERO
04445	R	104460	R	IN	JMS	OUTA	/PRINT CHARACTER
04446	R	440125	R		ISZ	TEMPO	/P CHARACTERS PRINTED?
04447	R	604435	R		JMP	BACK	/NO
04450	R	200415	R		LAC	PER	/YES,
					PRINTC		/PRINT DECIMAL POINT
04451	R	103362	R	GEN*	JMS	XOUTL	
04452	R	604435	R		JMP	BACK	
				/////			
04453	R	750001	A		CLA:CMA		
04454	R	340124	R		TAD	PLCE	
04455	R	040124	R		DAC	PLCE	
04456	R	750000	A		CLA		
04457	R	604402	R		JMP	RET	
				/////			
04460	R	000000	A		OUTA		
04461	R	105036	R		JMS	OUTDG	/PRINT CHARACTER
04462	R	440124	R		ISZ	FCOUNT	/F CHARACTERS PRINTED?
04463	R	624460	R		JMP*	OUTA	/NO, RETURN
04464	R	624344	R		JMP*	TGO	/YES, NUMBER FINSHED
				/////			
04465	R	750001	A		CLA:CMA		
04466	R	340132	R		TAD	BEXP	/REDUCE E, BY 1
04467	R	040132	R		DAC	BEXP	
04470	R	440123	R		ISZ	SCOUNT	/ARE ALL SIG. FIGS. USED?
04471	R	604476	R		JMP	DIGP5	/NO
04472	R	750001	A		CLA:CMA		/YES,
04473	R	040123	R		DAC	SCOUNT	/RESET COUNT TO -1
04474	R	750000	A		CLA		
04475	R	604445	R		JMP	IN	/AND LEAVE C(AC) = 0
04476	R	220014	A	DIGP5	LAC*	FLT XR	/TAKE NEXT DIGIT FROM BUFFER
04477	R	604445	R		JMP	IN	
				/////			
					/DO FLOATING OUTPUT		
04500	R	200040	R		XXX	LAC	FISH
04501	R	741000	A			SKP	
04502	R	200055	R		FLOPX	LAC	DECP

04503	R	103752	R		JMS	TWOS	
04504	R	741200	A		SNA		
04505	R	777767	A	MD	LAW	-DIGITS	/SET COUNT TO PRINT
04506	R	040124	R		DAC	FCOUNT	/6 DIGITS AFTER DECIMAL POINT
04507	R	750000	A		CLA		
04510	R	105036	R		JMS	OUTDG	/PRINT "0"
04511	R	200415	R		LAC	PER	
					PRINTC		/PRINT "."
04512	R	103362	R	GEN*	JMS	XOUTL	
04513	R	444344	R		ISZ	TGO	/SEND RETURN
04514	R	220014	A		LAC*	FLT XR	/TAKE NEXT DIGIT FROM BUFFER
04515	R	104460	R		JMS	OUTA	/PRINT IT
04516	R	440123	R		ISZ	SCOUNT	/TEST FOR END OF INPUT
04517	R	604514	R		JMP	.-3	/AND REPEAT
04520	R	750001	A		CLA:CHA		
04521	R	040123	R		DAC	SCOUNT	/OUTPUT EXTRA ZEROS.
04522	R	750000	A		CLA		
04523	R	604515	R		JMP	.-6	
					.EJECT		

```

/DOUBLE PRECISION DECIMAL-BINARY
/INPUT AND CONVERSION FOR + OR - XXX...
04524 R 000000 A DECONV 0
04525 R 165522 E DZM* .AB
04526 R 165523 E DZM* .AC
04527 R 140037 R DZM OVER2
04530 R 140131 R DZM DNUMBR
04531 R 750001 A CLA:CHA
04532 R 040130 R DAC ISIGN
04533 R 777525 A LAW -253
04534 R 340061 R TAD CHAR
04535 R 741200 A SNA
04536 R 604543 R JMP .+5 /+SIGN: GET NEXT
04537 R 341367 R TAD M2 /CHECK - SIGN
04540 R 750200 A SZA:CLA
04541 R 604544 R JMP .+3
04542 R 040130 R DAC ISIGN
04543 R 104662 R JMS INPUT /GET NEXT
04544 R 540416 R SAD C240
04545 R 604543 R JMP .-2
/FORMERLY WAS JMP .-4
04546 R 104550 R JMS DECON
04547 R 624524 R JMP* DECONV
/////
04550 R 000000 A DECON 0
04551 R 540414 R SAD C305 /TEST LEAD CHARACTER FOR TERMINATOR
04552 R 624550 R JMP* DECON
TESTN
04553 R 102242 R GEN* JMS XTESTN
04554 R 624550 R JMP* DECON
04555 R 604565 R JMP DTST
04556 R 040127 R DSAVE DAC DIGIT /YES - SORTCN IN AC
04557 R 440131 R ISZ DNUMBR /INDEX NUMBER OF DIGITS
04560 R 104600 R JMS MULT10 /REMAIN MUST =0 SINCE OVERFLOW IS CHECKED
04561 R 740200 A SZA ERROR 26 /INPUT-OVERFLOW ERROR
GEN* .DEC
04562 R 603443 R GEN* JMP ERR2-26
04563 R 104662 R JMS INPUT
04564 R 604551 R JMP DECON+1 /CONTINUE
MINUSA=.
04565 R 777477 A DTST LAW -301
04566 R 340061 R TAD CHAR
04567 R 751100 A SPA:CLA
04570 R 624550 R JMP* DECON
04571 R 777445 A LAW -333
/LAW -332 CHANGED TO LAW -333 ....WAD....JUNE 69 FOR 02 BUG
04572 R 340061 R TAD CHAR
04573 R 750100 A SMA:CLA
04574 R 624550 R JMP* DECON
04575 R 200061 R LAC CHAR
04576 R 500151 R AND C77
04577 R 604556 R JMP DSAVE
/////
04600 R 000000 A MULT10 0 /ROUTINE TO MULTIPLY

```

04601 R 200037 R
 04602 R 040036 R
 04603 R 225523 E
 04604 R 040035 R
 04605 R 225522 E
 04606 R 040034 R
 04607 R 140126 R
 04610 R 104623 R
 04611 R 104623 R
 04612 R 104641 R
 04613 R 104623 R
 04614 R 200127 R
 04615 R 040036 R
 04616 R 140035 R
 04617 R 140034 R
 04620 R 104641 R
 04621 R 200126 R
 04622 R 624600 R

04623 R 000000 A
 04624 R 200037 R
 04625 R 744010 A
 04626 R 040037 R
 04627 R 225523 E
 04630 R 740010 A
 04631 R 065523 E
 04632 R 225522 E
 04633 R 740010 A
 04634 R 065522 E
 04635 R 200126 R
 04636 R 740010 A
 04637 R 040126 R
 04640 R 624623 R
 04641 R 000000 A
 04642 R 754000 A
 04643 R 200037 R
 04644 R 340036 R
 04645 R 040037 R
 04646 R 750010 A
 04647 R 365523 E
 04650 R 340035 R
 04651 R 065523 E
 04652 R 750010 A
 04653 R 365522 E
 04654 R 340034 R
 04655 R 065522 E
 04656 R 750010 A
 04657 R 340126 R
 04660 R 040126 R
 04661 R 624641 R

04662 R 000000 A
 04663 R 200025 R
 04664 R 750200 A

LAC OVER2 /FROM TAD
 DAC OTEMP
 LAC* .AC /DOUBLE PRECISION WORD
 DAC LTEMP /BY TEN (DECIMAL)
 LAC* .AB /REMAIN=REMAINDER
 DAC HTEMP
 DZM REMAIN
 JMS MULT2 /CALL SUBROUTINE TO
 JMS MULT2 /MULTIPLY BY TWO
 JMS DUBLAD /CALL DOUBLE ADD
 MULT2
 LAC DIGIT /ADD LAST DIGIT RECEIVED
 DAC OTEMP
 DZM LTEMP
 DZM HTEMP
 JMS DUBLAD
 LAC REMAIN /EXIT WITH REMAINDER
 JMP* MULT10 /IN AC

////
 MULT2 0 /MULTIPLY OVER2, LORD, WORD BY 2

LAC OVER2
 CLL:RAL
 DAC OVER2
 LAC* .AC
 RAL
 DAC* .AC
 LAC* .AB
 RAL
 DAC* .AB
 LAC REMAIN
 RAL
 DAC REMAIN
 JMP* MULT2

DUBLAD 0 /TRIPLE PRECISION ADDITION

CLA:CLL
 LAC OVER2
 TAD OVER1
 DAC OVER2
 CLA:RAL
 TAD* .AC
 TAD AC1L
 DAC* .AC
 CLA:RAL
 TAD* .AB
 TAD AC1H
 DAC* .AB
 CLA:RAL
 TAD REMAIN
 DAC REMAIN
 JMP* DUBLAD

////
 /INPUT FROM TEXT OR KEYBOARD:
 INPUT 0

LAC INSUB /INPUT A CHARACTER
 SZA:CLA /NON-ZERO FOR KEYBOARD

04665 R 604670 R
04666 R 102752 R GEN*
04667 R 624662 R
04670 R 103322 R GEN*
04671 R 624662 R

JMP .+3
GETC
JMS UTRA
JMP* INPUT
READC
JMS XI33
JMP* INPUT
.EJECT

```

/FLOATING OUTPUT CONVERSION ROUTINE
FLOUTP 0
04672 R 000000 A LAC* .AB /NUMBER>0??
04673 R 225522 E SMA:CLA
04674 R 750100 A LAW 240-255 /PRINT DASH OR SPACE
04675 R 777763 A TAD SMIN
04676 R 340420 R PRINTC
04677 R 103362 R GEN* JMS XOUTL
04700 R 225522 E LAC* .AB /TAKE ABSOLUTE VALUE
04701 R 751100 A SPA:CLA
04702 R 125534 E JMS* .BA
04703 R 750001 A CLA:CMA /SUBTRACT 1 FROM BINARY EXPONENT
04704 R 365521 E TAD* .AA /COMPENSATE AT FG04
04705 R 065521 E DAC* .AA
04706 R 140132 R DZM BEXP /INITIALIZE DECIMAL EXPONENT
04707 R 225521 E FG02 LAC* .AA /IS -4<EXPONENT<-1?
04710 R 740100 A SMA
04711 R 604723 R JMP FG03
04712 R 340214 R TAD TEN
04713 R 750100 A SMA:CLA
04714 R 604727 R JMP FG04
04715 R 125530 E JMS* .AS /MULTIPLY
04716 R 000214 R TEN
04717 R 750001 A CLA:CMA
04720 R 340132 R FG03M2 TAD BEXP
04721 R 040132 R DAC BEXP
04722 R 604707 R JMP FG02
04723 R 125531 E FG03 JMS* .AT /DIVIDE
04724 R 000214 R TEN
04725 R 200167 R LAC C1
04726 R 604720 R JMP FG03M2

/////
04727 R 140127 R FG04 DZM DIGIT /MULTIPLY BY TWO
04730 R 104623 R JMS MULT2 /I.E. SHIFT LEFT
04731 R 200211 R LAC BUFST /INIT BUFFER POINTER
04732 R 065573 R DAC* (FLT XR
04733 R 225521 E LAC* .AA
04734 R 045036 R DAC OUTDG /TEMP COUNT
04735 R 104600 R JMS MULT10 /MULTIPLY BY TEN
04736 R 604753 R JMP FG05
04737 R 744020 A FG05A CLL:RAR
04740 R 044344 R DAC TGO /TEMP
04741 R 225522 E LAC* .AB
04742 R 740020 A RAR
04743 R 065522 E DAC* .AB
04744 R 225523 E LAC* .AC
04745 R 740020 A RAR
04746 R 065523 E DAC* .AC
04747 R 200037 R LAC OVER2
04750 R 740020 A RAR
04751 R 040037 R DAC OVER2
04752 R 204344 R LAC TGO /TEMP
04753 R 445036 R FG05 ISZ OUTDG /TEMP COUNT
04754 R 604737 R JMP FG05A
04755 R 741200 A SNA /IS FIRST DIGIT A ZERO

```

```

04756 R 605025 R JMP FG07 /YES, IGNORE
04757 R 060014 A DAC* FLT XR /MULTIPLICATIONS YIELD
04760 R 204770 R FG06 LAC DCOUNT
04761 R 065521 E DAC* .AA
04762 R 104600 R JMS MULT10 /IE. .672x10=6+.72.. ETC
04763 R 060014 A DAC* FLT XR
04764 R 465521 E ISZ* .AA /ALL DIGITS OUTPUT??
04765 R 604762 R JMP .-3 /NO: CONTINUE
04766 R 200211 R LAC BUFST /INIT BUFFER POINTER
04767 R 065573 R DAC* (FLT XR
004770 R 777766 A M12=. DCOUNT LAW -DIGITS-1
04771 R 104344 R JMS TGO /OUTPUT MANTISSA AND EXPONENT
04772 R 624672 R JMP* FLOUTP
04773 R 200414 R LAC C305
PRINTC
04774 R 103362 R GEN* JMS XOUTL
/OUTPUT THE EXPONENT
LAC BEXP /TAKE ABSOLUTE VALUE OF EXPONENT
SPA
04776 R 741100 A SPA
04777 R 103752 R JMS TWS
05000 R 065521 E DAC* .AA
05001 R 200132 R LAC BEXP /PRINT SIGN
05002 R 750100 A SMA:CLA
05003 R 777776 A LAW 253-255
05004 R 340420 R TAD SMIN
PRINTC
05005 R 103362 R GEN* JMS XOUTL
05006 R 165522 E DZM* .AB /CLEAR COUNT
05007 R 225521 E LAC* .AA
05010 R 465522 E ISZ* .AB
05011 R 340213 R TAD M144
05012 R 740100 A SMA
05013 R 605010 R JMP .-3
05014 R 340212 R TAD C144
05015 R 065521 E DAC* .AA /SAVE TENS AND UNITS
05016 R 750001 A CLA:CMA /OUTPUT HUNDREDS
05017 R 365522 E TAD* .AB
05020 R 740200 A SZA /UNLESS ZERO
05021 R 105036 R JMS OUTDG
05022 R 225521 E LAC* .AA /PRINT TWO DIGITS
05023 R 103101 R JMS PRNT
05024 R 624672 R JMP* FLOUTP
05025 R 750001 A FG07 CLA:CMA /IGNORE FIRST DIGIT
05026 R 340132 R TAD BEXP /SUBTRACT 1 FROM
05027 R 040132 R DAC REXP /DECIMAL EXPONENT
05030 R 225522 E LAC* .AB
05031 R 741200 A SNA /IS MANTISSA ZERO?
05032 R 365523 E TAD* .AC
05033 R 751200 A SNA:CLA
05034 R 040132 R DAC BEXP /YES:EXP=0
05035 R 604760 R JMP FG06
////
05036 R 000000 A OUTDG 0 /OUTPUT ONE DIGIT
05037 R 340152 R TAD C260

```

05040 R 103362 R GEN*
05041 R 625036 R

PRINTC
JMS XOUTL
JMP* OUTDG
.EJECT

```

05042 R 000000 A
05043 R 750200 A
05044 R 104662 R
05045 R 540416 R
05046 R 605044 R
05047 R 104524 R
05050 R 200061 R
05051 R 342243 R
05052 R 750200 A
05053 R 605061 R
05054 R 104662 R
05055 R 140131 R
05056 R 104550 R
05057 R 200131 R
05060 R 103752 R
05061 R 040133 R
05062 R 225523 E
05063 R 741100 A

/FLOATING POINT INPUT
FLINTP 0
      SZA:CLA
      JMS INPUT
      SAD C240
      JMP .-2
      JMS DECONV
      LAC CHAR
      TAD MPER
      SZA:CLA
      JMP FIG01
      JMS INPUT
      OZM ONUMBR
      JMS DECON
      LAC ONUMBR
      JMS TWOS
      DAC SEXP
      LAC* .AC
      SPA

/IF C(AC) = 0, USE CHAR
/IF C(AC) NON-ZERO, GET NEXT
/GET FIRST CHARACTER
/IGNORE LEADING SPACES
/READ FIRST DIGIT GROUP
/ENDED BY PERIOD?
/YES, READ 2ND GROUP
/NUMBER OF DIGITS IN SEXP
/NO

FIG01 DAC SEXP

/FORMERLY A LAC* .AB SZA SEQUENCE WHICH ALLOWED OVERFLOW INTO
/SIGN BIT
ERROR 27
      .DEC
      JMP ERR2-27
      LAC* .AC
      DAC* .AB
      LAC OVER2
      DAC* .AC
      LAC P43
      DAC* .AA
      JMS* .CD
      ISZ ISIGN
      JMS* .BA
      JMS* .AP
      PT1+400000
      LAW -305
      TAD CHAR
      SZA:CLA
      JMP ENDFIX
      JMS INPUT
      JMS DECONV
      LAC OVER2
      ISZ ISIGN
      JMS TWOS
      TAD SEXP
      DAC SEXP
      ENDFIX JMS* .AO
      PT1+400000
      ENDFIX LAC SEXP
      SNA
      JMP* FLINTP
      SMA:CLA
      JMP FIG04

/END OF FLOATING POINT INPUT
/COMPENSATE FOR DECIMAL EXPONENTS
/RESTORE MANTISSA
/TEST DECIMAL EXPONENT
/YES - CONVERT DECIMAL EXPONENT
/DECIMAL POINT IS
/C(SEXP)PLACES TO RIGHT
/OF LAST DIGIT

```



```

05122 R 105140 R
05123 R 400214 R
05124 R 620072 R
05125 R 000000 A
05126 R 200167 R
05127 R 605135 R
05130 R 105140 R
05131 R 300214 R
05132 R 620072 R
05133 R 000000 A
05134 R 750001 A
05135 R 340133 R
05136 R 040133 R
05137 R 605115 R
    
```

FIG04

FIG04X

```

JMS      FINT      /. IS TO THE LEFT:
FDIV     TEN       /TIMES .1000
FPUT*    PT1
FEXT
LAC      C1
JMP      FIG04X
JMS FINT      /. IS TO THE RIGHT:
FMPY TEN /MULTIPLY BY 10
FPUT*    PT1
FEXT
CLA!CMA
TAD      SEXP
DAC      SEXP
JMP      ENDFIX
.EJECT
    
```

```

005140 R
05140 R 000000 A
05141 R 225140 R
05142 R 445140 R
05143 R 741200 A
05144 R 625140 R
05145 R 040134 R
05146 R 500221 R
05147 R 040136 R
05150 R 205140 R

05151 R 505574 R

05152 R 240136 R
05153 R 040136 R
05154 R 200220 R
05155 R 500134 R
05156 R 751200 A
05157 R 605162 R
05160 R 220136 R
05161 R 040136 R
05162 R 200136 R
05163 R 500221 R
05164 R 741200 A
05165 R 605201 R
05166 R 200136 R
05167 R 040135 R
05170 R 200207 R
05171 R 065573 R
05172 R 201120 R
05173 R 040054 R
05174 R 220135 R
05175 R 060014 A
05176 R 440135 R
05177 R 440054 R
05200 R 605174 R
05201 R 140036 R
05202 R 200134 R
05203 R 742010 A
05204 R 742010 A
05205 R 500222 R
05206 R 340223 R
05207 R 045211 R
05210 R 200136 R
05211 R 000000 A
    
```

/BASIC FLOATING-POINT INTERPRETER.

```

FINT=
FPNT 0
LAC* FPNT /GET NEXT INSTRUCTION
ISZ FPNT
SNA
JMP* FPNT
DAC JUMP
AND MASK7 /GET 13 BIT ADDRESS
DAC ADDR
LAC FPNT
.IFDEF POP15
AND (70000 /MASK BK + PG BITS
.ENDC
.IFUND POP15
AND (60000 /MASK BK BITS
.ENDC
XOR ADDR
DAC ADDR
LAC INDRCT /INDIRECT BIT=1?
AND JUMP
SNA!CLA
JMP LOOP01 /NO-GO ON
LAC* ADDR /YES ,DEFER ,W/O AUTO-INDEX
DAC ADDR
LAC ADDR
AND MASK7
SNA
JMP FNULL
LAC ADDR
DAC JUMP2
LAC CEX1 /SAVE FLOATING ARGUMENT
DAC* (FLT XR
LAC HFLT
DAC CNTR
LAC* JUMP2
DAC* FLTXR
ISZ JUMP2
ISZ CNTR
JMP -4
FNULL DZM OVER1
LAC JUMP /GET COMMAND
RTL
RTL
AND C7 /GET BITS 0-2, IE OPCODE
TAD TABLE /LOOKUP IN TABLE
DAC +2
LAC ADDR
XX /GO THERE
.EJECT
    
```

LOOP01

FNULL

```

05212 R 045214 R      //
05213 R 125525 E      FLPT   DAC .+2
05214 R 000000 A      JMS* .AP           /STORE
05215 R 605141 R      XX
05216 R 045220 R      FLGT   JMP FPNT+1
05217 R 125524 E      DAC .+2
05220 R 000000 A      JMS* .AQ           /LOAD
05221 R 605141 R      XX
05222 R 045224 R      FLSU   JMP FPNT+1
05223 R 125527 E      DAC .+2
05224 R 000000 A      JMS* .AR           /SUBTRACT
05225 R 605141 R      XX
05226 R 045230 R      FLAD   JMP FPNT+1
05227 R 125526 E      DAC .+2
05230 R 000000 A      JMS* .AQ           /ADD
05231 R 605141 R      NORF   JMP FPNT+1
05232 R 125536 E      JMS* .CD           /NORMALIZE
05233 R 605141 R      JMP FPNT+1

05234 R 045310 R      //
05235 R 340167 R      EXITF  DAC      X.BH      /POWER
05236 R 040104 R      TAD      C1
05237 R 225522 E GEN*  DAC      BOX
05240 R 740200 A      GETSGN
05241 R 605245 R      LAC*     .AB
05242 R 165521 E      SZA
05243 R 165523 E      JMP      EXITFZ
05244 R 605141 R      DZM* .AA
05245 R 750100 A      DZM* .AC
05246 R 605251 R      EXITFZ  JMP      FPNT+1
05247 R 125534 E      SMA!CLA  /IS BASE NEGATIVE
05250 R 045372 R      JMP      .+3      /NO
05251 R 750001 A      JMS*     .BA      /YES
05252 R 205140 R      CLA!CMA
05253 R 101055 R GEN*  DAC      SAVPOW   /SAVE SIGN OF BASE
05254 R 105140 R      LAC      FPNT
05255 R 600137 R      PUSHA   /SAVE INTERPRETER ENTRY
05256 R 525310 R      JMS     XPUSHA
05257 R 000000 A      JMS     FPNT     /SAVE BASE AND GET POWER
05260 R 105343 R      FPUT    XY
05261 R 040037 R      FGET*   X.BH
05262 R 125527 E      FXIT
05263 R 405310 R      JMS     FIX      /GET INTEGER POWER
05264 R 225522 E GEN*  DAC      OVER2
05265 R 744202 A      JMS*     .AR
05266 R 744000 A      X.BH+400000 /SEE IF INTEGER
05267 R 205372 R      GETSGN
05270 R 740600 A      LAC*     .AB
05271 R 603441 R GEN*  SZA!CLL!CML /L=1 IF INTEGER POINTER
05272 R 740600 A      CLL      /L=0 IF NON-INTEG
05273 R 205372 R      LAC      SAVPOW
05274 R 740600 A      SNL!SZA /NON-INTEG POWER OF
05275 R 603441 R GEN*  ERROR    20      /NEG NUMBER
05276 R 603441 R GEN*  .DEC
05277 R 603441 R GEN*  JMP      ERR2-28

```

05272	R	200037	R	LAC	OVER2	/CHECK POWER
05273	R	740100	A	SMA		/POS OR NEG?
05274	R	605302	R	JMP	EXITFQ	/POS-
05275	R	125524	E	JMS*	.AO	/NEG - CHANGE SIGN
05276	R	405310	R	X.RH+400000		/WITH GET.NEGATE.PUT
05277	R	125534	E	JMS*	.BA	
05300	R	125525	E	JMS*	.AP	
05301	R	405310	R	X.RH+400000		
05302	R	220104	R	EXITFQ	LAC*	BOX
05303	R	741200	A	SNA		
05304	R	605331	R	JMP	ZERPOW	
05305	R	125524	E	JMS*	.AO	/GET BASE
05306	R	000137	R	XY		
05307	R	125535	E	JMS*	.BH	
05310	R	000000	A	X.BH	XX	
05311	R	205372	R	LAC	SAVPOW	/GET SIGN OF BASE
05312	R	740100	A	SMA		
05313	R	605320	R	JMP	EXITFP	/POS BASE
05314	R	200037	R	LAC	OVER2	/GET POWER
05315	R	740020	A	RAR		/PUT ODD OR EVEN IN LINK
05316	R	741400	A	SZL		
05317	R	125534	E	JMS*	.BA	/CHANGE SIGN
05320	R	200037	R	EXITFP	LAC	OVER2
05321	R	740100	A	SMA		
05322	R	605333	R	JMP	EXITFX	/NO CHANGE
05323	R	105140	R	JMS	FPNT	/DIVIDE ANSWER INTO ONE
05324	R	600137	R	FPUT	XY	
05325	R	500167	R	FGET	FLTONE	
05326	R	400137	R	FDIV	XY	
05327	R	000000	A	FXIT		
05330	R	605333	R	JMP	EXITFX	
05331	R	125524	E	ZERPOW	JMS*	.AO
05332	R	000167	R	EXITFX	FLTONE	/ANY BASE TO ZERO POWER
05333	R			GEN*	POPA	/IS ONE
05333	R	220013	A	LAC*	PDLXR	
05334	R	045140	R	DAC	FPNT	
05335	R	125536	E	JMS*	.CD	
05336	R	605141	R	JMP	FPNT+1	
				/////		
05337	R	045341	R	FLMY	DAC	+.2
05340	R	125530	E	JMS*	.AS	/MULTIPLY
05341	R	000000	A	XX		
05342	R	605141	R	JMP	FPNT+1	
				/ FORM AN	INTEGER FROM C(FLAC-FLAC+1)	
05343	R	000000	A	FIX	0	/VIA (INTEGER)
05344	R	125533	E	JMS*	.AX	/FIX
05345	R	040037	R	DAC	OVER2	
05346	R	125532	E	JMS*	.AH	/FLOAT
05347	R	200037	R	LAC	OVER2	
05350	R	140037	R	DZM	OVER2	
05351	R	625343	R	JMP*	FIX	
05352	R	045361	R	FLDV	DAC	FLDVX
05353	R	440136	R	ISZ	ADDR	
05354	R	220136	R	LAC*	ADDR	
05355	R	505575	R	AND		(377777)

05356 R 741200 A
GEN*
05357 R 603440 R GEN*
05360 R 125531 E
05361 R 000000 A FLOVX
05362 R 605141 R

SNA
ERROR 29 /DIVISION BY ZERO
.DEC
JMP ERR2-29
JMS* .AT /DIVIDE
XX
JMP FPNT+1
.EJECT

```

/FLOATING SQUARE ROOT FUNCTION
05363 R          XSQRT GETSGN
05363 R 225522 E GEN* LAC*      .AB
05364 R 751100 A      SPA:CLA  30      /NUMBER IS NEGATIVE=IMAGINARY ROOTS
          GEN*      .DEC
05365 R 603437 R GEN* JMP      ERR2-30
05366 R 125520 E      JMS*     DSQRT
05367 R 605371 R      JMP      .+2
05370 R 000076 R      FLARG
          RETURN
05371 R 602517 R GEN* JMP      EFUN3
/
/OTS ERROR ROUTINE
/
05372 R 005372 R SAVPOW=.
05372 R 000000 A .ER      0
          ERROR      36
          GEN*      .DEC
05373 R 603431 R GEN* JMP      ERR2-36
          .EJECT

```

/UNPACK INITIALIZATION

05374 R 000000 A
 05375 R 040041 R
 05376 R 777777 A
 05377 R 040042 R
 05400 R 625374 R

.GETI 0
 DAC GETP
 LAW -1
 DAC GETCX
 JMP* .GETI

/POINTER IN AC
 /INIT POINTER
 /INIT COUNTER

/UNPACK

05401 R 000000 A
 05402 R 440042 R
 05403 R 605414 R
 05404 R 220041 R
 05405 R 440041 R
 05406 R 040043 R
 05407 R 220041 R
 05410 R 440041 R
 05411 R 040044 R
 05412 R 777773 A
 05413 R 040042 R
 05414 R 777770 A
 05415 R 040045 R
 05416 R 200044 R
 05417 R 740010 A
 05420 R 440045 R
 05421 R 605424 R
 05422 R 500155 R
 05423 R 625401 R
 05424 R 040044 R
 05425 R 200043 R
 05426 R 740010 A
 05427 R 040043 R
 05430 R 605416 R

.GET 0
 ISZ GETCX
 JMP GET4
 LAC* GETP
 ISZ GETP
 DAC GET1X
 LAC* GETP
 ISZ GETP
 DAC GET2
 LAW -5
 DAC GETCX
 LAW -10
 DAC GET3
 LAC GET2
 RAL
 ISZ GET3
 JMP .+3
 AND P177
 JMP* .GET
 DAC GET2
 LAC GET1X
 RAL
 DAC GET1X
 JMP GET5

/WORD PAIR STARTED
 /NEED NEXT PAIR
 /FIRST PART
 /LAST PART
 /RESET CHAR COUNT
 /SHIFT LOOP 7 +1/2 TIMES
 /DOUBLE AC ROTATE LOOP
 /KEEP LOOPING
 /GOT CHARACTER
 /EXIT WITH CHARACTER IN AC

/PACK INITIALIZATION

05431 R 000000 A
 05432 R 040046 R
 05433 R 140047 R
 05434 R 625431 R

.PUTI 0
 DAC PUTP
 D2M PUTC
 JMP* .PUTI

/POINTER IN AC
 /INIT POINTER
 /INIT COUNTER

/PACK

05435 R 000000 A
 05436 R 500155 R
 05437 R 040050 R
 05440 R 744000 A
 05441 R 200047 R
 05442 R 345576 R
 05443 R 045445 R
 05444 R 200050 R
 05445 R 000000 A
 05446 R 005453 R

.PUT 0
 AND P177
 DAC PUT6
 CLL
 LAC PUTC
 TAD (JMP* PUT7
 DAC .+2
 LAC PUT6
 XX
 PUT7 PUT1

/CHARACTER IN AC
 /SAVE CHARACTER
 /CHARACTER POSITION
 /BUILD DISPATCH
 /GET CHARACTER
 /MODIFIED JMP. PUT7
 /CHAR1

05447	R	005461	R		PUT2		/CHAR2
05450	R	005464	R		PUT3		/CHAR3
05451	R	005477	R		PUT4		/CHAR4
05452	R	005502	R		PUT5		/CHAR5
05453	R	742020	A	PUT1	RTR		/8 RIGHT
05454	R	742020	A		RTR		
05455	R	742020	A		RTR		
05456	R	742020	A		RTR		
05457	R	160046	R	PUT8	DZM*	PUTP	/CLEAR DATA WORD
05460	R	605504	R		JMP	PUT9	
05461	R	742010	A	PUT2	RTL		/4 LEFT
05462	R	742010	A		RTL		
05463	R	605504	R		JMP	PUT9	
05464	R	742020	A	PUT3	RTR		/3 RIGHT - 1ST HALF
05465	R	740020	A		RAR		
05466	R	500144	R		AND	P17	/4 BITS ONLY
05467	R	260046	R		XOR*	PUTP	/FINISH 1ST WORD OF PAIR
05470	R	060046	R		DAC*	PUTP	
05471	R	440046	R		ISZ	PUTP	/TO LAST WORD OF PAIR
05472	R	200050	R		LAC	PUT6	/GET CHARACTER
05473	R	742020	A		RTR		/4 RIGHT - 2ND HALF
05474	R	742020	A		RTR		
05475	R	505552	R		AND	(700000	/3 BITS ONLY
05476	R	605457	R		JMP	PUT8	
05477	R			PUT4	RTL6		/8 LEFT
05477	R	100770	R GEN*		JMS	XRTL6	
05500	R	742010	A		RTL		
05501	R	605504	R		JMP	PUT9	
05502	R	140047	R	PUT5	DZM	PUTC	/RESET 5/7 COUNTER
05503	R	741010	A		SKP:RAL		/1 LEFT
05504	R	440047	R	PUT9	ISZ	PUTC	/TO NEXT CHARACTER
05505	R	260046	R		XOR*	PUTP	
05506	R	060046	R		DAC*	PUTP	/MERGE INTO WORD PAIR
05507	R	200047	R		LAC	PUTC	
05510	R	741200	A		SNA		
05511	R	440046	R		ISZ	PUTP	/2ND WORD COMPLETE
05512	R	625435	R		JMP*	.PUT	/RETURN
		003756	R		.END BEGIN		
05513	R	005513	E *ETV				
05514	R	005514	E *ETV				
05515	R	005515	E *ETV				
05516	R	005516	E *ETV				
05517	R	005517	E *ETV				
05520	R	005520	E *ETV				
05521	R	005521	E *ETV				
05522	R	005522	E *ETV				
05523	R	005523	E *ETV				
05524	R	005524	E *ETV				
05525	R	005525	E *ETV				
05526	R	005526	E *ETV				
05527	R	005527	E *ETV				
05530	R	005530	E *ETV				
05531	R	005531	E *ETV				
05532	R	005532	E *ETV				
05533	R	005533	E *ETV				

05534	R	005534	E	*ETV
05535	R	005535	E	*ETV
05536	R	005536	E	*ETV
05537	R	005537	E	*ETV
05540	R	000015	A	*LIT
05541	R	000016	A	*LIT
05542	R	000013	A	*LIT
05543	R	000777	A	*LIT
05544	R	000010	A	*LIT
05545	R	000011	A	*LIT
05546	R	000012	A	*LIT
05547	R	001570	R	*LIT
05550	R	601575	R	*LIT
05551	R	770000	A	*LIT
05552	R	700000	A	*LIT
05553	R	600000	A	*LIT
05554	R	000351	R	*LIT
05555	R	000005	A	*LIT
05556	R	000006	A	*LIT
05557	R	031000	A	*LIT
05560	R	001000	A	*LIT
05561	R	002000	A	*LIT
05562	R	603632	R	*LIT
05563	R	000100	A	*LIT
05564	R	000103	A	*LIT
05565	R	000102	A	*LIT
05566	R	000003	A	*LIT
05567	R	000002	A	*LIT
05570	R	003213	R	*LIT
05571	R	077777	A	*LIT
05572	R	000001	A	*LIT
05573	R	000014	A	*LIT
05574	R	060000	A	*LIT
05575	R	377777	A	*LIT
05576	R	625446	R	*LIT

NO ERROR LINES

FOCAL	PAGE 105		FOCAL	PAGE 106	FOCAL	PAGE 107	FOCAL	PAGE 108	
AC1H	00034	R	C242	00450	R FNUM	02417	R FINDN	02700	R
AC1L	00035	R	C250	00424	R EPAR	02376	R FINFIN	01507	R
ADD	00056	R	C251	00427	R FPAR2	02477	R FINT	005140	R
ADDR	00136	R	C252	00422	R FRASE	02633	R FISW	00040	R
ALIST	00447	R	C254	00432	R FRG	02657	R FIX	05343	R
ALLCM1	00316	R	C255	00420	R ERL	02650	R FLAC1	002360	R
ALLCM2	00322	R	C260	00152	R ERROR	MACRO	FLAC2	002420	R
ANYMAT	02045	R	C273	00433	R ERRPCX	03475	R FLAC3	002427	R
ARGNXT	02411	R	C275	00435	R FRR2	003475	R FLAD	05226	R
ARRAYN	00142	R	C277	00145	R ERR2CT	00106	R FLARG	00076	R
ARTN	04315	R	C305	00414	R FRR2T	03512	R FLARGP	00160	R
ASK	01603	R	C306	00163	R FRVX	02672	R FLARG2	00101	R
ATLIST	00436	R	C314	00164	R ER2T	00105	R FLOV	05352	R
ATSW	00053	R	C375	00475	R ESCA	03620	R FLDVX	05361	R
AXIN	000010	A	C377	00500	R ETERM	02335	R FLGT	05216	R
AXOUT	00014	R	C7	00222	R ETERMN	02332	R FLIMIT	01447	R
BACK	04435	R	C77	00151	R ETERM1	02306	R FLINTP	05042	R
REGIN	03756	R	DATAN	05513	E ETERM2	02342	R FLIST1	00466	R
REXP	00132	R	DCONT	01042	R FVAL	02272	R FLIST2	00463	R
RKI	000003	A	DCOS	05514	E EXITF	05234	R FLMY	05337	R
BKO	000005	A	DCOUNT	04770	R EXITFP	05320	R FLOG	04321	R
BLKIN	00030	R	DEBGSW	00067	R EXITFQ	05302	R FLOP	02362	R
RLKOUT	00031	R	DECON	04550	R EXITFX	05333	R FLOPX	04502	R
ROTTOM	00000	R	DECONV	04524	R EXITFZ	05245	R FLOUTP	04672	R
ROX	00104	R	DECP	00055	R EXTR	03011	R FLPT	05212	R
RUFFER	004046	R	DECR	04453	R EX1	00033	R FLSU	05222	R
RUF	00004	R	DELETE	02550	R FADD	100000	A FLTONE	00167	R
RUFST	00211	R	DEXP	05515	E FCONT	01453	R FLT XR	000014	A
RUFSTX	00001	R	DGRP	01003	R FCOS	04334	R FLTZER	00171	A
CCR	000501	R	DGRP1	01012	R FCOUNT	00124	R FMPY	300000	A
CEX1	00207	R	DIG	04465	R FDIV	400000	A FMUL	300000	A
CFRS	00157	R	DIGIT	00127	R FEND2	02721	R FNOR	700000	A
CFRSX	00162	R	DIGITS	000011	A FEND3	02722	R FNTABE	000376	R
CHAR	00061	R	DIGP5	04476	R FETVAR	02110	R FNTABF	00351	R
CLISTX	00471	R	DLOG	05516	E FEXP	04272	R FNULL	05201	R
CNTR	00054	R	DMPSW	00070	R FEXT	000000	A FOCAL9	00177	R
COMARY	01534	R	DNUMBR	00131	R FGET	500000	A FOR	01407	R
COMBOT	00010	R	DO	00776	R FG02	04707	R FPNT	05140	R
COMBUF	00007	R	DOK	02566	R FG03	04723	R FPOPJ	001501	R
COMDEC	01570	R	DONE	02605	R FG03M2	04720	R FPOW	000000	A
COMEIN	003756	R	OSAVE	04556	R FG04	04727	R FPRNT	04416	R
COMEND	01575	R	OSIN	05517	E FG05	04753	R FPUSHJ	001460	R
COMLST	00225	R	OSORT	05520	E FG05A	04737	R FPUT	600000	A
COMMEN	001210	R	DTST	04565	R FG06	04760	R FRST	00113	R
COMMON	01513	R	DUBLAD	04641	R FG07	05025	R FRSTCV	00005	R
COMMXX	01524	R	EFOP	000053	R FIG01	05061	R FSIN	04340	R
COMOUT	004066	R	EFUN	02431	R FIG04	05130	R FSUB	200000	A
C1	000167	R	EFUN3	02517	R FIG04X	05135	R FUNAME	00137	R
C100	00150	R	ELPAR	02475	R FILEXT	00161	R FUNCHK	02455	R
C140	00175	R	ENDCR	00020	R FILE01	004140	R FUNCTR	00140	R
C144	00212	R	ENDFI	05113	R FILE02	004173	R FUNERR	03515	R
C200	00154	R	ENDFIX	05115	R FILE03	004125	R FUNFND	02473	R
C212	00477	R	ENDLN	MACRO	R FINCR	01433	R FUNLOP	02463	R
C215	00501	R	ENDMAT	02060	R FINCRX	01437	R FUNPTR	00141	R
C240	00416	R	ENDT	00002	R FINDLN	MACRO	R FXIT	000000	A

FOCAL	PAGE 109	FOCAL	PAGE 110	FOCAL	PAGE 111	FOCAL	PAGE 112				
GENDX	03036	R	ISIGN	00130	R	M100	00153	R	PUT4	05477	R
GETARG	02135	R	ITABLE	00404	R	M11	02537	R	PUT5	05502	R
GETC	MACRO		JUMP	00134	R	M12	004770	R	PUT6	00050	R
GETCX	00042	R	JUMP2	00135	R	M137	00165	R	PUT7	05446	R
GETLN	MACRO		LASTCV	000003	R	M140	00176	R	PUT8	05457	R
GETP	00041	R	LASTLN	00066	R	M144	00213	R	PUT9	05504	R
GETSGN	MACRO		LASTOP	00052	R	M2	01367	R	P13	00143	R
GETVAP	02116	R	LASTV	00006	R	NAGSW	00060	R	P17	00144	R
GETVAR	02141	R	LBCLOS	04177	R	NORF	05232	R	P177	00155	R
GETVAX	02130	R	LBFIL	04234	R	NOTMAT	02077	R	P2	00147	R
GETVCT	00021	R	LBIEND	04260	R	ONE	01034	R	P3	00146	R
GET1	03026	R	LBIN	04105	R	OP	00107	R	P337	00166	R
GET1X	00043	R	LBINLP	04143	R	OPNEXT	02301	R	P40	00174	R
GET2	00044	R	LBIN01	04114	R	OPTABL	00224	R	P43	00217	R
GET3	00045	R	LBIN02	04136	R	OPTABS	00376	R	P7600	01347	R
GET4	05414	R	LBIN03	04266	R	OTBF01	003414	R	RANPT	00112	R
GET5	05416	R	LBIN1A	04123	R	OTBUFP	00013	R	RCAL01	03327	R
GEXIT	00755	R	LBKILL	04205	R	OTEMP	000036	R	RCAL02	03333	R
GFND1	02221	R	LBOU	04157	R	OUTA	04460	R	RCAL03	03310	R
GINC	00156	R	LBOU2	04214	R	OUTBUF	03211	R	RDEOM	04152	R
GLIST	00454	R	LBOU1	04165	R	OUTDG	05036	R	READC	MACRO	
GONE	00612	R	LBOU2	04171	R	OVER1	00036	R	RECOVR	03521	R
GOTO	01176	R	LBOU3	04202	R	OVER2	00037	R	RECVR	003527	R
GRPTST	01343	R	LBOU4	04210	R	PACBUF	03562	R	REMAIN	00126	R
GS1	02161	R	LBTEXT	04224	R	PACBXT	03611	R	RET	04402	R
GS1A	02160	R	LBTEXT	04227	R	PACKC	MACRO		RETURN	MACRO	
GS2	02174	R	LBRIT	04220	R	PACKST	00071	R	RETURX	02262	R
GS3	02163	R	LIBCMD	00332	R	PC	00063	R	RET1	04413	R
GS4	02171	R	LIBRAR	04067	R	PCHK	01067	R	RND2	00210	R
GTEM	00016	R	LIBRSW	00032	R	PCKTB	03632	R	ROT	03652	R
HSPX	04155	R	LINENO	00062	R	PCK1	03623	R	RTL6	MACRO	
HTEMP	000034	R	LISTG0	00510	R	PC1	01210	R	R6	04372	R
IBAR	00575	R	LIST3	000501	R	PDLXR	000013	R	SACH	00075	R
IBARX	00571	R	LIST6	00474	R	PD2	01113	R	SAVEOT	00022	R
IF	01356	R	LITX	01573	R	PD3	01132	R	SAVPOW	005372	R
IF1	01405	R	LOOP01	05162	R	PER	00415	R	SBAR	01744	R
IF3	01375	R	LPOPJ	001300	R	PLCE	000124	R	SCHAR	01731	R
IGNOR	00602	R	LPROCS	001037	R	POPA	MACRO		SCONT	01724	R
ILIST	00460	R	LPRTST	02536	R	POPF	MACRO		SCONTX	01726	R
IMAC	03306	R	LTEMP	000035	R	POPJ	MACRO		SCOUNT	00123	R
IMAGER	03307	R	MASK7	00221	R	PRINTC	MACRO		SCRUB	02003	R
IMAGEN	03273	R	MD	04505	R	PRNT	03101	R	SET	001407	R
IMBF01	003300	R	MFLT	01120	R	PRNTLN	MACRO		SETRCL	03553	R
IMBF02	003312	R	MINUSA	004565	R	PROC	001204	R	SETHCL	03544	R
IMRUFF	03123	R	MODBF1	000134	R	PROCES	01203	R	SEX	02035	R
IMBUFP	00011	R	MODBF2	000135	R	PT1	00072	R	SEXC	01337	R
IN	04445	R	MODBUF	00017	R	PUSHA	MACRO		SEXP	00133	R
INBF01	003331	R	MODIFY	01705	R	PUSHF	MACRO		SFOUND	01750	R
INRUF	03127	R	NORMAT	02050	R	PUSHJ	MACRO		SIGN2	00122	R
INRUF2	00012	R	MOVCOM	01146	R	PUTC	00047	R	SMIN	00420	R
INDRCT	00220	R	MOVUPX	01163	R	PUTCNT	00023	R	SORTB	02013	R
INPUT	04662	R	MPER	02243	R	PUPP	00046	R	SORTC	MACRO	
INPUTX	00640	R	MULT10	04600	R	PUT1	05453	R	SORTCN	00051	R
INSUB	00025	R	MULT2	04623	R	PUT2	05461	R	SORTJ	MACRO	
IRETN	00607	R	M1	00614	R	PUT3	05464	R	SORTJX	MACRO	

FOCAL	PAGE 113	FOCAL	PAGE 114	FOCAL	PAGE 115			
SPNOR	MACRO	WCAL01	03412	R	.AO	05524	E	
SRETN	01762	R	WCAL02	03422	R	.AP	05525	E
SRNLST	00503	R	WCAL03	03276	R	.AO	05526	E
START	00512	R	WCAL04	03523	R	.AR	05527	E
START0	00545	R	WORDS	000003	A	.AS	05530	E
STARTC	00541	R	WRITE	01234	R	.AT	05531	E
STARTL	00531	R	WRITED	01276	R	.AH	05532	E
STARTQ	00543	R	WTESTG	01271	R	.AX	05533	E
STARTV	00003	R				.BA	05534	E
STARTZ	00547	R	WTFST2	01255	R	.BH	05535	E
SURS	002233	R	WX	01273	R	.CD	05536	E
TABLE	00223	R	XABS	02514	R	.ER	05372	R
TASK	01606	R	XCTIN	00057	R	.GET	05401	R
TASK4	01657	R	XCTX	00015	R	.GETI	05374	R
TCRLF	01655	R	XENDLN	03051	R	.NEWF	05537	E
TCRLF2	01661	R	XFIND	02675	R	.PUT	05435	R
TDUMP	03657	R	XGETLN	00664	R	.PUTI	05431	R
TDUMPC	03725	R	XINT	02506	R	.RLXIT	MACRO	
TDUMPX	03716	R	XI33	03322	R	.SCOM	000100	A
TEMPK	00024	R	XI33NX	03345	R			
TEMPO	00125	R	XOUTL	03362	R			
TEN	00214	R	XOUTLI	03404	R			
TERMS	000416	R	XOUTLO	03377	R			
TESTA	00736	R	XOUTLX	003412	R			
TESTC	MACRO		XOUTLZ	03426	R			
TESTN	MACRO		XPOPJ	02265	R			
TEXTP	000014	R	XPRNT	03065	R			
TGO	04344	R	XPUSHA	01055	R			
TGRP2	01047	R	XPUSHJ	01100	R			
THISLN	00064	R	XRAN	03734	R			
THISOP	00065	R	XRT	000011	A			
TINTR	01670	R	XRTL6	00770	R			
TLIST	00455	R	XRT2	000012	A			
TLISTX	00456	R	XSBEGN	02644	R			
TQUOT	01645	R	XSGN	02510	R			
TR1	03602	R	XSORTC	01321	R			
TSTGRP	MACRO		XSORTX	02040	R			
TSTLPR	MACRO		XSPNOR	02233	R			
TTI	000776	A	XSQRT	05363	R			
TTIN	00026	R	XTESTC	01301	R			
TTO	000775	A	XTESTN	02242	R			
TTOUT	00027	R	XTESTX	01317	R			
TWOS	03752	R	XT3	01316	R			
TYPE	01604	R	XX	000000	A			
TYPE2	01636	R	XXX	04500	R			
T1	00073	R	XY	000137	R			
T2	00074	R	XYZ	03111	R			
UTE	02754	R	X.BH	05310	R			
UTO	02763	R	X15	000015	A			
UTRA	02752	R	X16	000016	A			
UTRAX	03007	R	YESMAT	02070	R			
UTX	03014	R	ZERPOW	05331	R			
UTXP6	03024	R	ZTESTN	02260	R			
VAL	003322	R	.AA	05521	E			
WALL	01266	R	.AP	05522	E			
			.A	05523	E			

FOCAL PAGE 116

ENDLN MACRO
 ERROR MACRO
 FINDLN MACRO
 GETC MACRO
 GETLN MACRO
 GETSGN MACRO
 PACKC MACRO
 POPA MACRO
 POPF MACRO
 POPJ MACRO
 PRINTC MACRO
 PRNTLN MACRO
 PUSHA MACRO
 PUSHF MACRO
 PUSHJ MACRO
 READC MACRO
 RETURN MACRO
 RTL6 MACRO
 SORTC MACRO
 SORTJ MACRO
 SORTJX MACRO
 SPNOR MACRO
 TESTC MACRO
 TESTN MACRO
 TSTGRP MACRO
 TSTLPR MACRO
 RLXIT MACRO
 BOTTOM 000000
 FEXT 000000
 FPOW 000000
 FXIT 000000
 XX 000000
 RUFSTX 000001
 ENDT 000002
 BKI 000003
 LASTCV 000003
 STARTV 000003
 WORDS 000003
 BUFR 000004
 BKO 000005
 FRSTCV 000005
 LASTV 000006
 COMBUF 000007
 AXIN 000010
 COMBOT 000010
 DIGITS 000011
 INBUFP 000011
 XRT 000011
 INBUFP 000012
 XRT2 000012
 OTBUFP 000013
 POLXR 000013
 AXOUT 000014
 FLTXR 000014
 TEXTP 000014

FOCAL PAGE 117

XCTX 00015
 X15 000015
 GTFM 00016
 X16 000016
 MODBUF 00017
 ENDCR 00020
 GETVCT 00021
 SAVEOT 00022
 PUTCNT 00023
 TEMPK 00024
 INSUR 00025
 TTIN 00026
 TTOUT 00027
 RLKIN 00030
 RLKOUT 00031
 LIBRSW 00032
 EX1 00033
 AC1H 00034
 HTEMP 000034
 AC1L 00035
 LTEMP 000035
 OTEMP 000036
 OVER1 00036
 OVER2 00037
 FISH 00040
 GETP 00041
 GETCX 00042
 GET1X 00043
 GET2 00044
 GET3 00045
 PUTP 00046
 PUTC 00047
 PUT6 00050
 R SORTCN 00051
 LASTOP 00052
 ATSW 00053
 EFOP 000053
 CNTR 00054
 DECP 00055
 ADD 00056
 XCTIN 00057
 NAGSW 00060
 CHAR 00061
 LINENO 00062
 PC 00063
 THISLN 00064
 THISOP 00065
 LASTLN 00066
 OEBGSW 00067
 OMPSW 00070
 R PACKST 00071
 PT1 00072
 T1 00073
 T2 00074
 R SACH 00075

FOCAL PAGE 118

FLARG 00076
 .SCOM 000100
 FLARG2 00101
 ROX 00104
 FR2T 00105
 R ERR2CT 00106
 OP 00107
 RANPT 00112
 R 00113
 SIGN2 00122
 R SCOUNT 00123
 R FCOUNT 00124
 R PLCE 000124
 R TEMP 00125
 R REMAIN 00126
 R DIGIT 00127
 R ISIGN 00130
 R DNUMBR 00131
 R REXP 00132
 R SEXP 00133
 R JUMP 00134
 R MOOBF1 000134
 R JUMP2 00135
 R MOOBF2 000135
 R ADOOR 00136
 R FUNAME 00137
 R XY 000137
 R FUNCTR 00140
 R FUNPTR 00141
 R ARRAYN 00142
 R P13 00143
 R P17 00144
 R C277 00145
 R P3 00146
 R P2 00147
 R C100 00150
 R C77 00151
 R C260 00152
 R M100 00153
 R C200 00154
 R P177 00155
 R GINC 00156
 R CFRS 00157
 R FLARGP 00160
 R FILEXT 00161
 R CFRSX 00162
 R C306 00163
 R C314 00164
 R M137 00165
 R P337 00166
 R C1 000167
 R FLTONE 00167
 R FLTZER 00171
 R P40 00174
 R C140 00175

FOCAL PAGE 119

M140 00176
 A FOCAL9 00177
 R CEX1 00207
 R RND2 00210
 R RUFST 00211
 R C144 00212
 R M144 00213
 R TEN 00214
 R P43 00217
 R INDRCT 00220
 R MASK7 00221
 R C7 00222
 R TABLE 00223
 R OPTABL 00224
 R COMLST 00225
 R ALLCM1 00316
 R ALLCM2 00322
 R LIBCMD 00332
 R FNTABF 00351
 R FNTABE 000376
 R OPTABS 00376
 R ITABLE 00404
 R C305 00414
 R PER 00415
 R C240 00416
 R TERMS 000416
 R C255 000420
 R SMIN 00420
 R C252 00422
 R C250 00424
 R C251 00427
 R C254 00432
 R C273 00433
 R C275 00435
 R ATLIST 00436
 R ALIST 00447
 R C242 00450
 R GLIST 00454
 R TLIST 00455
 R TLISTX 00456
 R ILIST 00460
 R FLIST2 00463
 R FLIST1 00466
 R CLISTX 00471
 R LIST6 00474
 R C375 00475
 R C212 00477
 R C377 00500
 R CCR 000501
 R C215 00501
 R LIST3 000501
 R SRNLST 00503
 R LISTG0 00510
 R START 00512
 R STARTL 00531

FOCAL	PAGE 120		FOCAL	PAGE 121		FOCAL	PAGE 122		FOCAL	PAGE 123	
STARTC	00541	R	FOR	01407	R	ZTESTN	02260	R	OUTBUF	03211	R
STARTQ	00543	R	SET	001407	R	RETURX	02262	R	IMAGEW	03273	R
STARTB	00545	R	FINCR	01433	R	XPOPJ	02265	R	WCAL03	03276	R
STARTZ	00547	R	FINCRX	01437	R	EVAL	02272	R	IMRF01	003300	R
IBARX	00571	R	FLIMIT	01447	R	OPNEXT	02301	R	IMAC	03306	R
IBAR	00575	R	FCONT	01453	R	ETERM1	02306	R	IMAGER	03307	R
IGNOR	00602	R	FPUSHJ	001460	R	FTERMN	02332	R	RCAL03	03310	R
IRETN	00607	R	FPOPJ	001501	R	FTFRM	02335	R	IMBF02	003312	R
GONE	00612	R	FINFIN	01507	R	ETERM2	02342	R	VAL	003322	R
M1	00614	R	COMMON	01513	R	FLAC1	002360	R	XI33	03322	R
INPUTX	00640	R	COMMX	01524	R	FLOP	02362	R	RCAL01	03327	R
XGETLN	00664	R	COMARY	01534	R	EPAR	02376	R	INBF01	003331	R
TESTA	00736	R	COMDEC	01570	R	ARGNXT	02411	R	RCAL02	03333	R
GEXIT	00755	R	LFTX	01573	R	ENUM	02417	R	XI33NX	03345	R
XRTL6	00770	R	COMEND	01575	R	FLAC2	002420	R	XOUTL	03362	R
TTO	000775	A	ASK	01603	R	FLAC3	002427	R	XOUTLQ	03377	R
DO	00776	R	TYPE	01604	R	EFUN	02431	R	XOUTLI	03404	R
TTI	000776	A	TASK	01606	R	FUNCHK	02455	R	WCAL01	03412	R
DGRP	01003	R	TYPE2	01636	R	FUNLOP	02463	R	XOUTLX	003412	R
DGRP1	01012	R	TQUOT	01645	R	FUNFND	02473	R	OTBF01	003414	R
ONE	01034	R	TCRLF	01655	R	ELPAR	02475	R	WCAL02	03422	R
LPROCS	001037	R	TASK4	01657	R	EPAR2	02477	R	XOUTLZ	03426	R
DCONT	01042	R	TCRLF2	01661	R	XINT	02506	R	ERRPCX	03475	R
TGRP2	01047	R	TINTR	01670	R	XSGN	02510	R	ERR2	003475	R
XPUSHA	01055	R	MODIFY	01705	R	XABS	02514	R	ERR2T	03512	R
PCHK	01067	R	SCONT	01724	R	EFUN3	02517	R	FUNERR	03515	R
XPUSHJ	01100	R	SCONTX	01726	R	LPRTST	02536	R	RECOVR	03521	R
PD2	01113	R	SCHAR	01731	R	M11	02537	R	WCAL04	03523	R
MFLT	01120	R	SBAR	01744	R	DELETE	02550	R	RECVR	003527	R
PD3	01132	R	SFOUND	01750	R	DDK	02566	R	SETWCL	03544	R
MOVCOM	01146	R	SRETN	01762	R	DONE	02605	R	SACTCL	03553	R
MOVUPX	01163	R	SCRUB	02003	R	ERASE	02633	R	PACBUF	03562	R
GOTO	01176	R	SORTB	02013	R	XSREGN	02644	R	TR1	03602	R
PROCES	01203	R	SEX	02035	R	ERL	02650	R	PACBXT	03611	R
PROC	001204	R	XSORTX	02040	R	FRG	02657	R	ESCA	03620	R
COMMEN	001210	R	ANYMAT	02045	R	FRVX	02672	R	PCK1	03623	R
PC1	01210	R	MORMAT	02050	R	XFIND	02675	R	PCKTB	03632	R
WRITE	01234	R	ENDMAT	02060	R	FINDN	02700	R	ROT	03652	R
WTEST2	01255	R	YESMAT	02070	R	FEND2	02721	R	TDUMP	03657	R
WALL	01266	R	NOTMAT	02077	R	FEND3	02722	R	TDUMPX	03716	R
WTESTG	01271	R	FETVAR	02110	R	UTRA	02752	R	TDUMPC	03725	R
WX	01273	R	GETVAP	02116	R	UTE	02754	R	XRN	03734	R
WRITED	01276	R	GETVAX	02130	R	UTO	02763	R	TWOS	03752	R
LPOPJ	001300	R	GETARG	02135	R	UTRAX	03007	R	BEGIN	03756	R
XTESTC	01301	R	GETVAR	02141	R	EXTR	03011	R	COMEIN	003756	R
XT3	01316	R	GS1A	02160	R	UTX	03014	R	RUFFER	004046	R
XTESTX	01317	R	GS1	02161	R	UTXP6	03024	R	COMOUT	004066	R
XSORTC	01321	R	GS3	02163	R	GET1	03026	R	LIBRAR	04067	R
SEX	01337	R	GS4	02171	R	XENDX	03036	R	LBIN	04105	R
GRPTST	01343	R	GS2	02174	R	XENDLN	03051	R	LBIN01	04114	R
P7600	01347	R	GFND1	02221	R	XPRNT	03065	R	LBIN1A	04123	R
IF	01356	R	SUBS	002233	R	PRNT	03101	R	FILE03	004125	R
M2	01367	R	XSPNOR	02233	R	XYZ	03111	R	LBIN02	04136	R
IF3	01375	R	XTESTN	02242	R	IMBUFF	03123	R	FILE01	004140	R
IF	01405	R	MPER	02243	R	INBUF	03127	R	LBINLP	04143	R

FOCAL	PAGE 123		FOCAL	PAGE 124		FOCAL	PAGE 125		FOCAL	PAGE 126	
OUTBUF	03211	R	RDEOM	04152	R	FG07	05025	R	.AP	05525	E
IMAGEW	03273	R	HSPX	04155	R	OUTOG	05036	R	.AQ	05526	F
WCAL03	03276	R	LBOU	04157	R	FLINTP	05042	R	.AR	05527	F
IMRF01	003300	R	LBOU1	04165	R	FIG01	05061	R	.AS	05530	F
IMAC	03306	R	LBOU2	04171	R	ENDFI	05113	R	.AT	05531	E
IMAGER	03307	R	FILE02	004173	R	ENDFIX	05115	R	.AW	05532	E
RCAL03	03310	R	LBCLOS	04177	R	FIG04	05130	R	.AX	05533	E
IMBF02	003312	R	LBOU3	04202	R	FIG04X	05135	R	.BA	05534	F
VAL	003322	R	LAKILL	04205	R	FINT	005140	R	.BH	05535	F
XI33	03322	R	LBOU4	04210	R	FPNT	05140	R	.CD	05536	E
RCAL01	03327	R	LBOU2	04214	R	LOOP01	05162	R	.NEWF	05537	E
INBF01	003331	R	LBWRIT	04220	R	FNULL	05201	R	FADD	100000	A
RCAL02	03333	R	LBTXT	04224	R	FLPT	05212	R	FSUB	200000	A
XI33NX	03345	R	LBTXX	04227	R	FLGT	05216	R	FMPY	300000	A
XOUTL	03362	R	LBFIL	04234	R	FLSU	05222	R	FMUL	300000	A
XOUTLQ	03377	R	LBIEND	04260	R	FLAD	05226	R	FDIV	400000	A
XOUTLI	03404	R	LBIN03	04266	R	NORF	05232	R	FGET	500000	A
WCAL01	03412	R	FEXP	04272	R	EXITF	05234	R	FPUT	600000	A
XOUTLX	003412	R	ARTN	04315	R	EXITFZ	05245	R	FNOR	700000	A
OT0F01	003414	R	FLOG	04321	R	EXITFQ	05302	R			
WCAL02	03422	R	FCOS	04334	R	X.BH	05310	R			
XOUTLZ	03426	R	FSIN	04340	R	EXITFP	05320	R			
ERRPCX	03475	R	TGO	04344	R	ZERPOW	05331	R			
ERR2	003475	R	R6	04372	R	EXITFX	05333	R			
ERR2T	03512	R	RET	04402	R	FLMY	05337	R			
FUNERR	03515	R	RET1	04413	R	FIX	05343	R			
RECOVR	03521	R	FPRNT	04416	R	FLOV	05352	R			
WCAL04	03523	R	BACK	04435	R	FLOVX	05361	R			
RECVR	003527	R	IN	04445	R	XSORT	05363	R			
SETNCL	03544	R	DECR	04453	R	SAVPOW	005372	R			
SETRCL	03553	R	OUTA	04460	R	.ER	05372	R			
PACBUF	03562	R	DIG	04465	R	.GETI	05374	R			
TR1	03602	R	DIGP5	04476	R	.GET	05401	R			
PACBXT	03611	R	XXX	04500	R	GET4	05414	R			
ESCA	03620	R	FLOPX	04502	R	GET5	05416	R			
PCK1	03623	R	MO	04505	R	.PUTI	05431	R			
PCKTB	03632	R	DECONV	04524	R	.PUT	05435	R			
ROT	03652	R	DECON	04550	R	PUT7	05446	R			
TDUMP	03657	R	NSAVE	04556	R	PUT1	05453	R			
TDUMPX	03716	R	DTST	04565	R	PUT8	05457	R			
TDUMPC	03725	R	MINUSA	004565	R	PUT2	05461	R			
XRAN	03734	R	MULT10	04600	R	PUT3	05464	R			
TWOS	03752	R	MULT2	04623	R	PUT4	05477	R			
BEGIN	03756	R	DUBLAD	04641	R	PUT5	05502	R			
COMEIN	003756	R	INPUT	04662	R	PUT9	05504	R			
RUFFER	004046	R	FLOUTP	04672	R	OATAN	05513	E			
COMOUT	004066	R	FG02	04707	R	DCOS	05514	E			
LIBRAR	04067	R	FG03M2	04720	R	NEXP	05515	E			
LBIN	04105	R	FG03	04723	R	NLOG	05516	E			
LBIN01	04114	R	FG04	04727	R	NSIN	05517	F			
LBIN1A	04123	R	FG05A	04737	R	NSQRT	05520	F			
FILE03	004125	R	FG05	04753	R	.AA	05521	F			
LBIN02	04136	R	FG06	04760	R	.AB	05522	F			
FILE01	004140	R	DCOUNT	04770	R	.AC	05523	E			
LBINLP	04143	R	#12	004770	R	.AO	05524	F			