

1C1 IDENT 7-17 TEMPORARY STORAGE AND EQU'S, COMPILER PART 1

\*  
\*TEMPORARY STORAGE \*\*\* ONE COPY PER USER

\* 2408

\$ACBBIT	ZR0	0	ACCEPT CHAR AND BRANCH BIT
\$ACBCHR	ZR0	0	
\$EOLTRG	ZR0	0	END OF LINE TRIGGER
\$FRSTCH	ZR0	0	FIRST CHAR READ BY NXACCH
\$INCXR	BSS	0	INCOMING X REGISTER
\$XITTP1	HLT	0	EXIT TEMP 1
\$INCAR	HLT	0	INCOMING A REGISTER
\$MWPTRG	HLT	0	MINUS WHILE PUNCHING TRIGGER
\$ITMPCT	HLT	0	INITIAL TEMP COUNT
\$WIMCH	HLT	0	WIMED CHAR
\$INDEX	ZR0	0	INPUT FILE INDEX, -1=NO FILE SPECIFIE
\$OUTDEX	ZR0	0	
\$INFLE	ZR0	0	INPUT FILE NO
\$OUTFLE	ZR0	0	OUTPUT FILE NO
\$CINF	ZR0	0	CMMD INPUT FILE
\$COUTF	ZR0	0	CMMD OUTPUT FILE
\$BP1	ZR0	0	BREAK POINT 1
\$BP2	ZR0	0	
\$FSTATS	ZR0	0	FORK STATUS WORD
\$TBLE	BSS	7	FORK TABLE
\$ICTTMP	ZR0	0	ICT POP TEMPOR
\$HOLIT1	ZR0	0	
\$HOLIT2	ZR0	0	
\$HOLIT3	ZR0	0	
\$HOLIT6	ZR0	0	
\$HOLIT9	ZR0	0	
\$VARBAS	ZR0	0	BASE ADDR OF DEBUG VARIABLE

\*FLAGS

\$FLC	ZR0	0	*1=LIST CODE (FOR DEBUGGING ONLY)
\$FCMDIN	ZR0	0	0=CMMD INCREMENT NEGATIVE
\$FDEBUG	ZR0	0	*1=PRODUCE DEBUG INFO FOR RUNTIME
\$FE0F	ZR0	0	*1=END OF FILE READ

\$FMAP ZR0 0  
\$FTTY0N ZR0 0

\*1=PRINT POST COMPILATION MAP  
\*1 = UNCONDITIONALLY TYPE ON TTY

\*LINKS  
\$LACB ZR0 0  
\$LNTERA ZR0 0  
\$LINITL ZR0 0  
\$LNXACC ZR0 0  
\$LNXINC ZR0 0  
\$LDBFXM ZR0 0  
\$LPONTR ZR0 0  
\$LFDLB ZR0 0  
\$LFEXTY ZR0 0  
\$LN0NES ZR0 0  
\$LTSTMT ZR0 0  
\$LFFF ZR0 0  
\$LCPYLS ZR0 0  
\$LNACS ZR0 0  
\$L0PC ZR0 0  
\$LRDYSC ZR0 0  
\$LPRINT ZR0 0  
\$LINITQ ZR0 0  
\$LSCAN ZR0 0  
\$LSERS ZR0 0  
\$LSELS ZR0 0  
\$LSTRED ZR0 0  
\$LQSNEX ZR0 0  
\$LTYTL ZR0 0  
\$LWI0PT ZR0 0  
\$LREASG ZR0 0  
\$LSTFIL ZR0 0  
\$CACBT1 ZR0  
\$CACBT2 ZR0  
\$LREAD ZR0 0

ACCEPT CHAR AND BRANCH

LIST FILE INDEX

\*MODIFIED INSTRUCTIONS EXECUTED HERE

\$A183A ZR0 0  
\$A184A ZR0 0  
\$A185A ZR0 0  
\$A187A ZR0 0  
\$A188A ZR0 0  
\$A232A ZR0 0  
\$A233A ZR0 0

\*

\*SYMBOLIC CENTRAL

\$SYMCEN ZR0 0  
\$CTL1 ZR0 0  
\$CTL2 ZR0 0  
\$SYMCN2 ZR0 0

CENTRAL 1  
CENTRAL 2

\*

\*

\*

\$FDLBCH ZR0 0  
\$FLDP1 ZR0 0  
\$FLDP2 ZR0 0  
\$FLDP3 ZR0 0  
\$FLDP31 ZR0 0  
\$FLDX ZR0 0  
\$FORKT1 ZR0 0  
\$FORKT2 ZR0 0  
\$S1INC ZR0 0  
\$S2INC ZR0 0  
\$SN1PT ZR0 0  
\$SN1PT1 ZR0 0  
\$SN2PT ZR0 0  
\$SN2PT1 ZR0 0  
\$SPNPT ZR0 0  
\$SPNPT1 ZR0 0  
\$BL0KCT ZR0 0  
\$SCANMD ZR0 0  
\$CRNTCH ZR0 0

MATCH CHAR

STATEMENT 1 INCREMENT

STATEMENT NO. 1 PTRS

STATEMENT NO. 2 PTRS

BL0CK CNT  
SCAN M0DE  
CRRNT CHAR

\$INCHCT ZR0	0	INPUT CHAR CNT
\$IICHCT ZR0	0	INITIAL INPUT CHAR CNT
\$INSTCT ZR0	0	INPUT ST0P CNT
\$RDCHCT ZR0	0	READ CHAR CNT
\$IRCHCT ZR0	0	INITIAL READ CHAR CNT
\$RDSTCT ZR0	0	READ ST0P CNT
\$LACHCT ZR0	0	LAST ACTIVE CHAR CNT
\$LINECT ZR0	0	LINE CNT
\$ILINCT ZR0	0	INITIAL LINE CNT
\$LINETP ZR0	0	LINES THIS PAGE
\$STLBPT ZR0	0	STA LBL PNTR
\$CKSUM ZR0	0	CHECK SUM
\$CKSMCT ZR0	0	CHECK SUM CNT
\$CHTSLN ZR0	0	CHARS THIS LINE
*MAX EXTIV IN COMMON		
\$MX0VNC BSS	0	DMY CNT
\$DMYCNT ZR0	0	
*COMMON BREAK		
\$COMBRK BSS	0	NAME 0F SPR0G PNTR
\$NAMESP ZR0	0	
*PROGRAM BREAK		
\$PR0GBK BSS	0	MAX TEMP CNT
\$MXTPCT ZR0	0	
*END 0F STA TRIGGER		
\$NSTTRG BSS	0	TEMP CNT
\$TMPCNT ZR0	0	
\$ACTEST ZR0	0	ACTIVE STA TRIGGER
\$LASTST ZR0	0	LAST STA TRIGGER
\$JPSTRG ZR0	0	JUMP STA TRIGGER
\$SPTRIG ZR0	0	SPR0G TRIGGER
\$SPRGLB ZR0	0	START 0F PR0G LBL
*DP MPY A		
*SAVE TEMP 1		
*READY CHAR TEMP 1		
\$DPMPYA BSS	0	COUNT 0F C0PY
\$SAVET1 BSS	0	
\$RDYCT1 BSS	0	
\$CTC0PY ZR0	0	
*DP MPY B		

*FAIL EXIT		
*ACTIVE CHAR TEMP 1		
\$DPMPYB BSS	0	MERG OF COPY
\$FAILEX BSS	0	
\$ACHTP1 BSS	0	
\$M8FCPY ZR0	0	
*DP MPY C		
*ACTIVE CHAR TEMP 2		
\$DPMPYC BSS	0	COUNT OF PLEX
\$ACHTP2 BSS	0	
\$CTPLEX ZR0	0	
*DP MPY D		
*INPUT CHAR TEMP 1		
\$DPMPYD BSS	0	TARGET LIST OF COPY
\$INCHT1 BSS	0	
\$TGTLCY ZR0	0	
*CONST TEMP A		
*DIMENS TEMP A		
*PACK TEMP 1		
\$CNSTPA BSS	0	SOURCE TOP OF COPY
\$DIMTPA BSS	0	
\$PAKTP1 BSS	0	
\$ST0CXX ZR0	0	
*CONST TEMP B		
*PACK TEMP 2		
*DIMENS TEMP B		
\$CNSTPB BSS	0	SOURCE LIST OF COPY
\$PAKTP2 BSS	0	
\$DIMTPB BSS	0	
\$SLSTCY ZR0	0	
*PACK TEMP 3		
\$PAKTP3 BSS	0	CHARACTERISTIC A
\$CHARA ZR0	0	
\$CHARB ZR0	0	CHARACTERISTIC B
\$DGSCTA ZR0	0	DIGITS SCANED CNT A
\$DGSCTB ZR0	0	DIGITS SCANED CNT B
\$DGUCTA ZR0	0	DIGITS USED CNT A
\$DGUCTB ZR0	0	DIGITS USED CNT B
\$SCDIGA ZR0	0	SCANED DIGIT A

\$SCDIGB ZR0	0	SCANED DIGIT B
*F0L TEMP A		
*BIM-BAM TEMP 1		
*C0F TEMP 1		
*FIC-FAC-FIP TEMP 1		
\$F0LTPA BSS	0	SER TEMP 1
\$BIAMT1 BSS	0	
\$C0FTP1 BSS	0	
\$FFFTP1 BSS	0	
\$SERTP1 ZR0	0	
*REASSIGN TEMP 1		
*F0L TEMP B		
*ADDRESS POINTED TO		
\$RSGNT1 BSS	0	SER TEMP 2
\$F0LTMB BSS	0	
\$ADRPT0 BSS	0	
\$SERTP2 ZR0	0	
*REASSIGN TEMP 2		
*DP TEMP 1		
\$RSGNT2 BSS	0	SER TEMP 3
\$DPTP1 BSS	0	
\$SERTP3 ZR0	0	
*REASSIGN TEMP 3		
\$RSGNT3 BSS	0	SER TEMP 4
\$SERTP4 ZR0	0	
*REASSIGN TEMP 4		
\$RSGNT4 BSS	0	SER TEMP 5
\$SERTP5 ZR0	0	
\$RSGNT5 ZR0	0	REASSIGN TEMP 5
\$QSFCTR ZR0	0	QSF COUNTER
\$QSFWRD ZR0	0	QSF WORD
\$QSFWAD ZR0	0	QSF WORD ADR
\$READSC ZR0	0	READ SPACE COUNT
\$PRITP1 ZR0	0	PRINT TEMP 1
\$ECHTP1 ZR0	0	ENTER CHAR TEMP 1
\$ECHTP2 ZR0	0	ENTER CHAR TEMP 2
\$NNTP1 BSS	1	
\$NNTP2 ZR0	0	NN TEMP 2

```

*
*LIST MEMORY
*SAVED ON DRUM FOR EACH SUBPROGRAM CREATED
$BASE BSS 0 BASE
*FL SCALAR BASE
$LAST BSS 0
$FLSBAS HLT 0 LIST MEMORY (INITIALIZED TO *LSTMEM)
*
*FLSLMT BSS 0 FL SCALAR LIMIT
$LIMIT BSS 0 LIMIT
*FX SCALAR BASE
$FXSBA HLT 7000B 7000
HLT 7400B 7400
$ HLT 7600B 7600
HLT 7700B 7700
HLT 7710B 7710
*GLOBAL SPRNG BASE
$GLSBAS HLT 7720B 7720
HLT 7730B 7730
HLT 7740B 7740
HLT 7750B 7750
HLT 7760B 7760
*EXTIV DATA BASE
$QVDTBS HLT 7770B 7770
HLT 7777B 7777
*CODE BASE
$CDBASE HLT 7777B 7777
HLT 7777B 7777
*PLEX BASE
$PLXBAS HLT 7777B 7777
*WORK BASE
$WRKBAS HLT 7777B 7777
*EXIT BASE
$XBASE HLT 7777B 7777
*EXIT LIMIT
$XLIMIT HLT 7777B 7777
HLT 7777B 7777

```

\$SVLIMI	BSS	0	SAVE LIMIT
*PUNCH	BASE		
\$PCHBAS	HLT	7777B	7777
\$START	HLT		START
	HLT		
	HLT		
	HLT		
	HLT		
	HLT		
	HLT		
	HLT		
	HLT		
\$QVDSTR	HLT	0	EXTIV DATA START
	HLT		
	HLT		
\$QVHSTR	HLT	0	EXTIV HOLD START
	HLT		
	HLT		
	HLT		
	HLT		
\$PCHSTR	HLT	0	PUNCH START
\$TOP	HLT	0	TOP
	HLT		
	HLT		
	HLT		
\$ARYTOP	HLT	0	ARRAY TOP
	HLT		
	HLT		
	HLT		
	HLT		
	HLT		
	HLT		
\$CODETOP	HLT	0	CODE TOP
\$QVHTOP	HLT	0	EXTIV HOLD TOP



	HLT		
	HLT		
	HLT		
	HLT		
	HLT		
\$PCHTOP	HLT	0	PUNCH TOP
\$BOTTOM	BSS	0	BOTTOM
\$FLSCBT	HLT	0	FL SCALAR BOTTOM
\$FXSBOT	HLT	0	FX SCALAR BOTTOM
	HLT		
	HLT		
	HLT		
\$GLSBOT	HLT	0	GLOBAL SPRNG BOTTOM
	HLT		
	HLT		
	HLT		
\$QVDBOT	HLT	0	EXTIV DATA BOTTOM
D0L0BM	EXT	*	D0 LOOPS OPEN BOTTOM
\$M0DBOT	HLT	0	MODE BOTTOM
\$C0DBOT	HLT	0	CODE BOTTOM
	HLT	0	
\$PLEXBT	HLT	0	PLEX BOTTOM
\$WRKBOT	HLT	0	WORK BOTTOM
\$EXIBOT	HLT	0	EXIT BOTTOM
	HLT		
\$SVBOT	HLT	0	SAVE BOTTOM
\$PNCHBT	HLT	0	PUNCH BOTTOM
NCARDS	EQU	9	
NCARDS	EXT	9	NUMBER OF CONTINUATION CARDS
IBSIZE	EQU	$73 * \text{NCARDS} / 4 + 39$	
IBSIZE	EXT	$73 * \text{NCARDS} / 4 + 39$	INPUT BUFFER SIZE
\$NPBUFF	BSS	IBSIZE	INPUT BUFFER
\$LSTMEM	HLT	0	START OF LIST MEMORY
*			
*E N D	OF	L I S T	M E M O R Y

```

*
PAGE      L I S T   D E F I N I T I O N S
FLSLST EXT    0      FL SCALAR LIST
FXSLST EXT    1      FX SCALAR LIST
LBLST  EXT    2      LBL LIST
*FL CONST LIST
FLCLST EXT    3      ERROR SYMBOL LIST
ERSYLS EXT    3
FXCLST EXT    4      FX CONST LIST
AYYLST EXT    5      ARRAY LIST
GLSLST EXT    6      GLOBAL SPRØG LIST
GDLST  EXT    7      GLOBAL DMY LIST
LØCSPL EXT   10B     LOCAL SPRØG LIST
LØCDML EXT   11B     LOCAL DMY LIST
*COMMON LIST
*SPRØG ARGUMENT LIST
*DIMENS LIST
*ERROR ØPUT LIST
CØLIST EXT   12B     SUM LIST
SPARGL EXT   12B
DIMLST EXT   12B
ERLIST EXT   12B
SUMLST EXT   12B
*EXTIV DATA LIST
*PRØD LIST
*TERM LIST
*SCRIPT EXP LIST
QDATLS EXT   13B     ERROR STØP LIST
PRØDLS EXT   13B
TERMLS EXT   13B
SCREXL EXT   13B
ERSTLS EXT   13B
*DO LØØPS ØPEN LIST
*EXTIV ØPUT LIST
DØLST  EXT   14B     MØDE LIST
QØPTLS EXT   14B
MØDLST EXT   14B
*CODE LIST
CØDLST EXT   15B     EXTIV TEMP LIST

```

QTPLS	EXT	15B	
*SCRIPT	LIST		
SCRLST	EXT	16B	EXTIV HOLD LIST
QHDLS	EXT	16B	
PLEXLS	EXT	17B	PLEX LIST
WORKLT	EXT	20B	WORK LIST
XITLS	EXT	21B	EXIT LIST
FAILDL	EXT	22B	FAIL DATA LIST
SAVELS	EXT	23B	SAVE LIST
PUNCHL	EXT	24B	PUNCH LIST
	PAGE	P0P, L0P, AND M0P	DEFINITIONS (DOUBLED)
SETUDP	EXT	200B	SET UP DMY P0P
STRP0P	EXT	210B	STORE P0P
DTRP0P	EXT	220B	D0 TRAP P0P
AGTP0P	EXT	224B	ASSIGN G0 TO P0P
FARGP0	EXT	230B	FIRST ARG P0P
NEXTAP	EXT	234B	NEXT ARG P0P
I0TP0P	EXT	240B	I0 TRAP P0P
FTCHP0	EXT	244B	FETCH P0P
0KFP0P	EXT	250B	0K FETCH P0P
ADDP0P	EXT	254B	ADD P0P
SUBP0P	EXT	264B	SUB P0P
MPYP0P	EXT	274B	MPY P0P
DIVP0P	EXT	304B	DIV P0P
CHSP0P	EXT	314B	CHS P0P
ASLP0P	EXT	320B	ASSIGN LBL P0P
LBL0P	EXT	6	LBL L0P
DELL0P	EXT	10B	DEL L0P
SYSL0P	EXT	12B	SYS L0P
BLKL0P	EXT	14B	BLK L0P
ABSL0P	EXT	22B	ABS L0P
BRUM0P	EXT	2	BRU M0P
BRRM0P	EXT	123B	BRR* M0P
BRR0P	EXT	122B	BRR 0P
ADDM0P	EXT	132B	ADD M0P
SKGM0P	EXT	166B	SKG M0P
HLTM0P	EXT	0	HLT M0P
HLTI0P	EXT	1	
HLTINS	EXT	23B	

BRM0P	EXT	11361B	SBRM* M0P
BRMM0P	EXT	106B	BRM M0P
N0PM0P	EXT	40B	N0P M0P
LDXM0P	EXT	162B	LDX M0P
LDAM0P	EXT	174B	LDA M0P
STAM0P	EXT	72B	
STLM0P	EXT	334B	STATEMENT LABEL M0P
EAX0PP	EXT	576B	EAX+M0P
MIN0P	EXT	142B	MIN M0P

\*  
\*

\* RUN-TIME LINK DEFINITIONS

STLINK	EXT	1	START OF DMYS LINK
NDYLNK	EXT	2	END OF DMYS LINK
ST0PSL	EXT	3	ST0P SPR0G LINK
WRITSL	EXT	4	WRITE SPR0G LINK
STPI0L	EXT	5	ST0P I=0 LINK
READSL	EXT	6	READ SPR0G LINK
PAUSSL	EXT	11B	PAUSE SPR0G LINK
DECLFSL	EXT	12B	DECLARE FILE SPR0G LINK
0PENSL	EXT	13B	0PEN FILE SPR0G LINK
CLFSL	EXT	14B	CLOSE FILE SPR0G LINK
CLAFSL	EXT	15B	CLOSE ALL FILES SPR0G LINK
CGTLNK	EXT	20B	COMPUTED GO TO SPR0G LINK
ISSLNK	EXT	22B	IF SENSE SWITCH LINK
ISLLNK	EXT	23B	IF SENSE LIGHT LINK
I0VLNK	EXT	24B	IF 0VERFLOW LINK
SENLSL	EXT	27B	SENSE LIGHT SPR0G LINK
EXPSPL	EXT	40B	EXP0N SPR0G LINK (POWER SUBR)
FIXSPL	EXT	31B	FIX SPR0G LINK
FL0TSL	EXT	32B	FL0AT SPR0G LINK
I0LUAL	EXT	33B	I0L UNSCRIPTED ARRAY LINK

PAGE FLAG AND PLEX DEFINITIONS

FLSIDF	EXT	0000	FL SCALAR ID FLAG
FXSIDF	EXT	0100B	FX SCALAR ID FLAG
FXCSTF	EXT	0400B	FX CONST FLAG
AYIDFG	EXT	0500B	ARRAY ID FLAG
GSPIDF	EXT	0600B	GLOBAL SPR0G ID FLAG
LSPIDF	EXT	1000B	LOCAL SPR0G ID FLAG

LDYIDF EXT	1100B	LOCAL DMY ID FLAG
MINUAP EXT	7741B	MINUS ADDRESSABLE PLEX
*MINUS NONADDRESSABLE PLEX		
MINNAP EXT	7701B	MINUS FLAG
MINUSF EXT	7701B	
*SUM TERM PLEX		
SUMTPX EXT	7600B	SUM FLAG
SUMFLG EXT	7600B	
ADDPLX EXT	7602B	ADD PLEX
*PRØD TERM PLEX		
PRØDTP EXT	7500B	PRØD FLAG
PRØDFG EXT	7500B	
MPYPLX EXT	7502B	MPY PLEX
*DIV TERM PLEX		
DIVTPX EXT	7402B	DIV FLAG
DIVFG EXT	7402B	
FCTFLG EXT	7302B	FUNCTION FLAG
FCTTPX EXT	7302B	FUNCTION TERM PLEX
*EXPØN TERM PLEX		
EXPTPX EXT	7202B	EXPØN FLAG
EXPFG EXT	7202B	
FIXTPX EXT	7101B	FIX TERM PLEX
*SINGLY SCRIPTED ARRAY PLEX		
SCRAPX EXT	6742B	SINGLY SCRIPTED ARRAY FLAG
SCRAFG EXT	6742B	
*SPEC SCRIPTED ARRAY PLEX		
SPSAPX EXT	6643B	SPEC SCRIPTED ARRAY FLAG
SPSAFG EXT	6643B	
*ARRAY DIMENS GRØUP PLEX		
AYDGP EXT	6500B	SPRØG ARG GRØUP PLEX
SPAGPX EXT	6500B	
	PAGE	EFFECTIVE ADDRESS DEFINITIONS
W0	EXT 0	W0
W1	EXT 37777B	W1
W2	EXT 37776B	W2
W3	EXT 37775B	W3
W4	EXT 37774B	W4
W5	EXT 37773B	W5
W6	EXT 37772B	W6

B0TSZ2	EXT	0	(B0TT0M=SIZE+2)
B0TSZ3	EXT	0	(B0TT0M=SIZE+3)
SIZEM1	EXT	0	(SIZE=1)
B0TP1	EXT	0	(B0TT0M+1)
B0TM1	EXT	0	(B0TT0M=1)
	END		

2C1 IDENT 7-17 FORTRAN COMPILER, PART 2

\*\*\*\*\*

\*  
\* ORIGIN AT ~~4000B~~ **30000B** \*

\*\*\*\*\*

\*  
\* R C H O P D E F I N I T I O N S

FCL	OPD	4600450B,2	FULL CYCLE LEFT
FCR	OPD	4600224B,2	FULL CYCLE RIGHT
BXC	OPD	4600022B,2	B TO X, CLB
XBC	OPD	24600040B,2	X TO B, CLX
CA	EQU	1	CLEAR A
CB	EQU	2	
*	CX	EQU 20000000B	ARPAS DEMANDS ,2
AB	EQU	4	A TO B
BA	EQU	10B	
BX	EQU	20B	
XB	EQU	40B	
XA	EQU	200B	
AX	EQU	400B	

\*  
\* M A C R O S

```

BEQ  MACRO  D  BRU IF EQUAL; SKE D(1); BRU **2; BRU D(2); ENDM
BNE  MACRO  D  BRU IF NOT EQUAL; SKE D(1); BRU D(2); ENDM
MOVE MACRO  D
MOVE1 NARG
      IF      'D(1$1,2)'='!$0'
      CL,D(MOVE1)
      ELSF   1
      LD,D(MOVE1) D(1)
      ENDF
MOVE2 EQU   2
      RPT    MOVE1-2
      ST,D(MOVE1) D(MOVE2)

```

MOVE2 EQU MOVE2+1  
ENDR  
ENDM

\*  
\*

\* P R O G R A M M E D O P E R A T O R D E F I N I T I O N S

BAM	OPD	10000000B,1,1
CSF	OPD	10100000B,1,1
CSA	OPD	10200000B,1,1
JAT	OPD	10300000B,1,1
JAF	OPD	10400000B,1,1
TRY	OPD	10500000B,1,1
GSF	OPD	10600000B,1,1
STD	OPD	10700000B,1,1
FET	OPD	11000000B,1,1
SWT	OPD	11100000B,1,1
JRS	OPD	11200000B,1,1
BIF	OPD	11300000B,1,1
MC0	OPD	11400000B,1,1
S0L	OPD	11500000B,1,1
S0F	OPD	11600000B,1,1
RSV	OPD	11700000B,1,1
CNT	OPD	12000000B,1,1
RLS	OPD	12100000B,1,1
T0T	OPD	12200000B,1,1
FIL	OPD	12300000B,1,1
M0N	OPD	12400000B,1,1
LDP	OPD	12500000B,1,1
BLF	OPD	12600000B,1,1
SAL	OPD	12700000B,1,1
PUL	OPD	13000000B,1,1
BNG	OPD	13100000B,1,1
BAF	OPD	13200000B,1,1
CAR	OPD	13300000B,1,1
PL0	OPD	13400000B,1,1
CAC	OPD	13500000B,1,1
CIC	OPD	13600000B,1,1
FIC	OPD	13700000B,1,1
FAC	OPD	14000000B,1,1



BOP	OPD	14100000B,1,1
MOP	OPD	14200000B,1,1
SNE	OPD	14300000B,1,1
SER	OPD	14400000B,1,1
LCF	OPD	14500000B,1,1
FIP	OPD	14600000B,1,1
ADR	OPD	14700000B,1,1
EOS	OPD	15000000B,1,1
SOB	OPD	15100000B,1,1
SOT	OPD	15200000B,1,1
SDP	OPD	15300000B,1,1
ADP	OPD	15400000B,1,1
REC	OPD	15500000B,1,1
COP	OPD	15600000B,1,1
LOC	OPD	15700000B,1,1
PRC	OPD	16000000B,1,1
PRQ	OPD	16100000B,1,1
SOC	OPD	16200000B,1,1
BIM	OPD	16300000B,1,1
FEX	OPD	16400000B,1,1
ICT	OPD	16500000B,1,1
*EXEC ENTRY, START AND RESTART		
\$PS	BRU	INITAZ
\$RS	LDA	ONES
	STA	INFLE
	STA	OUTFLE
	CLA	
	STA	FSTATS
	BRU	RUB1
*INITIALIZE MODIFIED CMMDS		
INITAZ	LDX	=4000B
	CLA	
	STA	4000B,2
	BRX	*=1
	MØVE	A183I,A183A,A
	MØVE	A184I,A184A,A
	MØVE	A185I,A185A,A
	MØVE	A187I,A187A,A
	MØVE	A188I,A188A,A

```

        MOVE    A232I,A232A,A
        MOVE    A233I,A233A,A
        MOVE    SCENI1,SYMCEN,A
        MOVE    SCENI2,SYMCN2,A
*POP LINKS
        LDX     MINS64           MINUS 64
PLI     LDA     PBRUE,2         POP BRU TABLE END
        STA     200B,2
        BRX     PLI
        SBRM    INITLS          INITIALIZE COMPILER LISTS
        MOVE    #1,COUTF,A      CMMD OUTPUT FILE
*SET #1 FLAGS
PS1     LDA     @NES
        STA     BP1             CODE OUTPUT ASSUMED
        STA     INDEX          SYMB INPUT FILE INDEX
        STA     OUTDEX         BINARY OUTPUT FILE INDEX
*SET 0 FLAG
        CLA
        STA     FDEBUG         NO DEBUGGING INFO ASSUMED
        STA     BP2           NO LISTING ASSUMED
        STA     FMAP
        LDA     #1
        STA     LSTFIL
        PRC     CARRET
*SET BREAK TABLE
        CLA
        LDX     #-1
        BRS     12
        MOVE    #PNEWS,TBLE,A
        LDA     FTBLE          ADDR OF FORK TBLE
        CAX
        BRS     9             OPEN FORK
        NOP     PNEWS
PWAIT  BRS     31
        STX     FSTATS        FORK STATUS
        BRS     17
        LDX     @NES
        BRS     29           CLEAR TTY OUTPUT BUFFER
        LDA     @NES

```

	STA	INFLE	
	STA	OUTFLE	
RUB1	PRC	CARRET	
	CLA		
	LDX	=-1	
	BRS	12	SET BREAK TABLE
	LDX	FSTATS	FORK STATUS
	BRU	*+1,2	
	BRU	RUB	
	BRU	BAD0P	
	PRQ	MEMQT	MEMORY PANIC
	BRU	PRL0C	
BAD0P	PRQ	RDERQT	READ ERROR QUOTE
PRL0C	LDA	TBLE	
	ETR	ADRMSK	
	LDB	EIGHT	
	LDX	0NE	
	BRS	36	PRINT LOCATION
	PRC	CARRET	
RUB	TC0	BELL	
	PRC	PLUS	
	LDX	0NES	
	BRS	11	
	CI0	ZER0	
	STA	FSTATS	(TEMP)
	MOVE	=CS2,TBLE,A	
	LDA	FTBLE	
	CAX		
	BRS	9	
	N0P	CS2	
	BRU	PWAIT	
PNEWS	PRQ	NEWS0T	<del>LATEST NEWS</del>
	BRU	CS	
CS2	LDA	FSTATS	
	BRU	CS5	
\$CSBAD	PRC	QMARK	
	LDX	0NES	
	BRS	11	CLEAR TTY INPUT BUFFER AFTER ERROR
*COMMAND	INPUT	START	

```

$CS   BRS      17
      LDA      0NES
      STA      INFLE
      STA      0UTFLE
      PRC      CARRET
CS7   PRC      PLUS
CS1   CI0      ZERO
      SKE      =155B; BRU CS3; BRU CS7-1
CS3   SKE      =152B; BRU CS5; BRU CS7-1
CS5   CLX; SBRM ACB
      DATA    'C'+CC0MP
      DATA    'M'+CMAP
      DATA    'I'+CSETIN
      DATA    '0'+CST0UT
      DATA    'N'+CN0
      DATA    'D'+CDEBUG
      DATA    'L'+CLIST
      DATA    ' '+CQUOTE
      DATA    6BIT+CSBAD
*SET INPUT FILE
CSETIN PRQ      INFQT
      PRQ      FRFQT
      CLA
      BRS 15
      BRU      CSBAD
      STA      INDEX
      BRU CS
      LDA 0NES
      STA INDEX
      BRU CSBAD
*SET OUTPUT FILE
CST0UT PRQ 0UTFQT
      PRQ T0FQT
CST0T1 CLA
      BRS      18
      BRU CST0T5
      STA 0UTDEX
      BRU CS
CST0T3 LDA 0NES

```

```

CST0T5 STA OUTDEX
PRQ WHATQT
BRU CST0T1
CMAP PRQ MAPQT
ICT PERIOD
BRU CSBAD
MOVE 0NES,FMAP,A
BRU CS
CDEBUG PRQ DBUGQT
ICT PERIOD
BRU CSBAD
LDA 0NES
CDB1 STA FDEBUG
BRU CS
*LIST STATEMENTS CMMD
CLIST PRQ LISTQT
PRQ T0FQT
CLA
BRB 18
BRU CSBAD
XAB
LDA 0NES
STA BP2
STB LSTFIL
STA FMAP
BRU CS
*N0 LIST CMMD
CNLIST PRQ LISTQT
ICT PERIOD
BRU CSBAD
CLA
LDB #1
STA BP2
STA FMAP
STB LSTFIL
BRU CS
*QUOTE MODE
CQUOTE LDA #2
LDA #=1

```

CQ2 BRS 12  
C10 ZERO  
BEQ =144B,CS  
BRU CQ2

\*N0 CMMD

CN0 PRC LETER0  
PRC SPACE  
C10 ZERO  
CLX, SBRM ACB  
DATA 'M'+CNMAP  
DATA 'L'+CNLIST  
DATA 'D'+CNDEBUG  
  
DATA 6BIT+CSBAD

CNMAP PRQ MAPQT  
ICT PERIOD  
BRU CSBAD  
MOVE =0,FMAP,A  
BRU CS

CNDEBUG PRQ DBUGQT  
ICT PERIOD  
BRU CSBAD  
CLA

CBADLT PRQ CDB1  
BRU BADLTQ BAD LIST OPENING  
BRU CS

CCOMP PRQ COMPQT  
ICT PERIOD  
BRU CSBAD  
PRC CARRET  
LDA =1  
SKE INDEX  
BRU \*\*2  
BRU CCOMP1  
SKE BUTDEX

CCOMP1	BRU	CCOMP3	
	PRQ	WI0QT	WHAT I/O QT
CCOMP2	BRU	CSBAD	
	PRQ	WRI0QT	WRONG I/O QT
CCOMP3	BRU	CSBAD	
	LDA	INDEX	
	BRS	16	
	BRU	CSBAD	
	STA	INFLE	
	CBA		
	SKE	THREE	SYMB FILE TYPE
	BRU	CCOMP2	
	LDA	0UTDEX	
	LDB	=1024	
	LDX	TW0	
	BRS	19	
	BRU	CSBAD	
	STA	0UTFLE	
	LDA	MINUS2	MINUS TW0
	STA	LINETP	LINES THIS PAGE
	LDA	LSTFIL	
	LDX	=3	
	SKG	=2	
	BRU	**3	
	BRS	19	
	BRU	CBADLT	
	STA	C0UTF	
	CLA		
	STA	FLC	LIST CODE (FOR DEBUGGING)
	STA	FE0F	FLAG END OF FILE READ
	STA	INCHCT	INPUT CHAR CNT
	STA	INSTCT	INPUT ST0P CNT
	STA	RDCHCT	READ CHAR CNT
	STA	RDSTCT	READ ST0P CNT
	LDA	MINUS1	MINUS ONE
	STA	READSC	READ SPACE COUNT
*START OF COMPILER			
\$STRCMP SBRM	INITLS		INITIALIZE LISTS
CLA			

S08	XITLS	EXIT LIST
STA	ITMPCT	INITIAL TEMP COUNT
STA	DMYCNT	DMY CNT
STA	CRNTCH	CRRT CHAR
STA	MXTPCT	MAX TEMP CNT
LDA	ONE	ONE
STA	LINECT	LINE CNT
LDA	MINUS1	MINUS ONE
STA	SPTRIG	SPR0G TRIGGER
STA	ACTEST	ACTIVE STA TRIGGER
STA	LASTST	LAST STA TRIGGER
STA	BL0KCT	BLOCK CNT
STA	CKSMCT	CHECK SUM CNT
*JOB CARD XLATE		
J0BCDX	JRS	SINIT
	CSA	ASTRSK
	JAF	N0JBCE
	JRS	RELPRT
	BRU	J0BCDX
*HEAD STA XLATE		
HEADSX	JRS	SINIT
*N0 JOB CARD ENTRY		
N0JBCE	TRY	LBFLDX
	JAF	ILLINX
	M0F	C0DLST
	M0F	C0DLST
	JRS	SVLSDT
*SUBROUTINE STA XLATE		
SUBRSX	FEX	FCTSXL
	QSF	SUBRQT
	JRS	SCLIDS
	JRS	SPRGHG
	CSA	LPAREN
	JAF	HEADEX
	BRU	DYSGSC
*FUNCTION STA XLATE		
FCTSXL	FEX	MAINPX
	QSF	FNCTQT
	JRS	SCLIDS
STA INITIALIZE		
STA INITIALIZE		
LBL FIELD XLATE		
ILLEGAL LINE XLATE		
SAVE LIST DATA		
FUNCTION STA XLATE		
SUBROUTINE QT		
SCALAR ID SCAN		
SPR0G HEAD GEN		
L PAREN		
HEAD EXIT		
DMY SEQ SCAN		
MAIN PROGRAM XLATE		
FUNCTION QT		
SCALAR ID SCAN		



	JRS	SPRGHG	SPR0G HEAD GEN
	CSF	LPAREN	L PAREN
*DMY SEQ SCAN			
DYSGSC	BLF	STLINK	START 0F DMYS LINK
A2	JRS	IDSCAN	ID SCAN
	JAT	INCNFA	ID CONFLICT FAIL
	SER	GDLIST	GLOBAL DMY LIST
	JAT	INCNFA	ID CONFLICT FAIL
	MC0	GDLIST	GLOBAL DMY LIST
	JRS	DYTPMK	DMY TEMP MAKER
	S0B	GDLIST	GLOBAL DMY LIST
	BAM	SETUDP	SET UP DMY P0P
	CSA	C0MMA	C0MMA
	JAT	A2	A1
	CSF	RPAREN	R PAREN
	BLF	NDYLNK	END 0F DMYS LINK
*HEAD EXIT			
HEADEX	ADR	C0DLST	CODE LIST
	CAR	W0RKLT	W0RK LIST
	CSF	CARET	
	BRU	A62	
*SPR0G HEAD GEN			
SPRGHG	LDA	W0,2	W0
	STA	NAMESP	NAME 0F SPR0G PNTR
	FIL	SPTITL	SPR0G TITLE
	FIL	ZER0	ZER0
	MC0	C0DLST	CODE LIST
	FIL	NMARKR	END MARKER
	JRS	LBLMAK	LBL MAKER
	STA	SPRGLB	START 0F PR0G LBL
	BIF	LBLLOP	LBL L0P
	FIL	HLT0P	
	BRU	CLR1AE	CLEAR 0NE AND EXIT
*MAIN PR0GRAM XLATE			
MAINPX	RLS	SAVELS	SAVE LIST
	FIL	MPTITL	MPR0G TITLE
	FIL	ZER0	ZER0
	FIL	ALLD0L	ALL D0LLARS
	FIL	ALLD0L	ALL D0LLARS

FIL	NMARKR	END MARKER
MIN	SPTRIG	SPRØG TRIGGER
ADR	CØDLST	CØDE LIST
CAR	WØRKLT	WØRK LIST
JRS	PCHCØD	PUNCH CØDE
BRU	MAINPE	MAIN PRØG ENTRY
*BØDY STA XLATE		
\$BØDYSX	JRS	SINIT
	TRY	LBFLDX
	JAF	ILLINX
	TRY	IBMEND
	SKR	LASTST
	NØP	Ø
*MAIN PRØG ENTRY		
MAINPE	JRS	SVLSDT
	FEX	ASSMSX
	CSA	G
	JAT	GØTØGP
	CSA	D
	JAT	DØRDIM
	CSA	LETERØ
	JAT	ØPEN SX
*IF STA XLATE		
IFSTAX	QSF	IFQT
	CSA	LPAREN
	JAF	FAIL
	JRS	EXPXLT
	JRS	MØDREL
	CSF	RPAREN
	JRS	LBCØMA
	FIL	SKARI
	BIF	BRUMØP
	JRS	LBCØMA
	FIL	SKGRZI
*IF STA EXITS		
\$IFSXIT	BIF	BRUMØP
	JRS	LBSCAN
	BIF	BRUMØP
	BRU	JPEØLE
		BRU MØP
		LBL SCAN
		BRU MØP
		JUMP EØL EXIT
		LBL FIELD XLATE
		ILLEGAL LINE XLATE
		LAST STATEMENT TRIG.
		SAVE LIST DATA
		ASSIGNMENT STA XLATE
		G
		GØ TØ GRØUP
		D
		DØ ØR DIMENSION
		ØPEN FILE STA XLATE
		IF QT
		L PAREN
		FAIL
		EXP REMEMBER MØDE XLATE
		MØDE RELEASE
		R PAREN
		LBL CØMMA SCAN
		SKA RUN TIME SIGN BIT INST
		BRU MØP
		LBL CØMMA SCAN
		SKG RUN TIME ZERØ INST

*OPEN FILE	STA	XLATE	
OPENSX	QSF	OPENQT	OPEN QT
	CSF	LPAREN	
	JRS	FXPXL	
	CSF	COMMA	
	BLF	DECFSL	DECLARE FILE SPR0G LINK
	CSA	I	
	JAF	OPENSX3	
	QSF	IFLQT	INPUT FILE QT
	FIL	INP0P	INPUT 0P
	BRU	OPENSX5	
OPENSX3	QSF	0FLQT	OUTPUT FILE QT
	FIL	0UTP0P	OUTPUT 0P
OPENSX5	CSF	COMMA	
	SBRM	NXACCH	NEXT ACTIVE CHARACTER, ADVANCE PAST ,
	FET	LACHCT	LAST ACTIVE CHAR CNT
OPENSX2	CSA	RPAREN	
	JAT	OPENSX4	
	SBRM	NXACCH	
	BRU	OPENSX2	
OPENSX4	LDA	ILINCT	INITIAL LINE CNT
	STA	LINECT	LINE CNT
	JRS	LMBRUG	LBL MAKER BRU GEN
	LDA	W1,2	
	XMA	INCHCT	INPUT CHAR CNT
	SUB	W1,2	
	ADD	F0UR	
	ABC		
	LSH	1	
	DIV	THREE	
	STA	W1,2	
	S0B	W0RKLT	W0RK LIST
	BAF	ABSL0P	W0RD C0UNT
	BRU	OPENSX8	
OPENSX7	JRS	PAKSTR	PACK STRING
	S0B	C0DLST	
OPENSX8	SKR	W1,2	
	BRU	OPENSX7	
	BIF	LBLL0P	

SKR WRKBOT  
BRU ACEBLE

\*D0 OR DIMENSION

D0RDIM CSA LETER0  
JAF DIMSX

LETTER 0  
DIMENSION STA XLATE

\*D0 STA XLATE

D0SX JRS LBSCAN  
CSA COMMA  
JRS SCLIDS  
CSF EQUAL  
JRS D0CTLX  
M0N D0LST  
BRU JPEBLE

LBL SCAN  
COMMA  
SCALAR ID SCAN  
EQUAL  
D0 CONTROL XLATE  
D0 LOOPS OPEN LIST  
JUMP EOL EXIT

\*D0 CONTROL XLATE

D0CTLX JRS M0DSET  
JRS XPMDX  
CSF COMMA  
FET W0,2  
BIM STRP0P  
JRS LMBRUG  
JRS LMLBLG  
FET W2,2  
JRS XPSCAN  
JRS SCRGEN  
JRS GRNM0D  
CSA COMMA  
JAT A3  
JRS ISMDFL  
JAT A4  
FET W1,2  
BIF MIN0P  
JRS XPBYMG  
FIL RTAFX1  
BIF SKGM0P  
BRU A6  
A3 JRS GRNM0D  
JRS XPMDX

MODE SET  
EXP BY MODE XLATE  
COMMA  
W0  
STORE POP  
LBL MAKER BRU GEN  
LBL MAKER LBL GEN  
W2  
EXP SCAN  
SCRIPT GEN  
GRNTEE ADDRESSABLE BY MODE GEN  
COMMA  
A1  
IS MODE FL  
A2  
W1  
MIN M0P  
EXP BY MODE GEN  
RUN TIME ADD FX 0NE  
SKG M0P  
A4  
GRNTEE ADDRESSABLE BY MODE GEN  
EXP BY MODE XLATE

A4	BRU	A5	A3
	JRS	GRNM0D	GRNTEE ADDRESSABLE BY M0DE GEN
	FIL	RTLFL1	RUN TIME LDP FL 0NE
A5	BAM	DTRP0P	D0 TRAP P0P
	BIF	HMTM0P	HLT M0P
A6	JRS	LMBRUG	LBL MAKER BRU GEN
	BIF	LBLL0P	LBL L0P
	BIF	BRUM0P	BRU M0P
	SKR	M0DB0T	M0DE B0TT0M
	M0F	C0DLST	C0DE LIST
	M0N	D0LST	D0 L00PS 0PEN LIST
	M0F	C0DLST	C0DE LIST
	M0N	D0LST	D0 L00PS 0PEN LIST
	BIF	LBLL0P	LBL L0P
	BRU	CLR1AE	CLEAR 0NE AND EXIT
*DIMENSION	STA	XLATE	
DIM SX	QSF	IMENGT	IMENSION QT
	RLS	PLEXLS	PLEX LIST
	RSV	DIMLST	DIMENS LIST
A7	JRS	IDSCAN	ID SCAN
	JAT	INCNFA	ID CONFLICT FAIL
	MC0	AYYLST	ARRAY LIST -- NAME 0F VARIABLE
	CSF	LPAREN	L PAREN
	FET	0NE	0NE
	FET	ZER0	ZER0
	FET	ZER0	ZER0
	BRU	A9	A3
A8	JRS	RGFXCS	REGISTER FX C0NST
	M0N	DIMLST	DIMENS LIST
A9	JRS	SIGNIS	SIGNED INTEGER SCAN
	CSA	VRGULE	VIRGULE
	JAT	A10	A4
	LDP	DP0NE	DP 0NE
	STD	DIMTPA	DIMENS TEMP A
	BRU	A11	A5
A10	LDP	CTL1	CENTRAL 1
	STD	DIMTPA	DIMENS TEMP A
	JRS	SIGNIS	SIGNED INTEGER SCAN
A11	LDA	DIMTPB	DIMENS TEMP B

MUL	W2,2	W2
RSH	1	1
ADP	W1,2	W1
STD	W1,2	W1
SOL	MAXASA	MAX ARRAY SIZE ALLOWED
JAT	ILALFA	ILLEGAL ALLOCATION FAIL
LDP	CTL1	CENTRAL 1
SDP	DIMTPA	DIMENS TEMP A
SKA	SIGNBT	SIGN BIT
BRU	ILALFA	ILLEGAL ALLOCATION FAIL
ADP	DPONE	DP ONE
STD	CTL1	CENTRAL 1
CBA		
MUL	W2,2	W2
RSH	1	1
STB	W2,2	W2
SOL	MAXASA	MAX ARRAY SIZE ALLOWED
JAT	ILALFA	ILLEGAL ALLOCATION FAIL
CSA	COMMA	COMMA
JAT	A8	A2
CSF	RPAREN	R PAREN
MON	AYYLIST	ARRAY LIST
SKR	WRKBOT	WORK BOTTOM
MON	AYYLIST	ARRAY LIST
PLB	AYDGP	ARRAY DIMENS GROUP PLEX
CAC	DIMLIST	DIMENS LIST
MON	AYYLIST	ARRAY LIST
CSA	COMMA	COMMA
JAT	A7	A1
CSA	CARRET	CAR RETURN
JAF	A7	A1
RLS	DIMLIST	DIMENS LIST
RSV	PLEXLS	PLEX LIST
BRU	CUAOUT	CLEAN UP AND OUTPUT
*G0 T0	GROUP	
G0T0GP	QSF	0T0GT
	S0C	DIGTFL
	JAF	C0RAGT
*G0 T0	STA XLATE	
		0T0 QT
		DIGIT FLAG
		COMPUTED OR ASSIGN G0 T0

G0T0SX	JRS	LBSCAN	LBL SCAN
	BIF	BRUM0P	BRU M0P
	BRU	JPE0LE	JUMR E0L EXIT
*COMPUTED 0R	ASSIGN G0	T0	
C0RAGT	CSA	LPAREN	L PAREN
	JAF	AGT0SX	ASSIGN G0 T0 STA XLATE
*COMPUTED G0	T0	STA XLATE	
CGT0SX	RSV	W0RKLT	W0RK LIST
A12	JRS	LBSCAN	LBL SCAN
	CSA	C0MMA	C0MMA
	JAT	A12	A1
	CSF	RPAREN	R PAREN
	CSA	C0MMA	C0MMA
	JRS	FXPXL	FX EXP XLATE
	BLF	CGTLNK	COMPUTED G0 T0 LINK
	CNT	W0RKLT	W0RK LIST
	BAF	HLT M0P	HLT M0P
A13	T0T	W0RKLT	W0RK LIST
	JAF	A14	A3
	BIF	BRUM0P	BRU M0P
	BRU	A13	A2
A14	RLS	W0RKLT	W0RK LIST
	BRU	JPE0LE	JUMP E0L EXIT
*ASSIGN G0	T0	STA XLATE	
AGT0SX	JRS	VARSCN	VAR SCAN
	JRS	SCRGEN	SCRIPT GEN
	BAM	AGTP0P	ASSIGN G0 T0 P0P
	CSA	C0MMA	C0MMA
	JAF	JPE0LE	JUMP E0L EXIT
	CSF	LPAREN	L PAREN
A15	JRS	LBSCAN	LBL SCAN
	CSA	C0MMA	C0MMA
	JAT	A15	A1
	CSF	RPAREN	R PAREN
	BRU	JPE0LE	JUMP E0L EXIT
*ASSIGNMENT	STA	XLATE	
ASSMSX	FEX	AFDFSX	ARITH FUN DEF STA XLATE
	JRS	VARSCN	VAR SCAN
	CSA	EQUAL	EQUAL

	JAF	FAIL	FAIL
	JRS	XPSCAN	EXP SCAN
	CSF	CARET	CAR RETURN
	SKN	BP1	
	BRU	ACEXIT	ACTIVE EXIT
	JRS	SETMTG	SET MODE THEN GEN
	JRS	STFMG	STORE FORGET MODE GEN
	BRU	ACEXIT	ACTIVE EXIT
*ARITH	FUN DEF	STA XLATE	
A16	AFDFSX	SKN	ACTIVE STA TRIGGER
	BRU	FORSXL	FORMAT STA XLATE
	FEX	FORSXL	FORMAT STA XLATE
	JRS	IDSCAN	ID SCAN
	JAT	FAIL	FAIL
	CSA	LPAREN	L PAREN
	JAF	FAIL	FAIL
	SOC	LETRFG	LETTER FLAG
	JAF	FAIL	FAIL
	FET	ZERØ	ZERØ
	JRS	STDMEV	STANDARD MODE EVAL
	MØN	MØDLST	MODE LIST
	MCØ	LØCSPL	LØCAL SPRØG LIST
	JRS	LMBRUG	LBL MAKER BRU GEN
	JRS	LMLBLG	LBL MAKER LBL GEN
	SØB	LØCSPL	LØCAL SPRØG LIST
	FIL	HLTØP	HALT INSTRUCTION
	BLF	STLINK	START ØF DMYS LINK
A16	JRS	SYMBSC	SYMBOL SCAN
	SER	LØCDML	LØCAL DMY LIST
	JAT	FAIL	FAIL
	MCØ	LØCDML	LØCAL DMY LIST
	JRS	DYTPMK	DMY TEMP MAKER
	SØB	LØCDML	LØCAL DMY LIST
	BAM	SETUDP	SET UP DMY PØP
	CSA	CØMMA	CØMMA
	JAT	A16	A1
	CSA	RPAREN	R PAREN
	JAF	FAIL	FAIL
	CSA	EQUAL	EQUAL



	JAF	FAIL	FAIL
	BLF	NDYLNK	END OF DMYS LINK
	JRS	XPMDX	EXP BY MODE XLATE
	RLS	L0CDML	LOCAL DMY LIST
	BIF	BRR0P	BRR 0P
	BIF	LBLL0P	LBL L0P
	SKR	M0DB0T	MODE B0TT0M
	LDA	TMPCNT	TEMP C0UNT
	STA	ITMPCT	INITIAL TEMP C0UNT
	BRU	IAE0LX	INACTIVE E0L EXIT
	*F0RMA T STA XLATE		
F0RSXL	FEX	F0RKSX	F0RK STA XLATE
	QSF	FRMTQT	F0RMA T QT
	CSF	LPAREN	L PAREN
	FET	LACHCT	LAST ACTIVE CHAR CNT
	JRS	FLSTSC	F0RMA T LIST SCAN
	LDA	ILINCT	INITIAL LINE CNT
	STA	LINECT	LINE CNT
	JRS	LMBRUG	LBL MAKER BRU GEN
	LDA	W1,2	W1
	XMA	INCHCT	INPUT CHAR CNT
	SUB	W1,2	W1
	ADD	F0UR	F0UR
	ABC		
	LSH	1	
	DIV	THREE	CALCULATE N0. 0F W0RDS
	STA	W1,2	W1
	S0B	W0RKLT	W0RK LIST
	BAF	ABS L0P	ABS L0P
	BRU	A18	A2
A17	JRS	PAKSTR	PACK STRING
	S0B	C0DLST	C0DE LIST
A18	SKR	W1,2	W1
	BRU	A17	A1
	BIF	LBLL0P	LBL L0P
	SKR	WRKB0T	W0RK B0TT0M
	BRU	IAE0LX	INACTIVE E0L EXIT
	*F0RMA T LIST SCAN		
FLSTSC	EQU	*	

A19	CSA	RPAREN	R PAREN
	JAT	EXIT	EXIT
A20	CSA	VRGULE	VIRGULE
	JAT	A21	A3
	JRS	FBASSC	FØRMAT BASIC SCAN
	CSA	VRGULE	VIRGULE
	JAF	A22	A4
A21	CSA	VRGULE	VIRGULE
	JAT	A21	A3
	CSA	CØMMA	CØMMA
	JAT	A20	A2
	BRU	A19	A1
A22	CSA	CØMMA	CØMMA
	JAT	A20	A2
	CSA	RPAREN	R PAREN
	JAF	A20	A2
	BRU	EXIT	EXIT
	*FØRMAT BASIC SCAN		
FBASSC	CSA	DØLLAR	DØLLAR
	JAT	A30	A8
	CSA	MINUS	MINUS
	JAT	A23	A1
	CSA	PLUS	PLUS
	JAF	A24	A2
A23	JRS	INSCAN	INTEGER SCAN
	CSF	P	P
	BRU	A25	A4
A24	SØC	DIGTFL	DIGIT FLAG
	JAF	A27	A5
	JRS	INSCAN	INTEGER SCAN
	CSA	P	P
	JAF	A26	A3
A25	SØC	DIGTFL	DIGIT FLAG
	JAF	A27	A5
	JRS	INSCAN	INTEGER SCAN
A26	LDA	CTL2	CENTRAL 2
	SKG	ZERØ	ZERØ
	BRU	ILNUFA	ILLEGAL NUMBER FAIL
	CSA	X	X

	JAT	A33	A11	
	CSA	H	H	
	JAT	A32	A10	
A27	CSA	E	E	
	JAT	A28	A6	
	CSA	F	F	
	JAT	A28	A6	
	CSA	LETERØ		
	JAT	INSCAN		
	CSA	I	I	
	JAT	INSCAN	INTEGER SCAN	
	CSA	A	A	
	JAT	INSCAN	INTEGER SCAN	
	CSF	LPAREN	L PAREN	CCA
	BRU	FLSTSC	FORMAT LIST SCAN	CAA
A28	JRS	INSCAN	INTEGER SCAN	
	CSF	PERIOD	PERIOD	
	BRU	INSCAN	INTEGER SCAN	
A29	SBRM	NXACCH	NEXT ACTIVE CHAR	
	SØC	TCRBFG	TAB-CAR RETURN-BACKSPACE	
	JAT	ILSYFA	ILLEGAL SYNTAX FAIL	
A30	CSA	DØLLAR	DØLLAR	
	JAF	A29	A7	
	BRU	A33	A11	
A31	LDA	UBLKFG	USE BLANKS FLAG	
	STA	SCANMD	SCAN MØDE	
	SØC	TCRBFG	TAB-CAR RETURN-BACKSPACE FLAG	
	JAT	ILSYFA	ILLEGAL SYNTAX FAIL	
	LDA	SKBLFG	SKIP BLANKS FLAG	
	STA	SCANMD	SCAN MØDE	
A32	CLA			
	STA	CRNTCH	CRRNT CHAR	
	SKR	CTL2	CENTRAL 2	
	BRU	A31	A9	
A33	SØC	CVRPFG	CØMMA-VIRGULE-R PAREN FLAG	
	JAT	EXIT	EXIT	
	BRU	FBASSC	FORMAT BASIC SCAN	
	*FØRK STA XLATE			
	FØRK SX FEX	WAIT SX	WAIT STA XLATE	

GSF FØRKQT  
JRS LBLMAK  
FET W0,2  
BIF LDAMØP  
JRS LBLMAK  
FET W0,2  
BIF STAMØP  
JRS LBLMAK  
FET W0,2  
BIF LDAMØP  
FIL BRS9  
JRS LMBRUG  
LDA W0,2  
STA FØRKT1  
SKR WRKBØT  
BIF LBLLØP  
FET W0,2  
BIF HLTMØP  
LDA W0,2  
XMA W1,2  
STA W0,2  
BIF LBLLØP  
JRS LBLMAK  
LDA W0,2  
STA FØRKT2  
BIF HLTMØP  
BIF LBLLØP  
FET SIX  
FIL ZERO  
SKR W0,2  
BRU FK5  
SKR WRKBØT  
FET FØRKT2  
BIF LBLLØP  
JRS IDSCAN  
JAF A34A  
SØT GSPIDF  
JAT A35A  
BRU INCNFA

LABEL MAKER

START FØRK  
LABEL MAKER BRU GEN

FK5

```

A34A  BØP  GLSLST
      MCØ  GLSLST
A35A  CSA  LPAREN
      JAF  A36A
      JRS  SPASQS
      CSF  RPAREN
      JRS  SCRGEN
      JRS  SPASQG
A36A  BIF  BRMØP
      FIL  BRS10
      FET  FØRKT1
      BIF  LBLLØP
      BRU  ACEØLE
*WAIT STA XLATE
WAITSX FEX  CALLSX
      QSF  WAITQT
      FIL  BRS31
      BRU  ACEØLE
*CALL STA XLATE
CALLSX FEX  CLFSX
      QSF  CALLQT
      JRS  IDSCAN
      JAF  A34
      SØT  GSPIDF
      JAT  A35
      BRU  INCNFA
A34   BØP  GLSLST
      MCØ  GLSLST
A35   CSA  LPAREN
      JAF  A36
      JRS  SPASQS
      CSF  RPAREN
      JRS  SCRGEN
      JRS  SPASQG
A36   BIF  BRMØP
      BRU  ACEØLE
*CLOSE FILE STA XTE
CLFSX FEX  CØNTSX
      QSF  CLOSQT

```

STØP FØRK

```

CALL STA XLATE

ACTIVE EØL EXIT

CLOSE FILE STA XLATE
CALL QT
ID SCAN
A1
GLØBAL SPRØG ID FLAG
A2
ID CØNFLECT FAIL
GLØBAL SPRØG LIST
GLØBAL SPRØG LIST
L PAREN
A4
SPRØG ARG SEQ SCAN
R PAREN
SCRIPT GEN
SPRØG ARG SEQ GEN
BRM* MØP
ACTIVE EØL EXIT

CONTINUE STA XLATE

```

	CSA	LPAREN	
	JAF	CLFSX7	
CLFSX3	JRS	FXPXL	FIXED EXPRESSION XLATE
	BLF	CLFSL	CLOSE FILE SPRØG LINK
	CSA	CØMMA	
	JAT	CLFSX3	
	CSF	RPAREN	
	BRU	ACEØLE	ACTIVE EØL EXIT
CLFSX7	BLF	CLAFSL	CLOSE ALL FILES SPRØG LINK
	BRU	ACEØLE	
	*CONTINUE STA XLATE		
CØNTSX	FEX	TYPESX	TYPE STA XLATE
	QSF	CØNTQT	CONTINUE QT
	BRU	ACEØLE	ACTIVE EØL EXIT
	*TYPE STA XLATE		
TYPESX	FEX	ACPAIR	ACCEPT PAIR
	QSF	TYPEQT	TYPE QT
TYPESX1	LDA	#1	TTY OUTPUT
	STA	CTL2	
	JRS	RGFXCS	REGISTER FIXED CØNSTANT
	BIF	LDAMØP	
	BLF	WRITSL	WRITE SPRØG LINK
	BRU	LCIØLX	LBL CØMMA IØL XLATE
	*ACCEPT PAIR		
ACPAIR	FEX	ACCTSX	ACCEPT STA XLATE
	QSF	ACPTQ	ACCEPT QT
	*ACCEPT TAPE STA XLATE		
ACTSX	QSF	TAPEQT	TAPE QT
	BLF	READSL	READ SUBPRØGRAM LINK
	BRU	LCIØLX	LBL CØMMA IØL XLATE
	*ACCEPT STA XLATE		
ACCTSX	FEX	READGP	READ GRØUP
	QSF	ACPTQ	ACCEPT QT
	FIL	CLAØP	I/O UNIT ZERO
	BLF	READSL	READ SPRØG LINK
	BRU	LCIØLX	LBL CØMMA IØL XLATE
	*READ GRØUP		
READGP	FEX	WRITGP	WRITE GRØUP
	QSF	READQT	READ QT

*TSS READ STA XLATE			
READSX	JRS	FXPXL	FX EXP XLATE
	BLF	READSL	READ SPR0G LINK
	BRU	CLCI0X	COMMA LBL COMMA I0L XLATE
*WRITE GROUP			
WRITGP	FEX	PHPAIR	PUNCH PAIR
	GSF	WRITEQ	
*TSS WRITE STA XLATE			
WRITSX	JRS	FXPXL	FX EXP XLATE
	BLF	WRITSL	WRITE SPR0G LINK
*COMMA LBL COMMA I0L XLATE			
CLCI0X	CSF	COMMA	COMMA
	BRU	LCI0LX	LBL COMMA I0L XLATE
*PUNCH PAIR			
PHPAIR	FEX	PRNTSX	PRINT STA XLATE
	GSF	PUNCHQ	PUNCH QT
	S0C	DIGTFL	DIGIT FLAG
	JAF	PCHTSX	PUNCH TAPE STA XLATE
*PUNCH STA XLATE			
	BRU	TYPSX1	
*PUNCH TAPE STA XLATE			
PCHTSX	GSF	TAPEQT	TAPE QT
	BRU	TYPSX1	
*PRINT STA XLATE			
PRNTSX	FEX	IFGRUP	IF GROUP
	GSF	PRINTQ	PRINT QT
	BRU	TYPSX1	
*LBL COMMA I0L XLATE			
LCI0LX	S0C	DIGTFL	DIGIT FLAG
	JAT	D36C	
	TRY	UNSCAS	UNSUBSCRIPTED ARRAY SCAN
	JAF	D36E	
*ARRAY NAME			
	PUL	W0RKLT	WORK LIST
	LDA	W1,2	
	STA	W0,2	
	JRS	L0DCTL	LOAD CENTRAL
	SER	GDLIST	GLOBAL DMY LIST
	JAF	D36A	

	BNG	TW0	(DUMMY POINTER)
	BIF	HLT10P	HLT* M0P (INDIRECT)
	SKR	WRKB0T	WORK B0TT0M
	BRU	CMI0LX	C0MMA I0L XLATE
D36A	FET	W0,2	
	S0F	FLFLAG	FLOATING FLAG
	BNG	TW0	(0FFSET)
	JAF	D36B	
	ADD	W0,2	(DOUBLE F0R FLOATING)
	STA	W0,2	
D36B	BAF	DELL0P	
	BRU	D36D	
	*STATEMENT LABEL		
D36C	JRS	LBSCAN	LBL SCAN
D36D	BIF	HLT M0P	HLT M0P
	BRU	CMI0LX	C0MMA I0L XLATE
	*SCALAR VARIABLE (T0 WHICH F0RMAT STA LABEL HAS BEEN ASSIGNED)		
D36E	JRS	VARSCN	
	S0F	SCA1DF	SCALAR 1D FLAG
	JAF	D36F	
	*UNSUBSCRIPTED SCALAR		
	BAM	AGTP0P	ASSIGN G0 T0 P0P
	BRU	CMI0LX	C0MMA I0L XLATE
	*UNSCRIPTED SCALAR		
D36F	JRS	LMBRUG	LBL MAKER BRU GEN
	JRS	LMLBLG	LBL MAKER LBL GEN
	SWT	W2,2	
	FIL	HLTINS	HLT INSTRUCTION
	JRS	SCRGEN	SCRIPT GEN
	BAM	FARGP0	FIRST ARG P0P
	BIF	LBL L0P	LBL L0P
	BIF	BRRM0P	BRR M0P
	*C0MMA I0L XLATE		
	CMI0LX	CSA	C0MMA
	JAF	I0LEND	I0L END
	JRS	I0LXLA	I0L XLATE
	*I0L END		



I0LEND	BLF	STPI0L	STOP I=0 LINK
	BRU	ACE0LE	ACTIVE E0L EXIT
*I0L XLATE			
I0LXLA	RSV	C0DLST	C0DE LIST
A37	CSA	LPAREN	L PAREN
	JAF	A38	A1
	JRS	I0LXLA	I0L XLATE
	CSF	RPAREN	R PAREN
	BRU	A41	A3
A38	TRY	UNSCAS	UNSCRIPTED ARRAY SCAN
	JAF	A39	A2
	FET	WO,2	WO
	BAM	FARGP0	FIRST ARG P0P
	PUL	W0RKLT	W0RK LIST
	SKR	WRKB0T	W0RK B0TT0M
	BNG	THREE	THREE
	BAF	NEXTAP	NEXT ARG P0P
	BLF	I0LUAL	I0L UNSCRIPTED ARRAY LINK
	BRU	A41	A3
A39	JRS	VARSCN	VAR SCAN
	JRS	SCRGEN	SCRIPT GEN
A40	BAM	I0TP0P	I0 TRAP P0P
A41	CSA	C0MMA	C0MMA
	JAF	A42	A7
	TRY	SCLIDS	SCALAR ID SCAN
	JAF	A37	A4
	CSA	EQUAL	EQUAL
	JAF	A40	A5
	ADR	D0LST	D0 L00PS 0PEN LIST
	C0F	C0DLST	C0DE LIST
	JRS	D0CTLX	D0 C0NTR0L XLATE
	BRU	A43	A6
A42	ADR	D0LST	D0 L00PS 0PEN LIST
	C0F	C0DLST	C0DE LIST
A43	ADR	C0DLST	C0DE LIST
	CAR	D0LST	D0 L00PS 0PEN LIST
	BRU	EXIT	EXIT
*IF GR0UP			
IFGRUP	FEX	ASSNSX	ASSIGN STA XLATE

QSF	IFQT	IF QT
CSA	LPAREN	L PAREN
JAT	IFSPAR	IF SENSE PAIR
*IF OVERFLOW STA XLATE		
IFBVX QSF	FLQT	FLOATING QT
QSF	OVFLQT	OVERFLOW QT
ADR	I0VLNK	IF OVERFLOW LINK
BRU	IFSOVX	IF SENSE=OVERFLOW XLATE
\$NLIST ZR0	PS=1	END OF COMPILER LISTS
6BIT EQU	400000B	
NEWSQT ASC	<del>NEWS NULL</del>	NEWS NULL, PUT MSG BETWEEN DOTS.
*		SEMICOLON FOR CR/LF.
END		

*'\$VERS. 5.041'*

3C1 IDENT 7=17 COMPILER, PART 3

SETT MACR0 D; LD.D(2) =1; ST.D(2) D(1); ENDM  
SETF MACR0 D; CL.D(2); ST.D(2) D(1); ENDM

BAM	0PD	10000000B,1,1
CSF	0PD	10100000B,1,1
CSA	0PD	10200000B,1,1
JAT	0PD	10300000B,1,1
JAF	0PD	10400000B,1,1
TRY	0PD	10500000B,1,1
GSF	0PD	10600000B,1,1
STD	0PD	10700000B,1,1
FET	0PD	11000000B,1,1
SWT	0PD	11100000B,1,1
JRS	0PD	11200000B,1,1
BIF	0PD	11300000B,1,1
MC0	0PD	11400000B,1,1
S0L	0PD	11500000B,1,1
S0F	0PD	11600000B,1,1
RSV	0PD	11700000B,1,1
CNT	0PD	12000000B,1,1
RLS	0PD	12100000B,1,1
T0T	0PD	12200000B,1,1
FIL	0PD	12300000B,1,1
M0N	0PD	12400000B,1,1
LDP	0PD	12500000B,1,1
BLF	0PD	12600000B,1,1
SAL	0PD	12700000B,1,1
PUL	0PD	13000000B,1,1
BNG	0PD	13100000B,1,1
BAF	0PD	13200000B,1,1
CAR	0PD	13300000B,1,1
PL0	0PD	13400000B,1,1
CAC	0PD	13500000B,1,1
CIC	0PD	13600000B,1,1
FIC	0PD	13700000B,1,1

FAC	0PD	14000000B,1,1
B0P	0PD	14100000B,1,1
M0F	0PD	14200000B,1,1
SNE	0PD	14300000B,1,1
SER	0PD	14400000B,1,1
LCF	0PD	14500000B,1,1
FIP	0PD	14600000B,1,1
ADR	0PD	14700000B,1,1
E0S	0PD	15000000B,1,1
S0B	0PD	15100000B,1,1
S0T	0PD	15200000B,1,1
SDP	0PD	15300000B,1,1
ADP	0PD	15400000B,1,1
REC	0PD	15500000B,1,1
C0F	0PD	15600000B,1,1
LC0	0PD	15700000B,1,1
PRC	0PD	16000000B,1,1
PRQ	0PD	16100000B,1,1
S0C	0PD	16200000B,1,1
BIM	0PD	16300000B,1,1
FEX	0PD	16400000B,1,1

\*IF SENSE PAIR

\$IFSPAR QSF       SENSEQ

CSA       S

JAF       IFSLX

\*IF SENSE SWITCH STA XLATE

IFSSSX QSF       WITCHQ

ADR       ISSLNK

BRU       IFSLSX

\*IF SENSE LIGHT STA XLATE

IFSLX QSF       LIGHTQ

ADR       ISLLNK

\*IF SENSE LIGHT=SWITCH XLATE

IFSLSX JRS       FXPXL

CSF       RPAREN

\*IF SENSE=OVERFLOW XLATE

\$IFS0VX BLF\*     W0,2

SKR       WRKB0T

JRS       LBC0MA

SENSE QT

S

IF SENSE LIGHT STA XLATE

WITCH QT

IF SENSE SWITCH LINK

IF SENSE LIGHT=SWITCH XLATE

LIGHT QT

IF SENSE LIGHT LINK

FX EXP XLATE

R PAREN

W0

WRKB0TT0M

LBL COMMA SCAN

	BRU	IFSXIT	IF STA EXITS
*ASSIGN	STA	XLATE	
\$ASSNSX	FEX	SENLSX	SENSE LIGHT STA XLATE
	QSF	ASSGNQ	ASSIGN QT
	JRS	LBSCAN	LBL SCAN
	QSF	T0QT	T0 QT
	JRS	VARSCN	VAR SCAN
	JRS	SCRGEN	SCRIPT GEN
	BAM	ASLP0P	ASSIGN LBL P0P
	BIF	BRUM0P	BRU M0P
	BRU	ACE0LE	ACTIVE E0L EXIT
*SENSE LIGHT	STA	XLATE	
SENLSX	FEX	C0MMSX	C0MM0N STA XLATE
	QSF	SENSEQ	SENSE QT
	QSF	LIGHTQ	LIGHT QT
	JRS	FXPXL	FX EXP XLATE
	BLF	SENLSL	SENSE LIGHT SPR0G LINK
	BRU	ACE0LE	ACTIVE E0L EXIT
*C0MM0N	STA	XLATE	
C0MMSX	FEX	EQUXLA	EQUIVALENCE STA XLATE
	QSF	CM0NQI	C0MM0N QT
A45	JRS	SYMBSC	SYMBOL SCAN
	SER	C0LIST	C0MM0N LIST
	JAT	ILALFA	ILLEGAL ALLOCATI0N FAIL
	ADR	QDATLS	EQUIV DATA LIST
	JRS	QLSRCH	EQUIV LIST SEARCH
	JAT	ILALFA	ILLEGAL ALLOCATI0N FAIL
	MC0	C0LIST	C0MM0N LIST
	CSA	C0MMA	C0MMA
	JAT	A45	A1
	BRU	IAE0LX	INACTIVE E0L EXIT
*EQUIVALENCE	STA	XLATE	
EQUXLA	FEX	RTRNSX	RETURN STA XLATE
	QSF	EQUIVQ	EQUIVALENCE QT
	JRS	CYEQDT	C0PY EQUIV DATA
	RSV	QTPLS	EQUIV TEMP LIST
A44	CSF	LPAREN	L PAREN
	JRS	QSEGX	EQUIV SEG XLATE
	CSF	RPAREN	R PAREN

	CSA	COMMA	COMMA
	JAT	A44	A1
	CSA	CARRET	CAR RETURN
	JAF	A44	A1
	RLS	QTPLS	EQUIV TEMP LIST
A46	RLS	QDATLS	EQUIV DATA LIST
	SNE	QDATLS	EQUIV DATA LIST
	JAT	A46	A2
A47	ADR	QDATLS	EQUIV DATA LIST
	C0F	QHDL5	EQUIV HOLD LIST
	JAT	A47	A3
	BRU	CUA0UT	CLEAN UP AND BUTPUT
*EQUIV	SEG	XLATE	
QSEGX	EQU	*	
A48	JRS	QSCSR	EQUIV SCAN AND SEARCH
	JAF	A52	A4
	JRS	APINSC	APPENDED INTEGER SCAN
	SUB	W0,2	W0
	STA	W0,2	W0
	BRU	A51	A3
A49	JRS	QSCSR	EQUIV SCAN AND SEARCH
	JAT	ILALFA	ILLEGAL ALLOCATI0N FAIL
A50	MC0	QTPLS	EQUIV TEMP LIST
	JRS	APINSC	APPENDED INTEGER SCAN
	S0B	QTPLS	EQUIV TEMP LIST
A51	CSA	COMMA	COMMA
	JAT	A49	A2
	FET	ZER0	ZER0
	BRU	A53	A1
A52	MC0	QTPLS	EQUIV TEMP LIST
	JRS	APINSC	APPENDED INTEGER SCAN
	S0B	QTPLS	EQUIV TEMP LIST
	CSA	COMMA	COMMA
	JAT	A48	A6
	RSV	QHDL5	EQUIV HOLD LIST
	FET	ZER0	ZER0
	FET	0NE	0NE
A53	LC0	QTPLS	EQUIV TEMP LIST
	JAF	CLR2AE	CLEAR TW0 AND EXIT

	MC0	QHDL5	EQUIV H0LD LIST
	T0T	QTPL5	EQUIV TEMP LIST
	SUB	W2,2	W2
	STA	W0,2	W0
	M0N	QHDL5	EQUIV H0LD LIST
	SER	C0LIST	C0MM0N LIST
	JAF	A53	A1
	SKR	WRKB0T	W0RK B0TT0M
	SKR*	WRKB0T	W0RK B0TT0M
	BRU	A53	A1
	BRU	ILALFA	ILLEGAL ALL0CATION FAIL
*EQUIV	SCAN	AND SEARCH	
QSCSR	JRS	SYMB5C	SYMBOL SCAN
	SER	QTPL5	EQUIV TEMP LIST
	JAT	ILALFA	ILLEGAL ALL0CATION FAIL
	ADR	QHDL5	EQUIV H0LD LIST
	JRS	QLSRCH	EQUIV LIST SEARCH
	JAF	XTF	EXIT FALSE
	BRU	EXITRU	EXIT TRUE
*EQUIV	LIST	SEARCH	
QLSRCH	RSV	QTPL5	EQUIV TEMP LIST
A54	SER*	W0,2	W0
	JAT	A56	A3
	ADR	QTPL5	EQUIV TEMP LIST
	C0F*	W1,2	W1
	JAT	A54	A1
A55	ADR*	W0,2	W0
	C0F	QTPL5	EQUIV TEMP LIST
	JAT	A55	A2
	RLS	QTPL5	EQUIV TEMP LIST
	BRU	CLR1EF	CLEAR 0NE EXIT FALSE
A56	BNG	TW0	TW0
	ADR	Q0PTLS	EQUIV 0PUT LIST
	C0F*	W2,2	W2
A57	ADR*	W1,2	W1
	C0F	QTPL5	EQUIV TEMP LIST
	JAT	A57	A4
	RLS	QTPL5	EQUIV TEMP LIST
	ADR*	W1,2	W1

	C0F	Q0PTLS	EQUIV 0PUT LIST
	SWT	W1,2	W1
	BRU	CLR1ET	CLEAR 0NE EXIT TRUE
	*APPENDED INTEGER SCAN		
	APINSC	FET	ZER0
	JRS	STDMEV	STANDARD M0DE EVAL
	CSA	LPAREN	L PAREN
	LDA	0NE	0NE
	STA	CTL2	CENTRAL 2
	JAF	A58	A1
	JRS	INSCAN	INTEGER SCAN
	S0L	MAXASA	MAX ARRAY SIZE ALL0WED
	JAT	ILALFA	ILLEGAL ALL0CATION FAIL
	CSF	RPAREN	R PAREN
A58	S0F	FLFLAG	FL FLAG
	LDA	CTL2	CENTRAL 2
	JAF	CLR1AE	CLEAR 0NE AND EXIT
	ADD	CTL2	CENTRAL 2
	BRU	CLR1AE	CLEAR 0NE AND EXIT
	*C0PY EQUIV DATA		
	CYEQDT	LDA	0NE
	STA	CTC0PY	C0UNT 0F C0PY
	BRU	A60	A2
A59	CAX		
	LDA	0,2	0
	S0B	QH0LS	EQUIV H0LD LIST
	MIN	CTC0PY	C0UNT 0F C0PY
A60	LDA	CTC0PY	C0UNT 0F C0PY
	ADD	QVDTBS	EQUIV DATA BASE
	SKG	QVDB0T	EQUIV DATA B0TT0M
	BRU	A59	A1
	LDA	QVDSTR	EQUIV DATA START
	SUB	QVDTBS	EQUIV DATA BASE
	ADM	QVHSTR	EQUIV H0LD START
	ADM	QVHT0P	EQUIV H0LD T0P
	BRU	EXIT	EXIT
	*RETURN STA XLATE		
	RTRNSX	FEX	PAUSSX
	QSF	RTURNQ	RETURN QT



	SKN	SPTRIG	SPRØG TRIGGER
	BLF	STØPSL	STØP SPRØG LINK
	SKN	SPTRIG	SPRØG TRIGGER
	BRU	JPEØLE	JUMP EØL EXIT
	FET	NAMESP	NAME ØF SPRØG PNTR
	BAM	ØKFPØP	ØK FETCH PØP
	FET	SPRGLB	START ØF PRØG LBL
	BIF	BRRMØP	BRR MØP
	BRU	JPEØLE	JUMP EØL EXIT
*PAUSE	STA	XLATE	
PAUSSX	FEX	STØPSX	STØP STA XLATE
	GSF	PAUSEQ	PAUSE QT
	BLF	PAUSSL	PAUSE SPRØG LINK
	TRY	INSCAN	INTEGER SCAN
	JAF	A61	A1
	FET	CTL2	CENTRAL 2
	BAF	NØPMØP	NØP MØP
	BRU	ACEØLE	ACTIVE EØL EXIT
A61	FIL	NØPINS	NØP INST
	BRU	ACEØLE	ACTIVE EØL EXIT
*STØP	STA	XLATE	
STØPSX	FEX	ILLSX	ILLEGAL STATEMENT XLATE
	GSF	STØPQT	STØP QT
	TRY	INSCAN	INTEGER SCAN
	BLF	STØPSL	STØP SPRØG LINK
	BRU	JPEØLE	JUMP EØL EXIT
*INACTIVE EØL EXIT	- FINISH STATEMENT PRØCESSING BUT SUPPRESS CØDE GEN		
\$IAEØLX	CSF	CARET	CAR RETURN
	BRU	CUAØUT	CLEAN UP AND ØUTPUT
*ACTIVE EØL EXIT			
\$ACEØLE	CSF	CARET	CAR RETURN
*ACTIVE EXIT			
\$ACEXIT	LDA	MINUS1	MINUS ØNE
	BRU	JUPACX	JUMP=ACTIVE EXIT
*JUMP EØL EXIT			
\$JPEØLE	CSF	CARET	CAR RETURN
	LDA	ZERØ	ZERØ
*JUMP=ACTIVE EXIT			
\$JUPACX	STA	JPSTRG	JUMP STA TRIGGER

```

MIN      ACTEST
*CLEAN UP AND OUTPUT
$CUAOUT SKN      BP2
BRU      ENDFIX
$A62     JRS      RELPRT
        JAF      A62
BRU      SFINAL
ENDFIX   LDA      LINECT
        STA      ILINCT
        LDA      INCHCT
        STA      IICHCT
        BRU      SFINAL
*ILLEGAL STA XLATE
$ILLSX   SNE      FAILDL
        JAT      0PDIAG
        FEX      0PDIAG
        JRS      VARSCN
        BRU      ILSYFA
*ILLEGAL LINE XLATE
$ILLINX  JRS      SVLSDT
*OUTPUT DIAGNOSTICS
$0PDIAG  FEX      ENDSX
        CSA      ASTRSK
        JAT      FAIL
        RSV      ERLIST
        RSV      ERSTLS
        SETT     FTTY0N,A
A63     JRS      PRTALN
        JRS      PRERMK
        JAF      A63
        RLS      FAILDL
A64     T0T      ERLIST
        JAF      A65
        PRQ*     W0,2
        SKR      WRKB0T
        PRC      CARET
        BRU      A64
A65     RLS      ERLIST
        RLS      ERSTLS

```

```

ACTIVE STA TRIGGER

RELEASE AND PRINT
A1
STA FINAL
LINE COUNT
INITIAL LINE COUNT
INPUT CHAR COUNT
INIT INPUT CHAR COUNT
STA FINAL

FAIL DATA LIST
OUTPUT DIAGNOSTICS
OUTPUT DIAGNOSTICS
VAR SCAN
ILLEGAL SYNTAX FAIL

SAVE LIST DATA

END STA XLATE
ASTERISK
FAIL
ERROR 0PUT LIST
ERROR STOP LIST

PRINT A LINE
PRINT ERROR MARKS
A1
FAIL DATA LIST
ERROR 0PUT LIST
A2
W0
W0RK B0TT0M
CAR RETURN
A3
ERROR 0PUT LIST
ERROR STOP LIST

```

	SETF	FTTY0N,A	
*STA FINAL			
\$SFINAL	JRS	STACUP	STA CLEAN UP
	JRS	REL PCH	RELEASE AND PUNCH
	BRU	B0DYSX	B0DY STA XLATE
*END STA XLATE			
\$ENDSX	FEX	N0NCDF	N0 END CARD FINISH
	CSF	ASTRSK	ASTERISK
\$IBMEND	MIN	LASTST	LAST STA TRIGGER
	QSF	ENDQT	END QT
	CSF	CARET	CARRAIGE RETURN
	JRS	REL PRT	RELEASE AND PRINT
	BRU	FINC0P	FINISH C0DE 0PUT
*N0 END CARD FINISH			
\$N0NCDF	LDA	IICHCT	INITIAL INPUT CHAR CNT
	STA	INCHCT	INPUT CHAR CNT
*FINISH C0DE 0PUT			
\$FINC0P	BLF	ST0PSL	ST0P SPR0G LINK
	JRS	REL PCH	RELEASE AND PUNCH
	BRU	PRTSUM	PRINT SUMMARY
*STA INITIALIZE			
\$SINIT	LDA	LINECT	LINE CNT
	STA	ILINCT	INITIAL LINE CNT
A66	LDA	INCHCT	INPUT CHAR CNT
	STA	IICHCT	INITIAL INPUT CHAR CNT
	STA	RDSTCT	READ ST0P CNT
	SBRM	NXINCH	NEXT INPUT CHAR
	CSA	CARET	CAR RETURN
	JAT	A66	A1
*C0MMENT STA XLATE			
\$C0MTSX	CSA	C	C
	JAF	EXIT	EXIT
	SKN	BP2	
	BRU	A67	A1
	JRS	REL PRT	RELEASE AND PRINT
	BRU	SINIT	STA INITIALIZE
A67	LDA	INCHCT	INPUT CHAR CNT
	STA	RDSTCT	READ ST0P CNT
	SBRM	NXINCH	NEXT INPUT CHAR

	CSA	CARET	CAR RETURN
	JAF	A67	A1
	MIN	LINECT	LINE CNT
	BRU	SINIT	STA INITIALIZE
	*LBL FIELD XLATE		
	\$LBFLDX FET	F0UR	F0UR
	JRS	DGCVIN	DIGIT CONV INITIAL
	BRU	A69	A2
A68	SBRM	NXINCH	NEXT INPUT CHAR
A69	S0C	DIGTFL	DIGIT FLAG
	JAF	A70	A3
	JRS	CNV1DG	CONV ONE DIGIT
	BRU	A71	A5
A70	CSF	SPACE	SPACE
A71	SKR	W0,2	W0
	BRU	A68	A1
	SBRM	NXINCH	NEXT INPUT CHAR
	CSA	SPACE	SPACE
	JAT	A72	A4
	CSF	N0	N0
A72	SKR	WRKB0T	W0RK B0TT0M
	LDA	CTL2	CENTRAL 2
	STA	STLBPT	STA LBL PNTR
	SKG	ZER0	
	BRU	A72A	INF0 SUPPRESSED
	JRS	REGLBL	REGISTER LBL
	LDA	W0,2	W0
	STA	STLBPT	STA LBL PNTR
	S0B	W0RKLT	W0RK LIST
	BIF	LBL0P	LBL L0P
	JRS	PRPNTR	PR0CESS PNTR
	LDA*	ADRPT0	ADDRESS P0INTED T0
	SKA	PDEFML	PREVIOUSLY DEF LBL MASK
	MRG	MPYDLM	MULTIPLY DEF LBL MASK
	MRG	PDEFML	PREVIOUSLY DEF LBL MASK
	STA*	ADRPT0	ADDRESS P0INTED T0
*PASS STATEMENT LABEL 0N T0 RUN-TIME			
A72A	SKN	FDBUG	
	BRU	EXIT	DEBUG INF0 SUPPRESSED

	FET	CTL2	INTEGER STATEMENT LABEL
	BAF	STLM0P	STATEMENT LABEL M0P
	BRU	EXIT	EXIT
	*STA CLEAN UP		
	\$STACUP FET	STLBPT	STA LBL PNTR
	SKN	JPSTRG	JUMP STA TRIGGER
	BRU	CLR1AE	CLEAR ONE AND EXIT
A73	SNE	D0LST	D0 LOOPS OPEN LIST
	JAF	CLR1AE	CLEAR ONE AND EXIT
	LDA	W0,2	W0
	SKE*	D0L0BM	D0 LOOPS OPEN BOTTOM
	BRU	CLR1AE	CLEAR ONE AND EXIT
	SKR	D0L0BM	D0 LOOPS OPEN BOTTOM
	M0F	D0LST	D0 LOOPS OPEN LIST
	M0F	D0LST	D0 LOOPS OPEN LIST
	M0N	C0DLST	C0DE LIST
	M0N	C0DLST	C0DE LIST
	BRU	A73	A1
	*RELEASE AND PUNCH		
	\$REL PCH RLS	SAVELS	SAVE LIST
	RLS	FAILDL	FAIL DATA LIST
	RLS	PLEXLS	PLEX LIST
	RLS	W0RKLT	W0RK LIST
	RLS	SCR LST	SCRIPT LIST
	BRU	PCHC0D	PUNCH C0DE
	*RELEASE AND PRINT		
	\$REL PRT RLS	FAILDL	FAIL DATA LIST
	*PRINT A LINE		
	\$PRTALN LDA	IICHCT	INITIAL INPUT CHAR CNT
	STA	INCHCT	INPUT CHAR CNT
A74	SBRM	NXINCH	NEXT INPUT CHAR
	SKA	CARRFG	CAR RETURN FLAG
	BRU	A74	
	LDA	0NES	
	STA	NSTTRG	END 0F STATEMENT TRIGGER
	STA	E0LTRG	END 0F LINE TRIGGER
	FET	ZER0	
A76	LDA	INCHCT	INPUT CHAR CNT
	BRU	A76A	

A75	SBRM	NXACCH	NEXT ACTIVE CHAR
	SKE	CARET	CAR RETURN
	BRU	A75A	
	LDA	INCHCT	INPUT CHAR CNT
	SKG	E0LTRG	END OF LN TRG(ADDR OF NO. 1 CHAR RET)
	MIN	NSTTRG	END OF STATEMENT TRIGGER
A75A	LDA	E0LTRG	END OF LINE TRIGGER
	SKG	0NES	END OF LINE.
	BRU	A75B	•NO
	LDB	CARET	CAR RETURN •YES
	BRU	A75C	
A75B	LDA	LACHCT	LAST ACTIVE CHAR CNT
	ADD	0NE	
	LDB	FRSTCH	FIRST CHAR READ BY NXACCH
A75C	STB	CRNTCH	CURRENT CHAR
	STA	INCHCT	INPUT CHAR CNT
A76A	STA	RDSTCT	READ ST0P CNT
	STA	CTL2	CENTRAL 2
	PRC	CRNTCH	CURRENT CHAR
	SER	FAILDL	FAIL DATA LIST
	JAF	A77	
	BNG	0NE	
	M0N	ERLIST	ERR0R 0UTPUT LIST
	M0N	ERSTLS	ERR0R ST0P LIST
	FET	MINUS1	MINUS 0NE
A77	MIN	W0.2	
	LDA	CRNTCH	CURRENT CHAR
	SKE	CARET	CAR RETURN
	BRU	A75	
	LDA	INCHCT	INPUT CHAR CNT
	STA	IICHCT	INITIAL INPUT CHAR CNT
	MIN	ILINCT	INITIAL LINE CNT
	LDA	ILINCT	INITIAL LINE CNT
	STA	LINCT	LINE CNT
	SKN	NSTTRG	END OF STATEMENT TRIGGER
	BRU	CLR1ET	CLEAR 0NE EXIT TRUE
	BRU	CLR1EF	CLEAR 0NE EXIT FALSE
*PRINT ERR0R MARKS			
\$PRERMK SNE	ERSTLS		ERR0R ST0P LIST

A80	JAF	EXIT	EXIT
	TOT	ERSTLS	ERROR STOP LIST
	JAT	A82	A1
	PRC	CARET	CAR RETURN
A81	BRU	EXIT	EXIT
	PRC	SPACE	SPACE
A82	SKR	W0,2	W0
	BRU	A81	A2
	PRC	ERMRK	ERROR MARK
	SKR	WRKBOT	WORK BOTTOM
	BRU	A80	A3
	*PRINT SUMMARY		
	\$PRTSUM SETT	FTTYON,A	
	SNE	D0LST	D0 LOOPS OPEN LIST
	JAF	A84	A2
	PRQ	D0NEQT	D0 NEST ERRORS QT
A83	M0F	D0LST	D0 LOOPS OPEN LIST
	JAF	A84	A2
	SKR	D0L0BM	D0 LOOPS OPEN BOTTOM
	SKR	D0L0BM	D0 LOOPS OPEN BOTTOM
	BNG	ZER0	ZER0
	ETR	ABSATM	ALL BUT SIGN AND TAG MASK
	STA	W0,2	W0
	JRS	PR5DEC	PRINT 5 DEC
	JRS	TSE0LP	TEST FOR E0L PRINT
	BRU	A83	A1
A84	M0F	LBLLST	LBL LIST
	JAF	A85	A3
	SKA	PDEFM	PREVIOUSLY DEF LBL MASK
	SKA	MPYDLM	MULTIPLY DEF LBL MASK
	FET	W0,2	W0
	SKR	WRKBOT	WORK BOTTOM
	BRU	A84	A2
A85	SNE	W0RKLT	WORK LIST
	JAF	0PCNST	OUTPUT CONSTANTS
	PRQ	LBLERQ	LABLING ERRORS QT
A86	JRS	PR5DEC	PRINT 5 DEC
	JRS	TSE0LP	TEST FOR E0L PRINT
	SNE	W0RKLT	WORK LIST

JAT A86  
 \*OUTPUT CONSTANTS  
 \$OPCNST CNT FXCLST  
 CNT FLCLST  
 ADD W1,2  
 SOB WØRKLT  
 BAF ABSLØP  
 ADR CØDLST  
 CAR FXCLST  
 ADR CØDLST  
 CAR FLCLST  
 JRS PCHCØD

\*ALLOCATE MEMORY  
 \$ALCMEM FET ZERO  
 LDA GLSBØT  
 SUB GLSBAS  
 RSH 1  
 SOB WØRKLT  
 ADD DMYCNT  
 SOB WØRKLT  
 ADD MXTPCT  
 SOB WØRKLT  
 STA PRØGBK  
 CLA  
 STA CØMBRK

\$CØMALC SETF FTTYØN,A  
 LDA =2  
 SKN FMAP  
 STA CØUTF  
 RSV LBLLST

\*CØMMØN ALLØCATION  
 SNE CØLIST  
 JAF PRØGAL  
 PRQ CMALQT  
 A87 LCØ CØLIST  
 JAF CØMØVL  
 JRS CMEMRQ

A4  
 FX CØNST LIST  
 FL CØNST LIST  
 W1  
 WØRK LIST  
 ABS LØP  
 CØDE LIST  
 FX CØNST LIST  
 CØDE LIST  
 FL CØNST LIST  
 PUNCH CØDE

ZERØ  
 GLØBAL SPRØG BØTTØM  
 GLØBAL SPRØG BASE  
 1  
 WØRK LIST  
 DMY CNT  
 WØRK LIST  
 MAX TEMP CNT  
 WØRK LIST  
 PRØGRAM BREAK  
 CØMMØN BREAK

(TEMP STØRAGE FØR DEBUG INFØ)

CØMMØN LIST  
 PRØGRAM ALLØCATION  
 CØMMØN ALLØCATION QT  
 CØMMØN LIST  
 CØMMØN ØVERLAY  
 CALC MEMORY REQUIRED



	LDA	W0,2	W0
	ADM	C0MBRK	C0MM0N BR0AK
	LDA	ZER0	ZER0
	SUB	C0MBRK	C0MM0N BR0AK
	STA	W0,2	W0
A88	ADR	QHDL5	EQUIV H0LD LIST
	C0F	QDATLS	EQUIV DATA LIST
	JAF	A89	A3
	SER	QHDL5	EQUIV H0LD LIST
	JAF	A88	A2
	BNG	TW0	TW0
	FET	W1,2	W1
	ADR	MX0VNC	MAX EQUIV IN C0MM0N
	JRS	PR0C0G	PR0CESS EQUIV GR0UP
A89	JRS	SETPUB	SET AND PUBLISH
A90	ADR	QDATLS	EQUIV DATA LIST
	C0F	QHDL5	EQUIV H0LD LIST
	JAT	A90	A4
	BRU	A87	A1
	*C0MM0N 0VERLAY		
	\$C0M0VL	FET 0NE	0NE
	LDA	ZER0	ZER0
	SUB	C0MBRK	C0MM0N BR0AK
	S0B	W0RKLT	W0RK LIST
	JRS	PR0G0P	PR0CESS EQUIV 0UTPUT
	*PR0GRAM ALL0CATION		
	\$PR0GAL	PRQ PR0GQT	PR0GRAM ALL0CATION QT
A91	SNE	QDATLS	EQUIV DATA LIST
	JAF	A94	A4
	RSV	QHDL5	EQUIV H0LD LIST
	FET	SIGN	
A92	LC0	QDATLS	EQUIV DATA LIST
	JAF	A93	A2
	MC0	QHDL5	EQUIV H0LD LIST
	T0T	QDATLS	EQUIV DATA LIST
	M0N	QHDL5	EQUIV H0LD LIST
	SKG	W0,2	W0
	BRU	A92	A1
	STA	W0,2	W0

A93	BRU	A92	A1
	FET	PR0GBK	PR0GRAM BREAK
	ADR	PR0GBK	PR0GRAM BREAK
	JRS	PR0CQG	PR0CESS EQUIV GR0UP
	RLS	QDATLS	EQUIV DATA LIST
	BRU	A91	A3
A94	FET	P0SFSC	P0S FULL SCALE
	FET	NEGFS	NEG FULL SCALE
	JRS	PR0Q0P	PR0CESS EQUIV 0UTPUT
	RLS	Q0PTLS	EQUIV 0PUT LIST
	FIL	NMARKR	END MARKER
*ARRAY ALL0CATION			
ARAYAL	EQU	*	
A95	LCF	AYYLST	ARRAY LIST
	JAF	A96	A2
	JRS	PUBALC	PUBLISH AND ALL0CATE
	FIL	CTL2	CENTRAL 2
*SAVE ARRAY INF0 F0R DEBUG			
	FET	0NE	
A95A	MIN	ARYT0P	
	LDA*	ARYT0P	
	SKN	VARBAS	
	S0B	LBLLST	
	SKR*	WRKB0T	
	BRU	A95A	
	LDA	VARBAS	
	SKN	VARBAS	
	S0B	LBLLST	
	LDA	THREE	
	ADM	ARYT0P	
	SKR	WRKB0T	
	BRU	A95	
A96	FIL	NMARKR	END MARKER
*SCALAR ALL0CATION			
SCALAL	FET	PR0GBK	PR0GRAM BREAK
	ADR	FXSLST	FX SCALAR LIST
	JRS	PR0SCA	PR0CESS SCALARS
	FIL	NMARKR	END MARKER
	FET	PR0GBK	PR0GRAM BREAK

	ADR	FLSLST	FL SCALAR LIST
	JRS	PR0SCA	PROCESS SCALARS
	FIL	NMARKR	END MARKER
	ADR	C0DLST	C0DE LIST
	CAR	W0RKLT	W0RK LIST
	FIL	PR0GBK	PR0GRAM BREAK
	FIL	C0MBRK	C0MM0N BREAK
	JRS	PCHC0D	PUNCH C0DE
*PROCESS LINKAGES			
	PR0CLK	SNE	GL0BAL SPR0G LIST
		JAF	A97A
		PRQ	SUBPR0GRAMS REQUIRED 0T
A97		LC0	GL0BAL SPR0G LIST
		JAF	A97A
		PRC	SPACE
		PRC	SPACE
		PRQ	SYMB0LIC CENTRAL
		JRS	TEST F0R E0L PRINT
		FIL	CENTRAL 1
		FIL	CENTRAL 2
		BRU	A1

*PASS VARIABLE NAMES TO RUN-TIME		
A97A	SKN	FDBUG
	BRU	A97B
	CNT	LBLLST
	BAF	BLKL0P
	ADR	C0DLST
	CAR	LBLLST
	BRU	TRMCPL
A97B	FIL	NMARKR

*TERMINATE C0MPILE			
\$TRMCPL	JRS	PCHC0D	PUNCH C0DE
	SKN	CKSMCT	CHECK SUM CNT
	JRS	PCHFIN	PUNCH FINISH

	LDA	PRGEND	PROGRAM END WORD
	WI0	0UTFLE	
	LDA	=1	
	STA	C0UTF	CMMD OUTPUT FILE
	PRC	CARET	
	PRC	CARET	
	CLA		
	STA	LINETP	LINES THIS PAGE
	LDA	INSTCT	INPUT START COUNTER
	SKG	INCHCT	INPUT CHAR COUNTER
	BRU	A98	TERMINATE COMPILE
	SKN	LASTST	=1=LAST STATEMENT OF FILE
	BRU	A98A	
	BRU	A98	
	SKN	FE0F	
	BRU	A98A	
	BRU	A98	E0F READ
A98A	SKN	BP2	=1=LISTING DESIRED
	BRU	STRCMP	START COMPL
	PRQ	10CRQT	TEN CARRIAGE RETURNS
	BRU	STRCMP	
A98	LDA	E0FWRD	END OF FILE WORD
	WI0	0UTFLE	
	WI0	0UTFLE	
	BRS	17	
	LDA	0NES	
	STA	INFLE	
	STA	0UTFLE	
	TC0	BELL	
	BRU	CS	COMMAND START
*			
*PROCESS	EQUIV	GROUP	
PR0CQG	EQU	*	
A99	LC0	QHDLS	EQUIV HOLD LIST
	JAF	A100	A2
	T0T	QHDLS	EQUIV HOLD LIST
	JRS	CMEMRQ	CALC MEMORY REQUIRED
	LDA	W4,2	W4
	ADD	W5,2	W5

	SUB	W2,2	W2
	STA	W2,2	W2
	ADM	W0,2	W0
	LDA*	W3,2	W3
	SKG	W0,2	W0
	LDA	W0,2	W0
	STA*	W3,2	W3
	SWT	W1,2	W1
	M0N	Q0PTLS	EQUIV 0PUT LIST
	M0N	Q0PTLS	EQUIV 0PUT LIST
	M0N	Q0PTLS	EQUIV 0PUT LIST
	BRU	A99	A1
A100	RLS	QHDL5	EQUIV H0LD LIST
	BRU	CLR3AE	CLEAR THREE AND EXIT
	*PROCESS EQUIV	OUTPUT	
	PR0Q0P	EQU	*
A101	T0T	Q0PTLS	EQUIV 0PUT LIST
	JAF	CLR2AE	CLEAR TWO AND EXIT
	LC0	Q0PTLS	EQUIV 0PUT LIST
	LDA	W2,2	W2
	SKG	CTL1	CENTRAL 1
	BRU	A102	A2
	LDA	W1,2	W1
	SKG	CTL2	CENTRAL 2
	BRU	A103	A3
A102	PRC	SPACE	SPACE
	PRC	SPACE	SPACE
	PRQ	ERR0RQ	ERR0R QT
	JRS	L0DCTL	L0AD CENTRAL
	JRS	PRTSYM	PRINT SYMBOL
	BRU	A101	A1
A103	FET	CTL2	CENTRAL 2
	JRS	SETPUB	SET AND PUBLISH
	BRU	A101	A1
	*SET AND PUBLISH		
SETPUB	FET	W1,2	W1
	JRS	L0DCTL	L0AD CENTRAL
	FET	W0,2	W0
	JRS	PUBLSH	PUBLISH

	JAF	CLR2AE	CLEAR TWO AND EXIT
	JRS	CFR0FS	CORRECT FOR OFFSET
A104	JRS	PRPNTR	PROCESS PNTR
	LDA	ALLD0L	ALL DOLLARS
	LDB	W0,2	W0
	STD*	ADRPT0	ADDRESS POINTED TO
	BRU	CLR1AE	CLEAR ONE AND EXIT
	*PUBLISH AND ALLCATE		
PUBALC	FET	PR0GBK	PROGRAM BREAK
	JRS	PUBLSH	PUBLISH
	JAF	XTF	EXIT FALSE
	JRS	CMEMRQ	CALC MEMORY REQUIRED
	LDA	W0,2	W0
	XMA	PR0GBK	PROGRAM BREAK
	ADM	PR0GBK	PROGRAM BREAK
	STA	W0,2	W0
	JRS	CFR0FS	CORRECT FOR OFFSET
	LDA	W1,2	W1
	STA	CTL2	CENTRAL 2
	BRU	CLR2ET	CLEAR TWO EXIT TRUE
	*PROCESS SCALARS		
PR0SCA	EQU	*	
A105	LCF*	W0,2	W0
	JAF	CLR1AE	CLEAR ONE AND EXIT
	JRS	PUBALC	PUBLISH AND ALLCATE
	JAT	A106	A2
	LDX	W0,2	W0
	LDA	T0P,2	T0P
	SUB	BASE,2	BASE
	RSH	0,2	0
	S0B	C0DLST	C0DE LIST
	FIL	CTL2	CENTRAL 2
A106	LC0*	W0,2	W0
	*PASS SCALAR INFO ON TO RUNTIME		
	SKN	VARBAS	
	BRU	**2	
	BRU	A105	DUMMY SYMBOL, SKIP
	LDA	CTL1	
	S0B	LBLLST	

	LDA	CTL2	
	S0B	LBLLST	
	LDA	VARBAS	VARIABLE BASE ADDRESS
	S0B	LBLLST	
	BRU	A105	A1
	*CORRECT FOR OFFSET		
CFR0FS	SWT	W1,2	W1
	S0T	AYIDFG	ARRAY ID FLAG
	JAF	EXIT	EXIT
	FET	W0,2	W0
	S0F	FLFLAG	FL FLAG
	BNG	TW0	TW0
	JAF	A107	A1
	ADM	W0,2	W0
A107	LDA	W2,2	W2
	ETR	SIGNBT	SIGN BIT
	XMA	W2,2	W2
	SUB	W0,2	W0
	ETR	ADRMSK	ADR MASK
	ADM	W2,2	W2
	BRU	CLR1AE	CLEAR ONE AND EXIT
	*CALC MEMORY REQUIRED		
CMEMRQ	JRS	IDCLAS	ID CLASSIFY
	JAF	A108	A1
	S0T	AYIDFG	ARRAY ID FLAG
	JAT	A110	A3
	S0F	SCAIDF	SCALAR ID FLAG
	JAT	A109	A2
	MC0	ERSYLS	ERROR SYMBOL LIST
	FET	ZER0	ZER0
	BRU	EXIT	EXIT
A108	JRS	RGSCID	REGISTER SCALAR ID
A109	S0F	FLFLAG	FL FLAG
	FET	0NE	0NE
	BRU	A111	A4
A110	FET	W0,2	W0
	S0F	FLFLAG	FL FLAG
	BNG	THREE	THREE
A111	JAF	EXIT	EXIT

	ADM	WO,2	WO
	BRU	EXIT	EXIT
*PUBLISH			
PUBLISH	LDA	CTL1	CENTRAL 1
	SKE	ALLD0L	ALL DOLLARS
	BRU	A112	A1
	BRU	CLR1EF	CLEAR ONE EXIT FALSE
A112	PRC	SPACE	SPACE
	PRC	SPACE	SPACE
	SER	GDLIST	GLOBAL DMY LIST
	JAT	A114	A4
	SER	ERSYLS	ERROR SYMBOL LIST
	JAF	A116	A2
A113	SETT	FTTY0N,A	
	PRQ	ERR0RQ	ERROR QT
	BRU	A115	A5
A114	PRQ	DYQT	DUMMY QT
A115	JRS	PRTSYM	PRINT SYMBOL
	LDA	=1	
	STA	VARBAS	
	SETF	FTTY0N,A	
	LDA	=2	
	SKN	FMAP	
	STA	C0UTF	
	CLA		
	STA	CTL2	CENTRAL 2
	BRU	CLR2EF	CLEAR TWO EXIT FALSE
A116	FET	WO,2	WO
	STA	VARBAS	VARIABLE BASE
	LSH	9	11
	STA	WO,2	WO
	FET	FIVE	FIVE
	JRS	P0CTAL	PRINT OCTAL
	JRS	PRTSYM	PRINT SYMBOL
	BRU	CLR1ET	CLEAR ONE EXIT TRUE
*PRINT SYMBOL			
PRTSYM	PRC	SPACE	SPACE
	PRQ	SYMCEN	SYMBOLIC CENTRAL
PRTSY7	JRS	TSE0LP	TEST FOR EOL PRINT



BRU	EXIT	EXIT
*TEST FOR EOL PRINT		
TSEOLP LDA	SIXTY	SIXTY
SKG	CHTSLN	CHARS THIS LINE
PRC	CARET	CAR RETURN
BRU	EXIT	EXIT
*FX EXP XLATE		
\$FXPXL FET	ZERØ	ZERØ
MØN	MØDLST	MØDE LIST
JRS	XPMDX	EXP BY MØDE XLATE
*MØDE RELEASE		
\$MØDREL SKR	MØDBØT	MØDE BØTTØM
BRU	EXIT	EXIT
*EXP REMEMBER MØDE XLATE		
\$EXPXLT JRS	XPSCAN	EXP SCAN
*SET MØDE THEN GEN		
\$SETMTG JRS	MØDSET	MØDE SET
BRU	SCAEGE	SCRIPT AND EXP GEN
*EXP BY MØDE XLATE		
\$XPMDX JRS	XPSCAN	EXP SCAN
*SCRIPT AND EXP GEN		
\$SCAEGE JRS	SCRGEN	SCRIPT GEN
BRU	XPBYMG	EXP BY MØDE GEN
END		

4C1 IDENT 7-17 COMPILER, PART 4

BAM	0PD	10000000B,1,1
CSF	0PD	10100000B,1,1
CSA	0PD	10200000B,1,1
JAT	0PD	10300000B,1,1
JAF	0PD	10400000B,1,1
TRY	0PD	10500000B,1,1
GSF	0PD	10600000B,1,1
STD	0PD	10700000B,1,1
FET	0PD	11000000B,1,1
SWT	0PD	11100000B,1,1
JRS	0PD	11200000B,1,1
BIF	0PD	11300000B,1,1
MC0	0PD	11400000B,1,1
S0L	0PD	11500000B,1,1
S0F	0PD	11600000B,1,1
RSV	0PD	11700000B,1,1
CNT	0PD	12000000B,1,1
RLS	0PD	12100000B,1,1
T0T	0PD	12200000B,1,1
FIL	0PD	12300000B,1,1
M0N	0PD	12400000B,1,1
LDP	0PD	12500000B,1,1
BLF	0PD	12600000B,1,1
SAL	0PD	12700000B,1,1
PUL	0PD	13000000B,1,1
BNG	0PD	13100000B,1,1
BAF	0PD	13200000B,1,1
CAR	0PD	13300000B,1,1
PL0	0PD	13400000B,1,1
CAC	0PD	13500000B,1,1
CIC	0PD	13600000B,1,1
FIC	0PD	13700000B,1,1
FAC	0PD	14000000B,1,1
B0P	0PD	14100000B,1,1

```

M0F 0PD 14200000B,1,1
SNE 0PD 14300000B,1,1
SER 0PD 14400000B,1,1
LCF 0PD 14500000B,1,1
FIP 0PD 14600000B,1,1
ADR 0PD 14700000B,1,1
E0S 0PD 15000000B,1,1
S0B 0PD 15100000B,1,1
S0T 0PD 15200000B,1,1
SDP 0PD 15300000B,1,1
ADP 0PD 15400000B,1,1
REC 0PD 15500000B,1,1
C0F 0PD 15600000B,1,1
LC0 0PD 15700000B,1,1
PRC 0PD 16000000B,1,1
PRQ 0PD 16100000B,1,1
S0C 0PD 16200000B,1,1
BIM 0PD 16300000B,1,1
FEX 0PD 16400000B,1,1

```

\*

\*

```

BEQ  MACR0  D  BRU IF EQUAL; SKE D(1); BRU **2; BRU D(2); ENDM
BNE  MACR0  D  BRU IF NOT EQUAL; SKE D(1); BRU D(2); ENDM
MOVE MACR0  D
      IF    'D(1$2)!'*!0'          ZERO
      CL,D(3)
      ELSF  1
      LD,D(3) D(1)
      ENDF
      ST,D(3) D(2)
      ENDM

```

\*

```

STZ  MACR0  A  STORE ZERO; CLA ; STA A(1) ; ENDM

```

\*

```

*SPR0G ARG SEQ SCAN
$SPASQS RSV      SPARGL          SPR0G ARGUMENT LIST
A117  TRY        UNSCAS          UNSCRIPTED ARRAY SCAN
      JAT        A118            A2
      JRS        XPSCAN          EXP SCAN

```

A118	MØN	SPARGL	SPRØG ARGUMENT LIST
	CSA	CØMMA	CØMMA
	JAT	A117	A1
	BRU	EXIT	EXIT
*EXP SCAN			
\$XPSCAN	RSV	SUMLST	SUM LIST
	RSV	PRØDLS	PRØD LIST
	JRS	MINUFS	MINUS FACTØR SCAN
	JAT	A119	A1
	CSA	PLUS	PLUS
	JRS	FACTSC	FACTØR SCAN
A119	JRS	PLUSFS	PLUS FACTØR SCAN
	JAT	A120	A2
	JRS	MINUFS	MINUS FACTØR SCAN
	JAF	A122	A4
A120	SWT	W1,2	W1
	MØN	SUMLST	SUM LIST
A121	MØN	SUMLST	SUM LIST
	JRS	PLUSFS	PLUS FACTØR SCAN
	JAT	A121	A3
	JRS	MINUFS	MINUS FACTØR SCAN
	JAT	A121	A3
	PLØ	SUMTPX	SUM TERM PLEX
	CIC	SUMLST	SUM LIST
A122	RLS	SUMLST	SUM LIST
	RLS	PRØDLS	PRØD LIST
	BRU	EXIT	EXIT
*PLUS FACTØR SCAN			
\$PLUSFS	CSA	PLUS	PLUS
	JAF	XTF	EXIT FALSE
	JRS	FACTSC	FACTØR SCAN
	BRU	EXITRU	EXIT TRUE
*MINUS FACTØR SCAN			
\$MINUFS	CSA	MINUS	MINUS
	JAF	XTF	EXIT FALSE
	JRS	FACTSC	FACTØR SCAN
	SØF	ADDRFG	ADDRESSABLE FLAG
	JAT	A123	A1
	FIC	MINNAP	MINUS NØNADDRESSABLE PLEX

A123	BRU	EXITRU	EXIT TRUE
	FIC	MINUAP	MINUS ADDRESSABLE PLEX
	BRU	EXITRU	EXIT TRUE
*FACTOR	SCAN		
*FACTSC	JRS	ELETSC	ELEMENT SCAN
A124	CSA	ASTRSK	ASTERISK
	JAF	A128	A5
	CSA	ASTRSK	ASTERISK
	JAF	A125	A2
	JRS	ELETSC	ELEMENT SCAN
	FIP	EXPTPX	EXPON TERM PLEX
	BRU	A124	A1
A125	M0N	PR0DLS	PR0D LIST
	JRS	ELETSC	ELEMENT SCAN
A126	CSA	ASTRSK	ASTERISK
	JAF	A127	A4
	CSA	ASTRSK	ASTERISK
	JAF	A125	A2
	JRS	ELETSC	ELEMENT SCAN
	FIP	EXPTPX	EXPON TERM PLEX
	BRU	A126	A3
A127	M0N	PR0DLS	PR0D LIST
	PL0	PR0DTP	PR0D TERM PLEX
	CIC	PR0DLS	PR0D LIST
A128	CSA	VRGULE	VIRGULE
	JAF	EXIT	EXIT
	JRS	ELETSC	ELEMENT SCAN
A129	CSA	ASTRSK	ASTERISK
	JAF	A131	A8
	CSA	ASTRSK	ASTERISK
	JAF	A130	A7
	JRS	ELETSC	ELEMENT SCAN
	FIP	EXPTPX	EXPON TERM PLEX
	BRU	A129	A6
A130	FIP	DIVTPX	DIV TERM PLEX
	BRU	A125	A2
A131	FIP	DIVTPX	DIV TERM PLEX
	BRU	A124	A1
*VAR	SCAN		

\$VARSCN	JRS	IDSCAN	ID SCAN
	JAF	RGSCID	REGISTER SCALAR ID
	S0F	SCAIDF	SCALAR ID FLAG
	JAT	EXIT	EXIT
	S0T	AYIDFG	ARRAY ID FLAG
	JAT	AYSCAN	ARRAY SCAN
	BRU	INCNFA	ID CONFLICT FAIL
*ELEMENT SCAN			
\$ELETSC	S0C	LETRFG	LETTER FLAG
	JAT	A132	A1
	CSA	LPAREN	L PAREN
	JAF	CSTSCN	CONST SCAN
	JRS	XPSCAN	EXP SCAN
	CSF	RPAREN	R PAREN
	BRU	EXIT	EXIT
A132	JRS	IDSCAN	ID SCAN
	JAT	A133	A3
	CSA	LPAREN	L PAREN
	JAF	RGSCID	REGISTER SCALAR ID
	B0P	GLSLST	GLOBAL SPR0G LIST
	JRS	STDMEV	STANDARD MODE EVAL
	M00	GLSLST	GLOBAL SPR0G LIST
	SER	GDLIST	GLOBAL DMY LIST
	JAT	INCNFA	ID CONFLICT FAIL
	BRU	FCTSC	FUNCTION SCAN
A133	S0F	SCAIDF	SCALAR ID FLAG
	JAT	EXIT	EXIT
	S0T	AYIDFG	ARRAY ID FLAG
	JAT	AYSCAN	ARRAY SCAN
	CSF	LPAREN	L PAREN
*FUNCTION SCAN			
\$FCTSC	JRS	SPASQS	SPR0G ARG SEQ SCAN
	CSF	RPAREN	R PAREN
	PL0	SPAGPX	SPR0G ARG GROUP PLEX
	CAC	SPARGL	SPR0G ARGUMENT LIST
	RLS	SPARGL	SPR0G ARGUMENT LIST
	FIC	FCTTPX	FUNCTION TERM PLEX
	BRU	EXIT	EXIT
*ARRAY SCAN			

\$AYSCAN	CSA	LPAREN	L PAREN
	JAT	A134	A6
	BRU	NUØFSF	NUMBER ØF SUBSCRIPTS FAIL
A134	RSV	DIMLST	DIMENS LIST
	FET	WØ,2	WØ
	BNG	FØUR	FØUR
	PUL	DIMLST	DIMENS LIST
A135	RSV	SCREXL	SCRIPT EXP LIST
A136	JRS	XPSCAN	EXP SCAN
	MØN	SCREXL	SCRIPT EXP LIST
	CSA	CØMMA	CØMMA
	JAT	A136	A2
	CSF	RPAREN	R PAREN
	MØF	SCREXL	SCRIPT EXP LIST
	JRS	GRTEFX	GRNTEE FIXED
A137	SNE	SCREXL	SCRIPT EXP LIST
	JAF	A138	A4
	MØF	DIMLST	DIMENS LIST
	JAF	NUØFSF	NUMBER ØF SUBSCRIPTS FAIL
	FIC	MPYPLX	MPY PLEX
	MØF	SCREXL	SCRIPT EXP LIST
	JRS	GRTEFX	GRNTEE FIXED
	FIC	ADDPLX	ADD PLEX
	BRU	A137	A3
A138	SNE	DIMLST	DIMENS LIST
	JAT	NUØFSF	NUMBER ØF SUBSCRIPTS FAIL
	RLS	SCREXL	SCRIPT EXP LIST
	RLS	DIMLST	DIMENS LIST
	SØT	MINUSF	MINUS FLAG
	JAT	A138A	
	SØF	ADDRFG	ADDRESSABLE FLAG
	JAT	A139	A5
A138A	MØN	SCRLST	SCRIPT LIST
	JRS	FXTMKR	FX TEMP MAKER
	FET	WØ,2	WØ
	MØN	SCRLST	SCRIPT LIST
A139	FIC	SCRAPX	SINGLY SCRIPTED ARRAY PLEX
	BRU	EXIT	EXIT
*SCALAR ID SCAN			

\$SCLIDS	JRS	IDSCAN	ID SCAN
	JAF	RGSCID	REGISTER SCALAR ID
	S0F	SCAIDF	SCALAR ID FLAG
	JAT	EXIT	EXIT
	BRU	INCNFA	ID CONFLICT FAIL
*UNSCRIPTED	ARRAY SCAN		
\$UNSCAS	S0C	LETRFG	LETTER FLAG
	JAF	FAIL	FAIL
	JRS	IDSCAN	ID SCAN
	JAF	FAIL	FAIL
	S0T	AYIDFG	ARRAY ID FLAG
	JAF	FAIL	FAIL
	CSA	LPAREN	L PAREN
	JAT	FAIL	FAIL
	FET	WO,2	WO
	MIN	WO,2	WO
	JRS	L0DCTL	LOAD CENTRAL
	JRS	RGFXCS	REGISTER FX CONST
	FIC	SCRAPX	SINGLY SCRIPTED ARRAY PLEX
	BRU	EXIT	EXIT
*GRNTEE	FIXED		
\$GRTEFX	S0F	FLFLAG	FL FLAG
	JAF	EXIT	EXIT
	FAC	FIXTPX	FIX TERM PLEX
	BRU	EXIT	EXIT
*ID SCAN			
\$IDSCAN	JRS	SYMBSC	SYMBOL SCAN
*ID CLASSIFY			
\$IDCLAS	SER	L0CDML	LOCAL DMY LIST
	JAT	A140	A2
	SER	FXSLST	FX SCALAR LIST
	JAT	EXITRU	EXIT TRUE
	SER	FLSLST	FL SCALAR LIST
	JAT	EXITRU	EXIT TRUE
	SER	AYYLST	ARRAY LIST
	JAT	A140	A2
	SER	GLSLST	GLOBAL SPR0G LIST
	JAT	A140	A2
	SER	L0CSPL	LOCAL SPR0G LIST



A140	JAF	XTF	EXIT FALSE
	JRS	STDMEV	STANDARD MØDE EVAL
	BRU	EXITRU	EXIT TRUE
	*STANDARD MØDE	EVAL	
	\$STDMEV LDA	CTL1	CENTRAL 1
	LRSH	16	
	ETR	CH4MSK	CHAR 4 MASK
	CAX		
	LDA	CHTTBL,2	CHAR TRANSL TABLE
	SKA	IJKLMN	IJKLMN FLAG
	BRU	A141	A1
	LDA*	WRKBØT	WØRK BØTTØM
	MRG	MØDEMS	MØDE MASK
	STA*	WRKBØT	WØRK BØTTØM
A141	LDA*	WRKBØT	WØRK BØTTØM
	BRU	EXIT	EXIT
	*SYMBOL SCAN		
	\$SYMBSC SØC	LETRFG	LETTER FLAG
	JAF	ILSYFA	ILLEGAL SYNTAX FAIL
	JRS	PAKSYM	PACK SYMBOL
	STA	CTL1	CENTRAL 1
	JRS	PAKSYM	PACK SYMBOL
	STA	CTL2	CENTRAL 2
A142	JAT	EXIT	EXIT
	JRS	PAKSYM	PACK SYMBOL
	BRU	A142	A1
	*PACK STRING		
	\$PAKSTR LDA	UBLKFG	USE BLANKS FLAG
	STA	SCANMD	SCAN MØDE
	LDA	NØTCRF	NØT CAR RETURN FLAG
	JRS	PAK3CW	PACK 3 CHARS/WØRD
	LDB	SKBLFG	SKIP BLANKS FLAG
	STB	SCANMD	SCAN MØDE
	BRU	EXIT	EXIT
	*PACK SYMBOL		
	\$PAKSYM LDA	LTRDGF	LETTER ØR DIGIT FLAG
	*PACK 3 CHAR/WØRD		
	\$PAK3CW STA	PAKTP3	TERMINAL CHARACTER INDICATOR
	CLA		

	STA	PAKTP1	CHARACTER COUNTER
	STA	PAKTP2	WORD ASSEMBLY REGISTER
A143	S0C	PAKTP3	
	JAF	A144	
	LDX	PAKTP1	
	EXU	CHR3ST,2	
	ETR	CHMT3B,2	
	ADM	PAKTP2	
	CLA		
	STA	CRNTCH	
	MIN	PAKTP1	
	LDA	PAKTP1	
	SKE	*3	
	BRU	A143	
	LDA	PAKTP2	
	BRU	XTF	
A144	LDX	PAKTP1	PACK TEMP 1
	CLA		PACK BLANKS
	MRG	PAKTP2	PACK TEMP 2
	BRU	EXITRU	EXIT TRUE
*LBL COMMA SCAN			
\$LBCOMA JRS	LBSCAN		LBL SCAN
	COMMA		COMMA
	BRU	EXIT	EXIT
*LBL SCAN			
\$LBSCAN JRS	INSCAN		INTEGER SCAN
	MAXLBL		MAX LBL ALLOWED
	ILNUFA		ILLEGAL NUMBER FAIL
	CTL2		CENTRAL 2
	ZER0		ZER0
	ILNUFA		ILLEGAL NUMBER FAIL
	REGLBL		REGISTERELBL
*CONST SCAN			
\$CSTSCN JRS	DGCVIN		DIGIT CONV INITIAL
	CLAB		
	STD	CHARA	CHARACTERISTIC A
	S0C	DIGTFL	DIGIT FLAG
	JAF	A145	A1
	JRS	DGCVSC	DIGIT CONV SCAN

	LDP	DGSCTA	DIGITS SCANED CNT A
	STD	CHARA	CHARACTERISTIC A
	CSA	PERIOD	PERIOD
	JAT	A146	A3
	CSA	E	E
	JAT	A147	A4
*HOLLERITH	LITERAL	XLATE	
	CSA	H	
	JAF	A145A	
HOLIT	LDA	CTL2	
	STA	HOLIT1	CT
	SUB	ONE	
	SKG	FIVE	
	BRU	*+2	
	BRU	A145A	NOT A HOL LIT
	STA	HOLIT9	CT TO TEST LATER FL OR FX
	LDA	CTL1	
	SKE	ZERO	
	BRU	A145A	
	STA	CRNTCH	CURRENT CHAR
	LDB	UBLKFG	USED BLANKS FLAG
	STB	SCANMD	SCAN MODE
	LDA	ONE	
	STA	HOLIT3	
HOLIT8	STB	CTL2	
	LDA	TWO	
	STA	HOLIT2	CT4
HOLIT7	LDA	SPACE	
	SKR	HOLIT1	
	BRU	HOLIT4	
HOLIT5	LRSH	2	
	LDB	HOLIT6	
	LCY	8	
	STB	HOLIT6	
	SKR	HOLIT2	
	BRU	HOLIT7	
	SKR	HOLIT3	
	BRU	HOLIT8	
	STB		CTL1

	LDA	SKBLFG	SKIP BLANKS FLAG
	STA	SCANMD	SCAN MODE
	LDA	H0LIT9	
	SKG	TW0	
	BRU	RGFXCS	REGISTER FIXED CONSTANT
	BRU	RGFLCS	REGISTER FL CONSTANT
H0LIT4	S0C	TCRBFG	TAB-CAR RETURN-BACKSPACE FLAG
	CLB		
	JAF	H0LITA	
	STB	H0LIT1	
	BRU	H0LIT7	
H0LITA	STB	CRNTCH	CURRENT CHAR
	BRU	H0LIT5	
A145A	LDP	CTL1	CENTRAL 1
	S0L	MAXINT	MAX INTEGER ALLOWED
	JAT	ILNUFA	ILLEGAL NUMBER FAIL
	BRU	RGFXCS	REGISTER FX CONST
A145	CSF	PERI0D	PERIOD
	S0C	DIGTFL	DIGIT FLAG
	JAF	ILSYFA	ILLEGAL SYNTAX FAIL
A146	JRS	DGCVSC	DIGIT CONV SCAN
	CSA	E	E
A147	LDP	CHARA	CHARACTERISTIC A
	SDP	DGUCTA	DIGITS USED CNT A
	STD	CHARA	CHARACTERISTIC A
	LDP	CTL1	CENTRAL 1
	JAF	FLCSTM	FL CONST MAKER
	STD	CNSTPA	CONST TEMP A
	JRS	SIGNIS	SIGNED INTEGER SCAN
	LDP	CHARA	CHARACTERISTIC A
	ADP	CTL1	CENTRAL 1
	STD	CHARA	CHARACTERISTIC A
	LDP	CNSTPA	CONST TEMP A
	STD	CTL1	CENTRAL 1
*FL CONST MAKER			
\$FLCSTM EAX		47	57
	N0D	48	60
	STD	CTL1	CENTRAL 1
A148	SKN	CHARB	CHARACTERISTIC B

	BRU	A150	A1
	MIN	CHARB	CHARACTERISTIC B
	LDP	TENTH3	ONE TENTH B-3
	\$BRM	DBFXMU	DOUBLE FX MULT
	NØD	47	57
	STD	CTL1	CENTRAL 1
	EAX	=3,2	=3
A149	BRU	A148	A2
	LDP	TENB4	TEN B4
	\$BRM	DBFXMU	DOUBLE FX MULT
	NØD	47	57
	STD	CTL1	CENTRAL 1
	EAX	4,2	4
A150	SKR	CHARB	CHARACTERISTIC B
	BRU	A149	A3
	SKG	ZERØ	ZERØ
	BRU	RGFLCS	REGISTER FL CONST
	LDE		
	STB	CTL2	CENTRAL 2
	CXA		
	ETR	ADRMASK	ADR MASK
	SKG	CLNEGL	CHARIS LOWER NEG LIMIT
	SKG	CUPØSL	CHARIS UPPER PØS LIMIT
	BRU	RGFLCS	REGISTER FL CONST
	BRU	ILNUFA	ILLEGAL NUMBER FAIL
	*SIGNED INTEGER SCAN		
	\$SIGNIS CSA	MINUS	MINUS
	JAF	A151	A1
	JRS	INSCAN	INTEGER SCAN
	CLAB		
	SDP	CTL1	CENTRAL 1
	STD	CTL1	CENTRAL 1
	BRU	EXIT	EXIT
A151	CSA	PLUS	PLUS
	BRU	INSCAN	INTEGER SCAN
	*INTEGER SCAN		
	\$INSCAN SØC	DIGTFL	DIGIT FLAG
	JAF	ILSYFA	ILLEGAL SYNTAX FAIL
	JRS	DGCVIN	DIGIT CONV INITIAL

JRS	DGCVSC	DIGIT CONV SCAN
LDP	CTL1	CENTRAL 1
SOL	MAXINT	MAX INTEGER ALLOWED
JAT	ILNUFA	ILLEGAL NUMBER FAIL
BRU	EXIT	EXIT
*DIGIT CONV INITIAL		
\$DGCVIN CLAB		
STD	DGSC TA	DIGITS SCANED CNT A
STD	DGUCTA	DIGITS USED CNT A
STD	SCDIGA	SCANED DIGIT A
STD	CTL1	CENTRAL 1
BRU	EXIT	EXIT
*DIGIT CONV SCAN		
DGCVSC EQU	*	
A152	SOC	DIGIT FLAG
	JAF	EXIT
	JRS	CONV ONE DIGIT
	BRU	A1
*CONV ONE DIGIT		
\$CNV1DG LDA	CRNTCH	CRRNT CHAR
MUL	RSH18	RIGHT SHIFTER 18
ETR	CH4MSK	CHAR 4 MASK
SUB	ASCNB	ASCII NUMBER BASE
STA	SCDIGB	SCANED DIGIT B
MIN	DGSC TB	DIGITS SCANED CNT B
CLA		
STA	CRNTCH	CRRNT CHAR
LDP	CTL1	CENTRAL 1
SKA	T0P5BM	T0P 5 BITS MASK
BRU	EXIT	EXIT
LSH	4	4
STD	CTL1	CENTRAL 1
LDP	TENB4E	TEN B4 EXACT
SBRM	DBFXMU	DOUBLE FX MULT
ADP	SCDIGA	SCANED DIGIT A
STD	CTL1	CENTRAL 1
MIN	DGUCTB	DIGITS USED CNT B
BRU	EXIT	EXIT
*DOUBLE FX MULT		

\$DBFXMU	HLT	LDBFXM	
	STD	DPMPYA	DP MPY A
	LDA	CTL2	CENTRAL 2
	CLB		
	RCY	1	1
	MUL	DPMPYA	DP MPY A
	RSH	23	27
	STD	DPMPYC	DP MPY C
	LDA	DPMPYB	DP MPY B
	CLB		
	RCY	1	1
	MUL	CTL1	CENTRAL 1
	RSH	23	27
	ADP	DPMPYC	DP MPY C
	STD	DPMPYC	DP MPY C
	LDA	CTL1	CENTRAL 1
	MUL	DPMPYA	DP MPY A
	ADP	DPMPYC	DP MPY C
	SBRR	DBFXMU	DOUBLE FX MULT
*EXP	REMEMBER	MODE GEN	
\$XP	REMG	JRS	MODE SET
*EXP	BY	MODE GEN	
\$XP	BYMG	JRS	ISTMSA
	JAT	FASGG	IS TERM SIGNED ADDRESSABLE FETCH AND SET SIGN GEN
*SIGNED	CMPLX	BY	MODE GEN
\$SIG	CMG	JRS	ISMODE OK
	JRS	MODE SET	MODE SET
	JRS	SUMTG	SUM TERM GEN
	JRS	SXPSGG	SET EXP SIGN GEN
	JRS	MODEREL	MODE RELEASE
	JAT	EXIT	EXIT
*SET	EXP	MODE GEN	
\$SEX	MG	JRS	ISMODE FL
	JAT	A153	A1
	BLF	FIXSPL	FIX SPR0G LINK
	BRU	EXIT	EXIT
A153	BLF	FL0TSL	FL0AT SPR0G LINK
	BRU	EXIT	EXIT
*FETCH	AND	SET	SIGN GEN

FASSG BIM FTCHP0  
\*SET EXP SIGN GEN

\$SXPSGG LDA W0,2  
SKE NEGTRT  
BRU CLR1AE  
LDA\* M0DB0T  
ADD\* M0DB0T  
ADD CGSGIN  
S0B C0DLST  
BRU CLR1AE

\*TERM BY M0DE GEN

\$TMBYMG JRS ISTMSA  
JAF CTBYMG  
BIM FTCHP0  
BRU EXIT

\*CMLX TERM BY M0DE GEN

\$CTBYMG JRS ISMD0K  
JAT SUMTG  
S0F FLFLAG  
JAF SUMTG  
JRS M0DSET  
JRS SUMTG  
BLF FIXSPL  
BRU M0DREL

\*SUM TERM GEN

\$SUMTG S0T SUMFLG  
JAF PR0DTG  
JRS SUMPSU  
A154 M0F TERMLS  
JAF SUMPEX  
JRS GRSAG  
LDA W1,2  
SKE W2,2  
BRU A155  
BIM ADDP0P  
BRU A156  
A155 BIM SUBP0P  
A156 SKR WRKB0T  
BRU A154

FETCH P0P

W0  
NEGATIVE TRAIT  
CLEAR 0NE AND EXIT  
M0DE B0TT0M  
M0DE B0TT0M  
CHANGE SIGN INST  
C0DE LIST  
CLEAR 0NE AND EXIT

IS TERM SIGNED ADDRESSABLE  
CMLX TERM BY M0DE GEN  
FETCH P0P  
EXIT

IS M0DE 0K  
SUM TERM GEN  
FL FLAG  
SUM TERM GEN  
M0DE SET  
SUM TERM GEN  
FIX SPR0G LINK  
M0DE RELEASE

SUM FLAG  
PR0D TERM GEN  
SUM-PR0D SET UP  
TERM LIST  
SUM-PR0D EXIT  
GRNTEE SIGNED ADDRESSABLE GEN  
W1  
W2  
A1  
ADD P0P  
A3  
SUB P0P  
W0RK B0TT0M  
A2



*PRØD TERM GEN			
\$PRØDTG SØT		PRØDFG	PRØD FLAG
	JAF	FCTTG	FUNCTION TERM GEN
	JRS	SUMPSU	SUM-PRØD SET UP
A157	MØF	TERMLS	TERM LIST
	JAF	SUMPEX	SUM-PRØD EXIT
	JRS	GRSAG	GRNTEE SIGNED ADDRESSABLE GEN
	BIM	MPYPØP	MPY PØP
	EØS	W1,2	W1
	BRU	A157	A5
*SUM-PRØD SET UP			
\$SUMPSU RSV		TERMLS	TERM LIST
	PUL	TERMLS	TERM LIST
	RSV	WØRKLT	WØRK LIST
A158	MØF	TERMLS	TERM LIST
	JAF	A159	A2
	SØF	ADDRFG	ADDRESSABLE FLAG
	JAT	A158	A1
A159	ADR	TERMLS	TERM LIST
	CAR	WØRKLT	WØRK LIST
	MØF	TERMLS	TERM LIST
	BRU	TMBYMG	TERM BY MØDE GEN
*SUM-PRØD EXIT			
\$SUMPEX EØS		W1,2	W1
	RLS	TERMLS	TERM LIST
	BRU	EXIT	EXIT
*FUNCTION TERM GEN			
\$FCTTG SØT		FCTFLG	FUNCTION FLAG
	JAF	DVTMGN	DIV TERM GEN
	PUL	WØRKLT	WØRK LIST
	RSV	SPARGL	SPRØG ARGUMENT LIST
	PUL	SPARGL	SPRØG ARGUMENT LIST
	JRS	SPASQG	SPRØG ARG SEQ GEN
	JRS	ISMØK	IS MØDE ØK
	JRS	SPJPGN	SPRØG JUMP GEN
	JAF	SEXMG	SET EXP MØDE GEN
	BRU	EXIT	EXIT
*SPRØG JUMP GEN			
\$SPJPGN SØT		LSPIDF	LOCAL SPRØG ID FLAG

	JAF	A160	A1
	BNG	TW0	TW0
	BIF	BRMM0P	BRM M0P
	BRU	EXIT	EXIT
A160	BIF	BRM0P	BRM* M0P
	BRU	EXIT	EXIT
	*DIV TERM GEN		
	\$DVTMGN S0T	DIVFG	DIV FLAG
	JAF	EXPNTG	EXP0N TERM GEN
	PUL	W0RKLT	W0RK LIST
	JRS	ISTMSA	IS TERM SIGNED ADDRESSABLE
	JAT	A161	A1
	JRS	CTBYMG	CMPLX TERM BY M0DE GEN
	JRS	SNTBMG	ST0RE IN TEMP BY M0DE GEN
A161	SWT	W2,2	W2
	JRS	TMBYMG	TERM BY M0DE GEN
	E0S	W1,2	W1
	SWT	W1,2	W1
	BIM	DIVP0P	DIV P0P
	E0S	W1,2	W1
	BRU	EXIT	EXIT
	*EXP0N TERM GEN		
	\$EXPNTG S0T	EXPFG	EXP0N FLAG
	JAF	FIXTGN	FIX TERM GEN
	PUL	W0RKLT	W0RK LIST
	JRS	GRNM0D	GRNTEE ADDRESSABLE BY M0DE GEN
	SWT	W1,2	W1
	JRS	GRNM0D	GRNTEE ADDRESSABLE BY M0DE GEN
	BAM	FARGP0	FIRST ARG P0P
	BAM	NEXTAP	NEXT ARG P0P
	BLF	EXPSPL	EXP0N SPR0G LINK
	BRU	EXIT	EXIT
	*FIX TERM GEN		
	\$FIXTGN JRS	M0DSET	M0DE SET
	PUL	W0RKLT	W0RK LIST
	JRS	TMBYMG	TERM BY M0DE GEN
	E0S	W1,2	W1
	BRU	M0DREL	M0DE RELEASE
	*SPR0G ARG SEQ GEN		

\$SPASQG RSV      W0RKLT  
 A162    T0T      SPARGL  
          JAF      A163  
          JRS      GA0MGN  
          BRU      A162  
 A163    RLS      SPARGL  
          T0T      W0RKLT  
          BAM      FARGP0  
 A164    T0T      W0RKLT  
          JAF      A165  
          BAM      NEXTAP  
          BRU      A164  
 A165    RLS      W0RKLT  
          BRU      EXIT  
 \*SCRIPT GEN  
 SCRGEN EXT      \*  
 SCRGEN EQU      \*  
 A166    T0T      SCRLST  
          JAF      EXIT  
          JRS      XPREMG  
          T0T      SCRLST  
          JRS      STFMG  
          BRU      A166  
 \*GRNTEE ADDRESSABLE 0WN M0DE GEN  
 \$GA0MGN JRS      M0DSET  
          JRS      GRNM0D  
          BRU      M0DREL  
 \*GRNTEE ADDRESSABLE BY M0DE GEN  
 \$GRNM0D JRS      ISTMSA  
          JAF      A167  
          LDA      P0STRT  
          SKE      W1,2  
          BRU      A168  
          JRS      ISMD0K  
          JAF      A168  
          SWT      W1,2  
          BRU      CLR1AE  
 A167    JRS      SIGCMG  
          BRU      SNTBMG

W0RK LIST  
 SPR0G ARGUMENT LIST  
 A2  
 GRNTEE ADDRESSABLE 0WN M0DE GEN  
 A1  
 SPR0G ARGUMENT LIST  
 W0RK LIST  
 FIRST ARG P0P  
 W0RK LIST  
 A4  
 NEXT ARG P0P  
 A3  
 W0RK LIST  
 EXIT  
  
 SCRIPT LIST  
 EXIT  
 EXP REMEMBER M0DE GEN  
 SCRIPT LIST  
 ST0RE F0RGET M0DE GEN  
 A1  
  
 M0DE SET  
 GRNTEE ADDRESSABLE BY M0DE GEN  
 M0DE RELEASE  
  
 IS TERM SIGNED ADDRESSABLE  
 A1  
 P0SITIVE TRAIT  
 W1  
 A2  
 IS M0DE 0K  
 A2  
 W1  
 CLEAR 0NE AND EXIT  
 SIGNED CMLX BY M0DE GEN  
 ST0RE IN TEMP BY M0DE GEN

A168	JRS	FASSG	FETCH AND SET SIGN GEN	
	BRU	SNTBMG	STORE IN TEMP BY MODE GEN	
	*GRNTEE	SIGNED ADDRESSABLE GEN		
	\$GRSAG	JRS	ISTMSA	IS TERM SIGNED ADDRESSABLE
		JAT	EXIT	EXIT
		JRS	SNTBMG	STORE IN TEMP BY MODE GEN
		SWT	W1,2	W1
		JRS	TMBYMG	TERM BY MODE GEN
		SWT	W3,2	W3
		SWT	W2,2	W2
		E0S	W3,2	W3
		BRU	EXIT	EXIT
	*STORE IN TEMP BY MODE GEN			
	\$SNTBMG	JRS	TPBMKR	TEMP BY MODE MAKER
		FET	W0,2	W0
		BIM	STRPOP	STORE POP
		BRU	EXIT	EXIT
	*STORE FORGET MODE GEN			
	\$STFMG	BIM	STRPOP	STORE POP
		BRU	MODREL	MODE RELEASE
	*IS TERM SIGNED ADDRESSABLE			
	\$ISTMSA	FET	POSTRT	POSITIVE TRAIT
		SWT	W1,2	W1
A169	S0T	MINUSF	MINUS FLAG	
	JAF	A170	A2	
	PUL	WORKLT	WORK LIST	
	FET	NEGTRT	NEGATIVE TRAIT	
	E0S	W2,2	W2	
	BRU	A169	A1	
A170	S0F	ADDRFG	ADDRESSABLE FLAG	
	JAF	XTF	EXIT FALSE	
	JRS	ISMD0K	IS MODE 0K	
	JAT	EXITRU	EXIT TRUE	
	S0T	FXCSTF	FX CONST FLAG	
	JAF	EXITRU	EXIT TRUE	
	JRS	L0DCTL	LOAD CENTRAL	
	CLB			
	EAX	23	27	
	N0D	23	27	

LDE	CTL1	CENTRAL 1
STD	RGFLCS	REGISTER FL CONST
JRS	EXITRU	EXIT TRUE
BRU		
*IS MØDE FL		
\$ISMDFL LDA	FLFLAG	FL FLAG
BRU	ØKFLTS	ØK=FL TEST
*IS MØDE ØK		
\$ISMØØK LDA*	WRKBØT	WØRK BØTTØM
ETR	MØDEMS	MØDE MASK
*ØK=FL TEST		
\$ØKFLTS SKE*	MØDBØT	MØDE BØTTØM
BRU	XTF	EXIT FALSE
BRU	EXITRU	EXIT TRUE
*MØDE SET		
\$MØDSET LDA*	WRKBØT	WØRK BØTTØM
ETR	MØDEMS	MØDE MASK
SØB	MØDLST	MØDE LIST
BRU	EXIT	EXIT
*DMY TEMP MAKER		
\$DYTPMK FET	DYPTCD	DMY PNTR CØDE
MIN	DMYCNT	DMY CNT
LDA	DMYCNT	DMY CNT
MIN	DMYCNT	DMY CNT
ADD	WØ,2	WØ
STA	WØ,2	WØ
BRU	STDMEV	STANDARD MØDE EVAL
*TEMP BY MØDE MAKER		
\$TPBMKR JRS	FXTMKR	FX TEMP MAKER
LDA*	WRKBØT	WØRK BØTTØM
MRG*	MØDBØT	MØDE BØTTØM
STA*	WRKBØT	WØRK BØTTØM
BRU	EXIT	EXIT
*FX TEMP MAKER		
\$FXTMKR FET	FXTPCD	FX TEMP PNTR CØDE
MIN	TMP CNT	TEMP CNT
LDA	TMP CNT	TEMP CNT
MIN	TMP CNT	TEMP CNT
ADM	WØ,2	WØ

ADD	ONE	ONE
SKG	MXTPCT	MAX TEMP CNT
BRU	EXIT	EXIT
STA	MXTPCT	MAX TEMP CNT
BRU	EXIT	EXIT
*LBL MAKER BRU	GEN	
\$LMBRUG JRS	LBLMAK	LBL MAKER
FET	W0,2	W0
BIF	BRUM0P	BRU M0P
LDA	W0,2	W0
BRU	EXIT	EXIT
*LBL MAKER LBL	GEN	
\$LMLBLG JRS	LBLMAK	LBL MAKER
FET	W0,2	W0
BIF	LBLLOP	LBL L0P
LDA	W0,2	W0
BRU	EXIT	EXIT
*LBL MAKER		
\$LBLMAK FET	MADELB	MADE LBL
BRU	RGRMLB	REGISTER 0R MAKE LBL
*REGISTER LBL		
\$REGLBL SER	LBLBST	LBL LIST
JAT	EXIT	EXIT
FET	CTL2	CENTRAL 2, MOVE LABEL TO BOT OF WORK LIST
*REGISTER 0R MAKE LBL		
\$RGRMLB B0P	LBLBST	LBL LIST
SWT	W1,2	W1
M0N	LBLBST	LBL LIST
LDA	W0,2	W0
BRU	EXIT	EXIT
*REGISTER SCALAR ID		
\$RGSCID FET	ZER0	ZER0
JRS	STDMEV	STANDARD MODE EVAL
S0F	FLFLAG	FL FLAG
JAT	A171	FL0ATING SCALAR
B0P	FXSLST	FX SCALAR LIST
MC0	FXSLST	FX SCALAR LIST
BRU	A172	A1
A171 B0P	FLSLST	FL SCALAR LIST

	MC0	FLSLST	FL SCALAR LIST
A172	SWT	W1,2	W1
	BRU	CLR1AE	CLEAR ONE AND EXIT
	*REGISTER FX CONST		
	\$RGFXCS	SER FXCLST	FX CONST LIST
	JAT	EXIT	EXIT
	B0P	FXCLST	FX CONST LIST
	LDA	CTL2	CENTRAL 2
	S0B	FXCLST	FX CONST LIST
	BRU	EXIT	EXIT
	*REGISTER FL CONST		
	\$RGFLCS	LDP CTL1	CENTRAL 1
	XAB		
	STD	CTL1	CENTRAL 1
	SER	FLCLST	FL CONST LIST
	JAT	EXIT	EXIT
	B0P	FLCLST	FL CONST LIST
	MC0	FLCLST	FL CONST LIST
	BRU	EXIT	EXIT
	*LOAD CENTRAL		
	\$L0DCTL	JRS PRPNTR	PROCESS PNTR
	LDP*	ADRPT0	ADDRESS POINTED TO
	STD	CTL1	CENTRAL 1
	BRU	EXIT	EXIT
	*INITIALIZE ALL LISTS FROM BOTTOM		
	\$INITLS	HLT LINITL	
	L0X	ZER0	
	LDA	=LSTMEM	START OF LIST MEMORY
	STA	FLSBAS	
	BRU	A174	
A173	CAX		
	LDA	NLIST	END OF LIST MEMORY
	STA	BASE,2	BASE
A174	STA	START,2	START
	STA	TOP,2	TOP
	STA	BOTTOM,2	BOTTOM
	CXA		
	ETR	ADRMSK	ADR MASK
	ADD	0NE	0NE

SKE	NOLIST	NUMBER OF LISTS
BRU	A173	A1
SBRR	INITLS	INITIALIZE LISTS
*OVERFLOW FAIL		
\$OVFAIL ADR	OVFLQT	OVERFLOW QT
BRU	SETLSA	SET LAST ACTIVE
*ILLEGAL ALLOCATION FAIL		
\$ILALFA ADR	ALOCAG	ALLOCATION QT
BRU	SETLSA	SET LAST ACTIVE
*ILLEGAL NUMBER FAIL		
\$ILNUFA ADR	NUMBQT	NUMBER QT
BRU	SETLSA	SET LAST ACTIVE
*NUMBER OF SUBSCRIPTS FAIL		
\$NUOFSF ADR	SUBSCQ	SUBSCRIPTS QT
BRU	SETLSA	SET LAST ACTIVE
*ID CONFLICT FAIL		
\$INCNFA ADR	IDECQT	ID DECLARATION QT
*SET LAST ACTIVE		
\$SETLSA LDA	LACHCT	LAST ACTIVE CHAR CNT
BRU	NOTEF	NOTE FAIL
*ILLEGAL SYNTAX FAIL		
\$ILSYFA ADR	SYTAXQ	SYNTAX QT
LDA	INCHCT	INPUT CHAR CNT
*NOTE FAIL		
\$NOTEF SOB	FAILDL	FAIL DATA LIST
M0N	FAILDL	FAIL DATA LIST
BRU	FAIL	FAIL
END		



5C1 IDENT 7-15 PROGRAMMED OPER'S, COMPILER PART 5

BAM	OPD	10000000B,1,1
CSF	OPD	10100000B,1,1
CSA	OPD	10200000B,1,1
JAT	OPD	10300000B,1,1
JAF	OPD	10400000B,1,1
TRY	OPD	10500000B,1,1
QSF	OPD	10600000B,1,1
STD	OPD	10700000B,1,1
FET	OPD	11000000B,1,1
SWT	OPD	11100000B,1,1
JRS	OPD	11200000B,1,1
BIF	OPD	11300000B,1,1
MCB	OPD	11400000B,1,1
SBL	OPD	11500000B,1,1
SBF	OPD	11600000B,1,1
RSV	OPD	11700000B,1,1
CNT	OPD	12000000B,1,1
RLS	OPD	12100000B,1,1
TBT	OPD	12200000B,1,1
FIL	OPD	12300000B,1,1
MEN	OPD	12400000B,1,1
LDP	OPD	12500000B,1,1
BLF	OPD	12600000B,1,1
SAL	OPD	12700000B,1,1
PUL	OPD	13000000B,1,1
BNG	OPD	13100000B,1,1
BAF	OPD	13200000B,1,1
CAR	OPD	13300000B,1,1
PLB	OPD	13400000B,1,1
CAC	OPD	13500000B,1,1
CIC	OPD	13600000B,1,1
FIC	OPD	13700000B,1,1
FAC	OPD	14200000B,1,1
BOP	OPD	14100000B,1,1

M0F	0PD	14200000B,1,1
SNE	0PD	14300000B,1,1
SER	0PD	14400000B,1,1
LCF	0PD	14500000B,1,1
FIP	0PD	14600000B,1,1
ADR	0PD	14700000B,1,1
E0S	0PD	15000000B,1,1
S0B	0PD	15100000B,1,1
S0T	0PD	15200000B,1,1
SDP	0PD	15300000B,1,1
ADP	0PD	15400000B,1,1
REC	0PD	15500000B,1,1
C0F	0PD	15600000B,1,1
LC0	0PD	15700000B,1,1
PRC	0PD	16000000B,1,1
PRQ	0PD	16100000B,1,1
S0C	0PD	16200000B,1,1
BIM	0PD	16300000B,1,1
FEX	0PD	16400000B,1,1
ICT	P0PD	16500000B,1,1
ICTP0P	CI0	ZER0
	SKE	=155B
	BRR	0
	MIN	0
	BRR	0

*FEX P0P		FAIL EXIT 164
FEX	P0PD	16400000B,1,1
FEXP0P	SBRM	P0NTRY
	CXA	
	STA*	SVB0T
	LDA	ITMPCT
	STA	TMPCNT
	LDA	SKBLFG
	STA	SCANMD
	RSV	PLEXLS
	BRU	EXIT
*TRY P0P		TRY 105
TRY	P0PD	10500000B,1,1

P0P ENTRY

SAVE B0TT0M
INITIAL TEMP C0UNT
TEMP CNT
SKIP BLANKS FLAG
SCAN M0DE
PLEX LIST
EXIT

TRYP0P	SBRM	P0NTRY	P0P ENTRY
	SBRM	FEXTYS	FEX-TRY SAVE
	BRU	XTF	EXIT FALSE
	JRS	0,2	0
	LDA	MINU17	MINUS SEVENTEEN
	ADM	SVB0T	SAVE B0TT0M
	BRU	EXITRU	EXIT TRUE
*FEX-TRY SAVE			
FEXTYS	HLT	LFEXTY	
	STA	SAVET1	SAVE TEMP 1
	SAL	FLSLST	FL SCALAR LIST
	SAL	FXSLST	FX SCALAR LIST
	SAL	W0RKLT	W0RK LIST
	SAL	XITLS	EXIT LIST
	LDA	LINECT	LINE CNT
	S0B	SAVELS	SAVE LIST
	LDA	INCHCT	INPUT CHAR CNT
	S0B	SAVELS	SAVE LIST
	LDA	CRNTCH	CRRNT CHAR
	S0B	SAVELS	SAVE LIST
	LDA*	FEXTYS	FEX-TRY SAVE
	S0B	SAVELS	SAVE LIST
	S0B	SAVELS	SAVE LIST
	LDA	SAVET1	SAVE TEMP 1
	MIN*	FEXTYS	FEX-TRY SAVE
	SBRR	FEXTYS	FEX-TRY SAVE
*FAIL			
\$FAIL	LDX	SVB0T	SAVE B0TT0M
	LDA	W0,2	W0
	SUB	0NE	0NE
	STA	FAILEX	FAIL EXIT
	LDA	W1,2	W1
	STA*	FEXTYS	FEX-TRY SAVE
	LDA	W2,2	W2
	STA	CRNTCH	CRRNT CHAR
	LDA	W3,2	W3
	STA	INCHCT	INPUT CHAR CNT
	LDA	W4,2	W4
	STA	LINECT	LINE CNT

	LDA	MINUS5	MINUS FIVE	
	ADM	SVB0T	SAVE B0TT0M	
	REC	XITLS	EXIT LIST	
	REC	W0RKLT	W0RK LIST	
	REC	FXSLST	FX SCALAR LIST	
	REC	FLSLST	FL SCALAR LIST	
	SBRR	FEXTYS	FEX-TRY SAVE	
	*SAVE LIST DATA			
	\$SVLSDT	LDA	N0FLSV	NUMBER 0F FIRST LIST SAVED
		STA	SAVET1	SAVE TEMP 1
		LDX	N0LSVB	NUMBER 0F LISTS SAVED BRXER
A175		SAL*	SAVET1	SAVE TEMP 1
		MIN	SAVET1	SAVE TEMP 1
		BRX	A175	A1
		SBRM	FEXTYS	FEX-TRY SAVE
		BRU	RCFAIL	RECOVER FROM FAIL
		BRU	EXIT	EXIT
	*RECOVER FROM FAIL			
RCFAIL		LDA	FAILEX	FAIL EXIT
		STA*	EXIB0T	EXIT B0TT0M
		LDA	N0LLSV	NUMBER 0F LAST LIST SAVED
		STA	SAVET1	SAVE TEMP 1
		LDX	N0LSVB	NUMBER 0F LISTS SAVED BRXER
A176		REC*	SAVET1	SAVE TEMP 1
		SKR	SAVET1	SAVE TEMP 1
		BRX	A176	A1
		LDA	N0ITSV	NUMBER 0F ITEMS SAVED
		ADM	SVB0T	SAVE B0TT0M
		BRU	EXIT	EXIT
	*SAL P0P		SAVE A LIST 127	
SAL	P0PD		12700000B,1,1	
SALP0P	SBRM		N0NESE	N0 NEST ENTRY
		LDA	START,2	START
		SUB	BASE,2	BASE
		S0B	SAVELS	SAVE LIST
		LDA	T0P,2	T0P
		SUB	BASE,2	BASE
		S0B	SAVELS	SAVE LIST
		LDA	B0TT0M,2	B0TT0M

	SUB	BASE,2	BASE
	SQB	SAVELS	SAVE LIST
	BRU	INDEX	INDEX EXIT
*REC POP		RECOVER	
REC POPD		15500000B,1,1	
RECPBP SBRM		NONESE	NO NEST ENTRY
LDA*		SVBOT	SAVE BOTTOM
ADD		BASE,2	BASE
STA		BOTTOM,2	BOTTOM
SKR		SVBOT	SAVE BOTTOM
LDA*		SVBOT	SAVE BOTTOM
ADD		BASE,2	BASE
STA		TOP,2	TOP
SKR		SVBOT	SAVE BOTTOM
LDA*		SVBOT	SAVE BOTTOM
ADD		BASE,2	BASE
STA		START,2	START
SKR		SVBOT	SAVE BOTTOM
BRU		INDEX	INDEX EXIT
*BAM POP		BUILD BY ADDRESS MODE 100	
BAM POPD		10000000B,1,1	
BAMPBP SBRM		PENTRY	POP ENTRY
STX		BIAMT1	BIM-BAM TEMP 1
BRU		BIMBAM	BIM-BAM
*BIM POP		BUILD INSTRUCTION BY MODE	
BIM POPD		16300000B,1,1	
BIMPBP SBRM		PENTRY	POP ENTRY
STX		BIAMT1	BIM-BAM TEMP 1
JRS		ISMDOK	IS MODE OK
*BIM-BAM			
BIMBAM SBF		FLFLAG	FL FLAG
LDA*		EXIBOT	EXIT BOTTOM
MUL		RSH16	RIGHT SHIFTER 16
ADM		BIAMT1	BIM-BAM TEMP 1
ADM		BIAMT1	BIM-BAM TEMP 1
BIF*		BIAMT1	BIM-BAM TEMP 1
BRU		EXIT	EXIT
*BLF POP		BUILD LINK AND FILE	
BLF POPD		12600000B,1,1	

BLFP0P	SBRM	P0NTRY	P0P ENTRY
	CXA		
	ADD	ST0FLK	START 0F LINKS
	S0B	W0RKLT	W0RK LIST
	BAF	SYSL0P	SYS L0P
	BRU	EXIT	EXIT
*BIF	P0P	BUILD INSTRUCTION AND FILE 113	
BIF	P0PD	11300000B,1,1	
BIFP0P	SBRM	P0NTRY	P0P ENTRY
	CXA		
	XMA*	WRKB0T	W0RK B0TT0M
	S0B	W0RKLT	W0RK LIST
	S0F	PTIDFG	PART ID FLAG
	JAT	A178	A4
	S0T	SCRAFG	SINGLY SCRIPTED ARRAY FLAG
	JAF	A180	A7
	PUL	W0RKLT	W0RK LIST
	BIF	LDXM0P	LDX M0P
	JRS	DYTEST	DMY TEST
	JAF	A177	A3
	MIN	W0,2	W0
	BRU	A179	A5
A177	LDA	W1,2	W1
	MRG	0PTAGB	M0P TAG BIT
	STA	W1,2	W1
	BRU	A180	A7
A178	JRS	DYTEST	DMY TEST
	JAF	A180	A7
A179	LDA	W1,2	W1
	MRG	0PINDB	M0P IND BIT
	STA	W1,2	W1
A180	LDA*	WRKB0T	W0RK B0TT0M
	MUL	RSH18	RIGHT SHIFTER 18
	MUL	KEYVAL	KEY VALUE
	CBA		
	ADD*	WRKB0T	W0RK B0TT0M
	SUB	0NE	0NE
	MUL	RSH14	RIGHT SHIFTER 14
	SKR	WRKB0T	W0RK B0TT0M

LDA*	WRKB0T	WORK BOTTOM
MRG	0NEA14	ONE IN 14TH BIT
BRU	BAFBIF	BAF=BIF
*BAF POP	BUILD ABSOLUTE AND FILE	
BAF POPD	13200000B,1,1	
BAFP0P SBRM	P0NTRY	P0P ENTRY
LDA*	WRKB0T	WORK BOTTOM
MUL	RSH14	RIGHT SHIFTER 14
CXA		
*BAF=BIF		
BAFBIF RSH	10	12
CBA		
S0B	C0DLST	CODE LIST
BRU	CLR1AE	CLEAR ONE AND EXIT
*DMY TEST		
DYTEST S0T	LDYIDF	LOCAL DMY ID FLAG
JAT	A181	A1
SKN	SPTRIG	SPR0G TRIGGER
BRU	XTF	EXIT FALSE
FET	W0,2	W0
JRS	L0DCTL	LOAD CENTRAL
SER	GDLIST	GLOBAL DMY LIST
JAF	XTF	EXIT FALSE
SWT	W1,2	W1
S0T	AYIDFG	ARRAY ID FLAG
BNG	TW0	TW0
SKR	WRKB0T	WORK BOTTOM
JAF	A181	A1
CNA		
S0B	W0RKL T	WORK LIST
BAF	EAX0PP	EAX+ 0P
A181 BNG	TW0	TW0
BRU	EXITRU	EXIT TRUE
*FIL POP	FILE INSTRUCTION 123	
FIL POPD	12300000B,1,1	
FILP0P SBRM	N0NESE	N0 NEST ENTRY
LDA	0,2	0
S0B	C0DLST	CODE LIST
BRU	B0TEXT	BOTTOM EXIT

	PAGE		
*SER P0P	SEARCH		
SER P0PD	14400000B,1,1		
SERP0P SBRM	P0NTRY	P0P ENTRY	
LDA	T0P,2	T0P	
SUB	B0TT0M,2	B0TT0M	
STA	SERTP1	SER TEMP 1	
SKN	SERTP1	SER TEMP 1	
BRU	XTF	EXIT FALSE	
LDA	C0DE,2	C0DE	
ETR	SIZMSK	SIZE MASK	
SUB	0NE	0NE	
STA	SERTP2	SER TEMP 2	
MRG	EAX0P	LIVE EAX+ M0P	
STA*	A183	A2	
STA*	A187	A6	
LDA	B0TT0M,2	B0TT0M	
SUB	SERTP2	SER TEMP 2	
ADD	0NE	0NE	
STA	SERTP3	SER TEMP 3	
MRG	SKE0P	LIVE SKE+ M0P	
STA*	A184	A3	
ADD	0NE	0NE	
STA*	A185	A4	
ETR	ADRMSK	ADR MASK	
SUB	0NE	0NE	
MRG	SKM0P	LIVE SKM+ M0P	
STA*	A188	A7	
STX	SERTP4	SER TEMP 4	
LDA	C0DE,2	C0DE	
SKA	INTGLM	INTEGER LIST MASK	
BRU	A186	A5	
LDX	SERTP1	SER TEMP 1	
LDA	CTL2	CENTRAL 2	
LDB	CTL1	CENTRAL 1	
A182	XAB		
A183	EXU	(SIZE+1)	
A184	EXU	(B0TT0M+SIZE+2)	
	BRX	A2	
		A183A	
		A184A	
		A183	



A185	XAB EXU	A185A	(BOTTOM=SIZE+3)
	BRX	A182	A1
	BRU	A189	A8
A186	LXD	SERTP1	SER TEMP 1
	LDB	ONES	
	SKA	LBLSTM	LABEL LIST MASK
	LDB	ABSATM	ALL BUT SIGN AND TAG MASK
	LDA	CTL2	CENTRAL 2
A187	EXU	A187A	(SIZE-1)
A188	EXU	A188A	(BOTTOM=SIZE+2)
	BRX	A187	A6
A189	CXA		
	ETR	ADRMSK	ADR MASK
	SKG	ONE	ONE
	BRU	XTF	EXIT FALSE
	MRG	ABAM	ALL BUT ADR MASK
	STA	SERTP5	SER TEMP 5
	LXD	SERTP4	SER TEMP 4
	LDA	CODE,2	CODE
	ETR	CDMASK	CODE MASK
	ADD	SERTP3	SER TEMP 3
	ADD	SERTP5	SER TEMP 5
	SUB	BASE,2	BASE
	SQB	WORKLT	WORK LIST
	BRU	EXITRU	EXIT TRUE
*BOP POP		BOTTOM POINTER	
BOP POPD		14100000B,1,1	
BOPPOP SBRM		NONESE	NO NEST ENTRY
	LDA	CODE,2	CODE
	ETR	CDMASK	CODE MASK
	ADD	BOTTOM,2	BOTTOM
	SUB	BASE,2	BASE
	ADD	ONE	ONE
	SQB	WORKLT	WORK LIST
	BRU	BOTEXT	BOTTOM EXIT
*MCP POP		MØVE CENTRAL 0N -- COPY CENTRAL TO BOTTOM OF ADDRESSED LIST	
MCP POPD		11400000B,1,1	
MCPPOP SBRM		NONESE	NO NEST ENTRY

LDA	CTL1	CENTRAL 1
SQB	0,2	0
LDA	CTL2	CENTRAL 2
SQB	0,2	0
BRU	BTEXT	BOTTOM EXIT
*SOF POP	SET ON FLAG 116	
SOF	POP	11600000B,1,1
SOFPOP	SBRM	POP ENTRY
LDA*	WRKBOT	WORK BOTTOM
SKA	0,2	0
BRU	EXITRU	EXIT TRUE
BRU	XTF	EXIT FALSE
*SOT POP	SET ON TEST	
SOT	POP	15200000B,1,1
SOTPOP	SBRM	POP ENTRY
CXA		
LSH	12	14
LDB	CH1MSK	CHAR 1 MASK
SKM*	WRKBOT	WORK BOTTOM
BRU	XTF	EXIT FALSE
BRU	EXITRU	EXIT TRUE
*BNG POP	BRING	
BNG	POP	13100000B,1,1
BNGPOP	SBRM	POP ENTRY
FET	0,2	0
SWT	W1,2	W1
JRS	PRPNTR	PROCESS PNTR
LDA	W0,2	W0
ADM	ADRPT0	ADDRESS POINTED TO
LDA*	ADRPT0	ADDRESS POINTED TO
STA*	WRKBOT	WORK BOTTOM
BRU	EXIT	EXIT
*PROCESS PNTR	- RETURN ABS ADDR OF LIST ELT GIVEN ITS REL ADDR IN LIST	
\$PRPNTR LDA*	WRKBOT	WORK BOTTOM
MUL	RSH18	RIGHT SHIFTER 18
ETR	CH4MSK	CHAR 4 MASK
CAX		
LDA*	WRKBOT	WORK BOTTOM
ETR	PNTRAM	PNTR ADR MASK

ADD	BASE,2	BASE
STA	ADRPT0	ADDRESS POINTED TO
BRU	CLR1AE	CLEAR ONE AND EXIT
PAGE		
*SNE P0P	SET NOT EMPTY	
SNE P0PD	14300000B,1,1	
SNEP0P SBRM	TSTMTY	TEST FOR EMPTY
BRU	EXITRU	EXIT TRUE
*CNT P0P	COUNT 120	
CNT P0PD	12000000B,1,1	
CNTP0P SBRM	NONESE	NO NEST ENTRY
LDA	BOTTOM,2	BOTTOM
SUB	T0P,2	T0P
S0B	W0RKLT	WORK LIST
BRU	B0TEXT	BOTTOM EXIT
*FET P0P	FETCH 110	
FET P0PD	11000000B,1,1	
FETP0P SBRM	NONESE	NO NEST ENTRY
LDA	0,2	0
S0B	W0RKLT	WORK LIST
BRU	B0TEXT	BOTTOM EXIT
*SWT P0P	SWITCH ADDRESSED WORD WITH WORD ON BOT OF WORK LIST	
SWT P0PD	11100000B,1,1	
SWTP0P SBRM	NONESE	NO NEST ENTRY
LDA	0,2	0
XMA*	WRKB0T	WORK BOTTOM
STA	0,2	0
BRU	B0TEXT	BOTTOM EXIT
*M0N P0P	MOVE 0N, MOVE WORD 0N W.L. BOT TO LIST NUMBER IN ADDR	
*P0P UP THE	WORK LIST	
M0N P0PD	12400000B,1,1	
M0NP0P SBRM	NONESE	NO NEST ENTRY
LDA*	WRKB0T	WORK BOTTOM
SKR	WRKB0T	WORK BOTTOM
S0B	0,2	0
BRU	B0TEXT	BOTTOM EXIT
*M0F P0P	MOVE 0FF -- MOVE WORD 0N BOTTOM OF ADDRESSED LIST TO W.L. BOTTOM	
M0F P0PD	14200000B,1,1	
M0FP0P SBRM	TSTMTY	TEST FOR EMPTY

LDA*	BOTTOM,2	BOTTOM
SKR	BOTTOM,2	BOTTOM
SQB	WORKLT	WORK LIST
BRU	EXITRU	EXIT TRUE
*TBT PBP	TAKE OFF TOP -- WORD ON TOP OF ADDRESSED LIST TO BOTTOM OF W.L.	
TBT	PBPD	12200000B,1,1
TBTBP	SBRM	TSTMTY
	MIN	TBP,2
LDA*	TBP,2	TBP
SQB	WORKLT	WORK LIST
BRU	EXITRU	EXIT TRUE
*LCF PBP	LOAD CENTRAL FROM -- COPY 2 WRDS FROM LIST M TO CENTRAL	
LCF	PBPD	14500000B,1,1
LCFBP	SBRM	TSTMTY
	SUB	ONE
BRU	LCBLCF	LCB=LCF COMMON
*LCB PBP	LOAD CENTRAL OFF -- UNSTACK 2 WRDS FROM LIST M. PUT IN CENTRAL	
LCB	PBPD	15700000B,1,1
LCBFBP	SBRM	TSTMTY
	ADD	ONE
*LCB=LCF COMMON		
LCBLCF	XMA	TBP,2
	CAX	
	LDP	1,2
	STD	CTL1
BRU	EXITRU	EXIT TRUE
*TEST FOR EMPTY		
TSTMTY	HLT	LTSTMT
	SBRM	PENTRY
	LDA	TBP,2
	ADD	ONE
	SKG	BOTTOM,2
	SBRR	TSTMTY
	LDA	START,2
	STA	TBP,2
	STA	BOTTOM,2
BRU	XTF	EXIT FALSE
*RSV PBP	SERESERVE 117	
RSV	PBPD	11700000B,1,1

RSVP0P	SBRM	NONESE	NO NEST ENTRY
	LDA	START,2	START
	SUB	BASE,2	BASE
	S0B	0,2	0
	LDA	T0P,2	T0P
	SUB	BASE,2	BASE
	S0B	0,2	0
	LDA	B0TT0M,2	B0TT0M
	STA	START,2	START
	STA	T0P,2	T0P
	BRU	B0TEXT	B0TT0M EXIT
*RLS	P0P	RELEASE 121	
RLS	P0PD	12100000B,1,1	
RLSP0P	SBRM	P0NTRY	P0P ENTRY
*RELEASE LIST,		LIST NUMBER IN X	
RELIST	LDA	START,2	START
	STA	T0P,2	T0P
	STA	B0TT0M,2	B0TT0M
	SKG	BASE,2	BASE
	BRU	EXIT	EXIT
	LDA*	START,2	START
	ADD	BASE,2	BASE
	STA	T0P,2	T0P
	SKR	START,2	START
	LDA*	START,2	START
	ADD	BASE,2	BASE
	SKR	START,2	START
	LDB	START,2	START
	STA	START,2	START
	STB	B0TT0M,2	B0TT0M
	BRU	EXIT	EXIT
	PAGE		
*FIP	P0P	FLESH AND INHERIT PLEX	
FIP	P0PD	14600000B,1,1	
FIPP0P	CLA		
	SBRM	FFF	FIC-FAC-FIP
	LDA	M0FCPY	MERG 0F C0PY
	MRG*	WRKB0T	W0RK B0TT0M
	STA*	WRKB0T	W0RK B0TT0M

	BRU	EXIT	EXIT
*FAC PBP		FLESH AND COUNT	
FAC	PBP	14000000B,1,1	
FACPBP	CLA		
	BRU	FFCOM	FIC-FAC COMMON
*FIC PBP		FLESH INHERIT AND COUNT	
FIC	PBP	13700000B,1,1	
FICPBP	LDA	MODEMS	MODE MASK
*FIC-FAC COMMON			
FFCOM	SBRM	FFF	FIC-FAC-FIP
	BRU	EXIT	EXIT
*FIC-FAC-FIP			
FFF	HLT	LFFF	
	SBRM	PENTRY	PBP ENTRY
	STA	FFFTP1	FIC-FAC-FIP TEMP 1
	STA	MFCPY	MERG OF COPY
	PL0	0,2	0
	STA	CTPLEX	COUNT OF PLEX
	STA	CTCOPY	COUNT OF COPY
A190	LDA	WRKBOT	WORK BOTTOM
	SUB	CTCOPY	COUNT OF COPY
	CAX		
	LDA	0,2	0
	S0B	PLEXLS	PLEX LIST
	ETR	MODEMS	MODE MASK
	MRG	MFCPY	MERG OF COPY
	STA	MFCPY	MERG OF COPY
	SKR	CTCOPY	COUNT OF COPY
	BRU	A190	A1
	LDA	WRKBOT	WORK BOTTOM
	SUB	CTPLEX	COUNT OF PLEX
	STA	WRKBOT	WORK BOTTOM
	LDA*	WRKBOT	WORK BOTTOM
	ETR	FFFTP1	FIC-FAC-FIP TEMP 1
	MRG*	PLEXBT	PLEX BOTTOM
	STA*	WRKBOT	WORK BOTTOM
	SKR	PLEXBT	PLEX BOTTOM
	SBRR	FFF	FIC-FAC-FIP
*PL0 PBP		PLEX OPEN	

PL0	P0PD	13400000B,1,1	
PL0P0P	SBRM	N0NESE	N0 NEST ENTRY
	CXA		
	LSH	12	14
	ETR	CDMASK	C0DE MASK
	ADD	PLEXBT	PLEX B0TT0M
	SUB	PLXBAS	PLEX BASE
	ADD	0NE	0NE
	S0B	W0RKLT	W0RK LIST
	CXA		
	ETR	L0W2BM	L0W 2 BITS MASK
	S0B	PLEXLS	PLEX LIST
	BRU	B0TEXT	B0TT0M EXIT
*PUL	P0P	PULL 130	
PUL	P0PD	13000000B,1,1	
PULP0P	SBRM	P0NTRY	P0P ENTRY
	STX	TGTLCY	TARGET LIST 0F C0PY
	LDA*	WRKB0T	W0RK B0TT0M
	SKR	WRKB0T	W0RK B0TT0M
	ETR	PNTRAM	PNTR ADR MASK
	STA	ST0CX	S0URCE T0P 0F C0PY
	ADD	PLXBAS	PLEX BASE
	CAX		
	LDA	0,2	0
	STA	CTC0PY	C0UNT 0F C0PY
	BRU	A192	A2
A191	MIN	ST0CX	S0URCE T0P 0F C0PY
	LDA	ST0CX	S0URCE T0P 0F C0PY
	ADD	PLXBAS	PLEX BASE
	CAX		
	LDA	0,2	0
	S0B*	TGTLCY	TARGET LIST 0F C0PY
A192	SKR	CTC0PY	C0UNT 0F C0PY
	BRU	A191	A1
	BRU	EXIT	EXIT
*ADR	P0P	ADDRESS	
ADR	P0PD	14700000B,1,1	
ADRP0P	SBRM	P0NTRY	P0P ENTRY
	CXA		

	SBB	W0RKLT	W0RK LIST
	BRU	EXIT	EXIT
*CAR P0P		C0PY AND RELEASE	
CAR	P0PD	13300000B,1,1	
CARP0P	LDA*	WRKB0T	W0RK B0TT0M
	SKR	WRKB0T	W0RK B0TT0M
	SBRM	CPYLS	C0PY LIST
	LDX	SLSTCY	SOURCE LIST 0F C0PY
	BRU	RELIS	RELEASE LIST
*C0F P0P		C0PY FILE	
C0F	P0PD	15600000B,1,1	
C0FP0P	SBRM	P0NTRY	P0P ENTRY
	STX	C0FTP1	C0F TEMP 1
	SNE*	C0FTP1	C0F TEMP 1
	JAF	CLR1EF	CLEAR 0NE EXIT FALSE
	RSV*	W0,2	W0
	CAR*	C0FTP1	C0F TEMP 1
	BRU	EXITRU	EXIT TRUE
*CAC P0P		C0PY AND C0UNT	
CAC	P0PD	13500000B,1,1	
CACP0P	LDA	LHPLIN	LIVE HLT PLEX LIST INST
	SBRM	CPYLS	C0PY LIST
	BRU	CICACT	CIC-CAC C0UNT
*CIC P0P		C0PY INHERIT AND C0UNT	
CIC	P0PD	13600000B,1,1	
CICP0P	LDA	LHPLIN	LIVE HLT PLEX LIST INST
	SBRM	CPYLS	C0PY LIST
	LDA	M0FCPY	MERG 0F C0PY
	MRG*	WRKB0T	W0RK B0TT0M
	STA*	WRKB0T	W0RK B0TT0M
*CIC-CAC C0UNT			
CICACT	LDA*	WRKB0T	W0RK B0TT0M
	ETR	PNTRAM	PNTR ADR MASK
	ADD	PLXBAS	PLEX BASE
	STA	ADRPT0	ADDRESS P0INTED T0
	LDA	CTC0PY	C0UNT 0F C0PY
	ADM*	ADRPT0	ADDRESS P0INTED T0
	BRU	EXIT	EXIT
*C0PY LIST			



CPYLST	HLT	LCPYLS	
	SBRM	PENTRY	PDP ENTRY
	STA	TGTLCY	TARGET LIST OF COPY
	STX	SLSTCY	SOURCE LIST OF COPY
	CLA		
	STA	CTCOPY	COUNT OF COPY
A193	STA	MFCOPY	MERG OF COPY
	TOT*	SLSTCY	SOURCE LIST OF COPY
	JAF	A194	A2
	MEN*	TGTLCY	TARGET LIST OF COPY
	MIN	CTCOPY	COUNT OF COPY
	ETR	MODEMS	MODE MASK
	MRG	MFCOPY	MERG OF COPY
	BRU	A193	A1
A194	SBRR	CPYLST	COPY LIST
	PAGE		
*EBS PDP		EXCLUSIVE OR TO STORAGE	
EBS	PDPD	15000000B,1,1	
EOSPDP	SBRM	PENTRY	PDP ENTRY
	LDA*	WRKBOT	WORK BOTTOM
	EOR	0,2	0
	STA	0,2	0
	BRU	CLR1AE	CLEAR ONE AND EXIT
*SBL PDP		SET OUT OF LIMIT 115	
SBL	PDPD	11500000B,1,1	
SOLPDP	SBRM	PENTRY	PDP ENTRY
	STD	FOLTPA	FOL TEMP A
	SKN	FOLTPA	FOL TEMP A
	CLAB		
	SKN	FOLTPA	FOL TEMP A
	SDP	FOLTPA	FOL TEMP A
	ADP	0,2	0
	SKA	SIGNBT	SIGN BIT
	BRU	EXITRU	EXIT TRUE
	LDP	FOLTPA	FOL TEMP A
	BRU	XTF	EXIT FALSE
*LDP PDP		LOAD DOUBLE 125	
LDP	PDPD	12500000B,1,1	
LDPPDP	SBRM	NONESE	NO NEST ENTRY

LDA	0,2	0
LDB	1,2	1
BRU	INDEXEX	INDEX EXIT
*STD P0P	STORE DOUBLE 107	
STD P0PD	10700000B,1,1	
STDP0P SBRM	NONESE	NO NEST ENTRY
STA	0,2	0
STB	1,2	1
BRU	INDEXEX	INDEX EXIT
*ADP P0P	ADD DOUBLE PRECISION	
ADP P0PD	15400000B,1,1	
ADPP0P SBRM	NONESE	NO NEST ENTRY
XAB		
ADD	1,2	1
XAB		
ADC	0,2	0
BRU	INDEXEX	INDEX EXIT
*SDP P0P	STORE DOUBLE PRECISION	
SDP P0PD	15300000B,1,1	
SDPP0P SBRM	NONESE	NO NEST ENTRY
XAB		
SUB	1,2	1
XAB		
SUC	0,2	0
BRU	INDEXEX	INDEX EXIT
PAGE		
*S0C P0P	SET ON CHARACTER	
S0C P0PD	16200000B,1,1	
S0CP0P SBRM	RDYSCH	READY SCAN CHAR
SKA	0,2	0
BRU	EXITRU	EXIT TRUE
BRU	XTF	EXIT FALSE
*CSF P0P	CHAR SCAN OR FAIL 101	
CSF P0PD	10100000B,1,1	
CSFP0P SBRM	P0NTRY	P0P ENTRY
CSA	0,2	0
JAT	EXIT	EXIT
BRU	ILSYFA	ILLEGAL SYNTAX FAIL
*CSA P0P	CHAR SCAN OR ALTERNATE 102	

CSA	P0PD	10200000B,1,1	
CSAP0P	SBRM	RDYSCH	READY SCAN CHAR
	SKE	0,2	0
	BRU	XTF	EXIT FALSE
	CLA		
	STA	CRNTCH	CRRNT CHAR
	BRU	EXITRU	EXIT TRUE
*PRC	P0P	PRINT CHARACTER	
PRC	P0PD	16000000B,1,1	
PRCP0P	SBRM	P0NTRY	P0P ENTRY
	LDB	0,2	0
	CLA		
	LSH	6	6
	SBRM	PRINT	PRINT
	BRU	EXIT	EXIT
*PRQ	P0P	PRINT QUOTE	
PRQ	P0PD	16100000B,1,1	
PRQP0P	SBRM	P0NTRY	P0P ENTRY
	SBRM	INITQS	INITIALIZE QUOTE SCAN
A195	SBRM	QSNEXC	QUOTE SCAN NEXT CHAR
	BRU	EXIT	EXIT
	RSH	18	22
	SBRM	PRINT	PRINT
	BRU	A195	A1
*QSF	P0P	QUOTE SCAN 0F FAIL	
QSF	P0PD	10600000B,1,1	
QSF0P	SBRM	RDYSCH	READY SCAN CHAR
	SBRM	INITQS	INITIALIZE QUOTE SCAN
A196	SBRM	QSNEXC	QUOTE SCAN NEXT CHAR
	BRU	EXIT	EXIT
	SKE	CRNTCH	CRRNT CHAR
	BRU	FAIL	FAIL
	SBRM	NXACCH	NEXT ACTIVE CHAR
	BRU	A196	A1
*S0B	P0P	STORE 0N B0TT0M	
S0B	P0PD	15100000B,1,1	
S0BP0P	STA	INCAR	INC0MING A REGISTER
	STX	INCXR	INC0MING X REGISTER
	EAX*	0	0

	LDA	BOTTOM,2	BOTTOM
	SKE	LIMIT,2	LIMIT
	BRU	A228	A1
	SBRM	REASGM	REASSIGN MEMORY == THIS LIST NEEDS MORE SPACE
	EAX	0,2	0
A228	LDA	INCAR	INCOMING A REGISTER
	MIN	BOTTOM,2	BOTTOM
	STA*	BOTTOM,2	BOTTOM
	LDX	INCXR	INCOMING X REGISTER
	BRR	0	0
*JAF	PBP	JUMP ANSWER FALSE	
JAF	PBPD	10400000B,1,1	
JAFPBP	STA	XITTP1	EXIT TEMP 1
	CLA		
	BRU	JATAFC	JAT-JAF COMMON
*JAT	PBP	JUMP ANSWER TRUE	
JAT	PBPD	10300000B,1,1	
JATPBP	STA	XITTP1	EXIT TEMP 1
	LDA	XTRUEM	EXIT TRUE MASK
*JAT-JAF	COMMON		
JATAFC	EOR*	EXIBOT	EXIT BOTTOM
	SKA	XTRUEM	EXIT TRUE MASK
	BRU	XTENZ0	EXIT ON ZERO
	EAX*	0	0
	BRU	JJJUMP	JRS-JAT-JAF JUMP
*JRS	PBP	JUMP RECURSIVELY TO SUBRU	
JRS	PBPD	11200000B,1,1	
JRSPBP	SBRM	PBENTRY	PBP ENTRY
*JRS-JAT-JAF	JUMP		
JJJUMP	STX	0	0
	SKR	0	0
	BRU	XTENZ0	EXIT ON ZERO
*PBP	LINK	INITIALIZATION	
\$PBPBRU	BRU	BAMPBP	
	BRU	CSFPBP	
	BRU	CSAPBP	
	BRU	JATPBP	
	BRU	JAFPBP	
	BRU	TRYPBP	

BRU	QSFP0P
BRU	STDP0P
BRU	FETP0P
BRU	SWTP0P
BRU	JRSP0P
BRU	BIFP0P
BRU	MC0P0P
BRU	S0LP0P
BRU	S0FP0P
BRU	RSVP0P
BRU	CNTP0P
BRU	RLSP0P
BRU	T0TP0P
BRU	FILP0P
BRU	M0NP0P
BRU	L0PP0P
BRU	BLFP0P
BRU	SALP0P
BRU	PULP0P
BRU	BNGP0P
BRU	BAFP0P
BRU	CARP0P
BRU	PL0P0P
BRU	CACP0P
BRU	CICP0P
BRU	FICP0P
BRU	FACP0P
BRU	B0PP0P
BRU	M0FP0P
BRU	SNEP0P
BRU	SERP0P
BRU	LCFP0P
BRU	FIPP0P
BRU	ADRP0P
BRU	E0SP0P
BRU	S0BP0P
BRU	S0TP0P
BRU	SDPP0P
BRU	ADPP0P

```
BRU    RECP0P
BRU    C0FP0P
BRU    LC0P0P
BRU    PRCP0P
BRU    PRQP0P
BRU    S0CP0P
BRU    BIMP0P
BRU    FEXP0P
BRU    ICTP0P
BSS    10
$PBRUE ZR0
END    0
```

```

6C1 IDENT 7-18 R.0. STORAGE, COMPILER PART 6 (LAST)
*
*
*
*
BFS MACR0 D; SKN D(1); BRU D(2); ENDM
DEC MACR0 D; DATA D(1); ENDM
OCT MACR0 D; DATA D(1); B; ENDM
BEQ MACR0 D BRU IF EQUAL; SKE D(1); BRU **2; BRU D(2); ENDM
BNE MACR0 D BRU IF NOT EQUAL; SKE D(1); BRU D(2); ENDM
NCARDS EQU 9
IBSIZE EQU 73*NCARDS/4+39
BAM 0PD 10000000B,1,1
CSF 0PD 10100000B,1,1
CSA 0PD 10200000B,1,1
JAT 0PD 10300000B,1,1
JAF 0PD 10400000B,1,1
TRY 0PD 10500000B,1,1
QSF 0PD 10600000B,1,1
STD 0PD 10700000B,1,1
FET 0PD 11000000B,1,1
SWT 0PD 11100000B,1,1
JRS 0PD 11200000B,1,1
BIF 0PD 11300000B,1,1
MC0 0PD 11400000B,1,1
S0L 0PD 11500000B,1,1
S0F 0PD 11600000B,1,1
RSV 0PD 11700000B,1,1
CNT 0PD 12000000B,1,1
RLS 0PD 12100000B,1,1
T0T 0PD 12200000B,1,1
FIL 0PD 12300000B,1,1
M0N 0PD 12400000B,1,1
LDP 0PD 12500000B,1,1
BLF 0PD 12600000B,1,1
SAL 0PD 12700000B,1,1
PUL 0PD 13000000B,1,1
BNG 0PD 13100000B,1,1
BAF 0PD 13200000B,1,1

```

CAR	0PD	13300000B,1,1
PL0	0PD	13400000B,1,1
CAC	0PD	13500000B,1,1
CIC	0PD	13600000B,1,1
FIC	0PD	13700000B,1,1
FAC	0PD	14000000B,1,1
B0P	0PD	14100000B,1,1
M0F	0PD	14200000B,1,1
SNE	0PD	14300000B,1,1
SER	0PD	14400000B,1,1
LCF	0PD	14500000B,1,1
FIP	0PD	14600000B,1,1
ADR	0PD	14700000B,1,1
E0S	0PD	15000000B,1,1
S0B	0PD	15100000B,1,1
S0T	0PD	15200000B,1,1
SDP	0PD	15300000B,1,1
ADP	0PD	15400000B,1,1
REC	0PD	15500000B,1,1
C0F	0PD	15600000B,1,1
LC0	0PD	15700000B,1,1
PRC	0PD	16000000B,1,1
PRQ	0PD	16100000B,1,1
S0C	0PD	16200000B,1,1
BIM	0PD	16300000B,1,1
FEX	0PD	16400000B,1,1

\*INITIALIZE QUOTE SCAN

\$INITQS HLT LINITG

STX QSFWAD

LDA TW0

STA QSFCTR

LDA\* QSFWAD

STA QSFWRD

A197 SBRM QSNEXC

SBRR INITQS

BRU A197

\*QUOTE SCAN NEXT CHAR

\$QSNEXC HLT LQSNEX

LDA QSFWRD

QSF WORD ADR

QSF COUNTER

QSF WORD ADR

QSF WORD

QUOTE SCAN NEXT CHAR

INITIALIZE QUOTE SCAN

A1

QSF WORD



	LDB	#77600000B	PERIOD
	SKM	#' .1*200000B	QUOTE SCAN NEXT CHAR
	MIN*	QSNEXC	
	ABC		
	LSH	8	
	STB	QSFWRD	QSF WORD
	CAX		
	SKR	QSFCTR	QSF COUNTER
	BRU	A198	A1
	MIN	QSFWAD	QSF WORD ADR
	LDA*	QSFWAD	QSF WORD ADR
	STA	QSFWRD	QSF WORD
	LDA	TW8	
	STA	QSFCTR	QSF COUNTER
A198	LDA	CHTTBL,2	CHAR TRANSL TABLE
	SBRR	QSNEXC	QUOTE SCAN NEXT CHAR
*PRINT 5 DEC			
\$PR5DEC FET	FIVE	FIVE	
	FET	ZER0	ZER0
A199	LDA	W2,2	W2
	ETR	FIVMSK	FIVE DEC MASK
	MUL	TEN	TEN
	DIV	0NEMIL	0NE MILLION
	STB	W2,2	W2
	CAX		
	MRG*	WRKB0T	WORK BOTTOB
	STA*	WRKB0T	WORK BOTTOB
	EAX	20B,2	ZER0 STARTS AT 20
	SKG	ZER0	ZER0
	CLX		SPACE
	PRC	CHTTBL,2	CHAR TRANSL TABLE
	SKR	W1,2	W1
	BRU	A199	A1
	PRC	SPACE	SPACE
	PRC	SPACE	SPACE
	BRU	CLR3AE	CLEAR THREE AND EXIT
*PRINT 0CTAL			
\$P0CTAL BRU	A201	A1	
A200 CLA			

	LDB	W1,2	W1
	LSH	3	3
	STB	W1,2	W1
	CAX		
	EAX	20B,2	LIFT NUMBER BASE FOR ASCII CODE
	PRC	CHTTBL,2	CHAR TRANSL TABLE
A201	SKR	W0,2	W0
	BRU	A200	A2
	BRU	CLR2AE	CLEAR TWO AND EXIT
	*PRINT,	6 BIT CHAR RT JUSTIFIED IN A, X SAVED	
	\$PRINT	HLT	LPRINT
	CLB		
	RSH	6	
	CBA		
	MIN	CHTSLN	CHARS THIS LINE
	SKE	CARBL0	CAR RETURN ZERO LOW
	BRU	A204	A4
A201A	LDA	=155B	
	CI0	C0UTF	
	BFS	FTTY0N,A201B	
	LDB	C0UTF	
	SKB	=77777776B	
	CI0	=1	
A201B	EQU	*	
A202A	MIN	LINETP	LINES THIS PAGE
	LDA	LINETP	LINES THIS PAGE
	SKE	LINEPP	LINES PER PAGE
	BRU	A203	A5
	LDA	0NE	
	STA	LINETP	LINES THIS PAGE
	LDA	TEN	TEN
	STA	PRTTP1	PRINT TEMP 1
A202	LDA	LF	LINE FEED
	CI0	C0UTF	
	BFS	FTTY0N,A203A	
	LDB	C0UTF	
	SKB	=77777776B	
	CI0	=1	
A203A	SKR	PRTTP1	PRINT TEMP 1

A203	BRU CLA	A202	A6
	STA	CHTSLN	CHARS THIS LINE
	SBRR	PRINT	
A204	LSH	6	
	CIØ	CØUTF	CMMD ØUTPUT FILE
	BFS	FTTYØN, A204A	
	LDB	CØUTF	
	SKB	=77777776B	
	CIØ	#1	
A204A	SBRR	PRINT	PRINT
*			
	*PUNCH FINISH		
	\$PCHFØN MIN	CKSMCT	CHECK SUM CNT
	*PUNCH CØDE		
	PCHCØD EXT	*	
	PCHCØD EQU	*	
A205	SNE	CØDLST	CØDE LIST
	JAF	EXIT	EXIT
	SKN	CKSMCT	CHECK SUM CNT
	BRU	A206	A1
	MIN	BLØKCT	BLØCK CNT
	LDA	BLØKCT	BLØCK CNT
*	WIØ	ØUTFLE	
	LDA	BLØKCT	BLØCK CNT
	STA	CKSUM	CHECK SUM
	LDA	BLØKLG	BLØCK LENGTH
	STA	CKSMCT	CHECK SUM CNT
A206	TØT	CØDLST	CØDE LIST
	JAF	A205	
	ADD	CKSUM	CHECK SUM
	ADC	ZERØ	ZERØ
	STA	CKSUM	CHECK SUM
	LDA*	WRKBØT	WØRK BØTTØM
	WIØ	ØUTFLE	BINARY ØUTPUT FILE
	SKR	WRKBØT	WØRK BØTTØM
	SKR	CKSMCT	CHECK SUM CNT
	BRU	A206	A1
	LDA	CKSUM	CHECK SUM

*	W10	OUTFLE	BINARY OUTPUT FILE
	BRU	A205	
*READY	SCAN	CHAR	
\$RDYSCH	HLT	LRDYSC	
	SBRM	P0NTRY	P0P ENTRY
	LDA	CRNTCH	CRRNT CHAR
	SKE	ZER0	ZER0
	SBRR	RDYSCH	READY SCAN CHAR
	STX	RDYCT1	READY CHAR TEMP 1
	SBRM	NXACCH	NEXT ACTIVE CHAR
	LDX	RDYCT1	READY CHAR TEMP 1
	SBRR	RDYSCH	READY SCAN CHAR
*NEXT	ACTIVE	CHAR	
\$NXACCH	HLT	LNXACC	
	LDA	INCHCT	INPUT CHAR CNT
	STA	LACHCT	LAST ACTIVE CHAR CNT
	SBRM	NXINCH	NEXT INPUT CHAR
	STA	FRSTCH	FIRST CHAR READ BY NXACCH
	BRU	A208A	
A208	SBRM	NXINCH	NEXT INPUT CHAR
A208A	SKE	SPACE	SPACE
	BRU	A209	A7
	LDA	SCANMD	SCAN M0DE
	SKE	UBLKFG	USE BLANKS FLAG
	BRU	A208	A1
	LDA	SPACE	SPACE
A209	SKE	CARET	CAR RETURN
	BRU	A215	A9
	SKN	E0LTRG	END 0F LINE TRIGGER
	BRU	A209A	
	LDB	INCHCT	INPUT CHAR CNT
	STB	E0LTRG	END 0F LINE TRIGGER
A209A	SKN	LASTST	LAST STA TRIGGER
	BRU	A215	A9
	LDA	INCHCT	INPUT CHAR CNT
	STA	ACHTP1	ACTIVE CHAR TEMP 1
	MIN	LINECT	LINE CNT
A211	SBRM	NXINCH	NEXT INPUT CHAR
	SKA	CARRFG	CAR RETURN FLAG

	BRU	A211	A8
	LDX	FOUR	FOUR
A213	STX	ACHTP2	ACTIVE CHAR TEMP 2
	SKE	SPACE	SPACE
	BRU	A214	A5
	SBRM	NXINCH	NEXT INPUT CHAR
	SKR	ACHTP2	ACTIVE CHAR TEMP 2
	BRU	A213	A6
	SKA	CNCHFG	CONTINUATION CHAR FLAG
	BRU	A208	A1
A214	LDA	ACHTP1	ACTIVE CHAR TEMP 1
	STA	INCHCT	INPUT CHAR CNT
	LDA	CARET	CAR RETURN
	STA	CRNTCH	CRRNT CHAR
A215	LDX	WRKBOT	WORK BOTTOM
	SBRR	NXACCH	NEXT ACTIVE CHAR
	PAGE		
	*NEXT INPUT CHAR		
	\$NXINCH HLT	LNXINC	
A216	LDA	INCHCT	INPUT CHAR CNT
	SKE	INSTCT	INPUT STOP CNT
	BRU	A218	A3
A217	SBRM	READ	READ
	BRU	OVFAIL	OVERFLOW FAIL
	LDA	INSTCT	INPUT STOP CNT
	SKG	INCHCT	INPUT CHAR CNT
	BRU	A217	A4
A218	MIN	INCHCT	INPUT CHAR CNT
	LDA	THREE	THREE
	SUB	INCHCT	INPUT CHAR CNT
	ETR	THREE	THREE
	STA	INCHT1	INPUT CHAR TEMP 1
	LDA	INCHCT	INPUT CHAR CNT
	RSH	1	1
	ABC		
	DIV	NPBFLG	INPUT BUFF LENGTH
	CBX		
	LDA	NPBUFF,2	INPUT BUFFER
	LDX	INCHT1	INPUT CHAR TEMP 1

	EXU	CHARST,2	CHAR SHIFTER TABLE
	ETR	CH4MSK	CHAR 4 MASK
	CAX		
A219	LDA	CHTTBL,2	CHAR TRANSL TABLE
	STA	CRNTCH	CRRNT CHAR
	LDX	WRKBOT	WORK BOTTOM
	SBRR	NXINCH	NEXT INPUT CHAR
	PAGE		
*READ			
\$READ	HLT	LREAD	
	LDA	RDSTCT	READ STOP CNT
	ADD	NBCHCT	IN BUFF CHAR CNT
	SKG	RDCHCT	READ CHAR CNT
	SBRR	READ	READ
	MIN*	READ	READ
A224	CLA		
	SKR	READSC	READ SPACE COUNT
	BRU	A222	
A220	CIO	INFLE	READ CHAR FROM INPUT FILE
	LDX	ONE	
	SBRM	ACB	ACCEPT CHAR AND BRANCH
	DATA	15500000B+A221	ASCII CR
	DATA	13700000B+A225	ASCII END OF FILE
	DATA	14400000B+A225	END OF TEXT (ALSO AN EOF)
	DATA	13500000B+A223	MULTIPLE SPACE
	DATA	9BIT+***1	
	SKG	CH4MSK	
	SKG	ONES	
	BRU	A220	DISCARD
A222	ETR	CH4MSK	CHAR 4 MASK
	CAX		
	LDA	CHTTBL,2	CHAR TRANSL TABLE
	STA	WIMCH	WIMED CHAR
	SBRM	ENTERAC	ENTER A CHAR
	MIN	INSTCT	INPUT STOP CNT
	LDA	RDSTCT	READ STOP CNT
	ADD	NBCH	IN BUFF CHAR CNT*1
	SKG	RDCHCT	READ CHAR CNT
	SBRR	READ	

A221	BRU LDA BRU	A224 ICR A222	INTERNAL RT JUST CR
A223	CIB STA BRU	INFLE READSC A224	READ CHAR FROM INPUT FILE READ SPACE COUNT
A225	SKR SBRR SBRR	FE0F READ READ	FLAG END OF FILE READ
	\$NPBFLG ZR0 \$NBCHCT ZR0 \$NBCH ZR0 *ENTER A CHAR \$ENTERAC HLT	IBSIZE 4*IBSIZE 4*IBSIZE=1 LNTERA	INPUT BUFFER LENGTH INPUT BUFFER CHAR CNT INPUT BUFFER CHAR CNT=1
A226	MIN LDA ETR STA LDA RSH ABC DIV CBX STB LDA LDX ETR STA LDA EXU ETR MRG LDX STA SBRR	RDCHCT RDCHCT THREE ECHTP1 RDCHCT 1 NPBFLG ECHTP2 NPBUFF,2 ECHTP1 N0TCMT,2 ECHTP1 WIMCH CHARST,2 CHMTBL,2 ECHTP1 ECHTP2 NPBUFF,2 NTERAC	READ CHAR CNT READ CHAR CNT THREE ENTER CHAR TEMP 1 READ CHAR CNT 1 INPUT BUFF LENGTH ENTER CHAR TEMP 2 INPUT BUFFER ENTER CHAR TEMP 1 N0T CHAR MASK TABLE ENTER CHAR TEMP 1 WIMED CHAR CHAR SHIFTER TABLE CHAR MASK TABLE ENTER CHAR TEMP 1 ENTER CHAR TEMP 2 INPUT BUFFER ENTER A CHAR
	*N0 NEST ENTRY \$N0NESE HLT STX XMA	LN0NES NNTP1 0	NN TEMP 1 0

	STA	NNTP2	NN TEMP 2
	XMA	0	0
	EAX*	0	0
	SBRR	NONESE	N0 NEST ENTRY
	*POP ENTRY		
	\$PENTRY HLT	LPONTR	
	STA	XITTP1	EXIT TEMP 1
	LDA	EXIB0T	EXIT B0TT0M
	SKE	XLIMIT	EXIT LIMIT
	BRU	A229	A2
	SBRM	REASGM	REASSIGN MEM0RY
	EAX	XITLS	EXIT LIST
A229	LDA	0	0
	ETR	ADRMSK	ADR MASK
	MIN	EXIB0T	EXIT B0TT0M
	STA*	EXIB0T	EXIT B0TT0M
	LDA	XITTP1	EXIT TEMP 1
DEBUG	N0P	0	(BRM DEBUG)
	STA	XITTP1	EXIT TEMP 1
	CXA		
	ETR	ADRMSK	ADR MASK
	SKG	NLIST	END 0F LISTS
	SKG	FLSBAS	FL SCALAR BASE, START 0F LISTS
	BRU	A230	A1
	LDA	WRKB0T	W0RK B0TT0M
A230	EAX*	0	0
	CXA		
	ETR	ADRMSK	ADR MASK
	CAX		
	LDA	XITTP1	EXIT TEMP 1
	SBRR	PENTRY	POP ENTRY
	PAGE		
	*REASSIGN MEM0RY		
	\$REASGM HLT	LREASG	
	STX	RSGNT3	REASSIGN TEMP 3
	LDA	FLSLMT	FL SCALAR LIMIT
	SUB	FLSCBT	FL SCALAR B0TT0M
	SKG	0NE	0NE
	BRU	PRESSM	PRESS MEM0RY



*MOVE LISTS UP			
\$MVLST	LDX	RSGNT3	REASSIGN TEMP 3
	MIN*	REASGM	REASSIGN MEMORY
	EXU*	LREASG	REASSIGN MEMORY
	CXA		
	ETR	ADRMSK	ADR MASK
	STA	RSGNT1	REASSIGN TEMP 1
	LDA	ONE	ONE
	BRU	A235	A5
A231	STA	RSGNT2	REASSIGN TEMP 2
	CAX		
	LDA	BOTTOM,2	BOTTOM
	ADD	ONE	ONE
	MRG	LIVEL0	LIVE LDA+ 0P
	STA*	A232	A1
	LDA	BOTTOM,2	BOTTOM
	SUB	ONE	ONE
	MRG	LIVES0	LIVE STA+ 0P
	STA*	A233	A2
	LDA	BASE,2	BASE
	SUB	BOTTOM,2	BOTTOM
	SUB	ONE	ONE
	STA	RSGNT4	REASSIGN TEMP 4
	LDA	MINUS2	MINUS TWO
	ADM	BASE,2	BASE
	ADM	START,2	START
	ADM	TOP,2	TOP
	ADM	BOTTOM,2	BOTTOM
	LDX	RSGNT4	REASSIGN TEMP 4
	BRU	A234	A3
A232	EXU	A232A	(BOTTOM+1)
A233	EXU	A233A	(BOTTOM-1)
A234	BRX	A232	A1
	LDA	RSGNT2	REASSIGN TEMP 2
	ADD	ONE	ONE
A235	SKG	RSGNT1	REASSIGN TEMP 1
	BRU	A231	A4
	LDX	RSGNT3	REASSIGN TEMP 3
	SBRR	REASGM	REASSIGN MEMORY

	*PRESS MEMORY		
	\$PRESSM	LDA	N0LASL
			N0 0F LAST LIST
		STA	RSGNT1
			REASSIGN TEMP 1
A236		LDA	RSGNT1
			REASSIGN TEMP 1
		LDA	LIMIT,2
			LIMIT
		STA	RSGNT2
			REASSIGN TEMP 2
		XMA	BOTTOM,2
			BOTTOM
		STA	RSGNT4
			REASSIGN TEMP 4
		SUB	BOTTOM,2
			BOTTOM
		CNA	
		ADM	START,2
			START
		ADM	TOP,2
			TOP
		ADD	BASE,2
			BASE
		STA	BASE,2
			BASE
		SUB	BOTTOM,2
			BOTTOM
		SUB	ONE
			ONE
		CAX	
		BRU	A238
			A5
A237		LDA*	RSGNT4
			REASSIGN TEMP 4
		STA*	RSGNT2
			REASSIGN TEMP 2
		SKR	RSGNT4
			REASSIGN TEMP 4
		SKR	RSGNT2
			REASSIGN TEMP 2
A238		BRX	A237
			A3
A239		SKR	RSGNT1
			REASSIGN TEMP 1
		LDA	RSGNT1
			REASSIGN TEMP 1
		SKE	ZERO
			ZERO
		BRU	A236
			A1
		LDA	FLSCBT
			FL SCALAR BOTTOM
		SUB	FLSLMT
			FL SCALAR LIMIT
		SKG	MINUS2
			MINUS TWO
		BRU	MVLST
			MOVE LISTS UP
	*MEMORY OVERFLOW		
	\$MEMOVF	SBRM	INITLS
			INITIALIZE LISTS
		SKN	LASTST
			LAST STA TRIGGER
		BRU	A240
			A1
		JRS	RELPR
			RELEASE AND PRINT
		PRC	CARET
			CAR RETURN
A240		PRQ	OVFLQT
			OVERFLOW QUOTE
		BRU	TRMCPL
			TERMINATE COMPILE

	PAGE		
*BOTTOM EXIT	• EXIT FOR POPS WHICH BEGAN WITH 'NO NEXT ENTRY'		
\$BOTEXT LDX	NNTP2	NN TEMP 2	
STX	0	0	
LDX	WRKBOT	WORK BOTTOM	
BRR	0	0	
*INDEX EXIT			
\$INDEX LDX	NNTP2	NN TEMP 2	
STX	0	0	
LDX	NNTP1	NN TEMP 1	
BRR	0	0	
*CLEAR TWO EXIT FALSE			
\$CLR2EF SKR	WRKBOT	WORK BOTTOM	
*CLEAR ONE EXIT FALSE			
\$CLR1EF SKR	WRKBOT	WORK BOTTOM	
*EXIT FALSE			
\$XTF EAX	ZERO	ZERO	
BRU	XTRXF	EXIT TRUE OR EXIT FALSE	
*CLEAR TWO EXIT TRUE			
\$CLR2ET SKR	WRKBOT	WORK BOTTOM	
*CLEAR ONE EXIT TRUE			
\$CLR1ET SKR	WRKBOT	WORK BOTTOM	
*EXIT TRUE			
\$EXITRU EAX	XTRUEM	EXIT TRUE MASK	
*EXIT TRUE OR EXIT FALSE			
\$XTRXF STA	XITTP1	EXIT TEMP 1	
LDA	0,2	0	
LDX	EXIBOT	EXIT BOTTOM	
ADD	W1,2	W1	
ETR	TESTNM	TEST NODE MASK	
ADM	W1,2	W1	
BRU	XONLST	EXIT ON LIST	
*CLEAR FOUR AND EXIT			
\$CLR4AE SKR	WRKBOT	WORK BOTTOM	
*CLEAR THREE AND EXIT			
\$CLR3AE SKR	WRKBOT	WORK BOTTOM	
*CLEAR TWO AND EXIT			
\$CLR2AE SKR	WRKBOT	WORK BOTTOM	
*CLEAR ONE AND EXIT			

\$CLR1AE SKR	WRKB0T	WORK B0TT0M
*EXIT		
\$EXIT STA	XITTP1	EXIT TEMP 1
*EXIT 0N LIST		
\$X0NLST LDA*	EXIB0T	EXIT B0TT0M
SKR	EXIB0T	EXIT B0TT0M
STA	0	0
*EXIT 0N ZERO		
\$XT0NZ0 LDA	XITTP1	EXIT TEMP 1
LDX	WRKB0T	WORK B0TT0M
BRR	0	0
*CHAR TRANSL TABLE		
CHTTBL EXT	*	
CHTTBL EQU	*	
\$SPACE DATA	00002600B	SPACE
\$TAPEMK 0CT	01003200	TAPE MARK
\$D0UB0T DATA	07003200B	PRIME
\$SHARP DATA	03003200B	
\$D0LLAR DATA	04003200B	D0LLAR
\$SCRPTM 0CT	05003200	SCRIPT M
\$L0261 DATA	06003200B	AND SIGN
\$PRIME 0CT	07003200	PRIME
LPAREN EXT	*	
LPAREN EQU	*	L PAREN
\$L026 0CT	10003200	026L
RPAREN EXT	*	
RPAREN EQU	*	R PAREN
\$R026 0CT	11003210	026R
ASTRSK EXT	*	
ASTRSK EQU	*	ASTERISK
\$ASTRCH 0CT	12003200	ASTERISK CHAR
PLUS EXT	*	
PLUS EQU	*	PLUS
\$PLUSCH 0CT	13003200	PLUS CHAR
C0MMA EXT	*	
C0MMA EQU	*	C0MMA
C0MACH 0CT	14003210	C0MMA CHAR
MINUS EXT	*	
MINUS EQU	*	MINUS

\$MINUSC	ØCT	15003200	MINUS CHAR
\$PERIOD	ØCT	16003200	PERIOD
VRGULE	EXT	*	
VRGULE	EQU	*	VIRGULE
\$SLASHC	ØCT	17003210	SLASH CHAR
\$NØ	ØCT	20002205	NØ
\$N1	ØCT	21003205	N1
\$N2	ØCT	22003205	N2
\$N3	ØCT	23003205	N3
\$N4	ØCT	24003205	N4
\$N5	ØCT	25003205	N5
\$N6	ØCT	26003205	N6
\$N7	ØCT	27003205	N7
\$N8	ØCT	30003205	N8
\$N9	ØCT	31003205	N9
\$CØLØN	ØCT	32003200	CØLØN
SEMICØ	EQU	*	SEMI CØLØN
CARRET	EXT	*	
CARRET	EQU	*	CAR RET
\$CARET	ØCT	33000320	CAR RETURN
\$LESSTH	ØCT	34003200	LESS THAN
EQUAL	EXT	*	
EQUAL	EQU	*	EQUAL
\$EQSNCH	ØCT	35003200	EQUAL SIGN CHAR
\$GRTHAN	ØCT	36003200	GREATER THAN
\$QMARK	DATA	37002200B	QUESTION MARK
\$ATSYM	DATA	40003200B	
\$A	ØCT	41003206	A
\$B	ØCT	42003206	B
\$C	ØCT	43003206	C
\$D	ØCT	44003206	D
\$E	ØCT	45003206	E
\$F	ØCT	46003206	F
\$G	ØCT	47003206	G
\$H	ØCT	50003206	H
\$I	ØCT	51003246	I
\$J	ØCT	52003246	J
\$K	ØCT	53003246	K
\$L	ØCT	54003246	L

\$M	θCT	55003246	M
\$N	θCT	56003246	N
\$LETERθ	θCT	57003206	LETTER θ
\$P	θCT	60003206	P
\$Q	θCT	61003206	Q
\$R	θCT	62003206	R
\$S	θCT	63003206	S
\$T	θCT	64003206	T
\$U	θCT	65003206	U
\$V	θCT	66003206	V
\$W	θCT	67003206	W
\$X	θCT	70003206	X
\$Y	θCT	71003206	Y
\$Z	θCT	72003206	Z
\$LBRAKT	θCT	<del>73003200</del>	L BRAKET
\$BKWDVR	θCT	74003200	BACKWARDS VIRGULE
\$RBRACK	θCT	<del>75003210</del>	R BRACKET
MθDECG EQU	*		MθDE CHANGE
\$ERM RK	θCT	76003200	ERRθR MARK
\$DELETE	θCT	77006000	DELETE

\*  
 \*ACCEPT CHAR AND BRANCH  
 \*I/ A=CHAR RT ADJ, X=0,1 FOR 6,9 BIT CHAR  
 \*B LBST, A SAVED

\$ACB	ZRθ	LACB	
	SKG	ACBLIM,2	LIMIT
	SKG	ZERθ	
	CLA		
	STA	ACBCHR	
	LDB	B6,2	
	STB	ACBBIT	
	CLB		
	LSH*	ACBSCT,2	SHIFT COUNT
	LDB	ACBMSK,2	MASK
	EAX*	LACB	
ACB1	EAX	1,2	
	SKM	0,2	
	BRU	*+2	
	BRU	ACBθUT	

	XMA	ACBBIT
	SKA	0,2
	BRU	ACBOUT
	XMA	ACBBIT
	BRU	ACB1
ACBOUT	LDA	ADRMSK
	ETR	0,2
	CAX	
	LDA	ACBCHR
	BRU	0,2

\*

\*WI0 PATCH == FOR SYSTEM DEBUGGING ONLY == TYPES FTC 0CTAL OUTPUT

\*CALL FROM A206+6

\$WI0PTH	ZR0	LWI0PT	
	SKN	FLC	LIST CODE
	BRR	LWI0PT	
	ABC		
	LDX	MIN8	
WI01	CLA		
	LSH	3	
	ADD	M20B	
	CI0	0NE	
	BRX	WI01	
	TC0	CR	
	BRR	LWI0PT	
\$MIN8	DATA	=8	
\$M20B	DATA	20B	

\*  
\*  
\*  
\*  
\*  
\*  
\*  
\*  
\*

*RUN TIME ADD FX ONE			
\$RTAFX1 ADD	RNTFX1		RUN TIME FX ONE
*RUN TIME LDP FL ONE			
\$RTLFL1 LDP	RNTFL1		RUN TIME FL ONE
*SKA RUN T	SIGN BIT INST		
\$SKARI SKA	RNTSBT		RUN TIME SIGN BIT
*SKG RUN TIME ZERO INST			
\$SKGRZI SKG	RUNTM0		RUN TIME ZERO
*LIVE EAX+ M0P			
\$EAX0P EAX	0,2	0	
*LIVE SKM+ M0P			
\$SKM0P SKM	0,2	0	
IN0P EXT	*	INPUT 0P	
\$N0P0P N0P	0	LIVE N0P M0P	
\$0UTP0P N0P	1	0UTPUT 0P	
*LIVE HLT PLEX LIST INST			
\$LHPLIN HLT	PLEXLS		PLEX LIST
*LIVE BRU M0P			
\$BRU0P BRU	0	0	
*LIVE SKE+ M0P			
\$SKE0P SKE	0,2	0	
\$CLA0P CLA			
*LIVE LDA+ 0P			
\$LIVEL0 LDA	0,2	0	
*LIVE STA+ 0P			
\$LIVES0 STA	0,2	0	
*			
\$EMPTQT ASC	!..!		
\$F0RKQT ASC	!.F0RK.!		
\$C0MPQT ASC	!.0MPILE.!		
\$MEMQT ASC	!.MEM0RY PANIC - .!		
\$RDERQT ASC	!.READ ERROR .!		
\$INFQT ASC	!.NPUT.!		
\$FRFQT ASC	!. FROM .!		
\$T0FQT ASC	!. T0 .!		
\$0HQT ASC	!.0.!		
\$0UTFQT ASC	!.UTPUT.!		
\$WHATQT ASC	!. .!	WHAT QT	
\$ECH0QT ASC	!.CH0.!	ECH0 QT	



\$WI0QT ASC	!.;SPECIFY I=0 FILES.!'	
\$WRI0QT ASC	!.;INPUT FILE NOT SYMBOLIC.!'	
\$SUBRQT ASC	!.SUBROUTINE.!'	
\$FNCTQT ASC	!.FUNCTION.!'	
\$IFQT ASC	!.IF.!'	
\$IMENQT ASC	!.IMENSION.!'	
\$0T0QT ASC	!.0T0.!'	0T0 QT
\$FRMTQT ASC	!.FORMAT.!'	FORMAT QT
\$CALLQT ASC	!.CALL.!'	
\$C0NTQT ASC	!.CONTINUE.!'	
\$TYPEQT ASC	!.TYPE.!'	
\$ACCPQT ASC	!.ACCEPT.!'	
\$TAPEQT ASC	!.TAPE.!'	
\$READQT ASC	!.READ.!'	
\$NPTPQT ASC	!.NPUTTAPE.!'	NPUTTAPE QT
\$WRITEQ ASC	!.WRITE.!'	
\$UTPTPQ ASC	!.UTPUTTAPE.!'	UTPUTTAPE QT
\$PUNCHQ ASC	!.PUNCH.!'	
\$PRINTQ ASC	!.PRINT.!'	
\$FLQT ASC	!.FLOATING.!'	
\$0VFLQT ASC	!.0VERFLOW.!'	
\$SENSEQ ASC	!.SENSE.!'	
\$WITCHQ ASC	!.WITCH.!'	
\$LIGHTQ ASC	!.LIGHT.!'	
\$ASSGNQ ASC	!.ASSIGN.!'	
\$CM0NQT ASC	!.COMMON.!'	
\$T0QT ASC	!.T0.!'	
\$EQUIVQ ASC	!.EQUIVALENCE.!'	
\$RTURNQ ASC	!.RETURN.!'	
\$PAUSEQ ASC	!.PAUSE.!'	
\$ST0PQT ASC	!.STOP.!'	
\$D0NEQT ASC	!.;D0 NEST ERR0RS;.!'	
\$LBLERQ ASC	!.;;LABELING ERR0RS;;.!'	
\$CMALQT ASC	!.;;COMMON ALLOCATI0N;;.!'	
\$PR0GQT ASC	!.;;PR0GRAM ALLOCATI0N;;.!'	
\$SBREQT ASC	!.;;SUBPR0GRAMS REQUIRED;;.!'	
\$ERR0RQ ASC	!.ERROR.!'	
\$DYQT ASC	!.DUMMY.!'	
\$AL0CAQ ASC	!.ALLOCATI0N.!'	

\$NUMBQT ASC	!.NUMBER.!	
\$IDECQT ASC	!.ID DECLARATION.!	
\$SYNTAXQ ASC	!.SYNTAX.!	
\$SUBSCQ ASC	!.SUBSCRIPTS.!	
\$TENDQT ASC	!.;;THE END.!	
\$ENDQT ASC	!.END.!	
\$LISTQT ASC	!.IST.!	
\$DEBUGQT ASC	!.EBUG.!	DEBUG QT
\$ASQT ASC	!.AS.!	AS QT
\$OPENQT ASC	!.PEN.!	OPEN QT
\$WAITQT ASC	!.WAIT.!	
\$IOCRQT ASC	!.;;;;;;;;;;.!	TEN CARRIAGE RETURNS
\$IFLQT ASC	!.NPUT.!	INPUT FILE QT
\$OFLQT ASC	!.OUTPUT.!	OUTPUT FILE QT
\$MAPQT ASC	!.AP.!	
\$CLOSEQT ASC	!.CLOSE.!	CLOSE FILE QT
\$BADLTO ASC	!.CANT OPEN LIST FILE.!	
\$PTIDFG OCT	00100000	PART ID FLAG
\$ADDRFG OCT	00400000	ADDRESSABLE FLAG
OPTAGB EXT	*	
OPTAGB EQU	*	MOP TAG BIT
\$TRBRSF OCT	400	TAB OR BLANK OR SPACE FLAG
\$NOTCRF OCT	2000	NOT CAR RETURN FLAG
INTGLM EXT	*	
INTGLM EQU	*	INTEGER LIST MASK
\$TCRBFG OCT	100	TAB-CAR RETURN-BACKSPACE FLAG
ASCNB EXT	*	
ASCNB EQU	*	ASCII NUMBER BASE
\$CARRFG OCT	20	CAR RETURN FLAG
\$NOLLSV OCT	17	NUMBER OF LAST LIST SAVED
\$NOLITSV DEC	59	NUMBER OF ITEMS SAVED
\$LINEPP DEC	55	LINES PER PAGE
\$NOLIST DEC	21	NUMBER OF LISTS
\$NOLASL DEC	20	NO OF LAST LIST
\$CGSGIN OCT	14600000	CHANGE SIGN INST
\$ABL2MK OCT	77777774	ALL BUT LOW 2 MASK
\$MPTITL OCT	04000000	MPROG TITLE
SPTITL EXT	*	
SPTITL EQU	*	SPROG TITLE

\$NØPINS ØCT	02000000	NØP INST
\$NMARKR ØCT	00600000	END MARKER
\$TØP5BM ØCT	76000000	TØP 5 BITS MASK
\$TENB4E ØCT	24000000	TEN B4 EXACT
ØCT	00000000	
\$ALLDØL ASC	'\$\$\$'	ALL DØLLARS
\$CARØLØ DATA	33000000B	CAR RETURN ZERO LOW
\$CR ZRØ	155B	CARRIAGE RETURN
\$LF ZRØ	152B	LINE FEED
\$ICR ZRØ	33B	INTERNAL CARRIAGE RETURN
\$TESTNM ØCT	77600000	TEST NØDE MASK
ABAM EXT	*	
ABAM EQU	*	ALL BUT ADR MASK
\$CDMASK ØCT	77740000	CØDE MASK
SCAIDF EXT	*	
SCAIDF EQU	*	SCALAR ID FLAG
\$XTRUEM ØCT	00200000	EXIT TRUE MASK
PDEFLM EXT	*	
PDEFLM EQU	*	PREVIOUSLY DEF LBL MASK
\$TAGBIT ØCT	20000000	TAG BIT
FLFLAG EXT	*	
FLFLAG EQU	*	FL FLAG
\$MØDEMS ØCT	00040000	MØDE MASK
\$MINU17 DEC	=17	MINUS SEVENTEEN
\$NØLSVB DEC	=14	NUMBER ØF LISTS SAVED BRXER
\$MINUS5 DEC	=5	MINUS FIVE
\$MINUS2 DEC	=2	MINUS TWO
NØFLSV EXT	*	
NØFLSV EQU	*	NUMBER ØF FIRST LIST SAVED
LDTPFG EQU	*	LOAD TAPE FLAG
TYBYWF EQU	*	TYPING BY W FLAG
LETRFG EXT	*	
LETRFG EQU	*	LETTER FLAG
\$TWØ DEC	2	TWØ
LØW2BM EXT	*	
LØW2BM EQU	*	LØW 2 BITS MASK
\$THREE DEC	3	THREE
LTRDGF EXT	*	
LTRDGF EQU	*	LETTER ØR DIGIT FLAG

\$FOUR	DEC	4	FOUR
\$FIVE	DEC	5	FIVE
\$SIX	DEC	6	SIX
SEVEN	EQU	*	SEVEN
\$OCTDMS	DEC	7	OCT DIGIT MASK
EIGHT	EXT	CVRPFG	EIGHT
\$CVRPFG	OCT	00000010	COMMA-VIRGULE-R PAREN FLAG
\$NINE	DEC	9	NINE
\$TEN	DEC	10	TEN
\$SIXTY	DEC	60	SIXTY
\$BLOCKG	DEC	90	BLOCK LENGTH
\$ONEMIL	DEC	1000000	ONE MILLION
\$TENTH3	OCT	31463146	ONE TENTH B*3
	OCT	31463160	
\$TENB4	OCT	24000000	TEN B4
	OCT	00000010	
\$KEYVAL	DEC	910	KEY VALUE
CHMTBL	EQU	*	CHAR MASK TABLE
\$CH1MSK	OCT	77000000	CHAR 1 MASK
\$CH2MSK	OCT	00770000	CHAR 2 MASK
\$CH3MSK	OCT	00007700	CHAR 3 MASK
SIZMSK	EXT	*	
SIZMSK	EQU	*	SIZE MASK
KYCMSK	EQU	*	KEY CODE MASK
\$CH4MSK	OCT	00000077	CHAR 4 MASK
\$NOTCMT	OCT	00777777	NOT CHAR MASK TABLE
	OCT	77007777	
	OCT	77770077	
\$MINS64	OCT	77777700	MINUS 64 (MUST FOLLOW NOTCMT)
\$CHARST	NOP	0	CHAR SHIFTER TABLE
	MUL	RSH6	RIGHT SHIFTER 6
	MUL	RSH12	RIGHT SHIFTER 12
	MUL	RSH18	RIGHT SHIFTER 18
\$BR9	BR	9	START FORK
\$BR10	BR	10	STOP FORK
\$BR31	BR	31	WAIT
\$CHR3ST	LRSH	2	CHAR SHIFTER TABLE , 3C/W
	LRSH	10	
	LRSH	18	

\$CHMT3B DATA	37700000B,00177400B,377B	
\$CLNEGL DATA	00037377B	CHARIS LOWER NEG LIMIT
CUP0SL EXT	*	
CUP0SL EQU	*	CHARIS UPPER POS LIMIT
\$TREMSK 0CT	00000377	TREE FIELD MASK
\$MAXLBL DEC	0	MAX LBL ALLOWED
DEC	100000	
\$MAXINT 0CT	00000000	MAX INTEGER ALLOWED
MADELB EXT	*	
MADELB EQU	*	MADE LBL
ALBUTS EQU	*	ALL BUT S
ABSM EQU	*	ALL BUT SIGN MASK
\$P0SFSC 0CT	37777777	P0S FULL SCALE
FIVMSK EQU	*	FIVE DEC MASK
\$ABSATM 0CT	17777777	ALL BUT SIGN AND TAG MASK
\$MAXASA 0CT	00000000	MAX ARRAY SIZE ALLOWED
\$RSH6 0CT	00400000	RIGHT SHIFTER 6
RSH12 EQU	*	RIGHT SHIFTER 12
\$DECHFG 0CT	00004000	DELETE CHAR FLAG
CNCHFG EQU	*	CONTINUATION CHAR FLAG
0NEA14 EXT	*	
0NEA14 EQU	*	0NE IN 14TH BIT
\$RSH14 0CT	00001000	RIGHT SHIFTER 14
LBLSTM EXT	*	
LBLSTM EQU	*	LABEL LIST MASK
STFLK EXT	200B	
STFLK EQU	200B	START 0F LINKS
ST0FLK EXT	*	
ST0FLK EQU	*	START 0F LINKS
\$RSH16 0CT	00000200	RIGHT SHIFTER 16
\$IJKLMN DATA	40B	IJKLMN FLAG
\$RSH18 0CT	00000040	RIGHT SHIFTER 18
\$F0URK DATA	00010000	4 THOUSAND
\$FXTPCD 0CT	07600000	FX TEMP PNTR C0DE
\$DYPTCD 0CT	10400000	DMY PNTR C0DE
\$C0DE 0CT	00740002	C0DE
0CT	01700002	
0CT	02400301	LABEL LIST C0DE
0CT	03440002	

0CT 04400101  
 0CT 05500005  
 0CT 06500002  
 0CT 07000003  
 0CT 10500003  
 0CT 11700003  
 0CT 12000002  
 0CT 13000003  
 0CT 14000101  
 0CT 15000003  
 0CT 16000003  
 0CT 17000101  
 0CT 20000101  
 0CT 21000101  
 0CT 22000102  
 0CT 23000101  
 0CT 24000101

\*  
 9BIT EXT 40000B  
 9BIT EQU 40000B  
 \$ACBLIM ZR0 77B  
           ZR0 177B  
 \$ACBSCT ZR0 18  
           ZR0 15  
 \$ACBMSK DATA 77400000B  
           DATA 77740000B  
 \$B6 DATA 400000B  
 \$B9 DATA 40000B  
 \$FTBLE 10B TBLE,1  
 \$BELL ZR0 147B  
 \$TW0H DATA 200  
 \$RTPERD ZR0 16B  
 \$C0MPL7 DATA 77777770B  
 \$HBPTI BSS 2  
 HLT0P EXT \*  
 HLT0P EQU \*  
 P0STRT EXT \*  
 P0STRT EQU \*  
 SKBLFG EXT \*

(MUST FOLLOW 77B  
 SHIFT COUNT  
 (MUST FOLLOW

F0RK TABLE ADDRESS

RIGHT JUSTIFIED PERIOD  
 HEADING BUFFER PTRS INITIAL  
 HLT INST  
 POSITIVE TRAIT

SKBLFG	EQU	*	SKIP BLANKS FLAG
DPONE	EXT	*	
DPONE	EQU	*	DP ONE
RUNTM0	EXT	414B	
RUNTM0	EQU	414B	RUN TIME ZERO
\$ZER0	ZR0	0	ZER0
DIGTFL	EXT	*	
DIGTFL	EQU	*	DIGIT FLAG
0PINDB	EXT	*	
0PINDB	EQU	*	M0P IND BIT
NEGTRT	EXT	*	
NEGTRT	EQU	*	NEGATIVE TRAIT
UBLKFG	EXT	*	
UBLKFG	EQU	*	USE BLANKS FLAG
RSH23	EQU	*	R SHIFTER 23
RNTFL1	EQU	415B	RUN TIME FL ONE
RNTFL1	EXT	415B	
RNTFX1	EQU	415B	RUN TIME FX ONE
RNTFX1	EXT	415B	
\$ONE	ZR0	1	ONE
RELADB	EQU	*	RELATIVE ADR BIT
MPYDLM	EXT	*	
MPYDLM	EQU	*	MULTIPLY DEF LBL MASK
SIGNBT	EXT	*	
SIGNBT	EQU	*	SIGN BIT
RNTSBT	EXT	417B	
RNTSBT	EQU	417B	RUN TIME SIGN BIT
\$SIGN	ZR0	0,4	SIGN BIT
ALONES	EQU	*	ALL ONES WORD
NEGFS	EXT	*	
NEGFS	EQU	*	NEG FULL SCALE
MINUS1	EXT	*	
MINUS1	EQU	*	MINUS ONE
\$ONES	DATA	=1	
PNTRAM	EXT	*	
PNTRAM	EQU	*	PNTR ADR MASK
\$ADRMSK	ZR0	37777B	ADDRESS MASK
\$A183I	EAX	SIZEM1,2	INITIAL CONTENTS FOR MODIFIED COMMAND
\$A184I	SKE	B0TSZ2,2	

\$A185I SKE B0TSZ3,2  
\$A187I EAX SIZEM1,2  
\$A188I SKM B0TSZ2,2  
\$A232I LDA B0TP1,2  
\$A233I STA B0TM1,2  
\$SCENI1 ASC ' . '  
\$SCENI2 ASC ' . '  
\$PRGEND DATA 32465152B  
\$EOFWRD DATA 27657537B  
END

SYMBOLIC CENTRAL INITIALIZE

PROGRAM END WORD  
END OF FILE WORD



\*  
 \*\*\*\*\* ORIGIN AT 400 \*\*\*\*\*  
 \*

\*  
 \*\*\*\*\*FIXED STORAGE LOCATIONS \*\*\*\*\*  
 \*

\*LOC 400  
 \*

\$EOADR ZR0 0  
 \$EOTAG ZR0 0  
 \$EOIND ZR0 0  
 \$EADR1 ZR0 0  
 \$EOSIZE ZR0 0

\$MPRHST ZR0	0	HEAD START OF MAIN PROGRAM
\$LNKXMN HLT	0	LINK X MIN
\$BLKSTL HLT	0	BLOCK START LOCATION
\$BOTLT2 HLT	0	BOTTOM LINK TBL=2
\$BOTLT1 HLT	0	BOTTOM LINK TBL=1
\$BOTLT HLT	0	BOTTOM LINK TBL
\$TEMP2 ZR0	0	TEMP 2
\$TEMP3 ZR0	0	
\$TVSTRT ZR0	0	TV START
\$AYTBST ZR0	0	ARRAY TABLE START
\$FXSPST ZR0	0	FX SPEC START
\$FLSPST ZR0	0	FL SPEC START
\$FLSPND BSS	0	FL SPEC END
\$FLAG ZR0	0	
\$RW0RD BSS	0	
\$STFLC0 ZR0	0	START FL C0N
\$LINKX ZR0	0	LINK X
\$MIFMPI ZR0	0	MINUS IF MAIN PR0G IN
\$LOWC0M ZR0	0	LOWEST C0MM0N ADR
\$BREAK ZR0	0	BREAK
\$PRTYPE ZR0	0	PR0G TYPE
\$MIFPRR ZR0	0	MINUS IF PR0GRAM REXTIRED
\$H0LDLX ZR0	0	H0LD LINK X

\$PRSTRT	ZR0	0	PR0G START
\$ENT	ZR0	0	ENT
\$NAME1	ZR0	0	NAME 1
\$NAME2	ZR0	0	NAME 2
\$NAME3	ZR0	0	INITIALIED WITH MSG TERMINATOR
\$WORD	ZR0	0	WORD
\$LABELX	ZR0	0	LABEL X
\$1STIAD	ZR0	0	FIRST INST ADR
\$ABSCNT	ZR0	0	ABS COUNT
\$TEMP1	ZR0	0	TEMP 1
\$L0CCNT	ZR0	0	L0C C0UNTER
\$LBXLES	ZR0	0	LABEL X LEAST
\$LNKPTR	BSS	0	
\$STFXC0	BSS	1	START FX C0N
\$STDUM	BSS	0	START 0F DUMMYS
\$HEADST	ZR0	0	HEAD START
\$STTEMP	BSS	0	START 0F TEMP
\$VADR	ZR0	0	V ADR
\$STXSCA	BSS	0	START 0F FX SCALARS
\$VC0UNT	ZR0	0	V C0UNT
\$STFSCA	BSS	0	START 0F FL SCALARS
\$C0UNT	ZR0	0	C0UNT
\$STNXPR	ZR0	0	START 0F NEXT PR0G
\$RUNT0P	ZR0	0	RUN-TIME T0P MEMORY
ENDABS EXT	250B		END ABS LINK TABLE
\$MS0PSZ	ZR0	0	MINUS SPEC 0P TABLE SIZE
\$CNVHWD	ZR0	0	C0NV H0LD W0RD
\$10RTW0	ZR0	0	0NE 0R TW0
\$CNVHSW	ZR0	0	C0NV H0LD SERCH W0RD
\$STSCAL	ZR0	0	START 0F FX 0R FL SCALARS
\$SCALTB	ZR0	0	SCAL TABLE ADR
\$TABLCT	BSS	0	TABLE C0UNT
\$VECADR	ZR0	0	VECT ADR
\$HITCNT	BSS	0	HIT C0UNT
\$VECCNT	ZR0	0	VECT C0UNT
\$C0RENT	ZR0	0	C0RRECTED ENT
\$BFWADR	ZR0	0	BUFFER W0RD ADR
\$BUFCNT	ZR0	0	BUFFER C0UNT
\$CHKSUM	ZR0	0	CHECKSUM

\$LASTWD	BSS	0	LAST WORD
\$1WD0BL	ZR0	0	FIRST WORD OF BLOCK
\$MLKKTS	ZR0	0	MACH LINK TBL SIZE
\$MULSR	ZR0	0	MINUS UNLESS LOAD SYSTEM REQ
\$TPMPLK	ZR0	0	LAST ADR OF MEMORY + LBL KEY
\$MEMSIZ	ZR0	0	
\$MIRMTE	ZR0	0	
\$RELWRD	ZR0	0	WORD OF ARPAS 3 BIT RELOCATION REGISTERS
\$RELWCT	ZR0	0	RELOCATION WORD REGISTERS COUNT
\$CURWRD	ZR0	0	CURRENT WORD
\$L0C	ZR0	0	
\$TMP1	ZR0	0	
\$TMP2	ZR0	0	
\$TMP3	ZR0	0	
\$FRUB	ZR0	0	
\$INFLE	ZR0	0	
\$CHAR	ZR0	0	
\$TBL	BSS	7	FORK TABLE
TBL6	EXT	TBL+6	
\$NSBRML	ZR0	0	NEXT SBRM LINK POINTER
\$FSTBRK	ZR0	0	
\$FDEBUG	ZR0	0	*1=PR0G CURRENTLY BEING LOADED HAS DEBUG
\$SYMPT1	ZR0	0	
\$SYMPT2	ZR0	0	
\$S0S	ZR0	0	START OF SYMBOLS AT END OF PROGRAM
*			
\$G0RUN	BSS	7	STORAGE FOR RUNTIME BOOTSTRAP
TSRLB	EQU	*	
TSRLB	EXT	*	
TSRLB1	EXT	**1	
*LINKS			
\$LACB	ZR0	0	
\$LNW0R	ZR0	0	
\$LSERHL	ZR0	0	
\$LILLF1	ZR0	0	
\$LILLTA	ZR0	0	
\$LINCBR	ZR0	0	

\$LSTBRK ZR0 0  
\$LNXPDP ZR0 0  
\$LC0NV ZR0 0  
\$LTRA VE ZR0 0  
\$LADJCB ZR0 0  
\$LREPAC ZR0 0  
\$LSRCHS ZR0 0  
\$LSRCHR ZR0 0  
\$LCKPRA ZR0 0  
\$LPVECL ZR0 0  
\$LOSUBF ZR0 0  
\$CESYMS ZR0 0  
\$CESYMP ZR0 0  
\$CESYM ZR0 0  
END

PTR TO 1ST UNUSED WORD IN CESYM VECTOR  
CURRENT EXT SYMBOLS

\*  
 \*\*\*\*\*  
 \*ORIGIN THIS PACKAGE AT 30000B \* \* \*  
 \*  
 \*\*\*\*\*

\*  
 PRC    θPD    17000000B,1,1  
 PRQ    θPD    17100000B,1,1  
 FCL    θPD    4600450B,2  
 FCR    θPD    4600224B,2        FULL CYCLE RIGHT  
 XBC    θPD    24600040B,2        X TO B, CLX  
 CA     EQU    1  
 CB     EQU    2  
 AB     EQU    4            A TO B  
 BX     EQU    20B  
 XB     EQU    40B  
 XA     EQU    200B  
 AX     EQU    400B  
 \$FθRL   EQU   5455B            FθS SMT RELABELING BYTES

\*  
 BNE    MACRθ   D; SKE D(1); BRU D(2); ENDM  
 BEQ    MACRθ   D; SKE D(1); BRU \*\*2; BRU D(2); ENDM  
 DEC    MACRθ   D; DATA D(1); ENDM  
 θCT    MACRθ   D; DATA D(1)θB; ENDM  
 MθVE   MACRθ   D  
 MθVE1   NARG  
       IF       'D(1\$1,2)'='θ'  
       CLθD(MθVE1)  
       ELSF     1  
       LDθD(MθVE1) D(1)  
       ENDF  
 MθVE2   EQU    2  
       RPT     MθVE1-2  
       STθD(MθVE1) D(MθVE2)  
 MθVE2   EQU    MθVE2+1  
 ENDR

ENDM

```

*
*
$PS BRU L1 PROGRAM START
$RS LDA RELAB
STA TSRLB1
BRU GORUN
L1 BRS 43 ***
ETR =770000B ***THESE INSTRUCTIONS NOT NEEDED FOR
BRS 44 ***SYSTEM 'B' (EXEC 8.0) OR HIGHER.
LDX =-4000B
CLA
STA 4000B,2
BRX *=1
MOVE =NEWS,TBLE,A
L2 LDA TOPMEM LAST ADDR OF OBJECT PR0G CORE
ADD ONE ONE
STA MEMSIZ SIZE OF MACHINE
MRG TAGBIT TAG BIT
STA BOTLT BOTTOM LINK TBL
SUB ONE ONE
STA BOTLT1 BOTTOM LINK TBL=1
SUB ONE ONE
STA BOTLT2 BOTTOM LINK TBL=2
MOVE ALLD0L,NAME1,NAME2,A
MOVE MSGTRM,NAME3,A MESSAGE TERMINATOR CHARACTER
LDA TOPMEM LAST ADR OF MEMORY
ADD LBLKEY LBL KEY
STA TPMLK LAST ADR OF MEMORY + LBL KEY
LDA MSIX MINUS SIX
STA MS0PSZ MINUS SPEC OF TABLE SIZE
SKR MS0PSZ MINUS SPEC OF TABLE SIZE
SKR MS0PSZ MINUS SPEC OF TABLE SIZE
LDA MTWELV MINUS TWELVE
SUB MS0PSZ MINUS SPEC OF TABLE SIZE
STA MS0PSZ MINUS SPEC OF TABLE SIZE
LDA SIGNBT SIGN BIT
STA MULSR MINUS UNLESS LOAD SYSTEM REQ
LDX =-7 SET RUNTIME RELABELING CODE

```

```

      LDA      RELAB,2          IN LOADER TS BLOCK
      STA      TSRLB,2
      BRX      *-2
*SBRM LINKS WORK DOWN FROM CODE START
      LDA      CODEST
      SUB      #1
      STA      NSBRML          NEXT SB RM LINK LOCATION
      MOVE     #CESYM,CESYMS,CESYMP,A
      CLAB
      LDX      MABSLS
L3     STA      ENDABS,2          END ABS LINK TABLE
      BRX      L3              L2
      STA      LINKX           LINK X
      STA      MIFMPI          MINUS IF MAIN PR OG IN
      STA      MIRMTE          MINUS IF READING MACH TO END
      LDA      TOPMEM          LAST ADR OF MEMORY
      STA      LOWCOM          LOWEST COMMON ADR
      LDA      CODEST          CODE STARTING ADDR
      STA      MPRHST
      STA      BREAK           BREAK
      STA      LBXLES          LABEL X LEAST
      MOVE     #1,FRUB,A
      MOVE     PRCLNK,170B,A
      MOVE     PRQLNK,171B,A
P3     BRU      P2
      BRS      17
P2     MOVE     #FORK,TBLE,A
      LDA      FTBLE1
      BRS      9
      NOP      FORK
$WAIT BRS      31
      LDX      TBLE6
      BRU      *+1,2
      BRU      RUB
      BRU      BADOP
      PRQ      MEMQT
      BRU      PRLQC
BADOP PRQ      ILLQT
PRLQC LDA      TBLE

```

```

ETR      =37777B
LDB      =8
LDX      =1
BRS      36
BRS      10
RUB      LDA      0
          ETR      =37777B
          BEQ      =NEWS,RUB10
          SKN      FRUB          =1=N0 2ND RUB0UT
          BRU      RUB3
          MOVE     =0,FRUB,A
P5       LDA      TBLE
          ETR      =37777B
*CHECK THAT USER WAS IN AN INTERRUPTABLE AREA WHEN HE INTERRUPTED LOADER
          SKG      =E0SUBF      OPEN SUBPROGRAM FILE ROUTINE
          SKG      =0SUBF
          BRS      10          NOT OK, GO BACK TO EXEC
          MOVE     =21,TBLE,A
          BRU      P2
RUB3     LDA      TBLE
          ETR      =37777B
          SKE      =ENDLD
          BRS      10          RUB0UT BUTTON WAS PUSHED
          BRU      G0RUN      RELABEL IN RUNTIME FROM TS BLOCK
*                               IN SYS2 SW RELABELING AND GO TO R

```

```

RUB10   LDX      =1
          BRS      29
          BRU      P3
*USER NEWS MESSAGE
NEWS    PRQ      NEWSQT
          BRU      F0RK
ERR1    PRQ      FNBQT          FILE NOT BINARY
F0RK    PRQ      LMPQT          LOAD MAIN PROGRAM
F0RK10  LDX      =1 }
          BRS      14 } N0P
          BRS      11 }
F0RK1   PRQ      FR0MGT          CLEAR THE TTY INPUT BUFFER

```



```

CLEAR
LDX      #10000B
BRS      15
BRU      FØRKE
MOVE     #1,FRUB,X
BRS      16
BRU      FØRKE
XAB
BNE      #2,ERR1
STB      INFLE
*READ NEXT PRØG BLØCK
RNXPLK LDA      MEMSIZ      SIZE OF MACHINE
SUB      LBXLES      LABEL X LEAST
ADD      LINKX      LINK X
STA      TEMP1      TEMP 1
CLAB
STA      FDBUG      SET FALSE
STA      FSTBRK
BRU      L5
L4 STA*      LBXLES      LABEL X LEAST
MIN      LBXLES      LABEL X LEAST
L5 SKR      TEMP1      TEMP 1
BRU      L4      L10
SBRM     NXWORD      NEXT WORD
STA      PRTYPE      PRØG TYPE
LDB      ØNES      ØNES
SKN      MIFMPI      MINUS IF MAIN PRØG IN
BRU      L6      L3
SKM      MPRTYP      MAIN PRØG TYPE
BRU      L7      L4
L6 BRU      RDENDP      READ TO END OF ALL BUT FIRST MAIN PRØG
SKM      MPRTYP      MAIN PRØG TYPE
BRU      MANMIS      MAIN PRØGRAM MISSING
LDA      SIGN      SIGN BIT
L7 STA      MIFMPI      MINUS IF MAIN PRØG IN
STA      MIFPRR      MINUS IF PRØGRAM REQUIRED
LDA      LINKX
STA      HOLDLX
LDA      BREAK
BREAK

```

	STA	PRSTRT	PR0G START
	SKN	PRTYPE	
	BRU	L7L	COMPILED
	BRU	ASSM	ASSEMBLED
L7L	SBRM	NXWORD	NEXT WORD
	STA	ENT	ENT
	SBRM	NXWORD	NEXT WORD
	STA	NAME1	
	LDX	BREAK	BREAK
	STA	1,2	1
	SBRM	NXWORD	NEXT WORD
L8L	STA	NAME2	
	STA	2,2	2
L8	LDA	BREAK	BREAK
	ADD	EIGHT	EIGHT
	STA	BREAK	BREAK
	SUB	TOPMEM	LAST ADR OF MEMORY
	STA	LNKXMN	LINK X MIN
L10	LDA	BREAK	BREAK
	STA	1STIAD	FIRST INST ADR
	STA	L0CCNT	L0C COUNTER
L11	SBRM	SERHLT	SERCH LINK TABLE
L12	SBRM	NXWORD	NEXT WORD
	LDB	8BIT0P	8 BIT 0P MASK
	SKM	BLKL0P	BLK L0P
	BRU	L13	L1
	BRU	RDTEXT	READ TEXT
L13	STA	ENT	ENT
	SKA	INDBIT	IND BIT
	BRU	L14	L7
	SKA	TAGP0P	TAG P0P BITS
	BRU	L11	L2
	SBRM	NXWORD	NEXT WORD
	STA	NAME1	NAME 1
	SBRM	NXWORD	NEXT WORD
	STA	NAME2	NAME 2
	BRU	L11	L2
L14	ETR	8BIT0P	8 BIT 0P MASK
	RSH	15	17

SKN	MIFPRR	MINUS IF PROGRAM REQUIRED
BRU	L12	L8
STA	TEMP1	TEMP 1
LDB	REQBIT	REQUIRED BIT
LDA	ONE	ONE
SKG*	TEMP1	TEMP 1
BRU	L12	L8
STB*	TEMP1	TEMP 1
BRU	L12	L8

\*

\*

	*READ COMPILED TEXT	
RDTEXT	ETR	ADR MASK
	STA	MACH LINK TBL SIZE
	SKN	MINUS IF PROGRAM REQUIRED
	BRU	READ TO END OF PROGRAM
L15	LDA	LAST ADR OF MEMORY
	ADD	LINK X
L16	STA	LABEL X LEAST
	SBRM	NXWORD
	LDB	OP MASK
	LDX	MINUS SPEC OP TABLE SIZE
L17	SKM	OP TABLE 1
	BRU	L3
	ETR	NOT OP MASK
	MRG	OP TABLE 2
L18	BRX	L2
	STA	WORD
	SKM	LBL LOP
	BRU	L5
	ETR	ADR MASK
	SKN	WORD
	ADD	LBL KEY
	SUB	LAST ADR OF MEMORY + LBL KEY
	EOR	ONES
	ADD	ONE
	ADD	LINK X
	SKG	LABEL X LEAST
	STA	LABEL X LEAST

	STA	LABELX	LABEL X
	LDA	LBXLES	LABEL X LEAST
	SKG	BREAK	BREAK
	BRU	PR2BIG	PROGRAM TOO BIG
	LDA	L0CCNT	L0C COUNTER
	STA*	LABELX	LABEL X
	BRU	L16	L4
L19	SKM	ABSL0P	ABS L0P
	BRU	L22	L6
	STA*	BREAK	BREAK
	ETR	ADRMSK	ADR MASK
	STA	ABSCNT	ABS COUNT
	SBRM	INCBRK	INCR BREAK
	BRU	L21	L7
L20	MIN	L0CCNT	L0C COUNTER
	SBRM	NXWORD	NEXT WORD
	STA*	BREAK	BREAK
	SBRM	INCBRK	INCR BREAK
L21	SKR	ABSCNT	ABS COUNT
	N0P		
	SKN	ABSCNT	ABS COUNT
	BRU	L20	L8
	BRU	L16	L4
L22	SKM	DELL0P	DEL L0P
	BRU	L23	L9
	BRU	L27	L10
L23	SKM	SYSL0P	SYS L0P
	BRU	L25	L11
	ETR	N0PMSK	N0T 0P MASK
	MRG	SBRMIND	SBRM M0P
	STA*	BREAK	BREAK
	ETR	ADRMSK	ADR MASK
	STA	TEMP1	TEMP 1
	LDA*	TEMP1	TEMP 1
	SKA	0NES	0NES
	BRU	L24	L14
	MRG	REQBIT	REQUIRED BIT
	STA*	TEMP1	TEMP 1
L24	MIN	L0CCNT	L0C COUNTER

L25	BRU SKM BRU BRU	L28 BLKL0P L26 RDTAIL	L12 BLK L0P L13 READ TAIL
L26	MIN	L0CCNT	L0C COUNTER
L27	STA*	BREAK	BREAK
L28	SBRM	INCBRK	INCR BREAK
L28A	BRU MOVE BRU	L16 H0LDLX,LINKX,A RDENDP	L4

*READ TAIL			
RDTAIL	LDA	TOPMEM	
	ADD	LABELX	
	SUB	#1	
	STA	SYMPT1	
	STA	SYMPT2	
	LDA	L0CCNT	L0C COUNTER
	STA	TVSTRT	TV START
	LDA	BREAK	BREAK
	STA	AYTBST	ARRAY TABLE START
	SBRM	STBRK	STORE TO BRK
	LDA	BREAK	BREAK
	STA	FXSPST	FX SPEC START
	SBRM	STBRK	STORE TO BRK
	LDA	BREAK	BREAK
	STA	FLSPST	FL SPEC START
	SBRM	STBRK	STORE TO BRK
	LDA	BREAK	BREAK
	STA	FLSPND	FL SPEC END
*PROCESS 10	SPECIAL WORDS AT END OF	PROGRAM FILE	
	LDX	PRSTRT	PR0G START
	SBRM	NXWORD	NEXT WORD
	STA	TEMP1	TEMP 1
	SBRM	NXWORD	NEXT WORD
	E0R	0NES	0NES
	ADD	0NE	0NE
	ADD	TVSTRT	TV START

STA	STFLC0	START FL C0N
SUB	TEMP1	TEMP 1
STA	STFXC0	START FX C0N
SBRM	NXW0RD	NEXT W0RD
ADD	TVSTRT	TV START
STA	TVSTRT	TV START
STA	3,2	3
SBRM	NXW0RD	NEXT W0RD
ADD	TVSTRT	TV START
STA	STDUM	START 0F DUMMYS
STA	4,2	4
SBRM	NXW0RD	NEXT W0RD
ADD	TVSTRT	TV START
STA	STTEMP	START 0F TEMP
SBRM	NXW0RD	NEXT W0RD
ADD	TVSTRT	TV START
STA	5,2	5
SBRM	NXW0RD	NEXT W0RD
ADD	TVSTRT	TV START
STA	STXSCA	START 0F FX SCALARS
STA	6,2	6
SBRM	NXW0RD	NEXT W0RD
ADD	TVSTRT	TV START
STA	STFSCA	START 0F FL SCALARS
STA	7,2	7
SBRM	NXW0RD	NEXT W0RD
ADD	TVSTRT	TV START
STA	STNXPR	START 0F NEXT PR0G
ADD	PRTYPE	PR0G TYPE
STA	0,2	0
SBRM	NXW0RD	NEXT W0RD
E0R	0NES	0NES
ADD	0NE	0NE
ADD	MEMSIZ	SIZE 0F MACHINE
SKG	L0WC0M	L0WEST C0MM0N ADR
STA	L0WC0M	L0WEST C0MM0N ADR
*SEC0ND PASS		
2NDPAS LDA	L0WC0M	L0WEST C0MM0N ADR
SKG	STNXPR	START 0F NEXT PR0G

	BRU	PR2BIG	PROGRAM TOO BIG
	LDA	1STIAD	FIRST INST ADR
	STA	L0CCNT	L0C COUNTER
	STA	BREAK	BREAK
*SBRM	LINK SET	UP IF COMPILED	SUBPROGRAM
	LDA	PRTYPE	PROGRAM TYPE
	SKE	MPRTYP	MAIN PROGRAM TYPE
	BRU	*+2	
	BRU	L29	MAIN PR0G
	LDA	NSBRML	NEXT SBRM LINK LOCATION
	STA*	1STIAD	1ST INSTRUCTION ADDR
	SKR	NSBRML	DECREMENT POINTER
L29	SBRM	NXWDP2	NEXT WORD PASS 2
	BRU	L35	L5
	LDB	0PMASK	0P MASK
	SKM	ABSL0P	ABS L0P
	BRU	L32	L1
	ETR	ADRMSK	ADR MASK
	STA	ABSCNT	ABS COUNT
	BRU	L31	L2
L30	SBRM	NXWDP2	NEXT WORD PASS 2
	SBRM	ILLTAP	ILLEGAL TAPE
	STA*	L0CCNT	L0C COUNTER
	MIN	L0CCNT	L0C COUNTER
L31	SKR	ABSCNT	ABS COUNT
	BRU	L30	L3
	BRU	L29	L4
L32	SKM	DELL0P	DEL L0P
	BRU	L34	L6
	ETR	ADRMSK	ADR MASK
	STA	TEMP2	
	SBRM	NXWDP2	NEXT WORD PASS 2
	SBRM	ILLTAP	ILLEGAL TAPE
	SBRM	CONV	CONV
	ADD	TEMP2	
	ETR	NINDBT	NOT IND BIT
L33	STA*	L0CCNT	L0C COUNTER
	MIN	L0CCNT	L0C COUNTER
	BRU	L29	L4

L34	SBRM	CONV	CONV
	BRU	L33	L7
L35	LDA	STNXPR	START OF NEXT PRGG
	SUB	L0CCNT	
	STA	TEMP1	
	CLAB		
L36	STA*	L0CCNT	
	MIN	L0CCNT	
	SKR	TEMP1	
	BRU	L36	
L36B	LDA	STNXPR	
	SUB	T0PMEM	LAST ADR OF MEMORY
	STA	LNKXMN	LINK X MIN
	CNA		
	ADD	LINKX	
	STA	TEMP2	
	MOVE	STNXPR, TEMP3, A	
	CLA		
L36L	STA*	TEMP3	
	MIN	TEMP3	
	SKR	TEMP2	
	BRU	L36L	
	LDB	TVSTRT	TV START
	LDA	STDUM	START OF DUMMYS
	SUB	TVSTRT	TV START
	SBRM	TRAVEC	PROCESS TRA VECT

\*LOAD DEBUGGING SYMBOLS

	SBRM	NXWORD	
	ETR	=37777B	
	STA	TEMP1	
	LDA*	PRSTRT	PROGRAM START
	ETR	=37777B	
	STA	BREAK	
	STA	TEMP2	
L36Y	SKR	TEMP1	
	BRU	**2	
	BRU	L36X	



	SBRM	NXWORD	
	STA*	BREAK	
	MIN	BREAK	
	LDA	LOWCOM	LOWEST COMMON ADDR
	SKG	BREAK	
	BRU	PR2BIG	
	BRU	L36Y	
L36X	MIN	BREAK	
	LDA	BREAK	
	STA	STNXPR	START OF NEXT PROGRAM
	XMA*	PRSTRY	PROGRAM START
	ETR	=77740000B	
	ADM*	PRSTRY	
	LDA	TEMP2	
	LDX	BREAK	
	STA	=1,2	
	BRU	CKALPG	CHECK ALL PROGRAMS IN

\*  
\*

	*NEXT WORD PASS 2		
NXWDP2	HLT	LNWDP	
	LDA	AYTBST	ARRAY TABLE START
	SKG	BREAK	BREAK
	SBRR	NXWDP2	NEXT WORD PASS 2
	LDA*	BREAK	BREAK
	MIN	BREAK	BREAK
	MIN	LNWDP	NEXT WORD PASS 2
	BRR	LNWDP	NEXT WORD PASS 2

\*  
\*

	*CONV		
CONV	HLT	LCONV	
	SKA	SIGN	SIGN BIT
	BRU	L37	L1
	SBRR	CONV	BIT ZERO NOT SET = RETURN IMMEDIATELY
L37	STA	CNVHWD	

	LDB	=57000000B	
	SKM	=57000000B	
	BRU	L37A	
	ETR	=37777B	SBRM OR SBRR
	SKG	C0DEST	
	BRU	**2	
	BRU	L37A	
	LDA	CNVHWD	
	SBRR	C0NV	
L37A	ETR	ADRMSK	ADR MASK
	SUB	DUMKEL	DUMMY KEY
	SKA	SIGN	SIGN BIT
	BRU	L38	L2
	ADD	STDUM	START OF DUMMYS
	BRU	L48	L3
L38	ADD	DUMTKY	DUMMY=TEMP KEY
	SKA	SIGN	SIGN BIT
	BRU	L39	L4
	ADD	STTEMP	START OF TEMP
	BRU	L48	L3
L39	ADD	TMPKY	TEMP=LINKAGE KEY
	SKA	SIGN	SIGN BIT
	BRU	L40	L5
	RSH	1	1
	ADD	TVSTRT	TV START
	BRU	L48	L3
L40	ADD	LNKARK	LINKAGE=ARRAY KEY
	SKA	SIGN	SIGN BIT
	BRU	L41	L6
	STA	TEMP1	TEMP 1
	ADD	TEMP1	TEMP 1
	ADD	TEMP1	TEMP 1
	STA	TEMP1	TEMP 1
	RSH	4	4
	ADD	TEMP1	TEMP 1
	RSH	4	4
	ADD	TEMP1	TEMP 1
	ADD	EIGHT	EIGHT
	RSH	4	4

	ADD	AYTBST	ARRAY TABLE START
	STA	TEMP1	TEMP 1
	LDX	TEMP1	TEMP 1
	LDA	0,2	0
	SBRM	ADJCOM	ADJUST FOR COMMON
	BRU	L48	L3
L41	ADD	ARFXCK	ARRAY=FX CONST KEY
	SKA	SIGN	SIGN BIT
	BRU	L42	L7
	ADD	STFXC0	START FX CON
	BRU	L48	L3
L42	ADD	XLC0KY	FX CONST=FL CONST KEY
	SKA	SIGN	SIGN BIT
	BRU	L43	L8
	ADD	STFLC0	START FL CON
	BRU	L48	L3
L44	ADD	XSLCKY	FL CONST=FX SCALAR KEY
	SKA	SIGN	SIGN BIT
	BRU	L47	L10
	RSH	1	1
	STA	CNVH5W	CONV HOLD SERCH WORD
	LDA	0NE	0NE
	STA	10RTW0	0NE OR TW0
	LDX	STXSCA	START OF FX SCALARS
	LDB	FXSPST	FX SPEC START
	LDA	FLSPST	FL SPEC START
	SBRM	SRCHST	SEARCH SCALAR TABLE
	BRU	L48	L3
L43	ADD	XSLBKY	FX SCALAR=LBL KEY
	SKA	SIGN	SIGN BIT
	BRU	L44	L9
	E0R	0NES	0NES
	ADD	0NE	0NE
	ADD	T0PMEM	LAST ADR OF MEM0RY
	ADD	LINKX	LINK X
	STA	TEMP1	TEMP 1
	LDA	LBXLES	LABEL X LEAST
	SKG	TEMP1	TEMP 1
	BRU	L46	L11

L45	LDA	UNDLAD	UNDEFINED LABEL ADR
	BRU	L48	L3
L46	LDA*	TEMP1	TEMP 1
	SKA	ONES	ONES
	BRU	L48	L3
	BRU	L45	L12
L47	ADD	FXSKEY	FX SCALAR KEY
	STA	CNVHSH	CNV HOLD SERCH WORD
	LDA	TW0	TW0
	STA	10RTW0	ONE OR TW0
	LDX	STFSCA	START OF FL SCALARS
	LDB	FLSPST	FL SPEC START
	LDA	FLSPND	FL SPEC END
	SBRM	SRCHST	SEARCH SCALAR TABLE
L48	STA	TEMP1	TEMP 1
	LDA	CNVHWD	CNV HOLD WORD
	ETR	ALBADR	ALL BUT ADR MASK
	ADD	TEMP1	TEMP 1
	LDB	*57600000B	
	SKM	*57000000B	EXCEPT FOR SBRM AND SBRR
	ETR	*37777777B	REMOVE ZERO BIT
	SBRR	C0NV	C0NV

\*  
\*

\*ADJUST FOR COMMON

ADJCOM	HLT	LADJCO	
	SKA	SIGN	SIGN BIT
	BRU	L49	L1
	ADD	TVSTRT	LOCAL VARIABLE
	BRU	L50	L2
L49	ADD	MEMSIZ	COMMON VARIABLE
L50	ETR	ADRMSK	ADR MASK
	SBRR	ADJCOM	ADJUST FOR COMMON

\*  
\*

\*SEARCH SCALAR TABLE

SRCHST	HLT	LSRCHS	
	STX	STSCAL	START OF FX OR FL SCALARS
	STB	SCALTB	SCAL TABLE ADR

	SUB	SCALTB	SCAL TABLE ADR
	RSH	1	1
	STA	TABLCT	TABLE COUNT
	CLAB		
	STA	HITCNT	INITIALIZE HIT COUNT
	BRU	L55	L1
L51	LDA*	SCALTB	SCAL TABLE ADR
	SKG	CNVHSW	CONV HOLD SERCH WORD
	BRU	L52	L2
	BRU	L54	L3
L52	LDB	ONES	ONES
	SKM	CNVHSW	CONV HOLD SERCH WORD
	BRU	L53	L4
	BRU	L56	L5
L53	LDA	1ORTW0	ONE OR TWO
	ADD	HITCNT	HIT COUNT
	STA	HITCNT	HIT COUNT
L54	MIN	SCALTB	SCAL TABLE ADR
	MIN	SCALTB	SCAL TABLE ADR
L55	SKR	TABLCT	TABLE COUNT
	BRU	L51	L6
	LDA	CNVHSW	CONV HOLD SERCH WORD
	SUB	HITCNT	HIT COUNT
	ADD	STSCAL	START OF FX OR FL SCALARS
	SBRR	SRCHST	SEARCH SCALAR TABLE
L56	MIN	SCALTB	SCAL TABLE ADR
	LDA*	SCALTB	SCAL TABLE ADR
	SBRM	ADJCOM	ADJUST FOR COMMON
	SBRR	SRCHST	SEARCH SCALAR TABLE

\*INCR BREAK

INCBRK	HLT	LINCBR	
	MIN	BREAK	BREAK
	LDA	LBXLES	LABEL X LEAST
	SKG	BREAK	BREAK
	BRU	PR2BIG	PROGRAM TOO BIG
	SBRR	INCBRK	INCR BREAK

\*STORE TO BRK

\*X SAVED

STBRK	HLT	LSTBRK	
L57	SBRM	NXWORD	NEXT WORD
	LDB	OPMASK	OP MASK
	SKM	BLKLOP	BLK LOP
	BRU	L58	L1
	SBRR	STBRK	STORE TO BRK
L58	STA*	BREAK	BREAK
	SBRM	INCBRK	INCR BREAK
	BRU	L57	L2

\*PROCESS TRA VECT

TRAVEC	HLT	LTRAVE	
	STB	VECADR	VECT ADR
	STA	VECCNT	VECT COUNT
	BRU	L61	L1
L59	SBRM	NXWORD	NEXT WORD
	STA	NAME1	NAME 1
	SBRM	NXWORD	NEXT WORD
	STA	NAME2	NAME 2
	SBRM	SRCHRP	SEARCH LINKS RETURN PNTR
	STA*	VECADR	VECT ADR
	STA	LNKPTR	LINK PNTR
	LDA*	LNKPTR	LINK PNTR
	SKA	ONES	ONES
	BRU	L60	L3
	LDA	REQBIT	REQUIRED BIT
	STA*	LNKPTR	LINK PNTR
L60	MIN	VECADR	VECT ADR
L61	SKR	VECCNT	VECT COUNT
	BRU	L59	L2
	SBRR	TRAVEC	PROCESS TRA VECT

```

*SERCH LINK TABLE
SERHLT HLT LSERHL
LDA ENT ENT
ETR ADRMSK ADR MASK
ADD 1STIAD FIRST INST ADR
MRG BRUM0P BRU M0P
STA C0RENT C0RRECTED ENT
LDA NAME1
LDB =70000000B
SKM =20000000B TEST F0R SYS SPR0G, 200 = 235
BRU **2
BRU L62
SBRM SRCHRP SEARCH LINKS RETURN PNTR
BRU L63 L1
L62 ETR 8BIT0P 8 BIT 0P MASK
RSH 15 17
L63 SBRM CKPRAD CHECK PNTR ADR
SBRR SERHLT SERCH LINK TABLE
*CHECK PNTR ADR
CKPRAD HLT LCKPRA LINK PNTR
STA LNKPTR
CLB
SKE TOPMEM
STB MIFPRR MINUS IF PR0GRAM REQUIRED
LDA* LNKPTR LINK PNTR
SKA REQBIT REQUIRED BIT
BRU L64 L1
SKA 0NES 0NES
SBRR CKPRAD CHECK PNTR ADR
LDA ENT ENT
SKA TAGP0P TAG P0P BITS
SBRR CKPRAD CHECK PNTR ADR
BRU L65 L2
L64 LDA SIGN SIGN BIT
STA MIFPRR MINUS IF PR0GRAM REQUIRED
L65 LDA C0RENT C0RRECTED ENT
STA* LNKPTR LINK PNTR
SBRR CKPRAD CHECK PNTR ADR

```

\*

```

*SEARCH LINKS RETURN PNTR
SRCHRP HLT      LSRCHR
        LDA      LINKX      LINK X
        SKA      0NES       0NES
        BRU      L66        L4
        BRU      L70        L3
L66     LDX      LINKX      LINK X
        LDB      0NES       0NES
L67     LDA      NAME1      NAME 1
L68     EAX      2,2        2
        SKM*     B0TLT2     B0TT0M LINK TBL*2
        BRX      L68        L2
        LDA      NAME2      NAME 2
        SKM*     B0TLT1     B0TT0M LINK TBL*1
        BRX      L67        L1
L69     STX      TEMP1      TEMP 1
        LDA      TEMP1      TEMP 1
        ETR      ADRMSK     ADDRESS MASK
        SKG      0NE       0NE
        BRU      L70        L3
        BRU      L71        L8

```

```

*MAKE ENTRY IN REQUIRED PROGRAMS LIST
L70     LDA      LINKX      LINK X
        SUB      THREE     THREE
        STA      LINKX      LINK X
        SKG      LNKXMN     LINK X MIN
        BRU      PR2BIG     PROGRAM TOO BIG
        STA      TEMP1      TEMP 1
        LDX      TEMP1      TEMP 1
        EAX      2,2        2
        LDA      NAME1      NAME 1
        STA*     B0TLT2     B0TT0M LINK TBL*2
        LDA      NAME2      NAME 2
        STA*     B0TLT1     B0TT0M LINK TBL*1
        CLAB
        BRU      L69        L7

```



L71 EAX\* B0TLT BOTTOM LINK TBL  
 CXA  
 ETR ADRMSK ADR MASK  
 SBRR SRCHRP SEARCH LINKS RETURN PNTR

\*CHECK FOR ALL PROGS IN  
 CKALPG LDX LINKX LINK X  
 LDA REQBIT REQUIRED BIT

\*NAMED SUBPROGRAMS  
 L72 EAX 2,2  
 SKA\* B0TLT BOTTOM LINK TBL  
 BRU RDENDP  
 BRX L72 L1

\*NUMBERED SYSTEM SUBROUTINES (240-277)  
 LDX #240B-277B  
 L73 SKA 277B,2  
 BRU RDENDP READ TO END OF PROGRAM  
 BRX L73

\*ALL PROGS IN  
 ALPRIN MOVE #CESYM,CESYMP,A  
 BRU L74G  
 L74B LDX CESYMP  
 LDA 0,2  
 STA NAME1  
 LDA 1,2  
 STA NAME2  
 SBRM SRCHRP

CAX  
 LDA 0,2  
 ETR #37777B  
 CAB VALUE OF THE LABEL  
 \*ADDR CHAIN HEADER  
 LDX CESYMP  
 LDX 2,2  
 \*GO DOWN CHAIN

L74C	CBA	0,2	
	XMA	TMP1	
	STA	#77740000B	
	ETR	0,2	
	ADM	TMP1	
	LDA	#37777B	
	ETR		
	CAX		
	BNE	ZERO,L74C	
L74D	LDA	THREE	NEXT EXT SYMB
	ADM	CESYMP	
L74G	LDA	CESYMS	
	SKG	CESYMP	
	BRU	L74A	
	BRU	L74B	
L74A	LDA	MPRHST	MAIN PR0G HEAD START
L74	STA	HEADST	HEAD START
	LDA*	HEADST	HEAD START
	SKA	0NES	0NES
	BRU	L75	L1
	BRU	STUPE0	SET UP E0
L75	SKA	MAPTYP	MACH PR0G TYPE
	BRU	L74	
	LDX	HEADST	HEAD START
	LDA	4,2	4
	SUB	3,2	3
	LDB	3,2	3
L76	SBRM	PVECLT	PR0CESS VECT LAST TIME
	LDA*	HEADST	HEAD START
	BRU	L74	L3

*PR0CESS VECT LAST TIME			
L78	PVECLT	HLT	LPVECL
		STB	VADR
		STA	VC0UNT
		BRU	L79
		LDX*	VADR
			V ADR
			V C0UNT
			L1
			V ADR

```

L79 LDA 0,2 0
    STA* VADR V ADR
    MIN VADR V ADR
    SKR VCOUNT V COUNT
    BRU L78 L2
    SBRR PVECLT PROCESS VECT LAST TIME

```

```

*SET UP EO
STUPE0 LDA HEADST HEAD START
    ETR ADRMSK ADR MASK
    STA EOADR 71
    MRG TAGBIT TAG BIT
    STA EOTAG 73
    LDA EOADR 71
    MRG INDBIT IND BIT
    STA EOIND 74
    LDA LOWCOM LOWEST COMMON ADDRESS
    SKG EOADR 71
    BRU PR2BIG PROGRAM TOO BIG
    SUB EOADR 71
    STA EOSIZE 72
    LDA TOPMEM LAST ADR OF MEMORY
    ADD ONE
    STA RUNTOP RUN-TIME TOP MEMORY
    SUB EOADR 71
    STA COUNT COUNT
    LDA EOADR 71
    STA TEMP1 TEMP 1
    MRG =3777B RELEASE UNUSED PAGES
STUP1 ADD =4000B
    SKG LOWCOM
    BRU **2
    BRU LDSYS
    SKE LOWCOM
    BRU **2
    BRU LDSYS
    BRS 4 RELEASE PAGE, NOT USED

```

```

BRU STUP1
*LOADING COMPLETE
LDSYS BRS 17
MOVE #0,FRUB,A
$ENDLD BRS 10 TERMINATE FORK AND CALL RUNTIME

```

```

*NEED MORE PR0GS
NDPR0G LDA STNXPR START 0F NEXT PR0G
STA 1STIAD
STA BREAK BREAK
BRU RNXPLK READ NEXT PR0GRAM BL0CK

```

```

*READ TO END 0F PR0GRAM
RDENDP LDA PRTYPE
LDB #70000000B
SKM MAPTYP
BRU L111
L112 SBRM NXWORD
SKE APEND ASSEMBLED PR0G END
BRU L112
BRU NDPR0G
L111 SBRM NXWORD
SKE CPEND COMPILED PR0G END
BRU L111
BRU NDPR0G

```

```

*NEXT WORD
NXWORD HLT LNXWOR
L81 W10 INFLE
L250 BNE E0LWRD,L83
BRU L84 END 0F LIBRARY, MISSING TYPE0UT
L83 BNE E0FWRD,L85
LDA APEND
SKN PRTYPE

```

```

L220 LDA CPEND
      BNE CURWRD,L85
      SBRM 0SUBF
      BRU L81
L85 STA CURWRD
     BRR LNXWOR

```

## \*MISSING TYPEOUT

```

L84 BRS 17
L200 LDX TOPMEM
      MOVE LINKX,TMP2,A
L205 LDA 0,2
      SKA REQBIT
      BRU L210
L207 EAX =3,2
      LDA TMP2
      ADD THREE
      STA TMP2
      BNE ZERO,L205
      BRU L220
L210 STX TMP1
      LDA =2,2
      STA NAME1
      LDA =1,2
      STA NAME2
      PRQ MISSQT MISSING QT
      PRQ NAME1
      LDX TMP1
      BRU L207

```

## \*PROGRAM T00 BIG

```

PR2BIG PRQ 0FL0QT
ERRPT LDB =70000000B
      LDA =20000000B
      SKM NAME1
      BRU *+2
      BRS 10
      LDA NAME1

```

SKM	ALLD0L	
BRU	**2	
BRS	10	
PRQ	ERRQT	ERROR POINT
LDA	=NAME1	
LDB	=-1	
LDX	=1	
BRS	34	TYPE SPR0G NAME
BRS	10	

\*ILLEGAL FILE

\$ILLTAP	HLT	LILLTA	
\$ILLTP1	PRQ	ILLFQT	ILLEGAL FILE
	BRU	ERRPT	
ILLF1	ZR0	LILLTA	
	PRQ	ILQT1	
	BRU	ILLTP1	
ILLF2	ZR0	LILLTA	
	PRQ	ILQT2	
	BRU	ILLTP1	

\*MAIN PROGRAM MISSING

\$MANMIS	PRQ	ILQT3
	BRS	10

\*OPEN SUBPROGRAM BINARY FILE

\$0SUBF	HLT	L0SUBF	
	BRU	Z1	
ERR2	PRQ	FNBQT	FILE NOT BIN
Z1	BRS	17	CLOSE PREVIOUS FILE
	PRQ	L5UBQT	
Z2	LDX	=-1	CLEAR THE TTY INPUT BUFFER

```

BRS      14
BRS      11
PRQ      FROMQT
CLEAR
LDX      #10000B
BRS      15
BRU      Z1
BRS      16
BRU      Z1
XAB
BNE      #2,ERR2
STB      INFLE
MOVE     #*1,FRUB,A
$E0SUBF SBRR 0SUBF

```

\*

\*

```

*READ ASSEMBLED TEXT
$ASSM  MOVE  LINKX,H0LDLX,B
        MOVE  CESYMP,CESYMS,B
        MOVE  ZERO,MIFPRR,B
        MIN   1STIAD
        LDA   1STIAD
        STA   LOC
        MOVE  ONE,RELWCT,A
        LDA   PRYPE
        STA*  1STIAD
        BRU   L125
ARPBIN LDA   RELWRD
        SKR   RELWCT
        BRU   L125
        SBRM  NXWORD
        MOVE  #7,RELWCT,B
L125   ABC
        LSH   3
        STB   RELWRD
        CAX
        BRU   #+1,2
*RELOCATION BITS TABLE
BRU    ABSBIN

```

```

RELOCATION WORD REGISTERS COUNT
FIRST WORD OF REL REGISTERS

```

0

BRU	L180	2	1 EXT SYM ADDR (MOD 2**14)
BRU	R14BIN	3	3 RELOCATE (MOD 2**24)
BRU	R24BIN	4	4 ESCAPE
BRU	CTRWD	5	5 EXT SYM ADDR (MOD 2**24) HANDLED AS 14 BIT
BRU	L180	6	6 EXT SYM REF
BRU	*	7	7 LITERAL REFERENCE
BRU	R14BIN		

  

*CONTROL WORD			
CTRWD	SBRM	NXWORD	
	ABC		
	LSH	3	
	CAX		
	BRU	**+1,2	
*CONTROL WORD	TABLE		
BRU	L135	0	BINARY FOLLOWS
BRU	POP	1	POP
BRU	L130		
BRU	SKIP	3	OP DEF
BRU	L160	4	EXT SYMBOL DEF
BRU	SKIP	5	IDENT
BRU	L170	6	EXT SYMBOL REF
BRU	SKIP	7	SYMBOL TABLE

  

L130	CLA		
	LSH	6	
	SKG	=2	
	BRU	**+2	
	CLA		
	CAX		
	BRU	**+1,2	
	BRU	L132	END THIS PR0G
	BRU	ARPBIN	
	BRU	ARPBIN	
SKIP	SBRM	NXWORD	
	SKE	=1	
	BRU	SKIP	
SKIP1	STA	RELWCT	SET TO GET NEW SET OF 3BIT REGISTERS
	BRU	ARPBIN	
R24BIN	SBRM	NXWORD	24 BIT RELOCATABLE
	ADD	1STIAD	1ST INSTRUCTION ADDR



```

ABS BIN BRU L133
L133 SBRM NXWORD
LDX LOC
STA 0,2
L134 MIN LOC
LDA TOPMEM
ADD LINKX
SKG LOC
BRU PR2BIG
BRU ARPBIN
R14 BIN SBRM NXWORD 14 BIT (ADDR FLD) RELOCATABLE
STA TMP1
ADD 1STIAD
ETR ADRMSK
XMA TMP1
ETR ALBADR
ADD TMP1
BRU L133
POP LSH 6 PROGRAMMED OPERATOR DEF
CAX POP LINK ADDR
LDA BRUMOP
MRG LOC
STA 0,2 NOTE: USER POP LINKS WHICH CONFLICT WITH
BRU ARPBIN RUNTIME POP'S WILL BE OVERWRITTEN BY RUNTIME
L132 SKN MIFPRR
BRU L131
MOVE LOC,STNXPR,A END LOADING THIS SPRBG
LDA MAPTYP
MRG LOC
STA* PRSTRT
* CLEAR CORE BETWEEN END OF PRG AND LINKS
LDA TOPMEM
ADD LINKX
SUB STNXPR
STA TEMP1
MOVE STNXPR,TEMP2,A
CLA
L129 STA* TEMP2
MIN TEMP2

```

SKR TEMP1  
 BRU L129  
 BRU CKALPG

CHECK ALL PROGRAMS IN

L131 MOVE HOLDLX, LINKX, X THIS PRG UN-NEEDED, DISCARD  
 BRU RDENDP

L135 LSH 21 BIN FOLLOWS, SET LOC COUNTER  
 ETR =37777B  
 ADM LOC  
 BRU ARPBIN

\*

\*EXTERNAL SYMBOL DEFINITION

L160 SBRM REPACK  
 L161 SBRM SRCHRP  
 STA LNKPTR  
 SBRM NXWORD  
 ADD 1STIAD  
 MRG BRUMOP  
 XMA\* LNKPTR  
 LDB ONES  
 SKM REQBIT  
 BRU \*\*2  
 STB MIFPRR  
 LDA NAME2  
 BNE SYSSUB, L162  
 LDA NAME1  
 SKE 240SYS  
 - BRU L162  
 STB MIFPRR  
 MOVE 1STIAD, 240B, A  
 L162 SBRM NXWORD  
 BEQ ONES, SKIP1  
 SBRM REPCK1  
 BRU L161

MINUS IF PROGRAM REQUIRED  
 1ST INSTRUCTION ADDR

\*EXTERNAL SYMBOL REFERENCE

\*ENTER IN CURRENT EXT. SYMB. TABLE

L170	SBRM	REPACK	
L171	SBRM	SRCHRP	SEARCH LINKS RETURN POINTER
	STA	LNKPTR	
	LDA*	LNKPTR	
	SKA	ZERØ	
	BRU	L172	
	LDA	REQBIT	
	STA*	LNKPTR	
L172	LDA	NAME1	
	STA*	CESYMP	
	MIN	CESYMP	
	LDA	NAME2	
	STA*	CESYMP	
	MIN	CESYMP	
	SBRM	NXWORD	
	CLA		
	STA*	CESYMP	
	MIN	CESYMP	
	SBRM	NXWORD	
	BEQ	ONES,SKIP1	
L173	SBRM	REPCK1	
	BRU	L171	

\*14 BIT EXT SYMB REF

L180	SBRM	NXWORD	
	CAB		
	ETR	ALBADR	ALL BUT ADDRESS
	STA*	LØC	
	CBA		
	ETR	ADRMSK	
	MUL	THREE	
	LSH	23	
	ADD	CESYMS	START OF C.E. SYM. TABLE
	ADD	TWØ	3RD WORD OF 3 WORD ENTRY
L183	CAX		
	LDA	Q,2	
	ETR	ADRMSK	
	SKE	ZERØ	

BRU L183  
LDA L0C  
XMA 0,2  
ETR ALBADR  
ADM 0,2  
BRU L134

\*REPACK NAME, 6-BIT CHARS TO 8-BIT CHARS

REPACK ZR0 LREPAC  
SBRM NXWORD  
REPK1 ABC  
LSH 6  
RCH BX+CB  
LSH 2  
CXB  
LSH 6  
RCH BX+CB  
LSH 2  
CXB  
LSH 6  
STA NAME1  
CBX  
SBRM NXWORD  
RCH CA+AX+XB  
LSH 6  
CLB  
LSH 2  
CXB  
LSH 6  
RCH BX+CB  
LSH 2  
CXB  
LSH 6  
STA NAME2  
BRR LREPAC

REPCK1 ZR0 LREPAC  
BRU BRU1

READ 2ND WORD ONLY

2L6 PAGE 33

\*  
\*INPUT CHAR AND TEST  
\$ICTP0P TCI CHAR  
EAX\* 0  
CXA  
ETR =377B  
SKE CHAR  
BRR 0  
MIN 0  
BRR 0

\*  
\*PRINT CHARACTER  
\$PRCLNK BRU PRCP0P  
\$PRCP0P EAX\* 0  
CXA  
ETR =377B  
PRC1 CIO =1  
SKE =155B  
BRR 0  
LDA =152B  
BRU PRC1

\*PRINT QUOTE  
\$PRQLNK BRU PRQP0P  
\$PRQP0P EAX\* 0  
CXA  
ETR ADRMSK  
LDX =1  
LDB =-1  
BRS 34  
BRR 0

\*  
\*  
\* Q U O T E S

```

$ILLQT ASC      '$I>>/'
$MEMQT ASC      '$M>>/'
$LMPQT ASC      '$LOAD MAIN PROGRAM/'
$LSUBQT ASC     '$LOAD SUBPROGRAMS/'
$FROMQT ASC     '$FROM FILE /'
$FNBQT ASC      '$FILE NOT BINARY/'
$OFL0QT ASC     '$PROGRAM TOO BIG/'
$ILLFQT ASC     '$ILLEGAL FILE/'
$ILQT1 ASC      '$EXTERNAL SYMBOLS NOT ALLOWED/'
$ILQT2 ASC      '$SPECIAL RELOCATION NOT ALLOWED/'
$ILQT3 ASC      '$LOAD MAIN PROGRAM FIRST/'
$ERRQT ASC      '$CURRENTLY LOADING /'
$MSGTRM ASC     '/ ' MESSAGE TERMINATOR
$MISSQT ASC     '$MISSING /'
$ALLD0L ASC     '$$$$' ALL DOLLARS
NEWSQT ASC      '$VERSION 6.0 $/' NEWS MSG
*
*
C O N S T A N T S
E0FWRD DATA    27657537B END OF FILE WORD
E0LWRD DATA    33266555B END OF LIBRARY WORD
FTBLE1 10B     TBLE,1 FORK TABLE ADDR
FTBLE ZR0      TBLE
$TOPMEM ZR0    27776B TOP OF RUNTIME MEMORY
MSEVEN DATA   =7
$ONES DEC      =1
$ZER0 DATA    0
$ONE DEC       1
$TWO DEC       2
$THREE DATA   3
MSIX DEC       =6 MINUS SIX
MTWELV DEC     =12 MINUS TWELVE
$SIGN ZR0      0,4
SIGNBT EQU     SIGN
EIGHT DATA    8
BRUM0P BRU     0 BRU M0P
SYSSUB ASC     'SYS'
240SYS ASC     '240'
$UNDLAD DATA  230B UNDEFINED LABEL ADDR
MABSLS DATA   =30B MINUS ABS LINK SIZE

```

STXM0P	STX	0	
EAXTAG	EAX	0,2	
8BIT0P	0CT	37700000	8 BIT 0P MASK
0PMASK	0CT	17700000	0P MASK
MPRTYP	0CT	04000000	MAIN PR0G TYPE
N0PMSK	0CT	60077777	N0T 0P MASK
LBL0P	0CT	00300000	LBL L0P
ABS0P	0CT	01100000	ABS L0P
DELL0P	0CT	00400000	DEL L0P
SYS0P	0CT	00500000	SYS L0P
STLM0P	DATA	15600000B	
#SBRMIND	SBRM*	0	SBRM M0P
MAPTY	DATA	40000000B	MACH PR0G TYPE
BLKL0P	0CT	00600000	BLK L0P
	0CT	14600000	
	0CT	10600000	
	0CT	12400000	
	0CT	13000000	
	0CT	13400000	
0PTAB1	CNA		0P TABLE 1
	STA	0	
	LDA	0	
	ADD	0	
	SUB	0	
0PTAB2	EQU	*	0P TABLE 2
REQBIT	0CT	10000000	REQUIRED BIT
ADRMSK	DATA	37777B	
FXSKEY	EQU	*	FX SCALAR KEY
ARFXCK	EQU	*	ARRAY=FX C0NST KEY
XLC0KY	EQU	*	FX C0NST=FL C0NST KEY
XSLCKY	EQU	*	FL C0NST=FX SCALAR KEY
XSLBKY	EQU	*	FX SCALAR=LBL KEY
LNKARK	EQU	*	LINKAGE=ARRAY KEY
TMPLKY	EQU	*	TEMP=LINKAGE KEY
DUMTKY	DEC	1820	DUMMY=TEMP KEY
LBLKEY	DEC	3640	LBL KEY
DUMKEL	DEC	14560	DUMMY KEY
ALBADR	0CT	77740000	ALL BUT ADR MASK
6BIT0P	0CT	07700000	6 BIT 0P MASK

TAGBIT EQU \*  
 NLSTBL HLT 0,2  
 INDBIT HLT\* 0  
 \*BUFFER START  
 TAGPBP BCT 30000000  
 NINDBT BCT 77737777  
 CODEST ZR0 1200B  
 \$APEND DATA 31062144B  
 \$CPEND DATA 32465152B  
 BRS 43  
 XAB  
 ETR =77770000B  
 MRG =F0RL  
 XAB  
 BRS 44  
 BRU PS  
 RELAB BRU PS+1  
 END

TAG BIT  
 NOT LAST BLOCK BIT  
 IND BIT  
 TAG PBP BITS  
 NOT IND BIT  
 LOAD POINT FOR COMPILED CODE  
 ASSEMBLED PROG END  
 COMPILED PROG END



\*  
 \*\*\*\*\* ORIGIN AT 400 \*\*\*\*\*  
 \*

\*\*\*\*\* FIXED STORAGE LOCATIONS \*\*\*\*\*

\* 1-77=TSS STORAGE, 100-177=FORTRAN POPS, 200-277=SYSTEM SUBROUTINES

\* 400-450 SYSTEM CONSTANTS  
 \*

\*R U N T I M E            T R A N S F E R            V E C T O R            LBC 400

\$EOADR ZR0            0                            ABS. BASE OF ERASABLE STORAGE

\$EOTAG ZR0            0

\$EOIND ZR0            0

\$EADR1 ZR0            0

\$EOSIZE ZR0           0

\$FLTIND ZR0           0

\$SENSLW ZR0           0

SENSE LIGHT WORD

\$SENSSW ZR0           0

SENSE SWITCH WORD

\$TA            ZR0            0

(410)

\$TB            ZR0            0

\$TX            ZR0            0

\$ERRCON ZR0           0

\$ZER0 BSS            1

CONSTANTS == INITIALIZED AT RUN-TIME

RTSH23 EXT           \*

RIGHT SHIFT 23

\$0NE EQU            \*

FIXED POINT ONE

\$FP0NE DATA        1

FLOATING POINT 1

DATA            20000000B

\$SIGN DATA        40000000B

\$0NES DATA        =1

(420)

\*L I N K S

\$LACB ZR0            0

\$LCLFLE ZR0          0

\$LCMPG0 ZR0          0

\$LD130 ZR0           0

\$LDECLF ZR0          0

\$LECH0 ZR0           0

\$LERR0R ZR0          0

\$LERMSG ZR0	0
\$LFDL00 ZR0	0
\$LFIX ZR0	0
\$LFLD00 ZR0	0
\$LFL0AT ZR0	0
\$LFXBND ZR0	0
\$LIFSNL ZR0	0
\$LIFSNW ZR0	0
\$LIFVFL ZR0	0
\$LI0LSA ZR0	0
\$LLOCST ZR0	0
\$LSENST ZR0	0
\$LST0P ZR0	0
\$LPOWER ZR0	0
\$LPAUSE ZR0	0
\$LI0SUB ZR0	0
\$LNDDMY ZR0	0
\$L030 ZR0	0
\$LADS00 ZR0	0
\$LARCO0 ZR0	0
\$LARD00 ZR0	0
\$LBFDO0 ZR0	0
\$LCDB00 ZR0	0
\$LDNS00 ZR0	0
\$LFDT00 ZR0	0
\$LFMT00 ZR0	0
\$LICD00 ZR0	0
\$LICF00 ZR0	0
\$LICU00 ZR0	0
\$LMPRO0 ZR0	0
\$LICW00 ZR0	0
\$LIMT00 ZR0	0
\$LRBI00 ZR0	0
\$LTIN ZR0	0
\$LTTI00 ZR0	0
\$LINFO0 ZR0	0
\$LWCS00 ZR0	0
\$LFCS00 ZR0	0
\$LW0T00 ZR0	0

```

$LSETT ZR0 0
$LSTRTD ZR0 0
$LRIT00 ZR0 0
$LRNF00 ZR0 0
$LFTS0R ZR0 0
$LDBLXR ZR0 0
$RETURN BSS 1
$LIB1 BSS 1
$LIB2 BSS 2
$LIB3 BSS 1
$LIB4 BSS 2
$LIB5 BSS 2
$LIB6 BSS 1
$LIB7 BSS 1

```

```

*
*

```

```

*F L A G S
$FARF ZR0 0
$FERR ZR0 0
$FE0R ZR0 0
$FE0F ZR0 0
$FI0DIR ZR0 0
$FNAME ZR0 0
$FRUB ZR0 0
$FSSW ZR0 0
$FDBG ZR0 0
$FRUN ZR0 0

```

```

-1=A,R F0RMAT, 0=LETTER 0 F0RMAT
-1=ERROR PAUSE, 0=PAUSE
-1=E0R REACHED 0N INPUT
-1=E0F READ
I/O DIRECTION -1=0UTPUT

-1=N0 2ND RUB0UT
SENSE SW SET/RESET
-1=DEBUG M0DE, 0=CMMD M0DE 0R PR0G RUN

```

```

$TRUNQ BSS 1
$PTMP BSS 1
BSS 1
$QTEMP BSS 1
$QTEMP1 BSS 1

```

```

$QB00T BSS 1
$TEMP1 BSS 1

```

AGTTP1	EXT	TEMP1	AGT
FIXTMP	EXT	TEMP1	FIX
\$TEMP2	BSS	1	
FLAXRG	EXT	TEMP2	FLBAT
FXBAXR	EXT	TEMP2	FIX
TMPIZ	EXT	TEMP2	FSCAN
\$TEMP3	BSS	1	
FIXTP2	EXT	TEMP3	FIX
TMPX	EXT	TEMP3	FSCAN
\$TEMP4	BSS	1	
FLAG	EXT	TEMP4	FLA FLS
DPTMP1	EXT	TEMP4	LDP STD DPA DPS
TMPA	EXT	TEMP4	FSCAN
\$XM	BSS	1	DPM
\$ZE	BSS	1	FLA-S-M-D
\$ZM	BSS	1	FLA-S-M-D DPM
\$PART1	BSS	1	FTA-S-M-D
CHSIZ	EXT	PART1	FSCAN 3 BIT OCTAL OR 8 BIT CHAR MASK
\$PART2	BSS	1	FTA-S-M-D
SHFCNT	EXT	PART2	FSCAN SHIFT COUNT
\$TEMPA	BSS	1	FLA-S-M-D
\$XREG	BSS	1	FLA-S-M-D-N DPM F-XSD/FA/NA AG/ALF-X
\$EADR	BSS	1	
\$EADR2	BSS	1	F-XSD STRTDM ENDDMY
\$OVFIND	BSS	1	OVERFLOW IND
\$DØXTEM	BSS	1	DØX
DØFTEM	EXT	DØXTEM	DØF
\$DØFTAD	BSS	1	DØF
\$DØFXIT	BSS	1	DØF
\$EOSIZT	BSS	1	F-XFA/NA
\$EXIT	BSS	1	LTX FTA-S-M-D
\$DBUF	BSS	1	FSCAN DEC. TO BINARY CONV. BUFFER
\$DBUF1	BSS	1	
\$TXI	BSS	1	FSCAN X INSIDE FØRMAT SCAN
\$TXØ	BSS	1	FSCAN X ØUTSIDE FØRMAT SCAN
MSAVEX	EXT	TXØ	I/O HANDLERS
\$CHCNT	BSS	1	FSCAN CHAR. COUNT
\$BE	BSS	1	FSCAN BINARY EXPONENT
\$CC	BSS	1	FSCAN ØUTPUT CHAR. COUNT

\$D	BSS	1	FSCAN	DECIMAL PLACES
\$DE	BSS	1	FSCAN	DECIMAL EXPONENT
\$DW	BSS	1	FSCAN	SPECIFIED DATA FIELD WIDTH
\$FW	BSS	1	FSCAN	FIELD WIDTH
\$P	BSS	1	FSCAN	P SCALE FACTOR
\$ARGSGN	BSS	1	FSCAN	ARGUMENT SIGN
\$ARGADR	BSS	1	FSCAN	ARGUMENT ADDRESS
\$MRTNAD	BSS	1	FSCAN	MEMORY RETURN ADDRESS
\$FCPNTR	BSS	1	FSCAN	FORMAT CHAR. POINTER
\$FWPNTR	BSS	1	FSCAN	FORMAT WORD POINTER
\$WCPNTR	BSS	1	FSCAN	I/OBUF CHAR. POINTER
\$WWPNTR	BSS	1	FSCAN	I/OBUF WORD POINTER
\$DS	BSS	1	FSCAN	DATA SPECS. IN FORMAT FLAG
\$EP	BSS	1	FSCAN	EDIT IN PROGBSSS FLAG
\$IF	BSS	1	FSCAN	I-TYPE DATA SPEC. FLAG
\$IZ	BSS	1	FSCAN	IMPLICIT ZERO IN FORMAT FLAG
\$OT	BSS	1	FSCAN	OUTPUT TRANSACTION FLAG
\$ACBBIT	BSS	1		
\$ACBCHR	BSS	1		
\$ARRAY	ZR0	0		
\$OVFLID	ZR0	0		
\$FIXLNK	ZR0	0		
\$FLTLNK	ZR0	0		
\$ELEMENTS	ZR0	0		
\$MPRX	ZR0	0		
\$CHAR	ZR0	0		
\$CCHR	ZR0	0		CURRENT CHARACTER
ERLSZE	EXT	ERLTBL=ERLLIM-1		ERROR LIST SIZE
\$ERLPT	ZR0	0		ERROR LIST PTR
\$ERLTBL	BSS	10		ERROR LIST TABLE
ERLLIM	EXT	*-1		
ERLLIM	EQU	*-1		

\*I/O FILE NUMBERS

\*ENTRY: (1 WORD): FILE NO. AND I/O DIRECTION BIT IN SIGN POSITION

\$FLNMT BSS 10

IOTEND EQU \*

IOTEND EXT	*		
IOTSIZE EXT	FLNMT-IOTEND+2	I/O TABLE SIZE=2	
\$FLNPTR ZR0	0	FILE NAME PTR	
\$FNPTR BSS	2	PTRS TO STRING DURING LOOK-UP FOR OPEN	
\$FNPTR2 BSS	2		
\$FNBUF BSS	2		
\$FNBUF3 BSS	1		
\$FNCTR BSS	1		
\$CURFILE ZR0	0	CURRENT FILE NUMBER	
\$LDPNT EQU	1200B	LOAD POINT, SEE CODEST IN LDR.	
LDPNT3 EXT	LDPNT+3		
0BPST EXT	LDPNT+10B	OBJECT PR0G START	
\$PR0GRS ZR0	0		
\$TBLE BSS	7	FORK TABLE	
TBLE1 EXT	TBLE+1		
TBLE3 EXT	TBLE+3		
TBLE6 EXT	TBLE+6		
\$MSCNT ZR0	0	MULTIPLE SPACECNTUNT	
\$FC BSS	1		
\$EXPSGN BSS	1		
\$LSTCHR BSS	1		
\$LCLAFI BSS	1		
\$XSDGTS BSS	1		
\$NC BSS	1		
\$G2 ZR0	0		
\$DC BSS	1	FSCAN DIGITS COUNT	
\$FT BSS	1	FSCAN FORMAT TYPE	
\$NXTCHR BSS	1	FSCAN NEXT CHARACTER	
\$PC BSS	1	FSCAN INTEGER DIGIT COUNT	
\$RC BSS	1	FSCAN REPEAT COUNT	
\$STKADR BSS	1	FSCAN LEFT PARENTHESIS STACK ADRBSSS	
\$WR BSS	1	FSCAN WIDTH RETIRED	
\$ADRLPL ZR0	0	FSCAN LPSTK ADR TO TEST TABLE FULL	
VADRLL EXT	LPSTK+3*NSTLVLL=6		
\$ADRLPS ZR0	0	FSCAN LPSTK ADRBSSS FOR INITIALIZE	
VADRLS EXT	LPSTK=3		
NSTLVLL EQU	4	FSCAN MAX. NESTING LEVEL IN FORMATS	

```

$LPSTK BSS      (R)NSTLVL
LPSTK3 EXT     LPSTK+3
LPSTK4 EXT     LPSTK+4

```

FSCAN LEFT PARENTHESES STACK TABLE

1R6

PAGE 7

\*DEBUGGER STORAGE

```

BKOADR EXT     *-1
$BK1ADR ZR0    0          BREAK POINT 1 ADDRESS
$BK2ADR ZR0    0
$BK3ADR ZR0    0
BK4ADR EXT     *
$FFLBAT ZR0    0
$FNEXT ZR0     0
$DADRIN ZR0    0          ADDRESS INCREMENT
$DBN   ZR0     0          BREAK PT NUMBER (1,2,3)
$DD0TA ZR0     0          ABS. DOT (STA LABEL NO)
$DD0TR ZR0     0          REL. DOT (NO. OF STATEMENTS FROM DD0TA)
$DNMCHR ZR0    0          NAME CHARACTER
$DNUMB ZR0     0
$DNAME1 ZR0    0
$DNAME2 ZR0    0
$LOCTMP ZR0    0
$MRSTL ZR0     0          MOST RECENT STATEMENT LABEL
$DTMP1 ZR0     0
$DTMP2 ZR0     0
$DVBASE ZR0    0          VARIABLE BASE ADDRESS
$NSUBSC ZR0    0          NUMBER OF SUBSCRIPTS
$SCMAX ZR0     0          SUBSCRIPT MAX
$UPR0GS ZR0    0          USER'S PROGRAM START
$UDATAS ZR0    0          USER'S DATA START
$UPR0GN ZR0    0          USER'S PROGRAM END

```

\*G0T0 FILE RETURN CODE STORAGE

```

$SG0T0 ZR0     0          SWITCH TO PREVENT NEW WRITE
$KG0T0 ZR0     0          CONSTANTS STORAGE
$KG0T01 ZR0    0
$KG0T02 ZR0    0
$WG0T0 BSS     7          CODE STORAGE

```

\$EWG EQU \*

END STOR.

1R6

PAGE 8

\$BRKRP ZR0 0

\$BRKRA BSS 2

\$BRKRX ZR0

BREAK AND ESCAPE RETURN LOC.

END



\*  
\* ORIGIN THIS PACKAGE AT 30000B (SAME AS 2L)  
\*

\$F0RL EQU 5455B F0S SMT RELABELING BYTES (RUNTIME)

\*RCH 0P DEFINITIONS

GCD 0PD 53700000B,1,1  
FCL 0PD 4600450B,2 FULL CYCLE LEFT  
FCR 0PD 4600224B,2 FULL CYCLE RIGHT  
BXC 0PD 4600022B,2 B TO X, CLB  
XBC 0PD 24600040B,2 X TO B, CLX  
CA EQU 1 CLEAR A  
CB EQU 2  
\* CX EQU 20000000B ARPAS DEMANDS ,2  
AB EQU 4 A TO B  
BA EQU 10B  
BX EQU 20B  
XB EQU 40B  
XA EQU 200B  
AX EQU 400B  
MOVE MACRO D  
MOVE1 NARG  
IF 'D(1\$1,2)'='=0'  
CL,D(MOVE1)  
ELSF 1  
LD,D(MOVE1) D(1)  
ENDF  
MOVE2 EQU 2  
RPT MOVE1=2  
ST,D(MOVE1) D(MOVE2)  
MOVE2 EQU MOVE2+1  
ENDR  
ENDM  
\*  
STZ MACRO A STORE ZERO; CLA ; STA A(1) ; ENDM  
BEQ MACRO D; SKE D(1); BRU \*\*2; BRU D(2); ENDM  
BNE MACRO D; SKE D(1); BRU D(2); ENDM  
SETT MACRO D; MOVE =-1,D(1),D(2); ENDM

SETF MACRO D; MOVE =0,D(1),D(2); ENDM

2R6

PAGE 2

```
*
*
PRC 0PD 10200000B,1,1
PRQ 0PD 10300000B,1,1
FLA 0PD 013100000B,1,1
FLS 0PD 013500000B,1,1
FLM 0PD 014100000B,1,1
FLD 0PD 014500000B,1,1
LFP 0PD 012500000B,1,1
STD 0PD 010700000B,1,1
AXC 0PD 04600401B,2
AXCE 0PD 04600501B,2
AXBA 0PD 04600450B,2
BXC 0PD 04600022B,2
BXAC 0PD 04600222B,2
LGR 0PD 06624000B,1,1,1
NSTLBL EQU 4 MAX. NESTING LEVEL IN FORMATS
PS BRU SYSINI *** ORIGIN 30000B ***
$RS MOVE =1,FRUB,A
BRU PS3
* S Y S T E M I N I T I A L I Z E
SYSINI MOVE =0,FRUN,SENSLW,SENSSW,OVFIND,BRKRP,A
MOVE =1,FRUB,0NES,BK1ADR,BK2ADR,BK3ADR,A
```

\*SYSTEM SUBROUTINE LINKS

```
LDX =LINKTL
LDA LINKTE,2
STA 234B,2
BRX **2
SBRM SLDPNT SET LOAD POINT ADDRESSES
MOVE =1,ERLPT,A
```

\*SET UP POP ADDRESSES

```
LDX **57B
LDA POPEND,2
STA 157B,2
BRX **2
```

## \*USER FILES INITIALLY CLOSED

```

BRS      17
LDX      =FLNMT
CLA
STA      0,2
LDA      =40000001B
STA      1,2
LDX      =IOTSZE
LDA      =37777777B
STA      IOTEND,2
BRX      *=1

```

## \*SET UP FORMAT PROCESSING STACK

```

MOVE     =VADRLL,ADRLPL,A
MOVE     =VADRLS,ADRLPS,A

```

## \*INITIALIZE CONSTANTS

```

MOVE     =0,ZERO,A
LDP      IFPONE
STP      FPONE
MOVE     =40000000B,SIGN,A
MOVE     FLIND,FLTIND,A
MOVE     =ERROR,413B,A

```

```

PRQ      LDCOMP   LOADING COMPLETE
LDB      =10
LDX      =1
LDA      EOSIZE   ERASABLE STORAGE SIZE
BRS      36
PRQ      SPCEQT   SPACE AVAIL QUOTE
PS3      MOVE     =CS,TBLE,A
PS2      LDA      FTBLE

```

```

CAX
BRS 9
NOP TBLE
$WAIT BRS 31
MOVE TBLE,PROGRS,A PROGRAM RESTART
LDX TBLE6 FORK STATUS
BRU *+1,2
BRU RUB
BRU BADOP
PRQ MEMQT
BRU PRL0C
BADOP PRQ ILLQT
PRL0C LDA TBLE
ETR =37777B
LDB =8
LDX =1
BRS 36
BRS 10 RETURN TO EXEC
$RUB SKN FRUB
BRU RUB3
MOVE =0,FRUB,A
SKN FRUN
BRU PS3
LDP TBLE1
STP BRKRA
LDX TBLE3
STX BRKRX
LDA 0
STA BRKRP SAVE ESCAPE RETURN ADR.
SBRM L0CLBL
MOVE =0,FRUN,A
BRU PS3 GO TO CMMD START
RUB3 BRS 10
$BAD PRC 37B Q=MARK
*CMMD INPUT START
$CS LDX ==1
CLA
BRS 12 SET BREAK ON EVERYTHING TBL.
CLA

```

*LDX 37777B  
BRU 37777B*

```

          STA      MSCNT
          SETF    FRUN,A
          PRQ     CMMDQT
CS1       CI0     ZER0
CS5       CLX
          MOVE   =-1,FRUB,B
          SBRM   ACB
          DATA  ' '+CS1
          DATA  ' '+CQUOTE
          DATA  6BIT***1
          STA    CCHR
          BRU    DEBUG

```

\*SET UP DEBUGGER'S PROGRAM ENVIRONMENT

```

$SLDPNT ZR0     LACB
          MOVE   =LDPNT,UPR0GS,A
          LDA*   UPR0GS
          ETR    =37777B
          SUB    =1
          STA    UPR0GN
          MOVE   LDPNT3,UDATAS,A
          LDA    =LDPNT
          SBRM   L0CLBL
          BRR    LACB

```

```

*QUOTE MODE
CQUOTE EQU      *
CG2     CI0     ZER0
          BEQ   =144B,CS
          BRU   CG2

```

\*

\*

\*

\*

E R R O R   N O T E   T Y P E O U T

\*

```

$ERR0R ZR0     LERR0R
          STA    Q2

```

\*1

```

      MOVE      0,MRSTL,A
      LDA       LERROR
      ADD       #2
      STA       LERMSG      LINK TO ERROR MESSAGE
      LDA       NEGBEX      NEG EXP BIT
      SKA*      LERROR
      LDA       NOPMOP      LIV NOP MOP
      XMA       Q2          *1
ERR1_  LDX       #ERLSZE
      PRQ       ERRQT1
      PRQ*      LERMSG
      LDA       MRSTL
      SBRM      LCLBL      LOCATE LABEL
      LDA       DDOTA
ERR3   BNE      #1,ERR6
      BRU       ERR8

ERR6   PRQ      INSTQT      IN STATEMENT QT
      LDA       DDOTA
ERR7   LDB      #10
      LDX       #1
      BRS      36          TYPE STATEMENT NO
      LDA       DDOTR
      SKG      #0
      BRU      ERR8
      TC0      #1      +1
      LDB      #10
      LDX      #1
      BRS      36
ERR8   MOVE     MRSTL,0,A
      BRR      LERROR

```

```

*      S U B R O U T I N E S      A N D      P O P S
*CLOSE A USER'S FILE
*LEAVES FORTRAN UNIT NUMBER IN X
$CLFLE ZR0      LCLFLE
      SKG      #9

```

SKG #1  
 BRR LCLFLE  
 CAX  
 LDA FLNMT,2  
 ETR #37777B  
 SKE #37777B  
 BRS 20  
 LDA #37777777B  
 STA FLNMT,2  
 BRR LCLFLE

UNACCEPTABLE FORTRAN UNIT NO

2R6

PAGE 7

\*CLOSE ALL USER'S FILES

\$CLAFLE ZR0 LCLAFI  
 LDA #2  
 CLAFI5 SBRM CLFLE  
 CXA  
 ADD #1  
 SKG #9  
 BRU CLAFI5  
 BRR LCLAFI

\*STOP STATEMENT

\$STPLNK ZR0 LSTOP  
 MOVE #LDPNT,MRSTL,A  
 STPSTP PRQ STOPQT  
 STPST3 LDA MRSTL  
 SBRM L0CLBL  
 BRU CS  
 \$PAULNK ZR0 LPAUSE  
 MOVE LPAUSE,MRSTL,A  
 MIN LPAUSE  
 PRQ PSEQT  
 LDA\* LPAUSE  
 ETR #37777B  
 LDB #10  
 LDX #1  
 SKE #0

RUNTIME CMMD START

	BRS	36
	CI0	ZER0
	CLX	
	SBRM	ACB
	DATA	'G'+PAUG0
	DATA	6BIT+CS5
PAUG0	TC0	' 0'
	TC0	=155B
	BRR	MRSTL

## \*PRINT CHARACTER

\$PRCP0P EAX*		0
	CXA	
	ETR	=377B
PRCP1	CI0	=1
	BRR	0

## \*PRINT QUOTE

\$PRQP0P EAX*		0
	CXA	
	ETR	=37777B
	LDX	=1
	LDB	=1
	BRS	34
	BRR	0

\*  
\*

*ACCEPT CHAR AND BRANCH	
*CHAR OUT OF BOUNDS TREATED AS ZER0	
*I/ A=CHAR RT ADJ., X=0,1 FOR 6,9 BIT CHAR	
*B DISTURBED A PRESERVED	
\$ACB	ZR0 LACB
	SKG ACBLIM,2 LIMIT
	SKG =0
	CLA
	STA ACBCHR



```

LDB      B6,2
STB      ACBBIT
CLB
LSH*     ACBSCT,2   SHIFT COUNT
LDB      ACBMSK,2
EAX*     LACB
ACB1     EAX        1,2
        SKM        0,2
        BRU        **2
        BRU        ACBOUT
        XMA        ACBBIT
        SKA        0,2
        BRU        ACBOUT
        XMA        ACBBIT
ACBOUT   BRU        ACB1
        LDA        =37777B
        ETR        0,2
        CAX
        LDA        ACBCHR
        BRU        0,2
*
*
*
        PAGE
*
*   P R O G R A M M E D   O P E R A T O R S
*
*BKP P0P (152) BREAK POINT
$BKPP0P LDA        0
        STA        BRKRP           SAVE FOR PROCEED COMMAND
        SBRM       L0CLBL         LOCATE LABEL GIVEN ADDR
        PRQ        BEAKQT
        LDA        DD0TA
        BEQ        =-1,BKP10      START OF CURRENT PR0G
        LDB        =10
        LDX        0NE
        BRS        36
BKP5    LDA        DD0TR
        SKG        =0

```

	BRU	CS	
	TC0	=1	+1
	LDA	DD0TR	
	LDB	=10	
	LDX	0NE	
	SKE	=0	
	BRS	36	
	BRU	CS	COMMAND START
BKP10	PRQ	STRTQT	PROGRAM START QT
	BRU	BKP5	

\*FLA P0P (131) FLOATING ADD  
\$FLAP0P STA TEMPA

CLA

BRU \*\*3

\*FLS P0P (135) FLOATING SUBTRACT  
\$FLSP0P STA TEMPA

LDA 0NES

STA FLAG

STX XREG

CXA

STB ZM

B

STE

STX ZE

SIGN EXTENDED EXP(A,B)

ADD XREG

DOUBLE X

AXC

EAX\* 0

PROCURE ARGUMENT ADDRESS

SKE 1,2

BRU Q5B

CLAB 0 M=ZERO

Q5B

BRU FLAD

SKE TEMPA

BRU Q5A

CLAB 0 A,B=ZERO

Q5A

BRU FLAC

CXA

M NON ZERO

LDB 0,2

	STE		EXTEND SIGN OF EXP (M)
	XXA		
	SUB	ZE	EXP (M) = EXP (A)
	SKG	0NES	
	BRU	FLAGM	/A/ .GR. /M/
	SKA	M770	/A/ .LE. /M/
	LDA	=DD39	MAX SHIFT =39
	XMA	TEMPA	
	LDB	ZM	
	RSH*	TEMPA	ALIGN FRACTIONS
FLAC	XAB		
	SKN	FLAG	FLAG NEGATIVE IFF FLS
	BRU	Q5C	FLA
	MRG	0777	FLS
	SUB	0,2	
	E0R	0777	
	XAB		
	SUC	1,2	
Q5C	BRU	FLAF	
	RCH	0101B	CLA(E)
	ADD	0,2	
	XAB		
	ADC	1,2	
FLAF	STE		
FLA0VT	0VT		
	BRU	0FSET	RIGHT NORMALIZE
	N0D	38	LEFT NORMALIZE
	SKA	0NES	
	BRU	FLANZ	IF FRACTION ZERO,
	CLX		THEN CLEAR EXPONENT
FLANZ	XXA		
	SKG	M257D	TEST FOR EXPONENT UNDERFLOW
FLA0F	BRR	FLA0FS	SET OVERFLOW DEVI0USLY
FLA0K	XXA		
FLAX	LDE	0,2	LDE,CLX
	STX	FLAG	BSSET FLA/FLS FLAG
	LDX	XREG	
	0VT		
	BRU	0FL0	OVER/UNDERFLOW

FLAGM	BRR	0	
	CNA		FORCE EXP DIFF POSITIVE
	SKA	M778	SKIP IF LESS THAN 64
	LDA	=DD39	
	LDX	1,2	
	XXA		
	RSH	0,2	ALIGN FRACTIONS
FLAD	RCH	0102B	
	SKN	FLAG	
	BRU	Q5D	FLA
	XMA	TEMPA	FLS
	XAB		
	XMA	ZM	
	SUB	ZM	
	XAB		
	SUC	TEMPA	
	BRU	FLAF	
Q5D	XAB		FLA
	ADD	ZM	
	XAB		
	ADC	TEMPA	
	BRU	FLAF	
0FSET	RSH	1	RIGHT NORMALIZE
	E0R	MINUS	
	BRX	FLAX	
	XXA		
	SKG	DD255	CHECK FOR OVERFLOW
	BRU	FLA0K	
	BRR	FLA0F	
DIV0F	LDA	TEMPA	
	E0R	1,2	DETERMINE SIGN OF QUOTIENT
	BRU	Q5E	OVERFLOW
0FL0	STE		
	BRX	Q5E	OVERFLOW
	CLAB	0	UNDERFLOW
	LDX	XREG	
	BRR	0	
Q5E	LDX	SIGN	SET OVERFLOW INDICATOR
	STX	0VFIND	

EOR	MINUS	
RSH	39	
EOR	MINUS	
EAX	255	
BRU	FLAX	
FLA0FS 0 FLA0F,4		
*FLM POP (141)	FL0ATING MULTIPLY	
\$FLMPOP STX	XREG	
STA	TEMPA	AH
CXA		
STE		EXTEND EXP (A,B)
STX	ZE	
BXC		
LCY	1	DOUBLE X
XXA		
EAX*	0	
LGR	2	FORCE +
MUL	1,2	
STA	ZM	AL * MH
LDA	0,2	
CXB		
AXCE		
XXA		EXTEND EXP (M)
ADM	ZE	AE+ME = EXP (BSSULT)
BXAC		CXA, BXC
LGR	2	FORCE +
MUL	TEMPA	AH * ML
ADD	ZM	AH * ML + AL * MH
MUL	TW0	SCALE
STB	ZM	
XMA	TEMPA	
MUL	1,2	AH * MH
XAB		
ADD	ZM	
XAB		
ADC	TEMPA	
LDX	ZE	EXP (BSSULT)
SKA	MAXPSS	
BRU	FLCOM	ALL RIGHT TO NORMALIZE

SKB	MLONES	
BRU	FLCOM	NOT =1 OR 0
SKE	MINUS	SEPARATE =1 FROM 0
BRU	FLMX	0, EXIT
RCY	1	=1, RIGHT NORMALIZE
BRX	FLCOM	
BRU	FLCOM	
*FLD POP (145)	FLOATING DIVIDE	
\$FLDPOP STX	XREG	
STA	TEMPA	
CXA		
STE		EXTEND EXP (A,B)
STX	ZE	
BXC		
LCY	1	$(AH+AL)/(MH+ML) =$
LDB	TEMPA	$((AH+AL)/MH)*(1-ML/MH) =$
AXBA		$(Q+R/MH)*(1-ML/MH) =$
EAX*	0	$Q+(R-Q*ML)/MH$
RSH	2	
DIV	1,2	
OVF		
BRU	DIVBF	OVERFLOW IFF DIVISOR =0
STA	ZM	SINGLE PRECISION QUOTIENT=Q
BAC		
RSH	1	
STA	TEMPA	REMAINDER = R
LDB	0,2	
CXA		
STE		EXTEND EXP (M)
XXA		
CNA		$EXP(BSSULT) = EXP(A)+2+EXP(M)$
ADD	TW0	
ADM	ZE	
BAC		
RCY	2	
CNA		=ML
MUL	ZM	Q
ADD	TEMPA	$R = Q*ML$
DIV	1,2	$(R-Q*ML)/MH$

	MUL	TW0	SCALE
	ADD	ZM	$Q+(R=Q*ML)/MH$
	LDX	ZE	EXP(BSSULT)
	SKA	0NES	
	BRU	FLC0M	
	BRU	FLMX	
	*FLN P0P (147)	FL0ATING NEGATE	
	\$FLNP0P STX	XREG	
	SKB	ML0NES	
	BRU	FLNA	LESS SIG. HALF N0N F0R,
	CNA		
	SKB	ZER0	
	SKA	M01777	
	BRR	0	
	STE		
	SKB	MINUS	PLACE SIGN BIT IN B
	BRU	FLC0M	
	RCY	1	SIGN BIT T0 A
	BRX	Q5F	
FLC0M	N0D	+	N0RMALIZE RESULT
Q5F	XXA		
	SKG	DD255	CHICK F0R EXP0NENT
	SKG	M257D	0VERFL0W/UNDERFL0W
	BRU	FLA0F	
	XXA		
FLNB	LDE		
FLMX	LDX	XREG	
	BRR	0	
FLNA	STE		
	XAB		
	CNA		
	XAB		
	E0R	0NES	
	BRU	FLNB	
	*DPM P0P (155)	D0UBLE PRECISION MULTIPLY	
	\$DPMP0P STX	XREG	
	EAX*	0	
	STA	XM	
	BAC		

	RCY	2	
	MUL	1,2	
	STA	ZM	
	LDA	Q,2	
	LGR	2	
	MUL	XM	
	ADD	ZM	
	MUL	TW0	
	STB	ZM	
	XMA	XM	
	MUL	1,2	
	XAB		
	ADD	ZM	
	XAB		
	ADC	XM	
	LDX	XREG	
	SKG	MINUS	
Q28	SKB	0NES	
	BRR	0	
	BRR	Q29	*2
Q29	HLT	Q28	
	*LTX P0P (123)	LOAD THEN FIX	
	\$LTXP0P LDA	0	
	STA	EXIT	
	LDP*	EXIT	
	SBRM	FXBND A	FIX BANDA
	BRR	EXIT	
	*XMP P0P (140)	FIXED MULTIPLY	
	\$XMPP0P MUL*	0	0
	RSH	1	1
	XAB		N0
	BRR	0	0
	*XDV P0P (144)	FIXED DIVIDE	
	\$XDVP0P MUL	RTSH23	RT SHIFTER 23
	DIV*	0	0
	BRR	0	0
	* XST P0P (104)	FIX AND STORE	
	\$XSTP0P SBRM	FXBND A	FIX B AND A
	STA*	0	



```

      BRR      0
* FST POP (105) FLOAT AND STORE
$FSTPOP SBRM  FL0ATA      FLOAT A
      BRU      STDPOP      STORE DOUBLE POP
* FIX B AND A
$FXBND ZR0    LFXBND
      STX      FXBAXR      SAVE X
      SKB      EMASK       SKIP IF =1 < EXP0N > 64
      CLAB     0           OTHERWISE INTEGER = 0
      STA      FIXTP2      RECORD SIGN
      SKD      D23        COMPUTE SHIFT AND DIRECTION
      BRU      Q31         EXP0N > 22
      RCH      0102B      EXP0N < 23 SO CLEAR EXP0N
      STB      FIXTMP      SAVE FRACTION
      RSH      0,2
      XAB
      MRG      FIXTMP      MERGE FRACTION
Q32   SKB      0NES       SKIP IF FRACTION = 0
      SKN      FIXTP2      CHECK SIGN
      BRU      Q33         DO NOT ADD ONE
      ADD      0NE        ADD 1 TO TRUNCATED
Q33   LDX      FXBAXR      NEGATIVE INTEGER
      SBRR     FXBND      EXIT
Q31   RCH      0102B      CLEAR EXPONENT
      LSH      0,2
      BRU      Q32
*LTF POP (122) LOAD THEN FLOAT
$LTFPOP LDA*  0
      SBRM     FL0ATA      FLOAT A
      BRR      0
*FLOAT A
$FL0ATA ZR0   LFL0AT
      STX      FLAXRG      FLOAT A XREG
      CLB
      LDX      D23
      N0D     23           27
      LDE
      LDX      FLAXRG      FLOAT A XREG

```

```

      SBRR      FL0ATA      FL0AT A
*FTA POP (126) FL0AT THEN ADD
$FTAP0P STB    PART1
      STA      PART2
      LDA      0           0
      STA      EXIT
      LDA*     0           0
      SBRM     FL0ATA      FL0AT A
      FLA      PART1
      BRR      EXIT
* FTM POP (136) FL0AT THEN MULTIPLY
$FTMP0P STB    PART1
      STA      PART2
      LDA      0
      STA      EXIT
      LDA*     0
      SBRM     FL0ATA      FL0AT A
      FLM      PART1
      BRR      EXIT
* FTS POP (132) FL0AT THEN SUBTRACT
$FTSP0P SBRM   FTS0RD      FTS 0R FTD
      FLS      PART1
      BRR      EXIT
* FTD POP (142) FL0AT THEN DIVIDE
$FTDP0P SBRM   FTS0RD      FTS 0R FTD
      FLD      PART1
      BRR      EXIT
* FTS 0R FTD (COMMON PART)
$FTS0RD ZR0    LFTS0R
      STB      PART1
      STA      PART2
      LDA      0
      STA      EXIT
      LDA*     0
      SBRM     FL0ATA
      XMA      PART2
      XAB
      XMA      PART1
      XAB

```

```

SBRR      FTSØRD
*DØX PØP (110) DØ FIXED
$DØXPØP LDB      0          0
          STB      DØXTEM    DØX TEMP
          MIN      0          0
          ADM*     0          0
          SKA      SIGN
          BRU      Q35        *1
          LDA*     0          0
          SKG*     DØXTEM    DØX TEMP
          MIN      0          0
          BRR      0          0
Q35      LDA*     DØXTEM    DØX TEMP
          SKG*     0          0
          MIN      0          0
          BRR      0          0
*DØF PØP (111) DØ FLOATING
$DØFPØP STA      DØFTEM    DØF TEMP
          LDA      0          0
          STA      DØFTAD    DØF TEMP ADR
          ADD      ØNE       ØNE
          STA      DØFXIT    DØF EXIT
          LDA      DØFTEM    DØF TEMP
          FLA*     DØFXIT    DØF EXIT
          STD*     DØFXIT    DØF EXIT
          SKN      DØFTEM    DØF TEMP
          BRU      Q36        *1
          LFP*     DØFTAD    DØF TEMP ADR
          FLS*     DØFXIT    DØF EXIT
          SKG      ZERØ      ZERØ
          MIN      DØFXIT    DØF EXIT
          BRR      DØFXIT    DØF EXIT
Q36      FLS*     DØFTAD    DØF TEMP ADR
          SKG      ZERØ      ZERØ
          MIN      DØFXIT    DØF EXIT
          BRR      DØFXIT    DØF EXIT
*XSD PØP (100) FIXED SET UP DUMMY
$XSDPØP CLA
          BRU      Q37        *1

```

```

*FSD P0P (101) FLOATING SET UP DUMMY
$FSDP0P LDA FLIND FLOAT IND
Q37 LDB FLIND FLOAT IND
STX XREG XREG
EAX* 0 0
SKM* EADR2 EADR 2
BRU Q39 *2
Q38 MRG* EADR2 EADR 2
MIN EADR2 EADR 2
STA 0,2 0
MRG TAGBIT TAG BIT
STA 1,2 1
LDX XREG XREG
BRR 0 0
$ARGM ZR0 0
Q39 SBRM ERROR
BRU CS
ASC ! ARGUMENT MODES DISAGREE IN SUBPROGRAM CALL!
*FFA P0P (115) FLOATING FIRST ARGUMENT
$FFAP0P SBRM DBLXRG DOUBLE XREG
LDA FLIND FLOAT IND
BRU Q40 *1
*XFA P0P (114) FIXED FIRST ARGUMENT
$XFAP0P CLA
STX XREG XREG
Q40 LDB EOADR EO ADR
STB EADR1 EADR 1
LDB EOSIZE EO SIZE
STB EOSIZT EO SIZE TEMP
BRU Q41 *2
*FNA P0P (117) FLOATING NEXT ARGUMENT
$FNAP0P SBRM DBLXRG DOUBLE XREG
LDA FLIND FLOAT IND
BRU Q41 *2
*XNA P0P (116) FIXED NEXT ARGUMENT
$XNAP0P CLA
STX XREG XREG
Q41 STA* EADR1 EADR 1
EAX* 0 0

```

CXA	ADRMSK	ADR MASK
ETR	EADR1	EADR 1
ADM*	EADR1	EADR 1
MIN	EADR1	EADR 1
LDX	XREG	XREG
SKR	EOSIZT	EO SIZE TEMP
BRR	0	0
SBRM	ERROR	
BRU	Q41A	
ASC	' ERASABLE STORAGE EXCEEDED/ '	
Q41A BRR	0	
*DOUBLE XREG		
\$DBLXRG ZR0	LDBLXR	
STX	XREG	XREG
XXA		
ADD	XREG	
XXA		
SBRR	DBLXRG	DOUBLE XREG
*ALF POP (151)	ASSIGN LABEL TO FLOATING	
\$ALFP0P SBRM	DBLXRG	DOUBLE XREG
BRU	Q42	*1
*ALX POP (150)	ASSIGN LABEL TO FIXED	
\$ALXP0P STX	XREG	XREG
Q42 LDA	0	0
ETR	ADRMSK	ADR MASK
STA*	0	0
LDX	XREG	XREG
MIN	0	0
BRR	0	0
*AGF POP (113)	ASSIGNED GO TO FLOATING	
\$AGFP0P SBRM	DBLXRG	DOUBLE XREG
BRU	Q43	*3
*AGX POP (112)	ASSIGNED GO TO FIXED	
\$AGXP0P STX	XREG	XREG
Q43 LDX*	0	0
LDA	0,2	0
LDB	P0PADR	P0P ADR MASK
SKM	AP0PCD	ASN P0P CODE
BRU	Q45	*1

```

Q44  STX      AGTTP1      AGT TEMP 1
      LDX      XREG        XREG
      BRR      AGTTP1      AGT TEMP 1
Q45  SBRM     ERROR
      BRU      Q44         *2
      ASC      ' ASSIGNED GO TO * UNASSIGNED VARIABLE/'
*LDP PBP (125) LOAD DOUBLE PRECISION
$LDPPBP STX  DPTMP1
      CXA
      ADD      DPTMP1
      CAX
      EAX*     0           0
      LDA      1,2        1
      LDB      0,2        0
      LDX      DPTMP1
      BRR      0
*STD PBP (107) STORE DOUBLE PRECISION
$STDPBP STX  DPTMP1
      STA      TEMP3
      CXA
      ADD      DPTMP1
      CAX
      LDA      TEMP3
      EAX*     0           0
      STA      1,2        1
      STB      0,2        0
      LDX      DPTMP1
      BRR      0
*DPA PBP (154) DOUBLE PRECISION ADD
$DPAPBP STX  DPTMP1
      EAX*     0           0
      XAB
      ADD      0,2        0
      XAB
      ADC      1,2        1
      LDX      DPTMP1
      BRR      0
*DPS PBP (153) DOUBLE PRECISION SUBTRACT
$DPSPBP STX  DPTMP1

```

EAX*	0	0
XAB		
SUB	0,2	0
XAB		
SUC	1,2	1
LDX	DPTMP1	
BRR	0	0

\*  
\*

\*INPUT OUTPUT LIBRARY ROUTINES  
\* SINCE THESE ROUTINES ARE ALWAYS REQUIRED, THEY ARE PLACED IN  
\* COMMON MEMORY

\$DECLFL ZR0	LDECLF	
SKG	=9	
SKG	=1	
BRR	LDECLF	UNACCEPTABLE UNIT NO
ADD	=FLNMT	FILE NAME TABLE
STA	FLNPTR	FILE NAME PTR
MIN	LDECLF	

\*DETERMINE DIRECTION OF INFORMATION TRANSFER

CLB		
LDA*	LDECLF	
SKA	=1	
LDB	=-1	OUTPUT
STB	FI0DIR	I/O DIRECTION

\*LOOK UP NAME

MIN	LDECLF	
LDA	LDECLF	START OF STRING
ETR	=37777B	
MUL	=3	
LRSH	1	
CBX		
LDA*	LDECLF	END OF STRING
SUB	=1	
ETR	=37777B	
STA	LDECLF	
MUL	=3	

```

      LSH      23
      FCR
      ETR      #177777B
      ADD      #2
DECL3  STP      FNPTR
      LDA      #FNBUF
      MUL      #3
      LSH      23
      SUB      #1
      CAB
DECL4  STP      FNPTR2
      GCI      FNPTR          ADVANCE TO FILE LEADING DELIM
      BRU      DECERR
      BEQ      #1 /1,DECL9
      BNE      #7,DECL4
DECL8  GCI      FNPTR
      BRU      DECERR
      BEQ      #1 /1,DECL5
      BEQ      #7,DECL5
DECL9  WCI      FNPTR2
      BRU      DECL8
DECL5  WCI      FNPTR2
      LDP      FNPTR2
      LDX      FNBUF3
      BRS      60
      BRU      DECERR
      LDB      #14000B
      LDX      #3
      SKN      FI0DIR
      BRU      DECL40
      BRS      19          OUTPUT FILE
      BRU      DECERR
      MRG      #40000000B
DECL10 STA*    FLNPTR
DECL50  BRR      LDECLF
DECL40  BRS      16          INPUT FILE
      BRU      DECERR
      BRU      DECL10

```



```

DECERR SBRM  ERROR
        BRU  DECL50
        ASC  ' FILE NAME STRING UNRECOGNIZED/'

*READ FILE
$READF ZR0  LI0SUB
        CLB
        SBRM  SETI0T          SET UP I/O TABLE
        LDA  LI0SUB
        BRU  IFS00          INITIALIZE FORMAT SCAN

*
* WRITE FILE
$WRITE ZR0  LI0SUB
        LDB  SIGN
        SBRM  SETI0T          SET UP I/O TABLES
        LDA  LI0SUB
        LDB  SIGN          OUTPUT FLAG
        BRU  IFS00          INITIALIZE FORMAT SCAN

*
*
* ENDI0L HLT          END INPUT-OUTPUT LIST
        SKN  EP          SKIP IF NO EDIT IN PROGRESS
        BRU  END01
        SBRM  ERROR
        BRU  END03
        ASC  ' I-O LIST/'
END03  SBRR  ENDI0L
END01  LDA  0NES
        STA  EP          RESET EDIT IN PROGRESS FLAG
        LDX  MSAVEX
        SBRR  ENDI0L

*
*
* SET I/O TABLES
SETI0T ZR0  LSETT
        STX  MSAVEX
        SKG  #9

```

```

SKG      =-1
BRU      ILLUNØ          ILLEGAL UNIT NØ
CAX
STB      ØT
*TEST IF THIS FILE IS ØPENED
SETT1 LDA  FLNMT,2
      SKE  =37777777B
      BRU  SETIT2
      LDA  =1              SELECT TTY BY DEFAULT
      SKN  ØT
      CLA
      STA  0,2
      STA  CURFLE
      BRU  SETIT7
SETIT2 SKN  ØT
      BRU  SETIT5
      SKN  FLNMT,2
      BRU  SETT10
SETIT3 LDA  FLNMT,2          FILE NAME TABLE
      ETR  =37777B
      STA  CURFLE          CURRENT FILE NØ
SETIT7 SKE  ZERØ
      BRU  SETIT9
      TCØ  =147B          SIGNAL TERMINAL INPUT
      TCØ  =147B
SETIT9 LDX  MSAVEX
      BRR  LSETT
ILLUNØ SBRM  ERRØR          ILLEGAL UNIT NUMBER
      BRU  CS              RUNTIME CMMD START
      ASC  ' INPUT=OUTPUT UNIT EXPRESSION NØT 0=9/'
SETIT5 SKN  FLNMT,2
      BRU  SETIT3
SETT10 SBRM  ERRØR          DIRECTION ERRØR
      BRU  CS              CMMD START
      ASC  ' INPUT TØ OUTPUT FILE ØR VICE VERSA/'

```

\*  
\*

```

*Q U B T E S
CMMDQT ASC '$+/'
$CONTQT ASC 'CONTINUE/'
$STRTQT ASC '*START*/'
$BEAKQT ASC '$BREAK /'
INSTQT ASC '$IN STATEMENT /'

```

```

*
ST0FLK EQU 200B START OF LINKS
$ACBLIM DATA 77B,177B
$ACBSCT DATA 18,15
$ACBMSK DATA 77400000B,77740000B
$FTBLE 10B TBLE,1
6BIT EQU 00400000B
9BIT EQU 00040000B
B6 DATA 00400000B
B9 DATA 00040000B

```

```

LINKTS EQU *
ZR0 0
ZR0 STRTDM START OF DUMMIES
ZR0 ENDDMY END OF DUMMIES
ZR0 STPLNK ST0P
ZR0 WRITF (4) WRITE FILE
ZR0 ENDI0L END I0 LIST
ZR0 READF READ FILE
ZR0 ARGM ARGUMENT MODE ERROR
ZR0 SETI0T SET INPUT/OUTPUT TABLES
ZR0 PAULNK
ZR0 DECLFL (12)
ZR0 0
ZR0 CLFLE
ZR0 CLAFLE
DATA 0,0
ZR0 COMPG0
ZR0 0
ZR0 IFSNSW (22)
ZR0 IFSNLT IF SENSE LIGHT
ZR0 IF0VFL

```

DATA	0,0	
ZR0	SENSLT	(27) SENSE LIGHT
BRU	UNDEF	UNDEFINED LABEL TRANSFERS
ZR0	FXBND	
ZR0	FL0ATA	
ZR0	I0LUSA	
LINKTE EQU	*	
LINKTL EQU	LINKTS=LINKTE	
CARRET EQU	155B	
END		

```

*
*
*
* RCH OP DEFINITIONS
FCL  OPD  4600450B,2  FULL CYCLE LEFT
FCR  OPD  4600224B,2  FULL CYCLE RIGHT
BXC  OPD  4600022B,2  B TO X, CLB
XBC  OPD  24600040B,2  X TO B, CLX
CA   EQU  1          CLEAR A
CB   EQU  2
*   CX   EQU  20000000B  ARPAS DEMANDS ,2
AB   EQU  4          A TO B
BA   EQU  10B
BX   EQU  20B
XB   EQU  40B
XA   EQU  200B
AX   EQU  400B
MOVE MACRO  D
MOVE1 NARG
      IF      'D(1$1,2)'='0'
      CL.D(MOVE1)
      ELSF    1
      LD.D(MOVE1) D(1)
      ENDF
MOVE2 EQU  2
      RPT     MOVE1-2
      ST.D(MOVE1) D(MOVE2)
MOVE2 EQU  MOVE2+1
      ENDR
      ENDM
*
STZ  MACRO  A  STORE ZERO; CLA ; STA A(1) ; ENDM
BFS  MACRO  D; SKN D(1); BRU D(2); ENDM
BTR  MACRO  D; SKN D(1); BRU ++2; BRU D(2); ENDM
BEQ  MACRO  D; SKE D(1); BRU ++2; BRU D(2); ENDM
BNE  MACRO  D; SKE D(1); BRU D(2); ENDM
SETT MACRO  D; MOVE =-1,D(1),D(2); ENDM
SETF MACRO  D; MOVE =0,D(1),D(2); ENDM

```



```

      BRR      MRTNAD
*FIO P0P (121) FLOATING INPUT/OUTPUT
FIO P0P CXB
      LSH      1          DOUBLE INDEX REGISTER
      XXB          AND SAVE X IN B
      LDA      SIGN      FLOATING MODE INDICATOR
      BRU      **2
*XIO P0P (120) FIXED INPUT/OUTPUT
XIO P0P RCH      41B      SAVE X IN B, INTEGER MODE IN A
      STB      TX0
      STA      ARGADR
      EAX*     0          GET ARGUMENT ADDRESS
      CXA
      ETR      ADRMSK
      ADM      ARGADR
      LDA      0
      STA      MRTNAD      SAVE EXIT
      LDX      TXI
      SKN      EP          SKIP IF NO EDIT IN PROGBSSS
      SBRR     FDL00
      SBRM     ERROR
      BRU      XIO P0P7
      ASC      'FXIO/'
XIO P0P7 BRR     MRTNAD   RETURN TO CALL TO INITFS

```

```

*
$IFS00 ADD      ONE      FROM READP AND WRITE
      STA      MRTNAD
      STB      0T        SET UP OUTPUT FLAG
      STX      TX0      SAVE INDEX
      LDX      MRTNAD
      LDB      0PMASK
      LDA      0,2
      SKM      ZERO      TEST POINTER IN I/O CALLING SEQ
      BRU      IFS06
IFS05  EAX*     0,2      NORMAL POINTER TO LABELED FORMAT OR
      BRU      IFS14     ARRAY CONTAINING RUN TIME FORMAT
IFS06  SKM      BRUM0P

```

BRUM0P	BRU	IFS08	RUN TIME F0RMAT = SUBSCRIBED ARRAY
	STA	MRTNAD	EVALUATE SUBSCRIPT
	SBRM	1,2	GET MODIFIED ADDRESS
	LDX*	E0ADR	
	BRU	IFS10	
IFS08	LDB	P0PADR	
	SKM	AGTP0P	
	BRU	IFS31	
	EAX*	0,2	RUN TIME F0RMAT = SCALAR VARIABLE
IFS10	LDB	P0PADR	
	LDA*	0,2	
	SKM	AP0PCD	TEST IF F0RMAT LABEL ASSIGNED
	BRU	IFS32	
	LDX	0,2	
IFS13	LDX	1,2	
IFS14	STX	TEMP1	
	LDA	TEMP1	
	ETR	ADRMSK	
	STA	TEMP1	
	LDX	TEMP1	
	LDB	0PMASK	
	LDA	0,2	
	SKM	BRUM0P	TEST F0RMAT STATEMENT
	BRU	IFS16	
	BRU	IFS20	
IFS16	SKM	STLM0P	STATEMENT LABEL M0P
	BRU	IFS24	
IFS17	BRX	**1	
	LDA	0,2	
	SKM	BRUM0P	
	BRU	IFS30	
IFS20	STX	FWPNTR	
IFS21	BRU	I0E00	
IFS24	SKM	BKPM0P	BREAK P0INT M0P
	BRU	IFS22	
	BRU	IFS17	
IFS22	LDA	TEMP1	
	SUB	0NE	
	STA	FWPNTR	



```

IFS30 BRU I0E00
SBRM ERROR
BRU IFS330
ASC ' FORMAT STATEMENT BEGAN WITH TRACE POP'
ASC ' BUT WAS NOT FOLLOWED BY A BRU/'
IFS330 STX TEMP1
LDA TEMP1
ETR ADRMSK
STA FWPNTR
BRU IFS21
IFS31 SBRM ERROR
BRU IFS05
ASC ' FORMAT STATEMENT POINTER IN I-0 LIST IS UNACCEPTABLE/'
IFS32 LDB 0,2
SBRM ERROR
BRU IFS332
ASC ' FORMAT STATEMENT LABEL HAS NOT BEEN SET UP IN'
ASC ' #A SCALAR VARIABLE BY AN ASSIGN STATEMENT/'
IFS332 STB TEMP1
LDX TEMP1
BRU IFS13
AGTP0P DATA 11200000B ASSIGNED GO TO FIXED POP

```

```

I0E00 CLAB
STA FE0R CLR E0R FLAG
STA P SET P-SCALE = 0
STA LPSTK4 PBSSET FOR 0 LEVEL
LDA ADRLPS 0N LEFT PAREN STACK TABLE
STA STKADR
LDA 0NES
STA FCPNTR PBSSET FORMAT CHARACTER POINTER
CLA; STA MSCNT
SBRM DNS00 GET FIRST SPEC. FROM FORMAT
SKN IZ
BRU I0E25
SBRM ERROR SYNTAX ERROR IF NUMBER
BRU I0E25

```

```

I0E25  ASC      ' NUMBER PRECEEDS 1ST LEFT-PARENTHESIS/'
      SKE      LB
      BRU      **2          TEST IF CHARACTER = LEFT BRACKET
      BRU      LP00        OR LEFT PARENTHESIS
      SKE      LP          GO PUT ON LP STACK
      BRU      **2
      BRU      LP00
      SBRM     ERROR
      BRU      I0E30
I0E30  ASC      ' NO LEFT ( IN FORMAT/'
      SBRM     DNS00        FETCH NEXT SPEC
I0E31  LDX      DM20
      SKE      CHRTBL+20,2  IDENTIFY CHARACTER
      BRX      **1
      BRU      JMPTBL+20,2  GO TO APPROPRIATE PROCESSOR
      *                               FORMAT PROCESSOR BRANCH VECTOR
JMPTBL BRU      ISS02      E
      BRU      ISS02      F
      BRU      ISS01      I
      BRU      ISS01      A
      BRU      ISS01      LETTER 0
      BRU      ISS01      R
      BRU      X00        X
      BRU      I0E30      , (IGNORE)
      BRU      DL00      $
      BRU      RP00      ]
      BRU      LP00      [
      BRU      E0R00     /
      BRU      H00       H
      BRU      P00       P
      BRU      MS00      =
      BRU      I0E30      + (IGNORE)
      BRU      RP00      )
      BRU      LP00      (
      BRU      I0E30      CR (IGNORE)
      BRU      I0E30      LF (IGNORE)
*
I0E38  SBRM     ERROR
      BRU      I0E30

```

```

*      ASC      ! ILLEGAL FORMAT CHARACTER/!
*                      A,R=FORMAT PROCESSOR
R00    EQU      *
A00    LDA      =1      !
      MOVE     =-1,FARF,B      A,R FORMAT
      LDB      =3
      SBRM     ADS00          ALPHA DATA SETUP FOR INPUT/OUTPUT
*                      E=FORMAT PROCESSOR
*
E00    SBRM     ARD00          GO GET NEXT DATUM
      LDA      D
      ADD     D6
      SKN     ARGSGN
      SUB     ONE            = 1 IF ARGUMENT POSITIVE
      STA     WR            SET REQUIRED WIDTH = D+6
      CLAB
      SKE     DBUF1         TEST IF ARGUMENT = 0
      BRU     **2
      BRU     E10          BYPASS SETUP FOR P=SCALE FACTOR
      LDA     P
      SKN     P
      ADM     WR            ADJUST WIDTH FOR P=SCALE FACTOR
      ADD     D
      SBRM     ARCOO        ADD ROUND-OFF CONSTANT
      LDA     DE
      SUB     P
      STA     DE            ADJUST DECIMAL EXPONENT FOR P=SCALE
E10    LDA     P
      STA     PC            SET UP INTEGER DIGIT COUNT
      SKN     PC
      LDA     ZERO
      ADD     D
      STA     DC            AND FRACTION DIGIT COUNT
      LDA     WR
      SKN     IZ            TEST FOR WIDTHLESS FORMAT
      LDA     FW
      SUB     FW
      SBRM     BFD00        CONVERT TO FORMATTED DECIMAL

```

	LDA	DE	PROCESS DECIMAL EXPONENT
	LDB	E	
	SKG	DM100	IF EXPONENT $\leq$ -99
	LDB	MS	PUT MINUS SIGN IN PLACE OF E
	XAB		
	SBRM	ICW00	PUT E IN WBUF
	LDA	DE	
	SKA	SIGN	
	CNA		MAKE NEGATIVE EXPONENT POSITIVE
	SKG	D99	TEST IF 3 DIGITS REQUIRED
	BRU	E24	-100 $<$ EXPONENT $<$ 100
E20	MUL	ONE	EXPONENT $>$ 99
	DIV	D100	
E22	SKE	ZER0	
	ADD	=1 0'	
E23	STB	DE	
	SBRM	ICW00	PUT HUNDREDS DIGIT IN WBUF
	LDA	DE	
	MUL	ONE	
	DIV	D10	
	STB	DE	
	SKE	ZER0	
	ADD	=1 0'	
	SBRM	ICW00	PUT TENS DIGIT IN WBUF
	LDA	DE	
	ADD	=1 0'	
	SBRM	ICW00	PUT UNITS DIGIT IN WBUF
	BRU	ADS15	
E24	XAB		
	LDA	MS	
	SKN	DE	IF EXPONENT IS POSITIVE
	LDA	BLNK	PUT BLANK IN PLACE OF MINUS SIGN
	BRU	E23	
*			F-FORMAT PROCESSOR
*			
F00	SBRM	AR000	G0 GET NEXT DATUM
	CLA		
	SKE	DBUF1	F0R NON-ZER0 ARGUMENTS
	LDA	P	ADJUST DECIMAL EXPONENT F0R P-SCALE

	ADD	DE	
	STA	DE	
	ADD	D	
	SBRM	ARCOO	ADD ROUND=OFF
	LDA	D	
	SKG	ZERO	
	BRU	I03	EXIT THRU I-PROCESSOR IF D = 0
	CNA		
	SKG	DE	
	LDA	DE	
	STA	PC	SET UP INTEGER DIGIT COUNT
	LDA	D	
	SKN	DE	
	BRU	**2	
	ADD	DE	
	STA	DC	AND FRACTION DIGIT COUNT
	LDA	D	
	ADD	D2	SET REQUIRED WIDTH = D+2
	SKN	DE	
	ADD	DE	+ DE IF DE > 0
F10	SKN	ARGSGN	
	SUB	ONE	= 1 IF ARGUMENT POSITIVE
	SKN	IZ	TEST IF WIDTHLESS FORMAT
	LDA	FW	
	SUB	FW	
	SBRM	BFD00	GO CONVERT TO FORMATTED DECIMAL
	BRU	ADS15	
	*		H-FORMAT PROCESSOR
	*		
H00	SKN	IZ	TEST FOR IMPLICIT COUNT
	LDB	ONE	MAKE IT ONE
	STB	RC	
H03	SKR	RC	TEST CHARACTER COUNT
	BRU	**2	NOT DONE
	BRU	H10	
	SKN	OT	TEST OPERATING MODE
	BRU	H06	INPUT MODE
	SBRM	FCS00	GET CHARACTER FROM FORMAT
	SBRM	ICU00	PUT CHARACTER IN WBUF UNCOND.

H06	BRU	H03	
	SBRM	WCS00	GET CHARACTER FROM WBUF
	SBRM	ICF00,4	MOVE POINTER AND PUT CHAR IN FORMAT
	BRU	H03	
H10	SKN	BT	
	BRU	I0E30	
H11	LDA	=377B	FLUSH MULTIPLE SPACES
	SBRM	ICW00	
	BRU	I0E30	
*			I=FORMAT PROCESSOR
*			
I00	SBRM	AR000	G0 GET NEXT DATUM
	LDA	DE	
	SBRM	ARC00	ADD ROUND-OFF
I03	CLAB		
	STA	DC	SET FRACTION DIGIT COUNT = 0
	LDA	DE	
	SKG	ZER0	TEST IF ABS(ARGUMENT) < 1
	BRU	I08	IT IS
I05	STA	PC	SET UP INTEGER DIGIT COUNT
	ADD	D2	
	SKN	IF	SKIP IF NON-INTEGERS MODE
	SUB	0NE	DEC. PT. NOT REQUIRED FOR INTEGER
	BRU	F10	EXIT THRU F-PROCESSOR
I08	LDA	0NE	
	STB	DBUF1	MAKE ARGUMENT = 0
	BRU	I05	
*			0=FORMAT PROCESSOR
*			
000	CLA		
	STA	FARF	NOT A,R FORMAT
	LDB	D8	
	SBRM	ADS00	ALPHA DATA SETUP FOR INPUT/OUTPUT
	SKN	IZ	TEST IF WIDTHLESS FORMAT
	BRU	015	IT IS. SETUP TO SCAN TO END OF WBUF
	LDA	FW	

```

005 STA CHCNT
SBRM 030 GET NEXT INPUT CHARACTER
BRU *+1 IGNORE IF LEADING BLANK OR SPACE
010 LDX D3
SUB =1 0' NUMBER BASE
SBRM ICD00 INSERT CHARACTER INTO DBUF
SBRM 030 GET NEXT INPUT CHARACTER
SKN IZ TEST IF WIDTHLESS FORMAT
BRU 032 IT IS. TERMINATE ON TRAILING BLANK
CLAB 0 MAKE BLANK OR SPACE =0
BRU 010
015 LDA WBLIM EXTEND INPUT FIELD
LSH 2
BRU 005
*
030 HLT L030
SKR CHCNT TEST IF FIELD EXHAUSTED
BRU 035 NOT YET
032 LDA DBUF1
LDB DBUF
SKN ARGADR TEST IF FLOATING POINT DATUM
XAB INTEGER DATUM
BRU ADS31 GO STORE DATA IN ARGUMENT ADDRESS
035 SBRM WCS00 GET CHARACTER FROM WBUF
SKG =27B TEST IF OCTAL
SKG =17B
BRU *+2
BRU 010 IT IS
SKE SPACES TEST IF SPACE
BRU *+2
SBRR 030 IT IS
SKE CMA TEST IF COMMA
BRU *+2
BRU 032 IT IS. GO STORE DBUF
SKE =155B
BRU 037
LDA =1
STA FE0R
BRR L030

```

037 SBRM ERROR NON-OCTAL CHARACTER ERROR  
 BRU 010  
 ASC ! NON-OCTAL CHARACTER IN OCTAL FIELD!  
 \* P- AND MINUS-FORMAT PROCESSOR  
 \*

P00 STB P SET P-SCALE FACTOR  
 BRU I0E30

\* MS00 SBRM DNS00 FETCH NEXT SPECIFICATION  
 XAB  
 CNA NEGATE COUNT  
 XAB  
 SKE PP TEST FOR P SPEC.  
 BRU \*\*2  
 BRU P00  
 SKE X TEST FOR X SPEC.  
 BRU I0E38 ERROR IF NOT P OR X  
 SKN IZ TEST IF WIDTHLESS FORMAT  
 LDB 0NES IT IS. SET COUNT = -1  
 BRU X02  
 \* R-FORMAT PROCESSOR  
 \*

\*R00 IDENTICAL TO A-FORMAT, SEE A00  
 \*  
 \*  
 \*  
 \*

X-FORMAT PROCESSOR

X00 SKN IZ TEST IF IMPLICIT COUNT  
 LDB 0NE MAKE IT 1  
 X02 XAB  
 CNA  
 SBRM MPROO MOVE POINTER RIGHT SPECIFIED AMOUNT  
 BRU H11 BUFFER OVERFLOW  
 BRU H11

\* LEFT-PARENTHESIS PROCESSOR  
 \*

LPO0 LDA STKADR



	SKG	ADRLPL	TEST IF LEFT PAREN STACK TABLE FULL
	BRU	LP02	
	SBRM	ERR0R	STACK FULL
	BRU	I0E30	
LP02	ASC	' EXCESSIVE F0RMAT NESTING/'	
	ADD	D3	MOVE STACK ADDBSSS TO NEXT SLOT
	STA	STKADR	
	LDX	0NES	
	SKG	ADRLP1	TEST IF LEVEL 0 OR 1
	STX	DS	IT IS: INDICATE NO DATA SPEC. FOUND
	LDX	STKADR	
	SKN	I2	TEST FOR IMPLICIT COUNT
	LDB	0NE	MAKE IT 1
	BAC		
	SUB	0NE	
	STA	0,2	SAVE GROUP REPEAT COUNT
	STA	2,2	
	LDA	FCPNTR	PACK CURRENT F0RMAT P0INTERS
	LSH	15	
	MRG	FWPNTR	
	STA	1,2	AND SAVE
	BRU	I0E30	RETURN TO MAIN SCAN
*			RIGHT-PARENTHESIS PROCESSOR
*			
RP00	SKR*	STKADR	TEST REPEAT COUNT
	BRU	RP03	NOT DONE: PUSH BACK
	LDA	STKADR	
	SKG	ADRLP0	TEST IF OUTERMOST RIGHT PAREN
	BRU	RP06	IT IS
	SUB	D3	IT IS NOT: SET STKADR BACK A SLOT
	STA	STKADR	
	BRU	I0E30	AND RETURN TO MAIN SCAN
RP03	LDX	STKADR	
	LDA	1,2	
	MUL	M0400	UNPACK F0RMAT P0INTERS
	STA	FCPNTR	
	BAC		
	RCY	9	
	STA	FWPNTR	BSSSET TO MATCHING LEFT-PARENTHESIS

```

RP06 BRU I0E30
      SBRM FDL00          OUTERMOST; TRY FOR MORE DATA
      LDA 0NES          THERE IS MORE; TERMINATE THIS RECORD
      ADM MRTNAD
      SKN DS            TEST IF MORE DATA SPECS. IN FORMAT
      BRU RP08
      SBRM ERROR
      BRU RP05
      ASC ' EFIA NEEDED IN FORMAT/'
RP05 SBRM FDL00          FLUSH REMAINING DATA FROM I/O LIST
      BRU **1
RP08 SKN LPSTK4         TEST IF RETURN IS TO 0 OR 1 LEVEL
      BRU RP10          0 LEVEL
      LDA ADRLP1        1 LEVEL
      STA STKADR        SET STACK ADDRESS AT 1 LEVEL
RP10 LDX STKADR
      LDA 2,2
      STA 0,2           BSSTORE GROUP REPEAT COUNT
      LDA 1,2           UNPACK FORMAT POINTERS
      MUL M0400
      STA FCPNTR
      BAC
      RCY 9
      STA FWPNTR        AND BSSTORE
      BRU E0R01        EXIT THRU /=PROCESSOR
*
*
DL00 SKN IZ
      BRU DL02
      SBRM ERROR        ERROR IF NUMBER PRECEEDS *
      BRU DL02
      ASC ' REPEAT COUNT PRECEEDS DOLLAR SIGN IN FORMAT/'
DL02 SBRM FCS00        GET NEXT CHARACTER FROM FORMAT
      SKE DL            TEST IF $
      BRU **2          IT IS NOT
      BRU DL10
      SKN 0T           TEST WHICH MODE
      BRU DL05        INPUT
      SBRM ICU00      PUT CHARACTER IN WBUF UNCOND.

```

DL05	BRU SBRM SKE BRU CLA SBRM BRU	DL02 WCS00 DL **2 ICF00 DL02	GET CHARACTER FROM WBUF REPLACE WITH BLANK IF IT IS A \$  PUT CHARACTER IN FORMAT
DL10	SKN BRU LDA SBRM BRU	BT I0E30 #377B ICW00 I0E30	FLUSH MULTIPLE SPACES
*			/-PROCESSOR (END OF RECORD)
E0R00	SKN	IZ	TEST FOR IMPLICIT COUNT
E0R01	LDB STB	0NE RC	MAKE IT 1
E0R03	SKR BRU BRU CLB SKN STB SKN BRU LDA SBRM BRU	RC **2 I0E30 BT MSCNT BT E0R07 #155B ICU00 E0R03	TEST COUNT NOT DONE
E0R07	SKN BRU BRU	FE0R E0R10 E0R03	TEST OPERATING MODE
E0R10	SBRM SKE BRU BRU	WCS00 #155B E0R10 E0R03	

## ALPHA DATA SETUP SUBROUTINE

HLT	LADS00	
STA	DBUF	(A)=SPACES FOR A FORMAT, =0 FOR 0,R
STB	CHCNT	(B)=3 FOR A,R FORMATS, =8 FOR 0
SBRM	FDL00	GO GET NEXT DATUM ADDRESS
LDA	CHCNT	
SKN	ARGADR	SKIP IF FLOATING POINT ARGUMENT
BRU	ADS02	INTEGER ARGUMENT
ADD	CHCNT	DOUBLE CHARACTER COUNT
STA	CHCNT	
SKN	IZ	TEST IF WIDTHLESS FORMAT
STA	FW	IT IS. SET FIELD WIDTH = DATA WIDTH
SUB	FW	
LDB	DBUF	
SKN	0T	SKIP IF OUTPUT MODE
BRU	ADS20	GO SET UP FOR INPUT
SKG	ZER0	SKIP IF CHCNT > FW
BRU	ADS04	
SKN	FARF	SKIP IF A FORMAT
BRU	**2	
BRU	ADS05	A FORMAT
STA	DW	
SKN	DW	SKIP IF FW > DATA WIDTH
BRU	ADS06	
SBRM	MPR00	MOVE POINTER TO 1ST OUTPUT CHARACTER
BRU	ISS10	BUFFER OVERFLOW
BRU	ADS06	
LDA	FW	SET DW TO OUTPUT
MRG	SIGN	FIRST FW CHARACTERS ONLY
STA	DW	
LDX	ARGADR	
LDB	0,2	PUT 1ST WORD IN DBUF
SKN	ARGADR	SKIP IF FLOATING POINT ARGUMENT
BRU	ADS08	INTEGER ARGUMENT
LDA	1,2	PUT 2ND WORD IN DBUF
SKN	FARF	SKIP IF A OR R FORMATS
XAB		INVERT WORD ORDER FOR 0 FORMAT

ADS08	STA	DBUF	
	STB	DBUF1	
	SKN	FARF	SKIP IF A OR R FORMATS
	BRU	ADS10	
	LDX	D8	SET UP FOR 8 BIT CHARACTERS
	LDA	#377B	
	BRU	ADS12	
ADS10	LDX	D3	SET UP FOR 3 BIT CHARACTERS
	LDA	D7	
ADS12	STA	CHSIZ	
ADS13	SKR	CHCNT	TEST IF LAST CHARACTER PROCESSED
	BRU	ADS17	NOT YET
	SKN	FARF	SKIP IF A OR R FORMAT
	BRU	ADS15	
	BRU	ISS10	GO TEST REPEAT COUNT
ADS15	LDA	ONES	
	SKN	IZ	TEST IF WIDTHLESS FORMAT
	SBRM	MPROO	IT IS. INSERT BLANK
	BRU	ISS10	
	BRU	ISS10	GO TEST REPEAT COUNT
ADS17	LDP	DBUF	GET DATUM FROM DBUF
	LCY	0,2	SHIFT IN CHARACTER
	STP	DBUF	
	ETR	CHSIZ	EXTRACT AND
	SKN	FARF	
	ADD	#' 0'	NUMBER BASE
	SBRM	ICW00	PUT IN WBUF
	BRU	ADS13	
ADS20	STB	DBUF1	SET UP DBUF TO RECEIVE INPUT
	SKN	FARF	
	SBRR	ADS00	RETURN TO 0 PROCESSOR FOR 0 INPUT
	SKA	SIGN	SKIP IF DATA WIDTH > FIELD WIDTH
	BRU	ADS22	
	LDX	FW	SET UP TO INPUT ONLY FW CHARACTERS
	STX	CHCNT	
*	SKB	ONES	SKIP IF R FORMAT
	BRU	ADS25	SET SHIFT COUNT = CHCNT = FW
	BRU	ADS24	SET SHIFT COUNT = 0 FOR R FORMAT
ADS22	SBRM	MPROO	MOVE POINTER TO 1ST INPUT CHARACTER

	NOP	0	DISREGARD BUFFER OVERFLOW
ADS24	CLAB		
ADS25	STA	SHFCNT	SET UP LEFT SHIFT COUNT
ADS26	SKR	CHCNT	TEST IF LAST CHARACTER PROCESSED
	BRU	ADS36	NOT YET
	LDP	DBUF	
ADS30	SKR	SHFCNT	SKIP IF TERMINAL SHIFTING DONE
	BRU	ADS34	NOT YET
ADS31	LDX	ARGADR	STORE DATA BY MODE
	SKN	ARGADR	SKIP IF FLOATING POINT ARGUMENT
	BRU	ADS33	INTEGER ARGUMENT
	STP	0,2	
	BRU	ISS10	GO TEST REPEAT CO
ADS33	STB	0,2	
	BRU	ISS10	
ADS34	LCY	8	LEFT JUSTIFY CHARACTERS IN A,B
	BRU	ADS30	
ADS36	SBRM	WCS00	GET NEXT INPUT CHARACTER
	SKE	=155B	
	BRU	**2	
	BRU	ADS40	
	LDX	D8	
	SBRM	ICD00	INSERT IN DBUF
	BRU	ADS26	
ADS40	LDA	=-1	
	STA	FE0R	
	BRU	ADS36	PROCESS CARRIAGE RET. AS SPACE
*			ADD ROUNDOFF CONSTANT SUBROUTINE
*			28-41 CYCLES (920,930)
ARC00	ZR0	LARC00	
	SKG	D13	TEST IF *1 < EXP < 14
	SKG	0NES	
	SBRR	ARC00	
	MUL	0NE	DOUBLE
	CBX		AND PUT IN X
	LDA	DBUF	
	ADD	RNDTBL,2	ADD ROUNDOFF

	STA	DBUF	
	LDA	DBUF1	
	ADC	RNDTBL+1,2	
	STA	DBUF1	
	ØVT		TEST FOR ROUND ØVERFLOW
	BRU	**2	GO CORRECT FOR IT
	SBRR	ARC00	
	MIN	DE	ADJUST DECIMAL EXPONENT
	LDA	ARC15	
	STA	DBUF	SET SUM TO 0.1 + EPSILON
	LDA	ARC15+1	
	STA	DBUF1	
	SBRR	ARC00	
	*		ACQUIRE AND REDUCE DATA SUBROUTINE
	*		
ARD00	ZRØ	LARD00	
	SBRM	FØL00	GO FETCH DATUM ADDRESS
	SKN	ØT	TEST ØPERATING MØDE (Ø = ØUTPUT)
	BRU	D2B00	DECIMAL TO 2 WORD BIN
	SKN	ARGADR	TEST MØDE ØF ARGUMENT
	BRU	ARD08	INTEGER
	LØX	ARGADR	FLOATING
	LDA	1,2	PICK UP DATA
	LDB	0,2	
	STE		EXTRACT AND SAVE EXPONENT
ARD05	STA	DBUF1	AND PUT COMPONENT PARTS AWAY
	STB	DBUF	
	STX	BE	
	STA	ARGSGN	SAVE SIGN ØF ARGUMENT
	SKA	SIGN	
	SBRM	CDB00	MAKE POSITIVE
	SKN	IF	TEST IF I-ØORMAT REQUESTED
	SKN	ARGADR	AND FLOATING POINT ARGUMENT GIVEN
	BRU	**2	
	SBRM	TTI00	TRUNCATE TO INTEGER
	SBRM	RNFO0	REDUCE ARGUMENT TO NORMAL FORM
	SBRR	ARD00	
ARD08	CLAB		
	LDA*	ARGADR	PICK UP INTEGER DATUM

	LDX	D23	
	N0D	23	FL0AT
	BRU	ARD05	
*			BINARY-TO-FORMATTED-DECIMAL ROUTINE
*			
BFD00	ZR0	LBFD00	
	STA	DW	
	SKN	DW	TEST IF GIVEN WIDTH > REQUIRED WIDTH
	BRU	**3	
	SBRM	MPR00	IT IS: MOVE POINTER OVER
	SBRR	BFD00	EXIT IF BUFFER OVERFLOW
	SKN	ARGSGN	
	BRU	**3	
	LDA	MS	INSERT MINUS SIGN
	SBRM	ICW00	
BFD05	LDA	PC	
	SKG	ZER0	
	BRU	BFD07	
	SKR	PC	PROCESS INTEGER DIGITS:
	SBRM	IMT00	MULTIPLY DBUF BY 10
	ADD	#20B	
	SBRM	ICW00	PUT CARRY IN WBUF
	BRU	BFD05	
BFD07	SKN	IF	SKIP IF NON-INTEG ER MODE
	SBRR	BFD00	OR EXIT
	LDA	PER	
	SBRM	ICW00	INSERT DECIMAL POINT
BFD08	SKN	PC	TEST FOR LEADING FRACTION ZEROS
	BRU	BFD10	
	MIN	PC	INSERT THEM
	CLB		
	LDA	= ' 0'	
	SBRM	ICW00	
	BRU	BFD08	
BFD10	SKR	DC	
	BRU	**2	
	SBRR	BFD00	
	SBRM	IMT00	PROCESS REMAINING FRACTION DIGITS
	ADD	#20B	



```

SBRM      ICW00
BRU       BFD10

*
*          COMPLEMENT DATA BUFFER SUBROUTINE
          24-27 CYCLES (920,930)
CDB00     ZR0      LCDB00
          CLAB
          SUB      DBUF      SUBTRACT ARGUMENT FROM ZERO
          ABC
          SUC      DBUF1
          LDX      BE        PICK UP BINARY EXPONENT
          0VT      **1 IF OVERFLOW
          BRU      CDB06
          N0D      1
CDB04     STA      DBUF1
          STB      DBUF
          STX      BE
          SBRR     CDB00
CDB06     RCY      1          RIGHT SHIFT AND MAKE POSITIVE
          BRX      CDB04     ADJUST EXPONENT
          BRU      CDB04

*
*          DECIMAL-T0-2-WORD BINARY CONVERSION
D2B00     CLAB
          STA      DBUF
          STA      DBUF1     SET BINARY SUM = 0
          STA      DE        DECIMAL EXPONENT = 0
          STA      XSDGTS    EXCESS DIGIT COUNT = 0
          LDA      0NES
          STA      ARGSGN     ARGUMENT SIGN = +
          STA      EXPSGN     EXPONENT SIGN = +
          LDA      D
          STA      FC        FRACTION DIGIT COUNT = D
          LDA      DM12     POINT INDEX AT TOP
          STA      NC        OF PUNCTUATION TABLE
          CLA
          STA      LSTCHR    LAST (PREVIOUS) CHARACTER = BLANK
          LDA      WBLIM     COMPUTE MAXIMUM ALLOWABLE
*          ADD      0NE      FIELD WIDTH BASED ON THE
                          NUMBER OF REMAINING WORDS

```

	LSH	2	IN THIS RECORD
	SKN	IZ	TEST IF WIDTHLESS FORMAT
	BRU	**2	
	LDA	FW	IT IS NOT
	STA	DW	
D2B05	SKR	DW	TEST IF FIELD EXHAUSTED
	BRU	D2B35	IT IS NOT; CONTINUE GETTING DATA
D2B06	LDA	DE	
	SKG	D511	
	BRU	**2	
	LDA	D511	SET DE=511 IF DE>511
	SKN	EXPSGN	
	CNA		ATTACH SIGN TO DECIMAL EXPONENT
	ADD	XSDGTS	
	SUB	FC	
	STA	DE	DE=DE+XSDGTS*FC
	LDA	DBUF1	
	LDB	DBUF	
	LDX	D47	
	N0D	47	NORMALIZE BINARY SUM
	STX	BE	SAVE BINARY EXPONENT
	SKG	ZER0	
	BRU	D2B13	PROCEED TO STORE IF ARGUMENT * 0
	STA	DBUF1	STORE NORMALIZED SUM
	STB	DBUF	
	LDA	NC	
	SKG	DM7	BYPASS P-SCALING IF EXPONENT GIVEN
	SKN	IF	OR IF I=FORMAT CALLED FOR
	BRU	D2B10	
	LDA	DE	
	SUB	P	
	STA	DE	OTHERWISE ADJUST DE BY P-SCALE
D2B10	SKN	DE	TEST SIGN OF DECIMAL EXPONENT
	BRU	D2B12	+; REDUCE BY MULTIPLICATION
	SBRM	FDT00	-; REDUCE BY DIVISION
	MIN	DE	
	BRU	D2B10	
D2B12	SKR	DE	
	BRU	D2B21	

D2B13	SKN	ARGADR	TEST INTERNAL MODE OF ARGUMENT
	BRU	D2B22	INTEGER
	SKN	IF	FLOATING POINT
	SBRM	TTI00	MAKE FLOATING INTEGER IF I-FORMAT
	SKN	ARGSGN	TEST SIGN OF ARGUMENT
	SBRM	CDB00	ATTACH SIGN TO ARGUMENT
	LDA	M0400	ROUND TO 39 BITS
	ADD	DBUF	
	ABC		
	ADC	DBUF1	
	QVT		TEST FOR ROUND OVERFLOW
	BRU	D2B25	
D2B15	STA	DBUF1	
	LDA	BE	
	SKG	DM257	
	BRU	D2B27	EXPONENT UNDERFLOW
	SKG	M0377	
	BRU	D2B17	
	SBRM	ERR0R	LIST OVERFLOW ERROR MESSAGE
	BRU	D2B16	
	ASC	' EXPONENT OFLOW ON INPUT DATUM/ '	
D2B16	SKN	ARGSGN	
	BRU	D2B30	
	LDA	M03777	
	LDB	DM257	SET RESULT TO PLUS FULL SCALE
	BRU	D2B19	
D2B17	LDA	DBUF1	
	LDX	BE	
	LDE		PACK EXPONENT INTO FRACTION
D2B19	LDX	ARGADR	
	STA	1,2	STORE ARGUMENT
	STB	0,2	
	BRU	ISS10	GO TEST REPEAT COUNT
D2B21	SBRM	FMT00	
	BRU	D2B12	
D2B22	LDA	D23	SCALE DBUF AT BINARY 23
	SUB	BE	
	SKA	SIGN	
	BRU	D2B26	ERR0R 0R =0 (40000000B)

	SKG	D24	
	BRU	*+2	
	LDA	D24	SET SHIFT LIMIT AT 24
	CAX		
	LDA	DBUF1	
	RSH	0,2	SCALE RIGHT
D2B24	LDX	ARGADR	
	SKN	ARGSGN	TEST SIGN OF ARGUMENT
	CNA		MAKE NEGATIVE
	STA	0,2	STORE ARGUMENT
	BRU	ISS10	GO TEST REPEAT COUNT
D2B25	LDA	TAGBIT	CORRECT FOR
	MIN	BE	ROUND OVERFLOW
	BRU	D2B15	
D2B26	SKE	=-1	
	BRU	D2B31	NOT SCALED AT 24
	SKN	ARGSGN	CK ARGUMENT SIGN
	BRU	*+2	NEGATIVE
	BRU	D2B31	
	LDB	=57777777B	CK FOR MANTISSA = 20000000B
	SKB	DBUF1	
	BRU	D2B31	NOT POSSIBLE
	LDA	=40000000B	SET UP =0
	LDX	ARGADR	
	BRU	D2B24+3	
D2B27	SBRM	ERROR	LIST UNDERFLOW ERROR MESSAGE
	BRU	D2B28	
	ASC	' EXPONENT UNDERFLOW ON INPUT DATA/!	
D2B28	CLAB		SET RESULT TO ZERO
	BRU	D2B19	
D2B30	LDA	SIGN	SET RESULT TO MINUS FULL SCALE
	LDB	M0377	
	BRU	D2B19	
D2B31	SBRM	ERROR	INTEGER OVERFLOW ERROR MESSAGE
	BRU	D2B33	
	ASC	' INPUT INTEGER OVERFLOW/!	
D2B33	LDA	M0377	LOAD MAXIMUM INTEGER
	BRU	D2B24	AND GO STORE

\*

D2B35	SBRM	WCS00	GET ANOTHER CHARACTER FROM INPUT	3R6	PAGE 25
	SKG	=31B			
	SKG	=17B			
	BRU	*+2			
	BRU	D2B45	IT IS		
	LDX	NC			
	SKE	PNCTBL+12,2	IDENTIFY INPUT CHARACTER		
	BRX	**1			
	BRU	BRTBL+12,2	BRANCH ACCORDINGLY		
*					
BRTBL	BRU	D2B05	LEADING BLANK		
	BRU	D2B05	LEADING SPACE		
	BRU	D2B60	+ (ARGUMENT)		
	BRU	D2B62	- (ARGUMENT)		
	BRU	D2B65	.		
	BRU	D2B66	E		
	BRU	D2B60	+ (EXPONENT)		
	BRU	D2B63	- (EXPONENT)		
	BRU	D2B06	,		
	BRU	D2B06	TAB(CONTROL I)		
	BRU	D2B68	SPACE		
	BRU	D2B00	Q-MARK START FLD OVER		
	SKE	=155B			
	BRU	D2B46			
	LDA	=-1			
	STA	FE0R			
	CLA				
	BRU	D2B68			
D2B46	SBRM	<del>ERROR-</del>	UNIDENTIFIED CHARACTER		
	BRU	D2B00	AND TERMINATE		
BENSG	ASC	'	BAD CHARACTER IN INPUT/		
*					
D2B45	SUB	=20B			
	STA	LSTCHR	LAST CHAR		
	LDA	NC			
	SKG	DM7	IDENTIFY DIGIT-TYPE		
	BRU	D2B47			
	LDA	DE	EXPONENT DIGIT		
	MUL	D5			

	R0V		
	LSH	24	
	ADD	LSTCHR	
	0VT		
	LDA	M03777	
	STA	DE	
	BRU	D2B05	
D2B47	SKE	DM7	
	BRU	D2B50	INTEGER DIGIT
	MIN	FC	FRACTION DIGIT
D2B48	LDA	DBUF1	
	SKG	D83885	TEST FOR POSSIBLE MULTIPLY OVERFLOW
	BRU	D2B52	NO SWEAT
	MIN	XSDGTS	IGNORE BUT RECORD EXCESS DIGIT
	BRU	D2B05	
D2B50	LDA	DM8	
	STA	NC	POINT TO . IN PUNCTUATION TABLE
	BRU	D2B48	
D2B52	SBRM	IMT00	MULTIPLY BINARY SUM BY 10
	LDA	LSTCHR	
	ADD	DBUF	AND ADD CURRENT CHARACTER
	ABC		
	ADC	DBUF1	
	STA	DBUF1	
	STB	DBUF	
	BRU	D2B05	
D2B60	EAX	2,2	M0VE NC D0WN 2 PLACES
	STX	NC	
D2B61	STA	LSTCHR	SAVE LAST CHARACTER
	BRU	D2B05	
D2B62	STA	ARGSGN	SET ARGUMENT SIGN NEGATIVE
	BRU	**2	
D2B63	STA	EXPSGN	SET EXPONENT SIGN NEGATIVE
	EAX	1,2	AND M0VE NC D0WN 1 PLACE
	BRU	D2B60+1	
D2B65	CLB		
	STB	FC	SET FRACTION DIGIT COUNT = 0
	BRU	D2B63+1	
D2B66	EAX	=6	SET NC TO POINT TO EXPONENT SIGNS

```

D2B68 BRU D2B60+1
      SKN IZ TEST IF WIDTHLESS FORMAT
      BRU **4 IT IS
      CLB
      LDA =1 0'
      BRU D2B45 AND PROCESS ACCORDINGLY
      LDA LSTCHR
      SKE PER TEST IF PREVIOUS CHARACTER WAS PERIOD
      BRU **2
      BRU D2B06 IT WAS: TERMINATE
      SKG =1 9'
      SKG =1 0'
      LDA =1 0'
      BRU D2B61
  
```

```

* PUNCTUATION TABLE
* PUNCTBL DATA 0 BLANK
* DATA 0 SPACE
* DATA ' +! +
* DATA ' =! *
* DATA 16B .
* DATA 45B E
* DATA ' +! +
* DATA ' =! *
* DATA ' ,! ,
* DATA ' I!+200B TAB(CONT I)
* DATA 0 SPACE
* DATA 37B Q-MARK (FIELD DELETE CHAR)
* DATA 0
  
```

```

*
*
*
*
DNS00 ZR0 LDNS00
      CLAB
      STA IZ SET IMPLICIT ZERO FLAG (+ * SET)
      STB DBUF SET ACCUMULATING SUM = 0
  
```

DECIMAL NUMBER SCAN SUBROUTINE  
 32+40N = 32+46N CYCLES (920,930)

DNS02	SBRM	FCS00	GET NEXT CHARACTER FROM F0RMAT
	SKG	=31B	TEST IF DIGIT
	SKG	=17B	
	BRU	**2	
	BRU	DNS05	IT IS
	SKE	BLNK	TEST IF BLANK
	BRU	**2	
	BRU	DNS02	IT IS; IGNORE
	STA	NXTCHR	SAVE AS NEXT CHARACTER
	LDB	DBUF	PICK UP COUNT
	SBRR	DNS00	
DNS05	SUB	=20B	
	SKR	IZ	BSSET IMPLICIT ZERO FLAG (= = BSSET)
	N0P		
	XMA	DBUF	BINARIZE
	MUL	D5	INCOMING
	BAC		DIGIT
	ADM	DBUF	
	BRU	DNS02	

F0RMAT CHARACTER SCAN SUBROUTINE

*	FCS00	ZR0	LFCS00
		LDX	FCPNTR
		BRX	**3
		MIN	FWPNTR
		LDX	DM3
		STX	FCPNTR
		LDA*	FWPNTR
		EXU	RSHTBL+3,2
		ETR	=377B
		SBRR	FCS00

FLOATING DIVIDE BY TEN SUBROUTINE  
91 CYCLES (920), 55 CYCLES (930)

*	FDT00	ZR0	LFDT00
*		LDA	BE
		SUB	D3
		CAX	ADJUST BINARY EXPONENT



LDA	DBUF1	
LDB	DBUF	
RSH	1	
DIV	M02400	COMPUTE FIRST WORD OF QUOTIENT
STA	DBUF	AND SAVE
BAC		
DIV	M02400	COMPUTE SECOND WORD OF QUOTIENT
MUL	ONE	POSITION IN B
LDA	DBUF	
NOD	1	POST-NORMALIZE
STA	DBUF1	
STB	DBUF	
STX	BE	
SBRR	FDT00	
* * FMT00		
ZR0	LFMT00	FLOATING MULTIPLY BY TEN SUBROUTINE 53 CYCLES (920,930)
LDA	D4	
ADM	BE	COMPUTE NEW BINARY EXPONENT
LDA	DBUF1	
MUL	M02400	MULTIPLY 1ST WORD
STA	DBUF1	
LDA	DBUF	
STB	DBUF	
CLB		
RCY	1	
MUL	M02400	MULTIPLY 2ND WORD
MUL	ONE	SHIFT RIGHT 23
XAB		
ADD	DBUF	
XAB		
ADC	DBUF1	
LDX	BE	
NOD	1	POST NORMALIZE IF NECESSARY
STA	DBUF1	
STB	DBUF	
STX	BE	
SBRR	FMT00	
* INSERT CHARACTER IN DBUF SUBROUTINE		

```

*
ICD00  ZR0      LICD00      20 CYCLES (920,930)
        XMA      DBUF        SAVE CHARACTER TO BE INSERTED
        LDB      DBUF1
        LSH      0,2         SHIFT DBUF (3 IF 0, 8 IF A OR R)
        XAB
        MRG      DBUF        INSERT CHARACTER
        STA      DBUF1      AND PUT BACK IN DBUF
        STB      DBUF
        SBRR     ICD00

*
*
ICF00  ZR0      LICF00
        XAB
        LDX      FCPNTR
        SKN*     ICF00      PUT CHARACTER IN B
        BRU      ICF02      TEST IF POINTER TO BE MOVED
        BRX      **+3       MOVE FORMAT POINTER
        MIN      FWPNTR
        LDX      DM3
        STX      FCPNTR
ICF02  EXU      LSHTBL+3,2  POSITION CHARACTER
        LDA      CHMSK+3,2
        ETR*     FWPNTR
        STB*     FWPNTR
        ADM*     FWPNTR      INSERT IT IN FORMAT
        SBRR     ICF00
    
```

```

*OUTPUT CHAR TO FILE
$ICW00 ZR0      LICW00
        SKR      DW         TEST IF FIELD WIDTH PERMITS INSERTION
        SBRR     ICW00      IT DOES NOT; EXIT
ICW02  BNE      #'         ',ICW10
        MIN      MSCNT     MULTIPLE SPACE COUNT
        BRR      LICW00
    
```

ICW10	XMA	MSCNT	
	SKG	#3	
	BRU	ICW30	
	CAB		
	LDA	#135B	MS ESCAPTE CHAR
	CI0	CURFLE	
	CBA		
	CI0	CURFLE	
ICW14	CLA		
ICW15	XMA	MSCNT	
ICW20	SKE	#377B	MULTIPLE SPACE FLUSH CHAR
	CI0	CURFLE	CURRENT FILE
	BRR	LICW00	
ICW30	XMA	MSCNT	
	RCH	AB+CA	
ICW32	SKR	MSCNT	
	BRU	**2	
	BRU	ICW35	
	CLA		
	CI0	CURFLE	
	BRU	ICW32	
ICW35	STA	MSCNT	
	CBA		
	BRU	ICW20	
*			
\$ICU00	ZR0	LICW00	OUTPUT CHARACTER UNCOND.
	BRU	ICW02	
*			
*			
IMT00	ZR0	LIMT00	INTEGER MULTIPLY BY TEN SUBROUTINE
	LDA	DBUF	40*41 CYCLES (920,930)
	MUL	D5	MULTIPLY 2ND WORD
	SKA	SIGN	
	ADD	D10	ADD CORRECTIVE 10 IF ARGUMENT < 0
	STB	DBUF	
	XMA	DBUF1	
	MUL	D5	MULTIPLY 1ST WORD
	XAB		
	ADD	DBUF1	ADD PREVIOUS PRODUCT

```

XAB
ADC      ZERO      PUSH CARRY OVER
LSH      1
STA      DBUF1
BAC
RCY      1          SEPARATE INTEGER
XMA      DBUF1     FROM BSST OF PRODUCT
SBRR     IMT00

*
*
ISS01    CLA        SET INTEGER FLAG (* = SET)
BRU      **2
ISS02    LDA        BSSET INTEGER FLAG (= = BSSET)
LDA      ONES
STA      IF
MIN      DS        INDICATE DATA SPEC. FOUND
STX      FT        SAVE TRANSFER INDEX
SKN      IZ        SKIP IF EXPLICIT ZERO COUNT
LDB      ONE       OTHERWISE MAKE COUNT = 1
STB      RC        AND SAVE REPEAT COUNT
SBRM     DNS00     GO GET FIELD WIDTH
STB      FW
SKN      IF        SKIP IF INTEGER FLAG BSSET
BRU      ISS12     OR BYPASS ACQUISITION OF D
SKE      PER       TEST IF NEXT CHARACTER = PERIOD
BRU      ISS12     IT IS NOT; SET D = 0
LDA      IZ        IT IS; GO GET D
STA      TMP1Z     SAVE STATUS OF IZ
SBRM     DNS00     GO GET D FROM FORMAT
LDA      TMP1Z
STA      IZ
ISS09    STB       D
*** COME HERE FROM D2B AND FROM ADS WHEN A,R FORMATS
ISS10    LDX       FT
SKR      RC        TEST REPEAT COUNT
BRU      ISS15+20,2  GO/RETURN TO APPROPRIATE PROCESSOR
CLAB
STA      IZ        SET IMPLICIT=ZERO FLAG
LDA      NXTCHR    PICK UP NEXT CHARACTER
BRU      I0E31     RETURN TO MAIN SCAN

```

ISS12 CLAB 0 SET D = 0  
 BRU ISS09

\*  
 ISS15 BRU E00 E  
 BRU F00 F  
 BRU I00 I  
 BRU A00 A  
 BRU 000 0  
 BRU R00 R

\*  
 \*  
 \* MOVE POINTER RIGHT SUBROUTINE  
 \* 36=40 CYCLES (920,930)

MPR00 ZR0 LMPR00  
 AXC  
 MPR04 STX MPRX  
 SKN 0T  
 BRU \*\*2  
 BRU MPR10  
 SBRM WCS00  
 BNE =155B,MPR05  
 MOVE =-1,FE0R,A  
 BRU MPR06  
 MPR05 LDX MPRX  
 BRX MPR04  
 MPR06 MIN LMPR00  
 SBRR MPR00  
 MPR10 LDA = ' '  
 SBRM ICU00  
 BRU MPR05

SET FOR NORMAL RETURN

\*  
 \*  
 \* REDUCE TO NORMAL FORM SUBROUTINE  
 \* 25+62E = 30+100E CYCLES (920)

RNF00 ZR0 LRNF00  
 CLAB  
 STA DE SET DECIMAL EXPONENT = 0  
 SKE DBUF1  
 BRU RNF03

RNF02	SBRR SBRM SKR NOP	RNF00 FMT00 DE	EXIT IF ARGUMENT = 0  DE=DE+1
RNF03	LDA SKG	BE DM3	TEST BINARY EXPONENT
RNF05	BRU SKG BRU SBRM MIN LDA BRU	RNF02 ZER0 RNF10 FDT00 DE BE RNF05	< =4: REDUCE BY MULTIPLICATION  IN RANGE; GO SCALE AT BINARY 0 > 0: REDUCE BY DIVISION DE=DE+1
RNF10	CNA CAX LDA LDB RSH STA STB SBRR	DBUF1 DBUF 0,2 DBUF1 DBUF RNF00	M0VE SHIFT COUNT TO X PICK UP REDUCED ARGUMENT  SCALE RIGHT TO BINARY 0
*			TRUNCATE TO INTEGER SUBROUTINE
*			17=80 CYCLES (920); 17=42 CYCLES(930)
TTI00	ZR0 LDA SKA LDA SUB SKA BRU SBRR STX CNA CAX LDA LDB RSH LSH STA	LTTI00 BE SIGN ONES D47 SIGN **2 TTI00 TMPX  DBUF1 DBUF 0,2 0,2 DBUF1	LOCATE POSITION OF BINARY POINT  ARGUMENT <1  =1 < BE < 48  TRUNCATE ALL BITS TO RIGHT OF BINARY

STB DBUF  
LDX TMPX  
SBRR TTIOO

POINT

3R6 PAGE 35

\*  
\*  
\*  
WCS00 ZR0 LWCS00  
\*ALL 8-BIT TRIMED ASCII CHARS ARE LEGAL EXCEPT(144,135,137,155)  
\*LINE FEED ECHOED FOR C.R.  
\*MULTIPLE SPACES EXPANDED  
\*EOF, CR, EOT CAUSE RETURN  
SKN FE0R  
BRU RIF1  
LDA =1  
RIF1 BRR LWCS00  
SKR MSCNT  
BRU \*\*2  
BRU RIF3  
CLA  
RIF3 BRU RIF20  
CIB CURFLE CURRENT FILE NO  
ABC  
SKE CURFLE  
BRU RIF4  
RIF4 MOVE =-1,FRUB,A  
CBA  
LDX =1  
SBRM ACB  
DATA 15200000B+RIF3 IGNORE L.F.  
DATA 15500000B+RIF16 C.R.  
DATA 14400000B+RIF13 E.O.T.  
DATA 13500000B+RIF7 M.S.  
DATA 13700000B+RIF14 E.O.F.  
DATA 9BIT+RIF5  
RIF5 SKG =377B  
SKG =-1  
BRU RIF3  
RIF20 SBRR WCS00  
RIF13 CLA

```

SKE      CURFLE
BRU      RIF16
TC0      =152B
RIF16   LDA      =CARRET
BRU      RIF20
RIF7    CI0      CURFLE
STA      MSCNT
BRU      RIF1
RIF14   MOVE     =-1,FRUB,A
SBRM     ERROR
BRU      CS
ASC      CMMD START
         ' END OF FILE READ/'

```

```

*
*
*   DATA TABLES AND CONSTANTS
*

```

CHARACTER TABLE

```

CHRTBL EQU *
E       DATA 45B      E
        DATA 46B      F
        DATA 51B      I
        DATA 41B      A
        DATA 57B      0
        DATA 62B      R
X       DATA 70B      X
CMA     DATA 14B      ,
DL      DATA 04B      $
RB      DATA 75B      ]
LB      DATA 73B      [
SLASH   DATA 17B      /
H       DATA 50B      H
PP      DATA 60B      P
MS      DATA 15B      =
        DATA 13B      +
RP      DATA 11B      )
LP      DATA 10B      (
PER     DATA 16B      .
        DATA 155B,152B CR/LF (IGNORE)

```



\* CHARACTER RIGHT SHIFT TABLE  
 \$RSHTBL LRSB 16  
           LRSB 8  
           NOP 0

\* CHARACTER LEFT SHIFT TABLE  
 \$LSHTBL LSH 16  
           LSH 8  
 \$NOPMOP NOP 0,2

\* ROUND OFF TABLE  
 \$RNDTBL DATA 0,020000000B  
           DATA 031463147B,01463146B 5\*10(=2)  
           DATA 02436560B,0121727B 5\*10(=3)  
           DATA 023351362B,010142B 5\*10(=4)  
           DATA 033427262B,0643B 5\*10(=5)  
           DATA 074265422B,051B 5\*10(=6)  
           DATA 014336750B,04B 5\*10(=7)  
           DATA 032657712B,0 5\*10(=8)  
           DATA 02536310B,0 5\*10(=9)  
           DATA 0211340B,0 5\*10(=10)  
           DATA 015574B,0 5\*10(=11)  
           DATA 01300B,0 5\*10(=12)  
           DATA 0106B,0 5\*10(=13)  
 \$D8 DATA 010B,0 5\*10(=14)

\* CONSTANTS  
 \$D13 DATA 13  
 D24 DATA 24  
 D39 EQU 39  
 DD39 EXT 39  
 \$D99 DATA 99  
 \$D100 DATA 100  
 \$D511 DATA 511  
 \$D83885 DATA 83885  
 \$DM2 DATA =2  
 \$DM3 DATA =3  
 \$DM7 DATA =7  
 \$DM8 DATA =8  
 \$DM12 DATA =12

```

$DM20 DATA =20
$DM100 DATA =100
M040 EQU 40B
$M04000 DATA 04000B
$M040L DATA 0400000B
$M02400 DATA 024000000B
$ARC15 DATA 063146316B,03146314B
ERRFCH DATA 026233000B
ERR0CT DATA 046236300B
$ADRLP0 ZR0 LPSTK FSCAN LPSTK ADDRBS55 0F 0 LEVEL (
$ADRLP1 ZR0 LPSTK3 FSCAN LPSTK ADDRBS55 0F 1 LEVEL (

```

```

$IFP0NE DATA 1
$TAGBIT DATA 020000000B

```

\*  
\* P R O G R A M M E D O P E R A T O R L I N K A G E S  
\*

```

$P0PLST BRU XSDP0P 100 FIXED SET UP DUMMY
BRU FSDP0P 101 FLOATING SET UP DUMMY
BRU PRCP0P 102 PRINT CHAR
BRU PRQP0P 103 PRINT QUOTE
BRU XSTP0P 104 FIX AND STORE
BRU FSTP0P 105 FLOAT AND STORE
BRU * 106
BRU STDP0P 107 STORE DOUBLE PRECISION
BRU D0XP0P 110 D0 FIXED
BRU D0FP0P 111 D0 FLOATING
BRU AGXP0P 112 ASSIGNED G0 TO FIXED
BRU AGFP0P 113 ASSIGNED G0 TO FLOATING
BRU XFAP0P 114 FIXED FIRST ARGUMENT
BRU FFAP0P 115 FLOATING FIRST ARGUMENT
BRU XNAP0P 116 FIXED NEXT ARGUMENT
BRU FNAP0P 117 FLOATING NEXT ARGUMENT
BRU XI0P0P 120 FIXED INPUT/OUTPUT
BRU FI0P0P 121 FLOATING INPUT/OUTPUT
BRU LTFP0P 122 LOAD THEN FLOAT
BRU LTXP0P 123 LOAD THEN FIX
BRU * 124

```

BRU	LDPP0P	125	LOAD DOUBLE PRECISION
BRU	FTAP0P	126	FLOAT THEN ADD
BRU	*	127	(DIV = 910)
BRU	*	130	(SKB = 910)
BRU	FLAP0P	131	FLOATING ADD
BRU	FTSP0P	132	FLOAT THEN SUBTRACT
BRU *		133	
BRU	*	134	(CAX = 910)
BRU	FLSP0P	135	FLOATING SUBTRACT
BRU	FTMP0P	136	FLOAT THEN MULTIPLY
BRU	*	137	
BRU	XMPP0P	140	FIXED MULTIPLY
BRU	FLMP0P	141	FLOATING MULTIPLY
BRU	FTDP0P	142	FLOAT THEN DIVIDE
BRU	*	143	(STE = 910)
BRU	XDVP0P	144	FIXED DIVIDE
BRU	FLDP0P	145	FLOATING DIVIDE
BRU	*	146	(CNA = 910)
BRU	FLNP0P	147	FLOATING NEGATE
BRU	ALXP0P	150	ASSIGN LABEL TO FIXED
BRU	ALFP0P	151	ASSIGN LABEL TO FLOATING
BRU	BKPP0P	152	BREAKPOINT P0P
BRU	DPSP0P	153	DOUBLE PRECISION SUBTRACT
BRU	DPAP0P	154	DOUBLE PRECISION ADD
BRU	DPMP0P	155	DOUBLE PRECISION MULTIPLY
BRR	0	156	STATEMENT LABEL P0P
\$P0PEND BRU	*	157	(CXB = 910)

```

*
* LINKAGES TO USER CODED PROGRAMME
* OPERATORS 161 SYSTEM THRU 177 SYS AND
* LINKAGES TO SYSTEM SUBROUTINES
* 200 SYS THRU 247 SYS AS REQUIRED
*

```

```

*
* C O N S T A N T S
*

```

\$WBLIM DATA 70  
 \$MINUS DATA 40000000B  
 SIGN EQU MINUS  
 \$ADRMSK DATA 37777B  
 \$P0PADR DATA 017600000B  
 \$AP0PCD DATA 015000000B  
 CARRET EQU 155B  
 SPACES DATA 0  
 BLNK EQU SPACES  
 \$FLIND DATA 01000000B  
 M077 DATA 077B  
 CHMSK DATA 00177777B  
 DATA 77600377B  
 DATA 77777400B  
 \$M770 DATA 077777700B  
 \$0PMASK DATA 37700000B  
 MSK3C EQU CHMSK  
 \$M01777 DATA 017777777B

MAXIMUM FIELD WIDTH ALLOWED

3R6 PAGE 40

FLOATING INDICATOR

\*  
 \* FIX, FLOAT AND ERROR LINKAGES  
 \*

\*UNDEFINED LABEL \*\* ALL UNSATISFIED TRANSFERS POINT HERE

\$UNDEF SBRM ERROR  
 BRU CS COMMAND START  
 ASC ' UNDEFINED LABEL REFERENCED/'

\*  
 \* I / 0 BUFFER PARAMETERS  
 \*

\*  
 \* C O N S T A N T S  
 \*

D2 DATA 2  
 TW0 EXT D2  
 TW0 EQU D2  
 D3 DATA 3  
 D4 DATA 4  
 D5 DATA 5

D6	DATA	6
D7	DATA	7
D9	DATA	9
D10	DATA	10
\$D23	DATA	23
D47	DATA	47
DM4	DATA	=4
M03777	DATA	037777777B
MAXP8S	EXT	M03777
MAXP8S	EQU	M03777
M0377	DATA	0377B
DD255	EXT	M0377
DD255	EQU	M0377
M0400	DATA	0400B
NEGBEX	EXT	M0400
NEGBEX	EQU	M0400
\$EMASK	DATA	0700B
M0777	DATA	0777B
0777	EXT	M0777
0777	EQU	M0777
\$ML8NES	DATA	077777000B
DM257	DATA	=257
M257D	EXT	DM257
M257D	EQU	DM257
9BIT	EQU	00040000B

\*  
\* Q U O T E S  
\*

\$CRLF	ASC	!\$ /!
\$LDCOMP	ASC	!\$LOADING COMPLETE\$ /!
\$SPCEQT	ASC	! WORDS OF STORAGE UNUSED /!
\$MEMQT	ASC	!\$M>> /!
\$ILLQT	ASC	!\$I>> /!
\$ERRQT1	ASC	! \$ERROR: /!
\$ERRQT2	ASC	!\$EITHER JG TO GO, JD TO DISREGARD, OR RUBOUT /!
\$PSEQT	ASC	!\$PAUSE /!
\$ST8PQT	ASC	!\$*ST8P*\$ /!

END

CIT 0PD 13400000B,1,1,0,1 CHAR. INPUT AND TEST

\*RCH 0P DEFINITIONS

```

FCL 0PD 4600450B,2 FULL CYCLE LEFT
FCR 0PD 4600224B,2 FULL CYCLE RIGHT
BXC 0PD 4600022B,2 B TO X, CLB
XBC 0PD 24600040B,2 X TO B, CLX
CA EQU 1 CLEAR A
CB EQU 2
* CX EQU 20000000B ARPAS DEMANDS ,2
AB EQU 4 A TO B
BA EQU 10B
BX EQU 20B
XB EQU 40B
XA EQU 200B
AX EQU 400B
MOVE MACRO D
MOVE1 NARG
      IF 'D(1$1,2)'='0'
      CL,D(MOVE1)
      ELSF 1
      LD,D(MOVE1) D(1)
      ENDF
MOVE2 EQU 2
      RPT MOVE1-2
      ST,D(MOVE1) D(MOVE2)
MOVE2 EQU MOVE2+1
      ENDR
      ENDM
*
STZ MACRO A STORE ZERO; CLA ; STA A(1) ; ENDM
BFS MACRO D; SKN D(1); BRU D(2); ENDM
BTR MACRO D; SKN D(1); BRU **2; BRU D(2); ENDM
BEQ MACRO D; SKE D(1); BRU **2; BRU D(2); ENDM
BNE MACRO D; SKE D(1); BRU D(2); ENDM
SETT MACRO D; MOVE =-1,D(1),D(2); ENDM
    
```

SETF MACRO D; MOVE =0,D(1),D(2); ENDM

```

*
*
ICT      0PD      10600000B,1,1
PRC      0PD      10200000B,1,1
PRQ      0PD      10300000B,1,1
FLA      0PD      013100000B,1,1
FLS      0PD      013500000B,1,1
FLM      0PD      014100000B,1,1
FLD      0PD      014500000B,1,1
LFP      0PD      012500000B,1,1
STD      0PD      010700000B,1,1
AXC      0PD      04600401B,2
AXCE     0PD      04600501B,2
AXBA     0PD      04600450B,2
BXC      0PD      04600022B,2
BXAC     0PD      04600222B,2
XI0      0PD      12000000B,1,1
FI0      0PD      12100000B,1,1
XMP      0PD      14000000B,1,1
FFA      0PD      11500000B,1,1
FLN      0PD      14700000B,2

```

```

6BIT EQU 00400000B
9BIT EQU 00040000B

```

```

*****
** SYSTEM SUBROUTINES **
*****

```

```

*
$STRTDM HLT START OF DUMMIES
LSTRTD

```

LDA EOADR  
 STA EADR2  
 BRR LSTRTD

```
*          940  END OF DUMMIES
$ENDDMY HLT  LNDDMY
          LDA  EADR2
          SKE  EADR1
          BRU  E1
E3        BRR  LNDDMY
E1        SBRM ERROR
          BRU  E3
          ASC  ' WRONG NUMBER OF ARGUMENTS IN SUBROUTINE CALL/'
```

```
*          940  COMPUTED GO TO
$CMPG0 ZR0  LCMPG0
          MIN  LCMPG0
          SKG* LCMPG0
          SKG  ZER0
          BRU  CGERR
          SUB  0NE
          ADM  LCMPG0
C0G04    BRR  LCMPG0
CGERR    SBRM ERROR
          BRU  C0G04
          ASC  ' EXPRESSION IN COMPUTED GO TO IS OUT OF BOUNDS/'
```

```
*          224  IF0VFL          IF OVERFLOW
$IF0VFL HLT  LIFVFL
          SKN  0VFIND          *1* OVERFLOW
          MIN  LIFVFL
          CLAB
          STA  0VFIND
          BRR  LIFVFL
```



```

*          940  INPUT-OUTPUT LIST UNSCRIPTED ARRAY
$I0LUSA HLT  LI0LSA
          LDA  E0ADR
          STA  EADR
          LDA* EADR
          STA  ARRAY
          MIN  EADR
          LDA* EADR
          ETR  #37777B
          STA  ELEMTS
          LDA  ARRAY
          SKA  FLTIND
          BRU  F3
F2        SKR  ELEMTS
          BRU  F1
          BRR  LI0LSA
F1        XI0* ARRAY
          MIN  ARRAY
          BRU  F2
F3        SKR  ELEMTS
          BRU  F4
          BRR  LI0LSA
F4        FI0* ARRAY
          MIN  ARRAY
          MIN  ARRAY
          BRU  F3

```

```

*
*
*          IF SENSE SWITCH (204)
$IFSNSW ZR0  LIFSNSW
          STX  TX
          SKG  DEC24
          SKG  ZER0

```

	BRU	C2
	CAX	
	LDA	=40000000B
	CAB	
	LCY	0,2
	LDX	TX
	SKM	SENSSW
	MIN	LIFSNW
C3	BRR	LIFSNW
C2	SBRM	ERROR
	BRU	C3
	ASC	' SENSE SWITCH EXPRESSION OUT OF BOUNDS/'

*	IF SENSE LIGHT (205)	
\$IFSNLT	HLT	LIFSNL
	STX	TX
	SKG	DEC24
	SKG	ZERO
	BRU	D1
	CAX	
	LDA	=40000000B
	LDB	=40000000B
	LCY	0,2
	LDX	TX
	SKM	SENSLW
	MIN	LIFSNL
	EOR	=1
	ETR	SENSLW
	STA	SENSLW
D1	BRR	LIFSNL
	SBRM	ERROR
	BRU	D2
	ASC	' SENSE LIGHT EXPRESSION OUT OF BOUNDS/'
D2	MIN	LIFSNL
	BRR	LIFSNL

\*

```

*
* SENSE LIGHT (227)
$SENSLT HLT LSENST
    STX TX
    CLB
    SKE ZERO
    LDB SENSLW
    STB SENSLW
    SKG DEC24
    SKG =1
    BRU B1
    RCH AX+CA
    LDB =40000000B
    LSH 0,2
    MRG SENSLW
    STA SENSLW
    LDX TX
    BRR LSENST
B1 LDA LSENST
    SUB =1
    STA LIFSNL
    BRU D1
DEC24 DATA 24

```

```

*****
**          D E B U G G E R          **
*****
DCS BRU CS COMMAND START
$DEBUG NOP 0
D10 BRU D50
    $BRM TIN
    STA CCHR
    BRU D50
*TEST FOR END OF LINE
DTE0L LDA CCHR
DTE0L2 SKE =155B

```

```

BRU      D50
SBRM     TIN
STA      CCHR
BRU      DTE0L2
*GET PROGRAM OR VARIABLE NAME
D50      SETT  FNAME,A
        SETF  FFLOAT,A
        LDX   =6
        LDA   CCHR
        BEQ   =' $',D65
*TEST IF INTEGER VARIABLE
        SKG   =' N'
        SKG   =' I'-1
BRU      **2
BRU      D57
SETT     FFLOAT,B
D53      BRU   D57          1ST CHAR MUST BE LETTER
        BFS   FNAME,D63
        SBRM  TIN
        SKG   =' 9'
        SKG   =' 0'-1
D55      BRU   D57
        LRSH  8
        LDA   DNAME2
        LCY   8
        STA   DNAME2
        LSH   16
        LDA   DNAME1
        LSH   8
        STA   DNAME1
        BRX   D53
        BFS   FNAME,D70      BUT
*SKIP REST OF NAME AFTER FIRST 6 CHARACTERS
D56      SBRM  TIN
        STA   CCHR
        SKG   =' Z'
        SKG   =' A'-1
BRU      D58
BRU      D56

```

```

D58   SKG      =!   9!
      SKG      =!   0!-1
      BRU      D70
      BRU      D56
D57   SKG      =!   Z!
      SKG      =!   A!-1
      BRU      D60
      BRU      D55
*BLANK OUT REST OF NAME IF LESS THAN 6 CHARACTERS
D60   LDB      =!   '
      STA      CCHR
D62   STB      DNMCHR
D63   SETF     FNAME,B
      LDA      DNMCHR
      BRU      D55
*DOLLAR OUT REST OF NAME
D65   LDB      =!   $!
      SBRM     TIN
      STA      CCHR
      BRU      D62

*EXAMINE NAME TERMINATOR
D70   LDA      DNAME1
      BNE     =!   ',D400
      MOVE    ZERO,DNUMB,DADRIN,A
      LDA     CCHR
      BEQ     =!   ',D230   FORCES MAIN PROGRAM START
D75   SKG      =!   9!
      SKG      =!   0!-1
      BRU      D80
      SUB     =!   0!
      XMA     DNUMB
      MUL     =5
      CBA
      ADM     DNUMB
D77   SBRM     TIN
      BRU      D75
D80   CLX

```

	SBRM	ACB	
	DATA	01000000B+D100	EXCLAMATION PT
	DATA	'=' + D85	
	DATA	'/' + D1000	STATEMENT MODIFICATION
	DATA	';' + D200	
	DATA	'.' + D90	
	DATA	'+' + D95	PBS. RELATIVE ADDRESS INCREMENT
	DATA	'-' + D94	NEG. REL. ADDR. INC.
	DATA	'I' + D1120	COMMON INTEGER TYPEOUT
	DATA	'R' + D1100	COMMON REAL
	DATA	'F' + D1100	
DBAD	DATA	6BIT+DBAD	
	BRU	BAD	

*TYPE	VALUE	
D85	LDB	=10
	LDX	=1
	LDA	DNUMB
	BRS	36
	LDA	DADRIN
	BEG	ZER0,CS
	SKG	ZER0
	BRU	D87
	TC0	' ' +'
	LDA	DADRIN
D86	LDB	=10
	LDX	=1
	BRS	36
	BRU	CS
D87	TC0	' ' -'
	LDA	DADRIN
	CNA	
	BRU	D86
D90	LDA	DD0TA
	SKE	=1
	BRU	D92

D92 BRU DBAD DBT UNASSIGNED  
 STA DNUMB  
 MOVE DD0TR,DADRIN,A  
 BRU D77

\*NEGATIVE RELATIVE ADDR

D94 LDB =10  
 CLX  
 BRS 38  
 CNA  
 BRU D97

\*POSITIVE RELATIVE ADDR

D95 LDB =10  
 CLX  
 BRS 38

D97 ADM DADRIN ADDRESS INCREMENT  
 CBA TERMINATING CHAR  
 BRU D80

\*BREAK POINT

D100 SBRM TIN  
 BNE =155B,D105  
 STA CCHR  
 BRU D125

REMOVE ALL BREAK POINTS

D105 SUB = ' 0'  
 SKG =3  
 SKG =0  
 BRU DBAD  
 STA DBN  
 LDA DNUMB  
 SKE ZERO  
 BRU \*\*2  
 BRU \*\*3  
 SBRM D150  
 BRU CS

\*REMOVE SOMETHING

SKE DBN  
 BRU D120

## \*REMOVE ALL BREAK POINTS

```

D109 LDX      =1
D110 SKN      BKOADR,2
      SBRM    D130
      EAX     1,2
      CXA
      SKG     =3
      BRU    D110
      BRU    CS

```

## \*REMOVE A BREAK POINT

```

D120 LDX      DBN
      SBRM    D130
      BRU    CS

```

MUST HAVE A, NOTHING MORE  
REMOVE ALL B.P.

```

D125 LDA      DNUMB
      BNE     =0,DBAD
      BRU    D109

```

## \*(X)=BP NUMBER

```

D130 ZR0      LD130
      LDA     BKOADR,2
      SKE     =-1
      BRU    *+2
      BRR    LD130
      ETR     =37777B
      FCL
      LDA     0,2
      ETR     =37777B
      MRG    STLM0P
      STA     0,2
      FCR
      LDA     =-1
      STA     BKOADR,2
      BRR    LD130

```

## \*SET A BREAK POINT

```

D150 ZR0      LCLFLE
      SBRM    L0CSTA

```



```

CXA
STA DTMP2
LDA DBN
SKE ZERO
BRU *+2
LDA ONE
CAX
SBRM D130 REMOVE OLD BP IF SET
LDA DTMP2
STA BKOADR,2
CAX
LDA 0,2
ETR #77740000B
SKE STLMOP
BRU DBAD ANOTHER BREAK POINT ALREADY SET HERE
LDA 0,2
ETR #37777B
MRG BKPMOP
STA 0,2
SBRR D150

```

```

D200 SETF FNEXT,A
SBRM TIN
D210 CLX
SBRM ACB
DATA !S!+CSET
DATA !R!+CRSET
DATA !O!+DOUT OPEN OUTPUT FILE
DATA !I!+DIN OPEN INPUT FILE
DATA !C!+DCLOSE CLOSE FILE
DATA !G!+D250
DATA !N!+D260 NO-DEBUG
DATA !W!+D1200 WRITE GOTO BINARY PROGRAM
DATA !P!+D1300 PROCEED AFTER BREAK OR ESCAPE
DATA 6BIT+DBAD

```

\*CLOSE FILE

```

DCL0SE LDA      DNUMB
      BNE      =0,DCL0S5
      SBRM     CLAFLE      CLOSE ALL FILES
      BRU      CS
DCL0S5 SBRM     CLFLE
      BRU      CS

```

```

*OPEN OUTPUT FILE
DOUT  LDA      DNUMB
      SKG      =9
      SKG      =1
      BRU      DBAD
      SBRM     CLFLE      IF FILE OPENED, CLOSE IT
      TC0      =1
      CLEAR
      BRS      18
      BRU      DBAD
      XAB
      SKE      =155B
      BRU      DBAD
      XAB
      LDX      =3          SYMBOLIC
      BRS      19
      BRU      DBAD
      MRG      =40000000B
      LDX      DNUMB
      STA      FLNMT,2
      BRU      CS

```

```

*OPEN INPUT FILE
DIN   LDA      DNUMB
      SKG      =9
      SKG      =1
      BRU      DBAD
      SBRM     CLFLE
      TC0      =1
      CLEAR
      BRS      15

```

```

BRU   DBAD
BRU   16
BRU   DBAD
ETR   =37777777B
LDX   DNUMB
STA   FLNMT,2
BRU   CS

```

\*CMMD, SENSE SWITCH (SET/RESET)

```

$CSET MOVE =0, FSSW, A
BRU     CSSW
$CRSET MOVE =1, FSSW, A
CSSW   LDA   DNUMB
        SKG  =24
        SKG  ZERO
BRU     BAD
        LDB  SIGN
        RCH  AX+BA
        LCY  0,2
        SKN  FSSW
BRU     CSSW3
        EOR  =1
        ETR  SENSSW
BRU     CSSW4
CSSW3  MRG  SENSSW
CSSW4  STA  SENSSW
BRU     CS

```

```

D230   SETF  FNEXT, A
        SBRM TIN
        BNE  =1 G1, D210
$D240  SBRM  SLDPNT
        LDX  =1

```

NOT SEMI-COLON G  
FORCE TO MAIN PROGRAM

```

D250  CLA          12          SET BREAK ON EVERYTHING (ON GO TO RETURN)
      BRS          ZER0,DNUMB,A
      MOVE         =*1,MRSTL,A
      MOVE         CRLF
      PRQ          DADRIN,DD0TR,A  SET D0T
      MOVE         L0CSTA        LOCATE STATEMENT
      SBRM        0,2
      LDA          =17700000B
      LDB          BKPM0P
      SKM          **2
      BRU          1,2          FIRST WORD AFTER B.P.
      STX          DTMP1
      MOVE         =*2,FRUN,A
      CLEAR
      STA          BRKRP        CLEAR BREAK RETURN ADR.
      BRU*        DTMP1
D260  SKN          BK1ADR
      BRU          **2
      BRU          DBAD
      LDA          BK1ADR
      ETR          =37777B
      CAX
      LDB          =77400000B
      LDA          1,2
      SKM          =100000B
      BRU          **4
      ETR          =37777B
      SBRM        D150
      BRU          D250
      CAX
      SBRM        L0CLB4
      LDA          =1
      STA          DBN
      ADD          DD0TR
      STA          DADRIN
      LDA          DD0TA
      STA          DNUMB
      SBRM        D150

```

D400 SKR DADRIN  
 BRU D250  
 LDA CCHR  
 BNE =14B,D450

\*SWITCH SUBPROGRAM

D410 LDX =LDPNT LOAD POINT  
 D405 LDA DNAME1  
 SKE 1,2  
 BRU D420  
 LDA DNAME2  
 SKE 2,2  
 BRU D420

\*NAME MATCHED

CXA  
 ETR =37777B  
 STA UPR0GS USER PROGRAM START  
 LDA 0,2  
 ETR =37777B  
 SUB =1  
 STA UPR0GN USER PROGRAM END  
 LDA 3,2  
 ETR =37777B  
 STA UDATAS USER DATA START  
 BRU D10

D420 LDA 0,2  
 ETR =07700000B  
 BEQ ZERO,DBAD SPR0G NOT FOUND  
 LDX 0,2  
 BRU D405

\*FIND VARIABLE BASE  
 D450 LDX UPR0GN USER PROGRAM END

```

D455  LDX      0,2
      LDA      DNAME1
      SKE      0,2
      BRU      D460
      LDA      DNAME2
      SKE      1,2
      BRU      D460
      LDA      2,2
      STA      DVBASE      VARIABLE BASE
      BRU      D500
D460  CXA
      ADD      =3
      CAX
      SKG      UPR0GN
      BRU      D455
      BRU      BAD      NO SUCH VARIABLE IN THIS SPR0G

```

## \*ADJUST FOR VARIABLE SUBSCRIPTS

```

D500  LDA      CCHR
      BEQ      =10B,D510      LEFT (
*SCALER VARIABLE
      LDA      2,2
      ADD      UDATAS      USER DATA START
      STA      DVBASE
      LDA      CCHR
      BRU      D610

```

## \*ARRAY VARIABLE

```

D510  CLX
      LDB      =10
      BRS      38
      XAB
      BNE      =11B,DBAD      NOT RT. PAREN
      STA      CCHR
      LDA      UDATAS
      ADM      DVBASE
      CBA
      SUB      =1      ZERO=ORIGIN INDEXING

```

	ADM	DVBASE
	BFS	FFLBAT,D530
	ADM	DVBASE
D530	SBRM	TIN
D610	CLX	
	SBRM	ACB
	DATA	'/' + D620
	DATA	'[' + D700
	DATA	'!' + D800
	DATA	6BIT + DBAD

	*INTEGER	==	FL	PT.
D620	BTR			FFLBAT,D625
	SKN*			DVBASE
	BRU			D621
	TC0		'	'
	LDA*			DVBASE
	CNA			
	BRU			D622
D621	TC0		'	'
	LDA*			DVBASE
D622	LDB			=10
	LDX			0NE
	BRS			36
	TC0		'	'

## \*INTEGER INPUT

	TC0		'	'
	LDA			=155B
	CIT			ZER0
	BRU			**+2
	BRU			DCS
	LDB			=10
	CLX			
	BRS			38
	STA*			DVBASE
	BRU			DCS

CHAR. INPUT AND TEST ON CARRIAGE RET.  
 NEXT CHAR. NOT CAR. RET.  
 CAR. RET., NO CHANGE TO VARIABLE  
 INPUT DECIMAL INTEGER

## \*FLOATING PT

```

D625  LDX      =21100001B
      LDP*    DVBASE
      XAB
      BRS     53
      TCO     = ' '
      LDX      =21300000B
      BRS     52
      BRU    DCS
      XAB
      STP*    DVBASE
      BRU     DCS

```

## \*OCTAL OUTPUT

```

D700  TCO     = ' '
      BTR     FFLOAT,D710
      MOVE    =7,TEMP1,A
      LDA*    DVBASE
      BRU     D730
D710  MOVE    =15,TEMP1,A
      LDP*    DVBASE
D730  STP     DNAME1
D740  LDP     DNAME1
      LCY     3
      STP     DNAME1
      CBA
      ETR     =7
      ADD     = ' 0'
      CIO     ONE
      SKR     TEMP1
      BRU     D740
      TCO     = ' '

```

## \*OCTAL INPUT

```

      SBRM    TIN
      BEQ     =155B,DCS
      MOVE    =0,DNAME1,DNAME2,B

```



```

D920 BRU D925
      SBRM TIN
      BEQ =155B,D940
D925 SUB =' 0'
      SKG =7
      BRU **2
      BRU DBAD
      XMA DNAME2
      LDB DNAME1
      LCY 3
      ETR =77777770B
      ADM DNAME2
      STB DNAME1
      BRU D920

```

```

*ASCII == OUTPUT
D800 TC0 =' '
      LDA =3
      BFS FFL0AT,D805
      LDA =6
D805 STA TEMP1
      LDP* DVBASE
      STP DNAME1
D810 LDP DNAME1
      LCY 8
      STP DNAME1
      CBA
      ETR =377B
      CIO 0NE
      SKR TEMP1
      BRU D810
      TC0 =' '
      SBRM TIN
      BEQ =155B,DCS
      BRU D955

```

```

*CHARACTER INPUT
D950 SBRM TIN

```

D955 BEQ #155B,D940  
 XMA DNAME2  
 LDB DNAME1  
 LCY 8  
 ETR #77777700B  
 ADM DNAME2  
 STB DNAME1  
 BRU D950

D940 LDP DNAME1  
 D960 BTR FFL0AT,D970  
 STB\* DVBASE  
 BRU CS  
 D970 STP\* DVBASE  
 BRU CS

\* STATEMENT MODIFICATION\*

D1000	TC0	= ' ' !	
	TC0	= ' ' !	
	SBRM	TIN	TERMINAL INPUT
	BEQ	#155B,CS	CLOSE
	BNE	= ' C ',DBAD	
	PRQ	CONTQT	CONTINUE QUOTE

\*PLANT BRU AROUND STATEMENT\*

SBRM	L0CSTA	
EAX	1,2	
STX	LACB	
MIN	DADRIN	NEXT STATEMENT
SBRM	L0CSTA	
CXA		
MRG	#00100000B	
XXA		
BEQ	LACB,CS	
STX*	LACB	
BRU	CS	

\*CONVERTS COMMON ADDRESS BACK TO TRUE

D1120 SETF FFLBAT,B  
D1100 CLX  
LDA DNUMB CONVERTED DEC. BACK TO BINARY  
SKG #0  
BRU D1130  
LRSH 23  
DIV #10  
STA DNUMB  
LSH 24,2  
ADM DNAME1  
EAX 3,2  
BRU D1100+1  
D1130 LDA DNAME1  
SUB #50001B  
SKG #27777B MUST BE IN PROGRAM RANGE  
SKG #1177B CODE START =1 (SEE CODEST IN LDR.)  
BRU DBAD  
STA DVBASE  
BRU D530

\*WRITES OUT PROGRAM IN G0T0 FORM WITH BOOTSTRAP ROUTINE

D1200 LDA #55555555B CK SW IF ALREADY IN G0T0  
SKE SG0T0  
BRU \*+2  
BRU DBAD YES, NOT ALLOWED AGAIN  
STA SG0T0 SET SW.  
TC0 #155B  
LDX #9 MOVE BOOTSTRAP TO TEMPORARY STORAGE.  
LDA EG0T0,2  
STA EWG,2  
BRX \*+2  
CLEAR  
BRS 18 OPEN OUTPUT FILE  
BRU DBAD  
LDX #1 TYPE 1 FILE  
BRS 19  
BRU DBAD  
STA DNUMB FILE NO.

CLA		
WIO	DNUMB	FIRST CORE LOC,
LDA	#27777B	LAST CORE LOC.
WIO	DNUMB	
LDA	#WG0T0	STARTING LOCATION
WIO	DNUMB	
CLX		
LDA	#30000B	
BIO	DNUMB	
NBP		
LDA	#27657537B	
WIO	DNUMB	
WIO	DNUMB	
LDA	DNUMB	
BRS	20	
BRU	CS	

## \*BOOTSTRAP ROUTINE FOR G0T0 RETURN

	DATA	77770000B	KG0T0 IN 1R
	DATA	#F0RL	KG0T01 IN 1R
	BRS	43	
	XAB		
	ETR	KG0T0	
	MRG	KG0T01	
	XAB		
	BRS	44	
	BRU	D240	
EG0T0	EQU	*	
D1300	LDA	BRKRP	PROCEED AFTER BREAK OR ESCAPE
	BEG	ZER0,BAD	CK IF ALLOWED
	TC0	#155B	
	MOVE	#2,FRUN,A	
	LDP	BRKRA	
	LDX	BRKRX	
	BRR	BRKRP	

```

*LOCATE STATEMENT
*EXAMINES CURRENT SPR0G
*I/ (DNUMB)=STA LABEL NO, (DADRIN)=REL ADDR INC. IN NO. OF STATEMENTS
*0/ X=ADDR
$LOCSTA ZR0      LL0CST
                  LDA      UPR0GS      USER CURRENT PR0G START
                  ADD      =10B
                  CAX
                  LDA      DNUMB
                  BEQ      =-1,L0CST9
                  SKE      =0
                  BRU      L0CST8
                  BRR      LL0CST
L0CST1 CBA
L0CST8 LDB      =37777B
L0CST2 SKM      0,2
                  BRU      L0CST4
                  CAB
                  LDA      0,2
                  ETR      =77740000B
                  BNE      BKPM0P,L0CST3
                  BRU      LREL1
L0CST3 BNE      STLM0P,L0CST6
                  BRU      LREL1
L0CST4 EAX      1,2
                  XXA
                  SKG      UDATAS      USER DATA START = END OF CURRENT PR0G
                  BRU      L0CST5
                  BRU      DBAD
L0CST5 XXA
                  BRU      L0CST2
L0CST6 EAX      1,2
                  XXA
                  SKG      UDATAS
                  BRU      L0CST7
                  BRU      DBAD
L0CST7 XXA
                  BRU      L0CST1

```

L0CST9 LDX UPR0GS  
BRR LL0CST

\*LOCATE LABEL GIVEN STA ADDR  
\*I/A=PR0G ADDR  
\*0/ SETS D0T

\$L0CLBL ZR0 LL0CST  
ETR =37777B  
CAX  
MOVE =0,DD0TR,B  
L0CLB1 LDA 0,2  
ETR =77740000B  
BNE STLMP,L0CLB5  
L0CLB3 LDA 0,2  
ETR =37777B  
BNE ZER0,L0CLB4  
MIN DD0TR  
L0CLB9 EAX =1,2  
CXA  
SKG UPR0GS  
BRU \*\*2  
BRU L0CLB1  
SKE UPR0GS  
BRU \*\*2  
BRU L0CLB1  
LDA =-1  
L0CLB4 STA DD0TA  
BRR LL0CST  
L0CLB5 BEQ BKMP0P,L0CLB3  
BRU L0CLB9

\*RELATIVE ADDRESS

LREL1 MOVE DADRIN,L0CTMP,A  
SKN L0CTMP  
BRU LREL19  
\*GO BACK TO EARLIER STATEMENT  
LREL8 EAX =1,2  
CXA

```

      SKG      UPR0GS
      BRU      **2
      BRU      LREL12
      SKE      UPR0GS
      BRU      DBAD
LREL12 LDA      0,2
      ETR      =77740000B
      SKE      STLM0P
      BRU      LREL10
LREL11 MIN L0CTMP
      SKN      L0CTMP
      BRR      LL0CST
      BRU      LREL8
LREL10 SKE      BKPM0P
      BRU      LREL8
      BRU      LREL11
*GO FORWARD TO LATER STATEMENT
LREL19 SKR      L0CTMP
LREL20 EAX      1,2
      CXA
      SKG      UDATAS
      BRU      **2
      BRU      DBAD
      LDA      0,2
      ETR      =77740000B
      SKE      STLM0P
      BRU      LREL30
LREL21 SKR L0CTMP
      BRU      LREL20
      BRR      LL0CST
LREL30 SKE      BKPM0P
      BRU      LREL20
      BRU      LREL21

```

\*TERMINAL INPUT

\*X SAVED

```

$TIN  ZR0      LTIN
TIN10 CI0      ZER0

```

SKG       =77B  
SKG       =1  
BRU       \*\*2  
BRR       LTIN  
BEQ       =152B,TIN10  
BRR       LTIN

\*READ ONLY STORAGE

\$BKPM0P DATA   15200000B

\$STLM0P DATA   15600000B

END

BREAK POINT M0P  
STATEMENT LABEL M0P



SYM. PG.LN. IDENT.

10RTW0	2	28	1L6	1STIAD	2	8	1L6	1WD0BL	3	2	1L6	240SYS	35	3	2L6
2NDPAS	10	38	2L6	6BIT0P	36	5	2L6	8BIT0P	35	8	2L6	ABSL0P	35	13	2L6
AB	1	17	2L6	ABSBIN	29	6	2L6	ABSCNT	2	9	1L6	ALLD0L	34	20	2L6
ADJCOM	16	30	2L6	ADRMASK	35	32	2L6	ALBADR	36	4	2L6	ARPBIN	27	32	2L6
ALPRIN	21	28	2L6	APEND	36	13	2L6	ARFXCK	35	34	2L6	BAD0P	3	38	2L6
ASSM	27	22	2L6	AX	1	21	2L6	AYTBST	1	26	1L6	BLKSTL	1	19	1L6
BEQ	1	25	2L6	BFWADR	2	37	1L6	BLKL0P	35	19	2L6	B0TLT	1	20	1L6
BNE	1	24	2L6	B0TLT	1	22	1L6	B0TLT1	1	21	1L6	BX	1	18	2L6
BREAK	1	36	1L6	BRUM0P	35	1	2L6	BUFCNT	2	38	1L6	CESYM	4	14	1L6
CA	1	15	2L6	CALQT	34	12	2L6	CB	1	16	2L6	CHKSUM	2	39	1L6
CESYMP	4	13	1L6	CESYMS	4	12	1L6	CHAR	3	17	1L6	CNVHWD	2	27	1L6
CKALPG	21	12	2L6	CKPRAD	19	24	2L6	CNVHSW	2	29	1L6	C0UNT	2	22	1L6
C0DEST	36	12	2L6	C0NV	13	39	2L6	C0RENT	2	36	1L6	DEC	1	26	2L6
CPEND	36	14	2L6	CTRWD	28	13	2L6	CURWRD	3	10	1L6	E0ADR	1	10	1L6
DELL0P	35	14	2L6	DUMKEL	36	3	2L6	DUMTKY	36	1	2L6	EADR1	1	13	1L6
EOIND	1	12	1L6	EOSIZE	1	14	1L6	EOTAG	1	11	1L6	ENDLD	24	9	2L6
EAXTAG	35	7	2L6	EIGHT	34	39	2L6	ENDABS	2	25	1L6	E0SUBF	27	18	2L6
ENT	2	2	1L6	E0FWRD	34	24	2L6	E0LWRD	34	25	2L6	ERRQT	34	17	2L6
ERR1	4	34	2L6	ERR2	27	1	2L6	ERRPT	25	38	2L6	FLAG	1	30	1L6
FCL	1	12	2L6	FCR	1	13	2L6	F0DBUG	3	22	1L6	F0RK	4	35	2L6
FLSPND	1	29	1L6	FLSPST	1	28	1L6	FNBQT	34	10	2L6	FR0MQT	34	9	2L6
F0RK1	4	39	2L6	F0RK10	4	36	2L6	F0RL	1	22	2L6	FTBLE1	34	26	2L6
FRUB	3	15	1L6	FSTBRK	3	21	1L6	FTBLE	34	27	2L6	HEADST	2	16	1L6
FXSKEY	35	33	2L6	FXSPST	1	27	1L6	G0RUN	3	27	1L6	ILLF1	26	21	2L6
HITCNT	2	34	1L6	H0LDLX	1	39	1L6	ICTP0P	33	13	2L6	ILLTAP	26	18	2L6
ILLF2	26	24	2L6	ILLFUT	34	13	2L6	ILLQT	34	5	2L6	ILQT3	34	16	2L6
ILLTP1	26	19	2L6	ILQT1	34	14	2L6	ILQT2	34	15	2L6	L1	2	8	2L6
INCBRK	17	38	2L6	INDBIT	36	8	2L6	INFLE	3	16	1L6	L112	24	25	2L6
L10	6	19	2L6	L11	6	22	2L6	L111	24	29	2L6	L13	6	28	2L6
L12	6	23	2L6	L125	27	37	2L6	L129	30	3	2L6	L133	29	7	2L6
L130	28	27	2L6	L131	30	9	2L6	L132	29	29	2L6	L15	7	17	2L6
L134	29	9	2L6	L135	30	12	2L6	L14	6	38	2L6	L162	30	36	2L6
L16	7	20	2L6	L160	30	18	2L6	L161	30	19	2L6	L172	31	14	2L6
L17	7	23	2L6	L170	31	6	2L6	L171	31	7	2L6	L183	32	1	2L6
L173	31	26	2L6	L18	7	27	2L6	L180	31	30	2L6				

L19	8	8	2L6	L2	2	16	2L6	L20	8	15	2L6	L200	25	15	2L6
L205	25	17	2L6	L207	25	20	2L6	L21	8	19	2L6	L210	25	26	2L6
L22	8	24	2L6	L220	25	7	2L6	L23	8	27	2L6	L24	8	39	2L6
L25	9	2	2L6	L250	24	39	2L6	L26	9	5	2L6	L27	9	6	2L6
L28	9	7	2L6	L28A	9	9	2L6	L29	11	15	2L6	L29E	11	23	2L6
L3	3	11	2L6	L30	11	25	2L6	L31	11	29	2L6	L32	11	32	2L6
L33	12	2	2L6	L34	12	5	2L6	L35	12	7	2L6	L36	12	11	2L6
L36B	12	15	2L6	L36L	12	23	2L6	L36X	13	12	2L6	L36Y	13	2	2L6
L37	14	4	2L6	L37A	14	14	2L6	L38	14	20	2L6	L39	14	25	2L6
L4	5	20	2L6	L40	14	31	2L6	L41	15	11	2L6	L42	15	16	2L6
L43	15	33	2L6	L44	15	21	2L6	L45	16	5	2L6	L46	16	7	2L6
L47	16	11	2L6	L48	16	19	2L6	L49	16	35	2L6	L5	5	22	2L6
L50	16	36	2L6	L51	17	11	2L6	L52	17	15	2L6	L53	17	19	2L6
L54	17	22	2L6	L55	17	24	2L6	L56	17	30	2L6	L57	18	10	2L6
L58	18	15	2L6	L59	18	25	2L6	L6	5	32	2L6	L60	18	37	2L6
L61	18	38	2L6	L62	19	19	2L6	L63	19	21	2L6	L64	19	38	2L6
L65	20	1	2L6	L66	20	12	2L6	L67	20	14	2L6	L68	20	15	2L6
L69	20	21	2L6	L7	5	36	2L6	L70	20	30	2L6	L71	21	5	2L6
L72	21	15	2L6	L73	21	21	2L6	L74	22	21	2L6	L74A	22	20	2L6
L74B	21	30	2L6	L74C	22	5	2L6	L74D	22	14	2L6	L74G	22	16	2L6
L75	22	26	2L6	L76	22	32	2L6	L78	23	4	2L6	L79	23	8	2L6
L7L	6	5	2L6	L8	6	14	2L6	L81	24	38	2L6	L83	25	2	2L6
L84	25	14	2L6	L85	25	9	2L6	L8L	6	12	2L6	LABELX	2	7	1L6
LACB	3	34	1L6	LADJCO	4	5	1L6	LASTWD	3	1	1L6	LBLKEY	36	2	2L6
L6LL0P	35	12	2L6	LBXLES	2	12	1L6	LCKPRA	4	9	1L6	LC0NV	4	3	1L6
LDSYS	24	7	2L6	LILLF1	3	37	1L6	LILLTA	3	38	1L6	LINCBR	3	39	1L6
LINKX	1	33	1L6	LMPQT	34	7	2L6	LNKARK	35	38	2L6	LNKPTR	2	13	1L6
LNKXMN	1	18	1L6	LNXWDP	4	2	1L6	LNXW0R	3	35	1L6	L0C	3	11	1L6
L0CCNT	2	11	1L6	L0SUBF	4	11	1L6	L0WC0M	1	35	1L6	LPVECL	4	10	1L6
LREPAC	4	6	1L6	LSERHL	3	36	1L6	LSRCHR	4	8	1L6	LSRCHS	4	7	1L6
LSTBRK	4	1	1L6	LSUBQT	34	8	2L6	LTRAVE	4	4	1L6	MABSLS	35	5	2L6
MANMIS	26	30	2L6	MAPTYP	35	18	2L6	MEMQT	34	6	2L6	MEMSIZ	3	6	1L6
MIFMPI	1	34	1L6	MIFPRR	1	38	1L6	MIRMTE	3	7	1L6	MISSQT	34	19	2L6
MLKKTS	3	3	1L6	M0VE	1	28	2L6	M0VE1	1	29	2L6	M0VE2	1	38	2L6
M0VE2	1	35	2L6	MPRHST	1	17	1L6	MPRTYP	35	10	2L6	MSEVEN	34	29	2L6
MSGTRM	34	18	2L6	MSIX	34	35	2L6	MS0PSZ	2	26	1L6	MTWELV	34	36	2L6
MULSR	3	4	1L6	NAME1	2	3	1L6	NAME2	2	4	1L6	NAME3	2	5	1L6
NDPR0G	24	14	2L6	NEWS	4	32	2L6	NEWSQT	34	21	2L6	NINDBT	36	11	2L6

NLSTBL	36	7	2L6
NXWORD	24	37	2L6
ONES	34	30	2L6
OSUBF	26	38	2L6
PBPD	29	23	2L6
PRCLNK	33	23	2L6
PRQLNK	33	33	2L6
PS	2	4	2L6
RDENDP	24	21	2L6
RELWCT	3	9	1L6
REPCK1	32	19	2L6
RUB	4	6	2L6
RWORD	1	31	1L6
SIGN	34	37	2L6
SOS	3	25	1L6
STDUM	2	15	1L6
STLMOP	35	16	2L6
STUP1	23	36	2L6
SYMPT1	3	23	1L6
TABLCT	2	32	1L6
TBLE6	3	19	1L6
THREE	34	34	2L6
TMPLKY	35	39	2L6
TSRLB	3	29	1L6
TW0	34	33	2L6
VECADR	2	33	1L6
XA	1	20	2L6
XSLBKY	35	37	2L6
ZER0	34	31	2L6

N0PMSK	35	11	2L6
0CT	1	27	2L6
0PMASK	35	9	2L6
P2	3	28	2L6
PR2BIG	25	37	2L6
PRCP0P	33	24	2L6
PRQP0P	33	34	2L6
PVECLT	22	39	2L6
RDTAIL	9	14	2L6
RELWRD	3	8	1L6
REQBIT	35	31	2L6
RUB10	4	28	2L6
SBRMIN	35	17	2L6
SIGNBT	34	38	2L6
SRCHRP	20	7	2L6
STFLC0	1	32	1L6
STNXPR	2	23	1L6
STUPE0	23	15	2L6
SYMPT2	3	24	1L6
TAGBIT	36	6	2L6
TEMP1	2	10	1L6
TMP1	3	12	1L6
T0PMEM	34	28	2L6
TSRLB	3	28	1L6
UNDLAD	35	4	2L6
VECCNT	2	35	1L6
XB	1	19	2L6
XSLCKY	35	36	2L6

NSBRML	3	20	1L6
0FL0QT	34	11	2L6
0PTAB1	35	25	2L6
P3	3	26	2L6
PRC	1	10	2L6
PRL0C	3	39	2L6
PRSTRT	2	1	1L6
R14BIN	29	15	2L6
RDTEXT	7	13	2L6
REPACK	32	17	2L6
RNXPLK	5	12	2L6
RUB3	4	20	2L6
SCALTB	2	31	1L6
SKIP	28	37	2L6
SRCHST	17	2	2L6
STFSCA	2	21	1L6
STSCAL	2	30	1L6
STXM0P	35	6	2L6
SYSL0P	35	15	2L6
TAGP0P	36	10	2L6
TEMP2	1	23	1L6
TMP2	3	13	1L6
TPMPLK	3	5	1L6
TSRLB1	3	30	1L6
VADR	2	18	1L6
WAIT	3	31	2L6
XBC	1	14	2L6
Z1	27	2	2L6

NXWDP2	13	28	2L6
0NE	34	32	2L6
0PTAB2	35	30	2L6
P5	4	12	2L6
PRC1	33	27	2L6
PRQ	1	11	2L6
PRTYPE	1	37	1L6
R24BIN	29	3	2L6
RELAB	36	22	2L6
REPCK1	33	6	2L6
RS	2	5	2L6
RUNT0P	2	24	1L6
SERHLT	19	6	2L6
SKIP1	29	1	2L6
STBRK	18	9	2L6
STFXC0	2	14	1L6
STTEMP	2	17	1L6
STXSCA	2	19	1L6
SYSSUB	35	2	2L6
TBLE	3	18	1L6
TEMP3	1	24	1L6
TMP3	3	14	1L6
TRAVEC	18	21	2L6
TVSTRT	1	25	1L6
VC0UNT	2	20	1L6
WORD	2	6	1L6
XLC0KY	35	35	2L6
Z2	27	4	2L6

SYM. PG.LN. IDENT.

6BIT	27	13	2R6	6BIT	2	26	4R6	9BIT	27	14	2R6	9BIT	41	25	3R6
9BIT	2	27	4R6	A00	7	5	3R6	AB	1	16	2R6	ACB1	9	7	2R6
AB	1	13	3R6	AB	1	14	4R6	ACB	8	35	2R6	ACBMSK	27	11	2R6
ACBBIT	5	18	1R6	ACBCHR	5	19	1R6	ACBLIM	27	9	2R6	ADRLP1	38	11	3R6
ACBOUT	9	16	2R6	ACBSCT	27	10	2R6	ADRLPO	38	10	3R6	ADS00	16	4	3R6
ADRLPL	6	35	1R6	ADRLPS	6	37	1R6	ADRMSK	40	4	3R6	ADS06	16	33	3R6
ADS02	16	13	3R6	ADS04	16	24	3R6	ADS05	16	30	3R6	ADS13	17	11	3R6
ADS08	17	2	3R6	ADS10	17	8	3R6	ADS12	17	10	3R6	ADS22	17	39	3R6
ADS15	17	16	3R6	ADS17	17	21	3R6	ADS20	17	29	3R6	ADS30	18	7	3R6
ADS24	18	2	3R6	ADS25	18	3	3R6	ADS26	18	4	3R6	ADS36	18	18	3R6
ADS31	18	9	3R6	ADS33	18	14	3R6	ADS34	18	16	3R6	AGTTP1	4	1	1R6
ADS40	18	25	3R6	AGFP0P	21	31	2R6	AGTP0P	5	22	3R6	AP0PCD	40	6	3R6
AGXP0P	21	34	2R6	ALFP0P	21	20	2R6	ALXP0P	21	23	2R6	ARD05	19	26	3R6
ARCOO	18	32	3R6	ARC15	38	7	3R6	ARD00	19	16	3R6	ARGSGN	5	6	1R6
ARD08	19	38	3R6	ARGADR	5	7	1R6	ARGM	20	15	2R6	AX	1	19	4R6
ARRAY	5	20	1R6	AX	1	21	2R6	AX	1	18	3R6	AXC	2	13	2R6
AXBA	2	15	2R6	AXBA	2	14	3R6	AXBA	2	15	4R6	AXCE	2	13	3R6
AXC	2	12	3R6	AXC	2	13	4R6	AXCE	2	14	2R6	B9	27	16	2R6
AXCE	2	14	4R6	B1	6	19	4R6	B6	27	15	2R6	BAD	4	34	2R6
BA	1	17	2R6	BA	1	14	3R6	BA	1	15	4R6	BEQ	1	37	2R6
BAD0P	4	12	2R6	BE	4	38	1R6	BEAKQT	27	5	2R6	BFD05	20	16	3R6
BEQ	1	36	3R6	BEQ	1	37	4R6	BFD00	20	6	3R6	BFS	1	34	3R6
BFD07	20	24	3R6	BFD08	20	28	3R6	BFD10	20	35	3R6	BK2ADR	7	10	1R6
BFS	1	35	4R6	BK0ADR	7	8	1R6	BK1ADR	7	9	1R6	BKP5	9	38	2R6
BK3ADR	7	11	1R6	BK4ADR	7	12	1R6	BKP10	10	9	2R6	BNE	1	38	2R6
BKPM0P	27	11	4R6	BKPP0P	9	29	2R6	BLNK	40	9	3R6	BRKRP	8	4	1R6
BNE	1	37	3R6	BNE	1	38	4R6	BRKRA	8	5	1R6	BTR	1	35	3R6
BRKRX	8	6	1R6	BRTBL	25	11	3R6	BRUM0P	4	1	3R6	BX	1	16	4R6
BTR	1	36	4R6	BX	1	18	2R6	BX	1	15	3R6	BXC	2	16	2R6
BXAC	2	17	2R6	BXAC	2	16	3R6	BXAC	2	17	4R6	BXC	2	16	4R6
BXC	1	11	2R6	BXC	2	15	3R6	BXC	1	8	3R6	CA	1	13	2R6
BXC	1	9	4R6	C2	5	10	4R6	C3	5	9	4R6	CARRET	40	7	3R6
CA	1	10	3R6	CA	1	11	4R6	CARRET	28	11	2R6	CC	4	39	1R6
CB	1	14	2R6	CB	1	11	3R6	CB	1	12	4R6	CDB06	21	18	3R6
CCHR	5	27	1R6	CDB00	21	5	3R6	CDB04	21	14	3R6				

CGERR	3	28	4R6
CHRTBL	36	18	3R6
CLAFLE	7	14	2R6
C8G84	3	27	4R6
CQU8TE	5	28	2R6
CS1	5	4	2R6
CSSW3	14	29	4R6
D1	5	32	4R6
D100	10	22	4R6
D110	11	3	4R6
D120	11	11	4R6
D130	11	21	4R6
D2	5	35	4R6
D230	14	35	4R6
D260	15	18	4R6
D2B10	22	33	3R6
D2B16	23	22	3R6
D2B22	23	36	3R6
D2B27	24	26	3R6
D2B33	24	37	3R6
D2B47	26	8	3R6
D2B60	26	27	3R6
D2B65	26	36	3R6
D39	37	29	3R6
D410	16	8	4R6
D460	17	11	4R6
D500	17	21	4R6
D530	18	4	4R6
D58	8	1	4R6
D62	8	12	4R6
D625	19	3	4R6
D70	8	24	4R6
D740	19	26	4R6
D80	8	39	4R6
D83885	37	34	3R6
D9	41	3	3R6
D925	20	4	4R6
D950	20	39	4R6

CHAR	5	26	1R6
CHSIZ	4	18	1R6
CLFLE	6	38	2R6
C9MPG8	3	20	4R6
CRLF	41	29	3R6
CS5	5	5	2R6
CSSW4	14	30	4R6
D10	41	4	3R6
D1000	21	20	4R6
D1100	22	3	4R6
D1200	22	23	4R6
D1300	23	31	4R6
D200	12	23	4R6
D24	37	28	3R6
D2B00	21	23	3R6
D2B12	22	38	3R6
D2B17	23	27	3R6
D2B24	24	7	3R6
D2B28	24	29	3R6
D2B35	25	1	3R6
D2B48	26	11	3R6
D2B61	26	29	3R6
D2B66	26	39	3R6
D4	40	38	3R6
D420	16	31	4R6
D47	41	6	3R6
D510	17	30	4R6
D55	7	23	4R6
D6	41	1	3R6
D620	18	14	4R6
D63	8	13	4R6
D700	19	18	4R6
D75	8	29	4R6
D800	20	17	4R6
D85	9	16	4R6
D90	9	37	4R6
D94	10	7	4R6
D955	21	2	4R6

CHCNT	4	37	1R6
CIT	1	4	4R6
CMA	36	26	3R6
C8NTQT	27	3	2R6
CRSET	14	16	4R6
CSET	14	14	4R6
CURFLE	6	11	1R6
D10	6	34	4R6
D105	10	26	4R6
D1120	22	2	4R6
D125	11	15	4R6
D150	11	38	4R6
D210	12	25	4R6
D240	14	38	4R6
D2B05	22	6	3R6
D2B13	23	1	3R6
D2B19	23	30	3R6
D2B25	24	12	3R6
D2B30	24	31	3R6
D2B45	25	33	3R6
D2B50	26	16	3R6
D2B62	26	31	3R6
D2B68	27	2	3R6
D400	16	3	4R6
D450	16	39	4R6
D5	40	39	3R6
D511	37	33	3R6
D56	7	34	4R6
D60	8	10	4R6
D621	18	21	4R6
D65	8	17	4R6
D710	19	23	4R6
D77	8	37	4R6
D805	20	21	4R6
D86	9	26	4R6
D92	10	2	4R6
D940	21	11	4R6
D960	21	12	4R6

CHMSK	40	12	3R6
CLAFLE5	7	16	2R6
CMMDQT	27	2	2R6
CQ2	5	29	2R6
CS	4	36	2R6
CSSW	14	17	4R6
D	5	1	1R6
D100	37	32	3R6
D109	11	2	4R6
D1130	22	14	4R6
D13	37	27	3R6
D2	40	34	3R6
D23	41	5	3R6
D250	15	4	4R6
D2B06	22	8	3R6
D2B15	23	13	3R6
D2B21	23	34	3R6
D2B26	24	15	3R6
D2B31	24	34	3R6
D2B46	25	29	3R6
D2B52	26	19	3R6
D2B63	26	33	3R6
D3	40	37	3R6
D405	16	9	4R6
D455	17	2	4R6
D50	7	6	4R6
D53	7	18	4R6
D57	8	5	4R6
D610	18	5	4R6
D622	18	23	4R6
D7	41	2	3R6
D730	19	25	4R6
D8	37	25	3R6
D810	20	24	4R6
D87	9	30	4R6
D920	20	2	4R6
D95	10	13	4R6
D97	10	16	4R6

D970 21 15 4R6  
DBLXRG 21 13 2R6  
DC 6 28 1R6  
DD255 41 13 3R6  
DD8TR 7 18 1R6  
DECERR 25 1 2R6  
DECL40 24 36 2R6  
DECL9 24 20 2R6  
DL 36 27 3R6  
DL10 15 8 3R6  
DM20 38 1 3R6  
DM7 37 37 3R6  
DNMCHR 7 19 1R6  
DNUMB 7 20 1R6  
D8FXIT 4 29 1R6  
DPAP8P 22 30 2R6  
DS 5 13 1R6  
DTMP2 7 26 1R6  
E00 7 11 3R6  
EOSIZT 4 30 1R6  
E20 8 12 3R6  
E3 3 11 4R6  
EG8T8 23 29 4R6  
END03 25 28 2R6  
E8R01 15 18 3R6  
EP 5 14 1R6  
ERLSZE 5 28 1R6  
ERR6 6 19 2R6  
ERRFCH 38 8 3R6  
ERRQT2 41 35 3R6  
F00 8 36 3R6  
F3 4 24 4R6  
FCL 1 9 2R6  
FCR 1 10 2R6  
FDEBUG 3 26 1R6  
FE8R 3 20 1R6  
FFL8AT 7 13 1R6  
FIXLNK 5 22 1R6

D99 37 31 3R6  
DBN 7 16 1R6  
DCL8S5 13 5 4R6  
DD255 41 12 3R6  
DE 5 2 1R6  
DECL10 24 34 2R6  
DECL5 24 22 2R6  
DECLFL 23 15 2R6  
DL00 14 28 3R6  
DM100 38 2 3R6  
DM257 41 22 3R6  
DM8 37 38 3R6  
DNS00 27 36 3R6  
D8FP8P 19 18 2R6  
D8UT 13 10 4R6  
DPMP8P 15 36 2R6  
DTE8L 6 38 4R6  
DVBASE 7 27 1R6  
EOADR 1 10 1R6  
EOTAG 1 11 1R6  
E22 8 14 3R6  
EADR 4 23 1R6  
ELEMENTS 5 24 1R6  
ENDDMY 3 7 4R6  
E8R03 15 20 3R6  
ERLLIM 5 32 1R6  
ERLTBL 5 30 1R6  
ERR7 6 21 2R6  
ERR8CT 38 9 3R6  
EWG 8 1 1R6  
F1 4 21 4R6  
F4 4 27 4R6  
FCL 1 6 3R6  
FCR 1 7 3R6  
FDL00 2 34 3R6  
FERR 3 19 1R6  
FI8 2 19 4R6  
FIXTMP 4 2 1R6

DADRIN 7 15 1R6  
DBUF 4 32 1R6  
DCL8SE 13 1 4R6  
DD39 37 30 3R6  
DEBUG 6 32 4R6  
DECL3 24 5 2R6  
DECL50 24 35 2R6  
DIN 13 32 4R6  
DL02 14 33 3R6  
DM12 37 39 3R6  
DM3 37 36 3R6  
DNAME1 7 21 1R6  
DNS02 28 1 3R6  
D8FTAD 4 28 1R6  
D8XP8P 19 3 2R6  
DPSP8P 22 39 2R6  
DTE8L2 6 39 4R6  
DW 5 3 1R6  
EOIND 1 12 1R6  
E1 3 12 4R6  
E23 8 16 3R6  
EADR1 1 13 1R6  
EMASK 41 17 3R6  
ENDI8L 25 22 2R6  
E8R07 15 31 3R6  
ERLLIM 5 31 1R6  
ERR1 6 10 2R6  
ERR8 6 31 2R6  
ERR8R 5 38 2R6  
EXIT 4 31 1R6  
F10 9 21 3R6  
FARF 3 18 1R6  
FCL 1 7 4R6  
FCR 1 8 4R6  
FDT00 28 36 3R6  
FFA 2 21 4R6  
FI8DIR 3 22 1R6  
FIXTP2 4 8 1R6

DBAD 9 13 4R6  
DBUF1 4 33 1R6  
DCS 6 31 4R6  
DD8TA 7 17 1R6  
DEC24 6 23 4R6  
DECL4 24 12 2R6  
DECL8 24 16 2R6  
DIV8F 12 30 2R6  
DL05 15 2 3R6  
DM2 37 35 3R6  
DM4 41 7 3R6  
DNAME2 7 22 1R6  
DNS05 28 12 3R6  
D8FTEM 4 27 1R6  
D8XTEM 4 26 1R6  
DPTMP1 4 12 1R6  
DTMP1 7 25 1R6  
E 36 19 3R6  
EOSIZE 1 14 1R6  
E10 7 30 3R6  
E24 8 29 3R6  
EADR2 4 24 1R6  
END01 25 29 2R6  
E8R00 15 17 3R6  
E8R10 15 34 3R6  
ERLPT 5 29 1R6  
ERR3 6 16 2R6  
ERRC8N. 1 21 1R6  
ERRQT1 41 34 3R6  
EXPSGN 6 22 1R6  
F2 4 18 4R6  
FC 6 21 1R6  
FCPNTR 5 9 1R6  
FCS00 28 24 3R6  
FE8F 3 21 1R6  
FFAP8P 20 20 2R6  
FI8P8P 3 3 3R6  
FLA 2 7 2R6

FLA	2	6	3R6
FLAF	11	24	2R6
FLA0F	11	33	2R6
FLAP0P	10	15	2R6
FLD	2	10	2R6
FLIND	40	10	3R6
FLMP0P	13	8	2R6
FLNB	15	26	2R6
FL0ATA	17	33	2R6
FLSP0P	10	19	2R6
FNAME	3	23	1R6
FNCTR	6	8	1R6
F0RL	1	5	2R6
FSDP0P	20	2	2R6
FTAP0P	18	3	2R6
FTS0RD	18	29	2R6
FXBAXR	4	5	1R6
H00	9	30	3R6
H11	10	7	3R6
I08	10	28	3R6
ICT	2	3	3R6
ICW02	30	37	3R6
ICW20	31	11	3R6
IF	5	15	1R6
IFS05	3	37	3R6
IFS13	4	15	3R6
IFS20	4	32	3R6
IFS30	5	2	3R6
IFS332	5	19	3R6
ILLUN0	26	28	2R6
I0E25	6	2	3R6
I0LUSA	4	6	4R6
ISS01	32	11	3R6
ISS12	33	1	3R6
KG0T0	7	36	1R6
LADS00	2	19	1R6
LBFD00	2	22	1R6
LCMP00	1	35	1R6

FLA	2	7	4R6
FLAG	4	11	1R6
FLA0FS	13	6	2R6
FLAX	11	35	2R6
FLD	2	9	3R6
FLM	2	9	2R6
FLMX	15	27	2R6
FLNMT	5	38	1R6
FLS	2	8	2R6
FLTIND	1	15	1R6
FNAP0P	20	32	2R6
FNEXT	7	14	1R6
FP0NE	1	25	1R6
FSSW	3	25	1R6
FTBLE	27	12	2R6
FTSP0P	18	21	2R6
FXBNDA	17	6	2R6
H03	9	33	3R6
I00	10	15	3R6
ICD00	30	2	3R6
ICT	2	4	4R6
ICW10	31	1	3R6
ICW30	31	14	3R6
IF0VFL	3	34	4R6
IFS06	3	39	3R6
IFS14	4	16	3R6
IFS21	4	33	3R6
IFS31	5	11	3R6
IFSNLT	5	16	4R6
IMT00	31	30	3R6
I0E30	6	11	3R6
I0TEND	6	1	1R6
ISS02	32	13	3R6
ISS15	33	4	3R6
KG0T01	7	37	1R6
LARC00	2	20	1R6
LCDB00	2	23	1R6
LD130	1	36	1R6

FLAC	11	12	2R6
FLAGM	12	2	2R6
FLA0K	11	34	2R6
FLAXRG	4	4	1R6
FLD	2	10	4R6
FLM	2	8	3R6
FLN	2	22	4R6
FLNP0P	15	8	2R6
FLS	2	7	3R6
FLT0NK	5	23	1R6
FNBUF	6	6	1R6
FNPTR	6	4	1R6
FRUB	3	24	1R6
FSTP0P	17	3	2R6
FTDP0P	18	25	2R6
FW	5	4	1R6
GCD	1	8	2R6
H06	10	2	3R6
I03	10	18	3R6
ICF00	30	13	3R6
ICU00	31	26	3R6
ICW14	31	9	3R6
ICW32	31	16	3R6
IFP0NE	38	14	3R6
IFS08	4	6	3R6
IFS16	4	26	3R6
IFS22	4	37	3R6
IFS32	5	14	3R6
IFSN0W	4	36	4R6
INST0T	27	6	2R6
I0E31	6	12	3R6
I0TEND	5	39	1R6
ISS09	32	31	3R6
IZ	5	16	1R6
KG0T02	7	38	1R6
LARD00	2	21	1R6
LCLAFL	6	24	1R6
LDBLXR	3	6	1R6

FLAD	12	9	2R6
FLANZ	11	31	2R6
FLA0VT	11	25	2R6
FLC0M	15	20	2R6
FLDP0P	14	9	2R6
FLM	2	9	4R6
FLNA	15	29	2R6
FLNPTR	6	3	1R6
FLS	2	8	4R6
FMT00	29	17	3R6
FNBUF3	6	7	1R6
FNPTR2	6	5	1R6
FRUN	3	27	1R6
FT	6	29	1R6
FTMP0P	18	12	2R6
FWPNTR	5	10	1R6
H	36	31	3R6
H10	10	5	3R6
I05	10	23	3R6
ICF02	30	22	3R6
ICW00	30	34	3R6
ICW15	31	10	3R6
ICW35	31	22	3R6
IFS00	3	28	3R6
IFS10	4	10	3R6
IFS17	4	28	3R6
IFS24	4	34	3R6
IFS330	5	6	3R6
ILL0T	41	33	3R6
I0E00	5	26	3R6
I0E38	6	38	3R6
I0T0ZE	6	2	1R6
ISS10	32	33	3R6
JMPTBL	6	17	3R6
LACB	1	33	1R6
LB	36	29	3R6
LCLFLE	1	34	1R6
LDC0MP	41	30	3R6

LDECLF	1	37	1R6
LDPPBP	22	8	2R6
LFCS00	2	38	1R6
LFLD00	2	4	1R6
LFP	2	10	3R6
LGR	2	18	2R6
LIB4	3	11	1R6
LICD00	2	27	1R6
LIFSNL	2	7	1R6
LINFO0	2	36	1R6
LI0LSA	2	10	1R6
LNDDMY	2	17	1R6
L0CLB4	25	28	4R6
L0CST1	24	15	4R6
L0CST5	24	31	4R6
L0CST9	25	2	4R6
LP00	12	39	3R6
LPSTK	7	1	1R6
LREL1	25	34	4R6
LREL19	26	18	4R6
LREL8	25	38	4R6
LSETT	3	1	1R6
LSTRTD	3	2	1R6
LTXPBP	16	23	2R6
M02400	38	6	3R6
M0400	41	14	3R6
M0777	41	18	3R6
MAXP0S	41	10	3R6
ML0NES	41	21	3R6
M0VE1	1	23	2R6
M0VE2	1	29	2R6
M0VE2	1	27	4R6
MPR06	33	26	3R6
MRTNAD	5	8	1R6
MSCNT	6	20	1R6
NEGBEX	41	15	3R6
NSUBSC	7	28	1R6
010	11	4	3R6

LDNS00	2	24	1R6
LECH0	1	38	1R6
LFDL00	2	2	1R6
LFL0AT	2	5	1R6
LFP	2	11	4R6
LIB1	3	8	1R6
LIB5	3	12	1R6
LICF00	2	28	1R6
LIFSNW	2	8	1R6
LINKTE	28	7	2R6
LI0SUB	2	16	1R6
L030	2	18	1R6
L0CLB5	25	30	4R6
L0CST2	24	17	4R6
L0CST6	24	33	4R6
L0CSTA	24	6	4R6
LP02	13	6	3R6
LPSTK3	7	2	1R6
LREL10	26	14	4R6
LREL20	26	19	4R6
LRIT00	3	3	1R6
LSHTBL	37	8	3R6
LTFPBP	17	29	2R6
LWCS00	2	37	1R6
M0377	41	11	3R6
M04000	38	4	3R6
M257D	41	24	3R6
MAXP0S	41	9	3R6
M0VE	1	22	2R6
M0VE1	1	20	3R6
M0VE2	1	29	3R6
MPR00	33	14	3R6
MPR10	33	28	3R6
MS	36	33	3R6
MSK3C	40	17	3R6
N0PM0P	37	10	3R6
NXTCHR	6	30	1R6
015	11	12	3R6

LDPNT	6	12	1R6
LERMSG	2	1	1R6
LFDT00	2	25	1R6
LFMT00	2	26	1R6
LFTS0R	3	5	1R6
LIB2	3	9	1R6
LIB6	3	13	1R6
LICU00	2	29	1R6
LIFVFL	2	9	1R6
LINKTL	28	8	2R6
LL0CST	2	11	1R6
L0CLB1	25	12	4R6
L0CLB9	25	19	4R6
L0CST3	24	24	4R6
L0CST7	24	38	4R6
L0CTMP	7	23	1R6
LPAUSE	2	15	1R6
LPSTK4	7	3	1R6
LREL11	26	10	4R6
LREL21	26	28	4R6
LRNF00	3	4	1R6
LSTCHR	6	23	1R6
LTIN	2	34	1R6
LW0T00	2	39	1R6
M03777	41	8	3R6
M040L	38	5	3R6
M257D	41	23	3R6
MEMQT	41	32	3R6
M0VE	1	19	3R6
M0VE1	1	21	4R6
M0VE2	1	26	3R6
MPR04	33	16	3R6
MPRX	5	25	1R6
MS00	12	10	3R6
NC	6	26	1R6
NSTLBL	2	19	2R6
000	10	33	3R6
030	11	16	3R6

LDPNT3	6	13	1R6
LEKR0R	1	39	1R6
LFIX	2	3	1R6
LFP	2	11	2R6
LFXBND	2	6	1R6
LIB3	3	10	1R6
LIB7	3	14	1R6
LICW00	2	31	1R6
LIMT00	2	32	1R6
LINKTS	27	19	2R6
LMPRO0	2	30	1R6
L0CLB3	25	15	4R6
L0CLBL	25	8	4R6
L0CST4	24	26	4R6
L0CST8	24	16	4R6
LP	36	36	3R6
LP0WER	2	14	1R6
LRBI00	2	33	1R6
LREL12	26	6	4R6
LREL30	26	31	4R6
LSENST	2	12	1R6
LST0P	2	13	1R6
LTTI00	2	35	1R6
M01777	40	18	3R6
M040	38	3	3R6
M077	40	11	3R6
M770	40	15	3R6
MINUS	40	2	3R6
M0VE	1	20	4R6
M0VE2	1	32	2R6
M0VE2	1	30	4R6
MPR05	33	24	3R6
MRSTL	7	24	1R6
MSAVEX	4	36	1R6
NEGBEX	41	16	3R6
NSTLVL	6	39	1R6
005	11	1	3R6
032	11	19	3R6



035 11 24 3R6  
0BPST 6 14 1R6  
0NES 1 28 1R6  
0VFLID 5 21 1R6  
PART2 4 19 1R6  
PER 36 37 3R6  
P0PLST 38 19 3R6  
PRC 2 5 4R6  
PR0GRS 6 15 1R6  
PRQP0P 8 22 2R6  
PSEQT 41 36 3R6  
Q29 16 21 2R6  
Q35 19 13 2R6  
Q39 20 16 2R6  
Q42 21 24 2R6  
Q5A 10 38 2R6  
Q5E 12 38 2R6  
QTEMP1 3 34 1R6  
READF 25 6 2R6  
RIF14 36 9 3R6  
RIF4 35 26 3R6  
RNF00 33 35 3R6  
RNF10 34 14 3R6  
RP05 14 10 3R6  
RS 2 21 2R6  
RUB3 4 33 2R6  
SENSSW 1 17 1R6  
SETI0T 25 37 2R6  
SETIT7 26 22 2R6  
SETT 1 39 4R6  
SHFCNT 4 20 1R6  
SLDPNT 5 16 2R6  
STD 2 11 3R6  
STLM0P 27 12 4R6  
STPST3 7 28 2R6  
STZ 1 36 2R6  
TA 1 18 1R6  
TBLE1 6 17 1R6

037 12 2 3R6  
0FL0 12 33 2R6  
0PMASK 40 16 3R6  
P 5 5 1R6  
PAUG0 8 7 2R6  
PNCTBL 27 19 3R6  
PP 36 32 3R6  
PRCP1 8 17 2R6  
PRQ 2 6 2R6  
PS 2 20 2R6  
PTEMP 3 31 1R6  
Q31 17 25 2R6  
Q36 19 33 2R6  
Q40 20 26 2R6  
Q43 21 35 2R6  
Q5B 10 34 2R6  
Q5F 15 21 2R6  
R00 7 4 3R6  
RETURN 3 7 1R6  
RIF16 36 4 3R6  
RIF5 35 35 3R6  
RNF02 34 2 3R6  
RP 36 35 3R6  
RP06 14 2 3R6  
RSHTBL 37 4 3R6  
SCMAX 7 29 1R6  
SETF 2 1 2R6  
SETIT2 26 15 2R6  
SETIT9 26 26 2R6  
SETT1 26 6 2R6  
SIGN 1 27 1R6  
SPACES 40 8 3R6  
STD 2 12 4R6  
ST0FLK 27 8 2R6  
STPSTP 7 27 2R6  
STZ 1 33 3R6  
TAGBIT 38 15 3R6  
TBLE3 6 18 1R6

0777 41 20 3R6  
0FSET 12 23 2R6  
0T 5 17 1R6  
P00 12 7 3R6  
PAULNK 7 31 2R6  
P0PADR 40 5 3R6  
PRC 2 5 2R6  
PRCP0P 8 14 2R6  
PRQ 2 5 3R6  
PS2 3 39 2R6  
Q2 6 27 1R6  
Q32 17 19 2R6  
Q37 20 3 2R6  
Q41 20 38 2R6  
Q44 22 1 2R6  
Q5C 11 20 2R6  
QB00T 3 38 1R6  
RB 36 28 3R6  
RIF1 35 16 3R6  
RIF20 35 38 3R6  
RIF7 36 6 3R6  
RNF03 34 5 3R6  
RP00 13 25 3R6  
RP08 14 12 3R6  
RTSH23 1 23 1R6  
SENSLT 6 3 4R6  
SETF 1 39 3R6  
SETIT3 26 19 2R6  
SETT 1 39 2R6  
SETT10 26 33 2R6  
SIGN 40 3 3R6  
SPCEQT 41 31 3R6  
STDP0P 22 18 2R6  
ST0PQT 41 37 3R6  
STRTDM 2 39 4R6  
STZ 1 34 4R6  
TB 1 19 1R6  
TBLE6 6 19 1R6

0777 41 19 3R6  
0NE 1 24 1R6  
0VFINO 4 25 1R6  
PART1 4 17 1R6  
PC 6 31 1R6  
P0PEND 39 27 3R6  
PRC 2 4 3R6  
PRL0C 4 13 2R6  
PRQ 2 6 4R6  
PS3 3 38 2R6  
Q28 16 18 2R6  
Q33 17 23 2R6  
Q38 20 8 2R6  
Q41A 21 11 2R6  
Q45 22 4 2R6  
Q5D 12 18 2R6  
QTEMP 3 33 1R6  
RC 6 32 1R6  
RIF13 35 39 3R6  
RIF3 35 21 3R6  
RNDTBL 37 12 3R6  
RNF05 34 8 3R6  
RP03 13 33 3R6  
RP10 14 16 3R6  
RUB 4 19 2R6  
SENSLW 1 16 1R6  
SETF 2 1 4R6  
SETIT5 26 31 2R6  
SETT 1 38 3R6  
SG0T0 7 35 1R6  
SLASH 36 30 3R6  
STD 2 12 2R6  
STKADR 6 33 1R6  
STPLNK 7 25 2R6  
STRTQT 27 4 2R6  
SYSINI 2 24 2R6  
TBLE 6 16 1R6  
TEMP1 3 39 1R6

TEMP2	4	3	1R6
TIN	26	38	4R6
TMPX	4	9	1R6
TW0	40	35	3R6
UDATAS	7	31	1R6
VADRLL	6	36	1R6
WCPNTR	5	11	1R6
WRITF	25	13	2R6
XO2	12	32	3R6
XB	1	19	2R6
XBC	1	9	3R6
XI0	2	18	4R6
XMP	2	20	4R6
XSDGTS	6	25	1R6
ZER0	1	22	1R6

TEMP3	4	7	1R6
TIN10	26	39	4R6
TRUND	3	30	1R6
TX	1	20	1R6
UNDEF	40	25	3R6
VADRLS	6	38	1R6
WCS00	35	7	3R6
WWPNTR	5	12	1R6
XA	1	20	2R6
XB	1	16	3R6
XBC	1	10	4R6
XI0P0P	3	9	3R6
XMPP0P	16	29	2R6
XSDP0P	19	38	2R6
ZM	4	16	1R6

TEMP4	4	10	1R6
TMPA	4	13	1R6
TTI00	34	24	3R6
TXI	4	34	1R6
UPR0GN	7	32	1R6
WAIT	4	4	2R6
WG0T0	7	39	1R6
X	36	25	3R6
XA	1	17	3R6
XB	1	17	4R6
XDVP0P	16	34	2R6
XI0PP7	3	24	3R6
XNAP0P	20	36	2R6
XSTP0P	16	38	2R6

TEMPA	4	21	1R6
TMPIZ	4	6	1R6
TW0	40	36	3R6
TX0	4	35	1R6
UPR0GS	7	30	1R6
WBLIM	40	1	3R6
WR	6	34	1R6
X00	12	30	3R6
XA	1	18	4R6
XBC	1	12	2R6
XFAP0P	20	24	2R6
XM	4	14	1R6
XREG	4	22	1R6
ZE	4	15	1R6