

XVIM

MUMPS VIA

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

/
/
/ COPYRIGHT (C) 1975, 1976
/ DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
/
/ THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY
/ ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH
/ THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. NO OTHER
/ SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PRO-
/ VIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON
/ EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO
/ THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE
/ SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.
/
/ THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
/ WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COM-
/ MITMENT BY DIGITAL EQUIPMENT CORPORATION.
/
/ DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
/ OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
/
/
/ .EJECT

```

24 /
25 /
26 /
27 /
28 /
29 /
30 /
31 /
32 /
33 /
34 /
35 /
36 /
37 /
38 /
39 /
40 /
41 /
42 /
43 /
44 /
45 /
46 /
47 /
48 /
49 /
50 /
51 /
52 /
53 /
54 /
55 /
56 /
57 /
58 /
59 /
60 /
61 /
62 /
63 /
64 /
65 /
66 /
67 /
68 /
69 /
70 /
71 /
72 /
73 /
74 /
75 /

```

X V M / M U M P S
- - - -
VERSION V1A000

COPYRIGHT 1971,1972,1973,1974,1975
BY
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASSACHUSETTS, U.S.A.

EDIT 12-18 10/15/75 JAH

(1) 12/4/74 - MODIFY GLOBAL PUT TO AVOID EXCESS NODE SCRAMBLING &
INSERT KILL ^GLO SYNTAX WHICH DEALLOCATES ALL BLOCKS &
LET %E=-.04 INDICATE NO "THANK YOU" OR ERROR PRINT OUT &
MODIFY XCOM TO ALLOW 777777 TO BE WRITTEN TO CORE BY X_SVE

(2) 1/17/75 - INSERT BUFFER POOL (RP) CODE

(3) 1/20/75 (JAH) CORRECT SS CALCULATIONS

(4) 1/24/75 INSERT BUFFER POOL (RF) CODE

(5) 1/31/75 (JAH) - CORRECT XCOM CORE WRITE CODE

(6) 2/3/75 (JAH) - CORRECT \$A FOLLOWED BY READ & \$K(-777777-) PROBLEM

(7) 2/5/75 (JAH) - ZERO WRITSW ON DISK ERROR WITHIN MUMPS INSTEAD OF HANDLERS

(12-8) 2/25/75 (JAH) - IMPLEMENT RP03 SUPPORT

(12-9) 2/26/75 (JAH) - IMPLEMENT \$Q(-1) SYNTAX

(12-10) 3/20/75 (JAH) - ELIMINATE INTERLOCK BETWEEN PHANTOMS & DSKACT,
MODIFY \$HIGH EMPTY TO BE -1 OR 0

(12-11) 6/20/75 (JAH) - CORRECT GLOBAL REPLACEMENT OF POINTER ONLY NODE

(12-12) 7/24/75 (JAH) SEPARATE MUMPS EXEC & INTER INTO 2 PARTS - 1ST

(12-13) 8/11/75 (JAH) IMPLEMENT \$JT AND \$JC FUNCTIONS

(12-14) 8/20/75 (JAH) ALTER CHARACTER HANDLING ROUTINES

(12-15) 9/4/75 (JAH) INSTALL XVM CODE TO ALLOW PARTITIONS TO 128K

(12-16) 9/12/75 (JAH) INSERT IMMEDIATE READ BUFFER POOL CODE

(12-17) 9/25/75 (JAH) ENHANCE SWAP SPEED, ALLOW MORE THAN READ ALL
ON ONE COMMAND LINE, CORRECT GO TO A PART WITHIN A CALLED PROGRAM,
TAKE %WATCHDOG AND CLOCK HANG CODE OFF INT SIDE, ALLOW EMERGENCY
CLOSE DURING DISK ACTIVITY, PERMIT CNTL U FROM TERMINAL ONLY,
AND CLEAR PROG BUFFER ON PROG SPACE OVERFLOW

(12-18) 10/15/75 CORRECT KEY DETECTION AND FOR-UNTIL SYNTAX

(12-17) 9/22/75 (JAH) SPEED SWAPPING & STRING PROCESSING

/ THIS SYSTEM WAS ORIGINALLY DEVELOPED AT THE LABORATORY
/ OF COMPUTER SCIENCE OF MASSACHUSETTS GENERAL HOSPITAL AND WAS
/ SUPPORTED BY RESEARCH GRANTS FROM THE NATIONAL CENTER FOR
/ HEALTH SERVICES RESEARCH AND DEVELOPMENT (HS 00240) AND FROM
/ THE NATIONAL INSTITUTE OF GENERAL MEDICAL SCIENCES (GM 15287).

.EJECT

```

77 /PARAMETERS FOR CONDITIONAL ASSEMBLY
78 /      ....NORMALLY ENTERED THROUGH .DAT SLOT -10 (SECONDARY INPUT)
79 /      .DEC
80 /THE FOLLOWING PARAMETERS MUST BE ENTERED THROUGH THE SECONDARY
81 /      INPUT ... THERE IS NO DEFAULT CONDITION FOR THEM
82 /      DC01          /NUMBER OF DC01EB'S
83 /      PRNUM        /NUMBER OF PARTITIONS
84 /      CORE         /CORE SIZE IN THOUSANDS (EG 32)
85 /      RP02        /NUMBER OF RP02 DISKS
86 /      RP03      /NUMBER OF RP03 DISKS
87 /      RF15        /NUMBER OF DECDISKS
88 /      DTA         /NUMBER OF DECTAPE TRANSPORTS
89 /      PRTSIZE     /SIZE OF A PARTITION IN WORDS
90 /      NOG         /NUMBER OF GLOBALS
91 /
92 /THE FOLLOWING MAY APPEAR ON THE SECONDARY INPUT, BUT IF
93 /NOT THERE, ARE ASSIGNED A DEFAULT VALUE
94 /
95 /      .IFUND  PFAIL
96 PFAIL=0          /POWER FAILURE OPTION
97 /      .ENDC
98 /
99 /      GARBAGE COLLECTOR PARAMETER TO INHIBIT RUNNING OF GARBAGE
100 /      COLLECTOR IN THE ROUND ROBIN
101 /
102 /      .IFUND  GGRW
103 GGRW=0
104 /      .ENDC
105 /      .IFUND  SUBMERGE
106 SUBMERGE=1
107 /      .ENDC
108 /      .IFUND  RP03          /*12JAH8*
109 RP03=0          /*12JAH8*
110 /      .ENDC          /*12JAH8*
111 /      .IFUND  API
112 API=0
113 /      .ENDC
114 /      .IFNOZ  API
115 INTON=ION
116 INTOFF=IOF
117 /      .ENDC
118 /      .IFUND  XVM          /*12JAH15*
119 XVM=0          /ASSUME NON-XVM SYSTEM          /*12JAH15*
120 /      .ENDC          /*12JAH15*
121 /      .IFUND  DCOUNT
122 DCOUNT=10
123 /      .ENDC
124 /      .IFUND  LP15
125 LP15=0
126 /      .ENDC
127 /      .IFUND  LP15.2
128 LP15.2=0

```

```

129          .ENDC
130          .IFUND  CR03B
131      CR03B=0
132      CRXX=0
133          .ENDC
134          .IFUND  CR15
135      CR15=0
136      CRXX=0
137          .ENDC
138          .IFPNZ  CR03B
139          .IFPNZ  CR15
140      WARNING DO NOT ASSEMBLE WITH CR03B AND CR15 BOTH POSITIVE AND NON-ZERO
141          .END
142          .ENDC
143          .ENDC
144          .IFPNZ  CR03B+CR15
145      CRXX=1
146          .ENDC
147          .IFUND  MONEY
148      MONEY=0          /NO MONEY FUNCTION
149          .ENDC
150          .IFUND  FMONEY
151      FMONEY=0
152          .ENDC
153          .IFPNZ  FMONEY
154      000001 A      MONEY=1
155          .ENDC
156          .IFUND  MTA90N
157      MTA90N=0          /NO 9-TRACK DRIVES
158          .ENDC
159          .IFUND  MTA70N
160      MTA70N=0          /NO 7-TRACK DRIVES
161          .ENDC
162      000001 A      MTA=MTA70N+MTA90N          /TOTAL NUMBER OF MAGTAPE DRIVES
163          .IFUND  CLKHZ
164      CLKHZ=60          /CLOCK FREQUENCY IN HERTZ
165          .ENDC
166          .IFUND  SLICE
167      SLICE=150          /TIME SHARING SLICE IN MILLISECONDS
168          .ENDC
169          .IFUND  WTIMER
170      WTIMER=60          /1 MIN DEFAULT SIGNOFF
171          .ENDC
172          .IFUND  DUMMY
173      DUMMY=0
174          .ENDC
175          .IFUND  TTCHAR
176      TTCHAR=25          /NUMBER OF CHARS IN EACH TT BUFFER
177          .ENDC
178          .IFUND  TTALEM
179      TTALEM=10          /TT BUFFER "ALMOST EMPTY" CHAR COUNT
180          .ENDC

```

```

181          .IFUND  TTALFL
182 TTALFL=5          /TT BUFFER "ALMOST FULL" CHAR COUNT
183          .ENDC
184          .IFUND  DTBSZ
185 DTBSZ=256
186          .ENDC
187          .IFUND  RPOPTD
188 RPOPTD=2          /RP02 PROGRAM FILE OPTIMUM SPACING
189          .ENDC
190          .IFUND  RPOPTG
191 RPOPTG=6          /RP02 GLOBAL FILE OPTIMUM SPACING
192          .ENDC
193          .IFUND  RFOPTD
194 RFOPTD=2          /RF15 PROGRAM FILE OPTIMUM SPACING
195          .ENDC
196          .IFUND  RFOPTG
197 RFOPTG=4          /RF15 GLOBAL FILE OPTIMUM SPACING
198          .ENDC
199          .IFUND  MODIFY
200 MODIFY=1
201          .ENDC
202          .IFUND  V11
203 V11=0            /*12JAH10*
                /*12JAH10* SET POS TO RETAIN OLD SHIGH SEMANTICS
204          .ENDC
                /*12JAH10*
205          .IFUND  PWRITE
                /*12JAH16*
206 PWRITE=0        /IMMEDIATE WRITE TO DISK      /*12JAH16*
207          .ENDC
                /*12JAH16*
208 /THE FOLLOWING MAY NOT BE CHANGED BY SECONDARY INPUT
209 /
210          000400 A  RPSECT=256
211          000100 A  BLOCK=64
212          000003 A  RP=RP03*2+RP02          /NUMBER OF LOGICAL RP UNITS *12JAH8*
213 /
214          .IFUND  COMMAND
215 COMMAND=73
216          .ENDC
217          .IFNOZ  COMMAND
218 COMMAND=73
219          .ENDC
220          .IFPNZ  COMMAND-169
221 COMMAND=169
222          .ENDC
223          000111 A  COMMAND=COMMAND+1/3*3+1
224          000031 A  COMBUFF=COMMAND+2/3
225 /
226          .IFUND  STRING
227 STRING=73
228          .ENDC
229          .IFNOZ  STRING-72
230 STRING=73
231          .ENDC
232          .IFPNZ  STRING-178

```

```

233 STRING=178
234 .ENDC
235 000111 A STRING=STRING+1/3*3+1
236 000031 A STRACC=STRING+2/3
237 /
238 .OCT
239 .IFPNZ API
240 705521 A INTDN=705521 /EBI
241 705522 A INTOFF=705522 /DBI
242 .ENDC
243 .IFUND MTCOL /*12JAH17*
244 MTCOL=73 /*12JAH17*
245 .ENDC /*12JAH17*
246 .IFPNZ MICOL-STRING /*12JAH17*
247 MTCOL=STRING /*12JAH17*
248 .ENDC /*12JAH17*
249 /
250 /SYSTEM PARAMETERS BASED ON CONFIGURATION TAPE
251 /
252 /
253 000050 A NUMTT=DC01*10 /TOTAL DEVICES ON DC01'S
254 000062 A DEV=NUMTT+1+DTA+1+LP15+LP15.2+CRXX+MTA+DUMMY /TOTAL DEVICES, INCLUDING
255 /1 CONSOLE TELLY AND 1 PTR-PTP
256 000051 A NUMAT=NUMTT+1 /TOTAL TELETYPES, INCLUDING CONSOLE TELLY
257 000015 A ITWORD=ITCHAR+1/2
258 .DEC
259 000003 A PERIOD=CLKHZ*SLICE/1000 /CLOCK COUNTS/TIME SLICE
260 .OCT
261 160000 A ..C=CORE*2000
262 .IFUND .32K
263 .32K=40
264 .ENDC
265 .IFUND HLOCK /IF HLOCK IS DEFINED AS OTHER THAN 0
266 HLOCK=0 /SYSTEM WILL NOT HALT UNTIL USER
267 .ENDC /TAKES ACTION AND ZERDES HALTON IN SYSTAB
268 /
269 .IFDEF FUNCT1 /IF USER IS SPECIFYING EITHER SPECIAL
270 KBBGFU=0 /FUNCTIONS
271 .ENDC /OR
272 .IFDEF CSSVT05 /IS USING COMPUTER SPECIAL SYSTEMS VT05
273 KBBGFU=0 /EQUIPPED WITH BADGE READER--
274 .ENDC /SET UP FLAG TO SAY THAT WE NEED %B VARIABLE.
275 .IFDEF BADGIN
276 KBBGFU=0
277 .ENDC /ALSO IF HE'S GOT A BADGIN OF HIS OWN

```

```

278 .TITLE GLOBAL DEFINITIONS
279 /*12JAH12*
280 /GLOBALS USED IN ALL SYSTEMS
281 /*12JAH12*
282 /INTERNAL
283 /
284 .GLOBL ACSWR0,ACTEMP,ASSEM,BEGIN,BKCHAR,BOOL,BOOLE
285 .GLOBL BUFF,BUFF1,BUNIT,CHAR1,CHAR2,CHECK
286 .GLOBL CLKCNT,CLOSE,CLOSE1,CONT,CRFIL,DABSPT,DECIN,DECOUT
287 .GLOBL DEGRAD,DIRPTR,DISLOP,DISPAT,DSKACT,DSKBFF,DSKCNT
288 .GLOBL DTAB1,DTABLE,DTATST,DVHANG,DVHNG1,END,EVAL,EVAR
289 .GLOBL EXIT,FORTST,GBAD,GBLTPT,GBWD2,GCHANP
290 .GLOBL GDISK,GEND,GETKEY,GHDOLD,GHEAD,GHI,GLFNCT,GLOBAL
291 .GLOBL GSYM,GSYMB0,GUNIT,HALTON,HANG,HANG1,HLTSWT
292 .GLOBL ICHARI,ID,INDTAB,IPLACE,IPOINT,IVAR,KEY,KEY.2,KEYCHK,KVAR
293 .GLOBL LNTBPT,LOC,LVAR,M,MONIT,MVAR,NBPNT,OBPNT,OCHAR,OCHARI
294 .IFPNZ XVM /*12JAH15*
295 .GLOBL MEXEC /*12JAH15*
296 .ENDC /*12JAH15*
297 .GLOBL OCTOUT,OUTPUT,OPLACE,OPNCNT,OPPOINT,OP1,OTEXT,OUT,OUTTAB,PACKEM
298 .GLOBL PTEMPX,PRDCNTX,PBOOLX /*12JAH17*
299 .GLOBL PART,PLACE1,PLACE2,PLENG,POINT1,POINT2,PRGDRT,PRINT1,PRNCDV,PRODA
300 .GLOBL PROG,PRTEND,PRIIDX,PRTIN,PRTNMB,PRTPTR,PTTRAN,PTWHO
301 .GLOBL PUSER,RDCNT,REMOVE,RUNDOWN,SAVE,SCRIP,SCRIPS,SECOND
302 .GLOBL SETIO,SPOINT,STACH4,STACH8,STACK,STATCH,STAT1,STATX,STEP
303 .GLOBL STOR1,STR4,SUBOPT,SURQUE,SUM,SVAR,SVE,SWAPSW,SWPCNT
304 .GLOBL SYM,SYMBEG,SYMBOL,SYMBOT,SYMS,SYNTAX,TEMP,TEMP1
305 .GLOBL TEMP2,TERM,USABLE,UNSAVE,XOUT5A,USER,USERS,VAL,VBLOCK
306 .GLOBL VEND,WHOTAB,WRPART,WRTCNT,WRTSW,WVAR,XFRDN,XOUT4A,XVAR,..JCFN1,..8M,%E
307 .GLOBL XCO1,XCI1,XCI3
308 /EXTERNAL - MUMPSB
309 /
310 .GLOBL BEGIO,BOOLX,CALLP,CALL7,CALL71,CFUN1,DABSXF,DISTR
311 .GLOBL DL2,DMP,DMPOLD,DSKRDE,DSKWRT,ERROR0,ERROR2,FILE /*12JAH17*
312 .GLOBL FILE6,FTABLE,FUNCT,FUNOUT,GARB,GETNEW,GFREE,HALT,HALT1,HALTER
313 .GLOBL MUMPS2 /*12JAH17*
314 .GLOBL HFGLO1,HKBENT,HOUT,HTPENT,LOCK,OF2,OF3,OF5,OLAY
315 .GLOBL QF7,QF8,RDCNTX,SNAFU,TEMPX,UNLOCK,XDISLD
316 /
317 /EXTERNAL - MUBF
318 /
319 .GLOBL CLOCK,DATE,DATE.1,DATE.2,DXKEY,GBUFF1,GCHAIN
320 .GLOBL MUBUFF /*12JAH17*
321 .GLOBL GTABLE,GTMBUF,GTMB.1,MONTH,PASTAB
322 /
323 /EXTERNAL - MUPT
324 /
325 .GLOBL PTPIN,PTPOUT,PTROUT
326 .GLOBL MUPT /*12JAH17*
327 /
328 .IFPNZ API
329 .GLOBL NOFLAG

```

```
330 .ENDC
331 /
332 /EXTERNAL - MDUMP
333 /
334 .GLOBL MDUMP
335 .EJECT
```

/*12JAH17*
/*12JAH17*
/*12JAH17*
/*12JAH17*

```
336      /*12JAH12*
337      /MONEY GLOBALS
338      /*12JAH12*
339          .IFPNZ MONEY
340          /
341          /INTERNAL
342          /
343          .GLOBL STR4,TEMP,IPDINT,MFUNCT,SVE,TEMP1,POINT1
344          .GLOBL CHAR1,SYNTAX,ICHAR1,OCHAR,OUTPUT,SAVE,UNSAVE
345          /
346          /
347          /EXTERNAL
348          /
349          .GLOBL MONEYM
350          .ENDC
351          /
352          .IFPNZ FMONEY
353          /
354          /INTERNAL
355          /
356          .GLOBL FLOAT,MONYSW,VALF,BUFF2
357          .ENDC
358          /
359          .EJECT
```

/*12JAH17*

/*12JAH17*

/*12JAH17*

/*12JAH17*

```
360      /*12JAH12*
361      /SCANNER GLOBALS
362      /*12JAH12*
363          .IFPNZ DC01
364          /
365          /INTERNAL
366          /
367          .GLOBL  XOUT3,MONTTY,CONSW,READ1A
368          .IFDEF  KBBGFU
369          .GLOBL  %B
370          .ENDC
371      /
372          /EXTERNAL - MUDC
373          /
374          .GLOBL  TTSET,SCAN0,SCANCT,KBOUT,TPIN
375          .GLOBL  MUDC
376          .GLOBL  INTSC0
377          .IFPNZ  DC01-1
378          .GLOBL  INTSC1
379          .ENDC
380          .IFPNZ  DC01-2
381          .GLOBL  INTSC2
382          .ENDC
383          .IFPNZ  DC01-3
384          .GLOBL  INTSC3
385          .ENDC
386          .IFPNZ  DC01-4
387          .GLOBL  INTSC4
388          .ENDC
389          .IFPNZ  DC01-5
390          .GLOBL  INTSC5
391          .ENDC
392          .IFPNZ  DC01-6
393          .GLOBL  INTSC6
394          .ENDC
395          .IFPNZ  DC01-7
396          .GLOBL  INTSC7
397          .ENDC
398          /
399          .ENDC
400          .EJECT
```

/*12JAH17*

```
401          /*12JAH12*
402          /RP GLOBALS
403          /*12JAH12*
404          .IFPNZ  RP
405          /
406          /INTERNAL
407          /
408          .GLOBL  BPWRIT,RPSTAT,RPERRS,NFLAG,GUNIT,RPERST,SUBQUES /*12JAH16*
409          .GLOBL  RPDRV,RPADDR,PRINMB,RPTRAN,RUNDOWN
410          /
411          /EXTERNAL - RP
412          /
413          .GLOBL  RPFLAG
414          .GLOBL  MURP                                     /*12JAH17*
415          /
416          .ENDC
417          .EJECT
```

```
418          /*12JAH12*
419          /RF GLOBALS
420          /*12JAH12*
421             .IFPNZ  RF15
422             /
423             /INTERNAL
424             /
425             .GLOBL  NFLAG,RFERRS,RFSTAT,RFADRO,RFADR1,PRINMB,RETRAN,BPWRT /*12JAH16*
426             /
427             /EXTERNAL - MURF
428             /
429             .GLOBL  RFFLAG
430             .GLOBL  MURF                                     /*12JAH17*
431             /
432             .ENDC
433             .EJECT
```

```
434          /*12JAH12*
435          /DECTAPE GLOBALS
436          /*12JAH12*
437              .IFPNZ  DTA
438              /
439              /INTERNAL
440              /
441              .GLOBL  HANG,PRTNMR,LOKDT,AVAR,XOUTDT,DTWHO1
442              .GLOBL  INPUT,NOFLAG,IN,ALTMOD,EXIT,DTERRA,DISUFF
443              .GLOBL  DTERRB,DTERRS,OUTPUT,DTJMPO,DTJMPI,DITRAN
444              .IFNOZ  API
445              .GLOBL  DTSKPC
446              .ENDC
447              /
448              /EXTERNAL - DT
449              /
450              .GLOBL  DTACT,DTOUT,DTIN,DTOUT1,DTIN1
451              .GLOBL  MUDT
452              .IFNOZ  API
453              .GLOBL  DTENT1,DTENT2
454              .ENDC
455              /
456              .ENDC
457              .EJECT
```

/*12JAH17*

```
458      /*12JAH12*
459      /MT GLOBALS
460      /*12JAH12*
461          .IFPNZ MTA
462          /
463          /INTERNAL
464          /
465          .GLOBL MTSTAT,MTERRS,MTUN1,MTWHO1,MTTRAN,INPUT,MTBUFF
466          .GLOBL MTPUT,MTGET,MTEDM
467          .GLOBL MTOVEL                                     /*12JAH17*
468          /
469          /EXTERNAL - MUMT
470          /
471          .GLOBL IENTRY
472          .GLOBL MUMT                                     /*12JAH17*
473          /
474          .ENDC
475          .EJECT
```

```
476      /*12JAH12*
477      /LP GLOBALS
478      /*12JAH12*
479          .IFPNZ LP15
480          /INTERNAL
481          /
482          .GLOBL LPWHO,LPTRAN,LPSTA,LPWHO
483          /
484          /EXTERNAL
485          /
486          .GLOBL LPBIN,LPQUT
487          .GLOBL MULP                                     /*12JAH17*
488          /
489          .IFPNZ LP15.2
490          /
491          /INTERNAL
492          /
493          .GLOBL LPWHO2,LPTRN2
494          /
495          /EXTERNAL - MULP
496          /
497          .GLOBL LPBIN2,LPQUT2
498          .GLOBL MULP2                                     /*12JAH17*
499          /
500          .ENDC
501          /
502          .ENDC
503          .EJECT
```

```
504      /*12JAH12*  
505      /CARD READER GLOBALS  
506      /*12JAH12*  
507          .IFPNZ CRXX  
508          /  
509          /INTERNAL  
510          /  
511          .GLOBL CDWHO,CDSTAT,CDTRAN  
512          /  
513          /EXTERNAL - MUCR  
514          /  
515          .GLOBL CDBOUT  
516          .GLOBL MUCR  
517          /  
518          .ENDC
```

/*12JAH17*

```

519             .TITLE  MNEMONIC DEFINITIONS
520             .EBREL
521             /
522             /
523             /SUBROUTINE CALLS TO THE SEVERAL REENTRANT SUBROUTINES ARE
524             /      NOTED BY "CALL" INSTEAD OF "JMS", MAKING THE LISTING
525             /      EASIER TO READ.
526             /
527             100000 A  CALL=JMS
528             /
529             /
530             000000 A  ERROR=000000             /CAL USED FOR ERROR CALLS
531             /
532             /      DEFINITIONS FOR ERROR CALLS
533             /
534             000001 A  ERR1=1
535             000002 A  ERR2=2
536             000003 A  ERR3=3
537             000004 A  ERR4=4
538             000005 A  ERR5=5
539             000006 A  ERR6=6
540             000007 A  ERR7=7
541             000010 A  ERR10=10
542             000011 A  ERR11=11
543             000012 A  ERR12=12
544             000013 A  ERR13=13
545             000014 A  ERR14=14
546             000015 A  ERR15=15
547             000016 A  ERR16=16
548             000017 A  ERR17=17
549             000020 A  ERR20=20
550             000021 A  ERR21=21
551             000022 A  ERR22=22             /22, 23 AND 26 DETECTED AT INTERRUPT LEVEL
552             000023 A  ERR23=23             /AND ARE INVOKED IN A SPECIAL MANNER
553             000024 A  ERR24=24
554             000025 A  ERR25=25
555             000026 A  ERR26=26
556             000027 A  ERR27=27
557             000030 A  ERR30=30
558             000031 A  ERR31=31
559             000032 A  ERR32=32
560             000033 A  ERR33=33
561             .EJECT

```

```

562 /TWO MICRO-INSTRUCTIONS NOT INCLUDED IN MACRO'S PERMANENT SYMBOL TABLE
563 641000 A CLAC=641000 /CLEAR ACCUMULATOR
564 660000 A GAS=660000
565 /
566 /CONSOLE TELETYPE IOT'S
567 /
568 / NOTE: CONSOLE TELETYPE IS OPERATED IN HALF-DUPLEX MODE
569 /
570 /.....KEYBOARD
571 700301 A KSF=700301 /SKIP IF KEYBOARD INTERRUPTING
572 700312 A KRB=700312 /READ KEYBOARD TO AC 10-17
573 /.....PRINTER
574 700401 A TSF=700401 /SKIP IN PRINTER INTERRUPTING
575 700402 A TCF=700402 /CLEAR PRINTER FLAG TO STOP ITS INTERRUPT
576 700406 A TLS=700406 /LOAD PRINTER BUFFER AND PRINT
577 /
578 /PC05 PAPER TAPE STATION SKIP IOT'S
579 /
580 700101 A RSF=700101 /SKIP IF READER INTERRUPTING
581 700201 A PSF=700201 /SKIP IF PUNCH INTERRUPTING
582 /
583 /CLOCK IOT'S
584 /
585 700001 A CLSF=700001 /SKIP IF CLOCK INTERRUPTING
586 700004 A CLDF=700004 /TURN CLOCK OFF
587 700044 A CLON=700044 /TURN CLOCK ON
588 /
589 /RP DISKPACK IOT'S *12JAH8*
590 /
591 706312 A DPRSA=706312 /READ FUNCTION REGISTER "A" TO AC
592 706332 A DPRSB=706332 /READ STATUS REGISTER "B" TO AC
593 706341 A DPSJ=706341 /SKIP IF "JOB DONE" INTERRUPTING
594 706321 A DPSA=706321 /SKIP ON UNIT ATTENTION
595 706361 A DPSE=706361 /SKIP IF "ERROR" INTERRUPTING
596 706364 A DPWC=706364 /LOAD WORD COUNT REG FROM AC
597 706344 A DPCA=706344 /LOAD CURRENT ADDRESS REG FROM AC
598 706304 A DPLA=706304 /LOAD CYLINDER, HEAD, AND SURFACE FROM AC
599 706464 A DPLF=706464 /LOAD AC TO FUNCTION REG 0-7 AND EXECUTE
600 706301 A DPSF=706301 /SKIP ON ANY RP DISK FLAG
601 706324 A DPCS=706324 /SET CONTROLLER TO POWER CLEAR STATE
602 /
603 /DECTAPE IOT'S
604 /
605 707544 A DTXA=707544 /XOR AC INTO STATUS REGISTER "A"
606 707545 A DTLA=707545 /LOAD STATUS REGISTER "A" FROM ACCUMULATOR
607 707552 A DTRA=707552 /READ STATUS REGISTER "A" INTO AC 0-11
608 707561 A DTEF=707561 /SKIP IF "ERROR" INTERRUPTING
609 707572 A DTRB=707572 /SKIP IF "DT JOB DONE" INTERRUPTING
610 707601 A DTDF=707601
611 707541 A DTCA=707541 /CLEAR STATUS REGISTER A
612 /
613 /RF15 DECDISK IOT'S

```

614		/	
615	707001 A	DSSF=707001	/SKIP ON RF15 INTERRUPTING
616	707021 A	DSCC=707021	/CLEAR DISK CONTROL AND DISABLE "FREEZE"
617	707022 A	DRAL=707022	/"OR" APO INTO ACCUM
618	707062 A	DRAH=707062	/"OR" AP1 INTO ACCUM
619	707024 A	DLAL=707024	/LOAD ACCUM INTO APO
620	707064 A	DLAH=707064	/LOAD ACCUM INTO AP1
621	707041 A	DSCF=707041	/CLEAR FUNCTION REGISTER
622	707042 A	DSFX=707042	/XOR AC 15,16,17 INTO FUNCTION REGISTER
623	707044 A	DSCN=707044	/EXECUTE FUNCTION
624	707202 A	DLOK=707202	/"OR" SEGMENT ADDRESS INTO ACCUM
625	707242 A	DSCD=707242	/CLEAR STATUS REGISTER
626	707262 A	DSRS=707262	/"OR" STATUS REGISTER WITH ACCUM
627		/	
628		/DC01 LINE SCANNER IOT'S	
629		/	
630	704401 A	SSF=704401	/SKIP ON SCANNER INTERRUPTING
631	704412 A	SRR=704412	
632	704404 A	SCL=704404	/DISABLE SCANNER
633	704421 A	SSL=704421	/STOP SCANNER AND READ
634			/SCANNER LINE REGISTER
635	704422 A	SLI=704422	/LOAD SELECTED TRANSMITTER LINE DATA
636	704424 A	SCR=704424	/CLEAR SCANNER FLAG AND START
637			/SCANNER AT PRESENT LINE
638		/	
639		/LP15F LINEPRINTER SKIP IOT	
640		/	
641	706501 A	LPSF=706501	/SKIP ON DONE AND ERROR
642		/	
643		/	
644		/SECOND LINEPRINTER	
645		/	
646	703501 A	LPSF.2=703501	/FOR SPECIAL SECOND LP
647		/	
648		/CR03B CARD READER SKIP IOT	
649		/	
650	706721 A	CRSI=706721	/SKIP IF CR03B INTERRUPT
651		/	
652		/CR15 CARD READER SKIP IOT	
653	706701 A	CRSKP=706701	/SKP IF CR15 INTERRUPT
654		/	
655		/	
656		/POWER FAIL IOT	
657		/	
658	703201 A	PFSF=703201	/SKIP ON POWER LOW FLAG.
659		/	
660		.JFPNZ XVM	/*12JAH15*
661	701742 A	MPEU=701742	/*12JAH15* ENTER USER MODE
662	700024 A	LDMM=700024	/*12JAH15* LOAD MM REGISTER FROM AC
663	700032 A	RDMM=700032	/*12JAH15* READ MM REGISTER INTO AC
664	701762 A	RDCLK=701762	/*12JAH15* READ XVM CLOCK INTO AC
665	701741 A	MPSNE=701741	/*12JAH15* SKIP IF NEXM FLAG UP

70444

666	701744	A	MPCNE=701744	/*12JAH15* CLEAR NEXM FLAG
667	701764	A	IPIH=701764	/*12JAH15*
668			.ENDC	/*12JAH15*

```

669           .TITLE  MACRO DEFINITIONS
670           /
671           /BLOCK TRANSFER MACRO
672           .DEFIN  XFR,DEV,UNIT,ADDR,CORE,LNG,DIR,RETRY
673           JMS     DEV
674           -RETRY          /RETRY COUNT
675           LAC     UNIT    /UNIT NUMBER
676           LAC     ADDR    /DEVICE ADDRESS
677           LAC     CORE    /CORE BUFFER ADDRESS
678           LAC     LNG     /LENGTH AS NEGATIVE W.C.
679           .IFZER  DIR
680           CLL                      /READ
681           .ENDC
682           .IFPNZ  DIR
683           SKP!CLL          /OR WRITE
684           JMP*    CDBIN
685           .ENDC
686           .ENDM
687           000000 A      R=0
688           000001 A      W=1
689           /DESCRIPTION OF CONSOLE SWITCH USAGE:
690           /
691           /           THE CONSOLE SWITCHES ARE USED BY THE OPERATOR TO DEMAND
692           /           CERTAIN FUNCTIONS OF THE MONITOR
693           /
694           /SWITCHES 0 AND 1 (TOGETHER)  SYSTEMATIC HALT
695           /
696           /SWITCH 3           MAILBOX CALL ACTIVE
697           /
698           /SWITCH 5           ENABLE XCOM
699           /
700           /SWITCH 6           EMERGENCY HALT - MUST BE ONLY SWITCH ON!
701           /
702           /SWITCH 12          WHEN 0, USE CONSOLE TELLY FOR MINI-DDT
703           /                   WHEN 1, USE AS MUMPS TERMINAL OR MONITOR
704           /
705           /SWITCHES 13-17 ONLY EFFECTIVE WHEN SWITCH 12=1
706           /                   IF 0, USE AS MUMPS TERMINAL
707           /                   IF LEGAL LINE NUMBER, MONITOR THAT LINE
708           /                   IF ILLEGAL, USE AS MUMPS TERMINAL
709           /                   IF SAME AS CONSOLE LINE NUMBER, USE AS MUMPS
710           /                   TERMINAL.

```

```

711          .TITLE  INTERRUPT ENTRY AND REGISTER DEFINITIONS
712          /INTERRUPT ENTRY POINT
713          00000 R 000000 A          0
714          00001 R 601046 R          JMP      INTERU
715          00002 R 601040 R          JMP      GOBEG      /*12JAH12* BOOTSTRAP ENTRY HERE TO START SYSTEM
716          00003 R 601042 R          JMP      GODMP      /*12JAH12* START-UP HERE FOR CRASH PROCEDURES
717          00004 R 601044 R          JMP      GOSNAF     /*12JAH12* START-UP HERE FOR CRASH PROCEDURES
718          /NO BUFFER CLEAN UP, JUST XFR DABS AND GET MUPAK
719          00005 R 000606 R          SYSTAB      /POINTER TO SYSTEM TABLE
720          00006 R          A          .BLOCK  12
721          /
722          /LOCATION 7 IS CLOCK CELL
723          /
724          000007 A          R07=7
725          /
726          /CELLS 10, 11, 12 AND 13 ARE NON-REENTRANT SYSTEM CELLS
727          /
728          000010 A          R10=10
729          000011 A          R11=11
730          000012 A          R12=12
731          000013 A          R13=13
732          /
733          /CELLS 14, 15, 16 AND 17 ARE REENTRANT, AND ARE SAVED IN SWAPPING IMAGE
734          /
735          000014 A          R14=14
736          000015 A          R15=15
737          000016 A          R16=16
738          000017 A          R17=17
739          /
740          /
741          /CELL 20, THE "CAL" ENTRY POINT, IS ALSO USED AS REENTRANT SCRATCH
742          /BUT IS PROCEDURALLY USED AS "ERROR" ENTRY POINT
743          /
744          000020 A          R20=20
745          /
746          00020 R 740040 A          XX
747          00021 R 601036 R          JMP      GOERR      /*12JAH12* GOERR MUST BE IN PAGE 0!!!
748          00022 R          A          .BLOCK  2          /22 AND 23 MAY BE USED BY CR15 AS WC AND CA
749          00024 R 001777 A          XFR.1  1777          /LITERALS FOR XFRDN BOOT IN OF MUMPS
750          00025 R 376000 A          XFR.2  376000
751          /
752          /VARIOUS DATA CHANNEL, ADDRESSES
753          /
754          000022 A          R22=22
755          000023 A          R23=23
756          000030 A          R30=30          /NEGATIVE WORD COUNT FOR DECTAPE
757          000031 A          R31=31
758          000032 A          R32=32
759          000033 A          R33=33
760          000035 A          R35=35          /CORE ADDR. -1 FOR LP15
761          000036 A          R36=36          /NEGATIVE WORD COUNT FOR RF15 DISK
762          000037 A          R37=37          /CORE ADDRESS -1 FOR RF15 DECDISK

```

763	00026 R	A	.BLOCK 10	
764	00036 R	777741 A	-37	
765	00037 R	000037 R	.	/CURRENT ADDRESS FOR DECDISK

766		.TITLE	API ADDRESS TABLE	
767		.IFPNZ	API	
768	00040 R 101253 R	JMS	ILLINT	
769	00041 R 101253 R	JMS	ILLINT	
770	00042 R 101253 R	JMS	ILLINT	
771	00043 R 101253 R	JMS	ILLINT	
772		.IFPNZ	DTA	
773	00044 R 120603 R	JMS*	DTTRAN	/DEC TAPE
774		.ENDC		
775		.IFNOZ	DTA	
776		JMS	ILLINT	
777		.ENDC		
778		.IFPNZ	MTA	
779	00045 R 120604 R	JMS*	MTTRAN	/MAG TAPE
780		.ENDC		
781		.IFNOZ	MTA	
782		JMS	ILLINT	
783		.ENDC		
784	00046 R 101253 R	JMS	ILLINT	
785	00047 R 101253 R	JMS	ILLINT	
786	00050 R 120602 R	JMS*	PTTRAN	/PAPER TAPE READER
787	00051 R 101324 R	JMS	GOCLK	/CLOCK
788		.IFPNZ	PFAIL	
789		JMS	GOFF	/POWER FAIL
790		.ENDC		
791		.IFNOZ	PFAIL	
792	00052 R 101253 R	JMS	ILLINT	
793		.ENDC		
794	00053 R 101253 R	JMS	ILLINT	/PARITY
795	00054 R 101253 R	JMS	ILLINT	/VT15
796		.IFPNZ	CRXX	
797		JMS*	CDTRAN	/CARD READER
798		.ENDC		
799		.IFNOZ	CRXX	
800	00055 R 101253 R	JMS	ILLINT	
801		.ENDC		
802		.IFPNZ	LP15	
803		JMS*	LPTRAN	/LINE PRINTER
804		.ENDC		
805		.IFNOZ	LP15	
806	00056 R 101253 R	JMS	ILLINT	
807		.ENDC		
808	00057 R 101253 R	JMS	ILLINT	/ A/D
809	00060 R 101253 R	JMS	ILLINT	/DR99A/DB98A
810	00061 R 101253 R	JMS	ILLINT	
811	00062 R 101253 R	JMS	ILLINT	/DATA PHONE (DP09A)
812		.IFPNZ	RF15	
813		JMS*	RFTRAN	/RF DISK
814		.ENDC		
815		.IFNOZ	RF15	
816	00063 R 101253 R	JMS	ILLINT	
817		.ENDC		

818		.IFPNZ	RP	/*12JAH8*
819	00064 R 120605 R	JMS*	RPTRAN	/RP DISK
820		.ENDC		
821		.IFNOZ	RP	/*12JAH8*
822		JMS	ILLINT	
823		.ENDC		
824	00065 R 101253 R	JMS	ILLINT	
825	00066 R 101253 R	JMS	ILLINT	
826		.IFPNZ	LP15.2	
827		JMS*	LP2TRN	/LINE PRINTER #2
828		.ENDC		
829		.IFNOZ	LP15.2	
830	00067 R 101253 R	JMS	ILLINT	
831		.ENDC		
832		.IFPNZ	DC01	
833	00070 R 101110 R	JMS	INTSC0	/SCANNER #0
834		.ENDC		
835		.IFNOZ	DC01	
836		JMS	ILLINT	
837		.ENDC		
838		.IFPNZ	DC01-1	
839	00071 R 101133 R	JMS	INTSC1	/SCANNER #1
840		.ENDC		
841		.IFNOZ	DC01-1	
842		JMS	ILLINT	
843		.ENDC		
844		.IFPNZ	DC01-2	
845	00072 R 101157 R	JMS	INTSC2	/SCANNER #2
846		.ENDC		
847		.IFNOZ	DC01-2	
848		JMS	ILLINT	
849		.ENDC		
850		.IFPNZ	DC01-3	
851	00073 R 101203 R	JMS	INTSC3	/SCANNER #3
852		.ENDC		
853		.IFNOZ	DC01-3	
854		JMS	ILLINT	
855		.ENDC		
856		.IFPNZ	DC01-4	
857	00074 R 101227 R	JMS	INTSC4	/SCANNER #4
858		.ENDC		
859		.IFNOZ	DC01-4	
860		JMS	ILLINT	
861		.ENDC		
862		.IFPNZ	DC01-5	
863		JMS	INTSC5	/SCANNER #5
864		.ENDC		
865		.IFNOZ	DC01-5	
866	00075 R 101253 R	JMS	ILLINT	
867		.ENDC		
868		.IFPNZ	DC01-6	
869		JMS	INTSC6	/SCANNER #6

870		.ENDC	
871		.IFNOZ	DC01-6
872	00076 R 101253 R	JMS	ILLINT
873		.ENDC	
874		.IFPNZ	DC01-7
875		JMS	INTSC7
876		.ENDC	/SCANNER #7
877		.IFNOZ	DC01-7
878	00077 R 101253 R	JMS	ILLINT
879		.ENDC	
880		.ENDC	

```

881          .TITLE  MUPAK-MUMPS OVERLAP
882          /"MUPAK" INITIATES A DISK TRANSFER TO BRING "MUMPS" INTO CORE,
883          /THEN TRANSFERS CONTROL TO AN IDENTICAL SET OF CODE WITHIN
884          /ITSELF AT THESE SAME ADDRESSES. AS THE TRANSFER PROGRESSES,
885          /THE "MUPAK" CODE IS OVERWRITTEN WITH THIS "MUMPS" CODE.
886          /WHEN THE TRANSFER IS COMPLETE, "MUMPS" DETECTS IT, TESTS FOR
887          /ERROS, CHECKSUMS, AND TRANSFERS CONTROL TO THE INITIALIZATION
888          /PROCEDURE.
889          .IFPNZ  RP          /*12JAH8*
890          XFRDN  DPSJ          /IS JOB DONE?
891          SKP          /NOT YET...
892          JMP      .+3         /YES -
893          DPSE          /...OR IS THERE AN ERRDR?
894          JMP      .-4         /NO...
895          DPRSB        /STATUS REG B
896          AND      XFR.1     /GET RID OF JUNK
897          LMQ          /HOLD IN MQ
898          DPRSA        /STATUS REG A
899          ALS      10
900          AND      XFR.2
901          OMQ          /OR BOTH TOGETHER
902          SZA
903          HLT          /BAD TRANSFER, AC HOLD STATUS BITS
904          .ENDC
905          .IFZER  RP          /IF NO RP, MUST BE ON DEDISK *12JAH8*
906          XFRDN  DSSF          /IS TRANSFER COMPLETE?
907          JMP      .-1         /NOT YET...
908          DSR+10        /LOAD STATUS REG
909          SPA
910          HLT          /BAD TRANSFER, AC HOLDS STATUS BITS
911          .ENDC
912          /CALCULATE THE CHECKSUM - CANNOT USE CORE LOCATIONS
913          /NOTE: BY CONVENTION, "MUPAK" HAS LEFT THE POSITIVE WORD COUNT
914          /      OF THE TRANSFER IN THE LIMIT REGISTER
915          00116 R 735000 A    CLX
916          00117 R 707762 A    DBA          /USE INDEX MODE
917          00120 R 750000 A    CLA
918          00121 R 350001 A    TAD      1,X    /THIS INSTRUCTION MUST BE IN PAGE 0!!!
919          00122 R 725001 A    AXS      1
920          00123 R 600121 R    JMP      .-2
921          /CHECKSUM IN ACCUM - IT MUST BE ZERO!
922          00124 R 707764 A    EBA          /RETURN TO BANK MODE
923          00125 R 740200 A    SZA
924          00126 R 740040 A    HLT          /BAD CHECKSUM--AC WILL CONTAIN CHECKSUM DIFFERENCE
925          .IFPNZ  RP          /*12JAH8*
926          00127 R 134050 E    JMS*    DABSXF  /*12JAH12*
927          00130 R 011000 A    011000  /READ UNIT 0 WITHOUT INTERRUPT
928          .ENDC
929          .IFZER  RP          /*12JAH8*
930          JMS*    DABSXF  /*12JAH12*
931          2          /READ WITHOUT INTERRUPT
932          .ENDC

```

PAGE 29

MUMPS 100

MUPAK-MUMPS OVERLAP

933

00131 R 634041 E

JMP*

BEGIN

/*12JAH12*

```

934          .TITLE PARTITION SWAPPING IMAGE
935          /
936          /PARAMETERS AND POINTERS FOR RUNNING PARTITION .
937          /
938          /
939          /THESE ARE ADDRESS POINTERS INTO THE ACTIVE PARTITION.
940          /THESE ARE NOT SAVED AS PART OF THE SWAPPING IMAGE
941          /SINCE THAT WOULD EAT UP STORAGE IN THE PARTITION, AND
942          /IT IS EASY ENOUGH TO COMPUTE THEM EACH TIME THE PARTITION
943          /IS SWAPPED IN.
944          /
945          /IMPORTANT!!!!!!
946          /      "AVAR" THROUGH "EVAR" MUST APPEAR IN SAME ORDER AS "VBLOCK" VARIABLES
947          /
948          /
949          /
950          00132 R 000000 A  BUFF1  0          /POINTER TO THE DIRECT COMMAND BUFFER
951          00133 R 000000 A  BUFF2  0          /POINTER TO THE STRING ACCUMULATOR
952          00134 R 000000 A  BUFF   0          /POINTER TO THE PROGRAM BUFFER
953          00135 R 000000 A  SYMBOI 0          /POINTER TO THE END OF THE USER DEFINED SYMBOLS
954          /
955          /FROM HERE DOWN TO .....
956          /
957          00136 R 000000 A  AVAR    0          /DECTAPE POSITION VARIABLE
958          00137 R 000000 A  KVAR    0          /KEY
959          00140 R 000000 A  PVAR    0          /PAGE VARIABLE
960          00141 R 000000 A  MVAR    0          /MARGIN VARIABLE
961          00142 R 000000 A  FVAR    0          /FORMAT (RIGHT MARGIN) VARIABLE.
962          00143 R 000000 A  IVAR    0          /I/O DEVICE VARIABLE
963          00144 R 000000 A  WVAR    0          /WATCH DOG (TIMED READ) VARIABLE.
964          00145 R 000000 A  LVAR    0          /CURRANT STEP
965          00146 R 000000 A  SVAR    0          /STORAGE - CHARACTERS OF ROOM LEFT
966          00147 R 000000 A  TVAR    0          /TIME (IN SEC. SINCE MIDNITE/100)
967          00150 R 000000 A  XVAR    0          /CHARACTER POSITION (IN I/O LINE)
968          00151 R 000000 A  YVAR    0          /LINE POSITION IN PAGE
969          .IFDEF  KBBGFU
970          BVAR    0          /FUNCTION VARIABLE FOR DC01
971          .ENDC
972          00152 R 000000 A  EVAR    0          /BREAK CONTROL SWITCH
973          /.....DOWN TO HERE ARE "SYSVAR" NUMBER OF SYSTEM VARIABLES
974          /
975          000015 A  SYSVAR=EVAR-AVAR+1
976          00153 R 000000 A  SYMEND  0          /USED FOR HANG TIMER
977          00154 R 000000 A  PRTEND  0          /USED FOR %W TIMER
978          .EJECT

```

```

979          /SWAPPING IMAGE STARTS HERE
980          /VARIABLES
981          /
982          /FROM HERE DOWN TO .....
983          /
984          00155 R 000000 A   PC      0          /NOTE- PC AND AC MUST BE THE FIRST
985          00156 R 000000 A   AC      0          /LOCATIONS OF BOTH THE SWAPPING IMAGE
986                                     /AND THE ACTUAL PARTITION.
987          00157 R 000000 A   MQ      0
988          00160 R 000000 A   XR      0
989          00161 R 000000 A   LR      0
990          /DEX10 0
991          /DEX13 0
992          /DEX14 0          /THESE ARE TREATED AS COMMENTS HERE, SINCE
993          /DEX15 0          /THEY OCCUPY REAL CELLS ELSEWHERE.
994          /DEX16 0          /HOWEVER, IN THE ACTUAL PARTITION THERE
995          /DEX17 0          /ARE 7 CELLS, APPEARING IN THE ORDER SHOWN
996          /ERR20 0          /TO HOLD THEM DURING SWAPOUT.
997          /.....DOWN TO HERE ARE 14(OCTAL) HARD REGISTERS
998          /
999          000014 A   REGS=14
1000         /
1001         /
1002         /AND FROM HERE DOWN TO .....
1003         00162 R 000000 A   PRNCDV 0          /PRINCIPLE DEVICE - LINE # SIGNED IN ON
1004         00163 R 000000 A   INPUT  0          /CONTAINS JMP INTAB+DEV
1005         00164 R 000000 A   OUTPUT 0          /CONTAINS OUTTAB+DEV
1006         00165 R 000000 A   STACK  0          /CONTAINS TOP OF STACK-1 POINTER
1007         00166 R 000000 A   SPOINT 0          /CURRENT STACK POINTER
1008         00167 R 000000 A   END     0          /POINTER TO THE END OF THE PROGRAM BUFFER
1009         00170 R 000000 A   SYMBEG 0          /ADDR - BEGINNING OF SYMBOL TABLE
1010         00171 R 000000 A   OBPNT  0          /"OLD" I.E. LAST, BLOCK READ
1011         00172 R 000000 A   WRTSW  0          /WRITE SWITCH - IF NON-ZERO, "PERSONAL" BUFFER
1012                                     /MUST BE WRITTEN OUT
1013         00173 R 000000 A   SWPCNT 0          /SWAP COUNT - NUMBER OF TIMES SWAPPED OUT
1014         00174 R 000000 A   CLKCNT 0          /CPU USAGE COUNT
1015         00175 R 000000 A   WRTCNT 0          /NUMBER OF DATA BASE WRITES (AS BLOCKS)
1016         00176 R 000000 A   RDCNT  0          /NUMBER OF BLOCK READS
1017         00177 R 000000 A   SYM     0          /COUNT OF LEVELS OF FORS
1018         00200 R 000000 A   SCRIP   0
1019         00201 R 000000 A   SAVPRD 0          /ACCUMULATOR SAVE DURING XPRODA
1020         00202 R 000000 A   TEMP    0
1021         00203 R 000000 A   FORTST  0
1022         00204 R 000000 A   SYMS    0
1023         00205 R 000000 A   SCRIPS  0
1024         00206 R 000000 A   LOC     0          /POINTER INTO STEP BUFFER
1025         00207 R 000000 A   VAL     0
1026         .IFPNZ MONEY
1027         00210 R 741000 A   MONYSW  SKP          /NOP=$M USING SYMBOL TABLE, SKP=OTHER
1028         .ENDC
1029         .IFPNZ FMONEY
1030         00211 R 000000 A   VALF   0          /F.P. ARG LEGALITY INDICATOR; 0 ILL, 1 LEGAL

```

```

1031      00212 R 000000 A   FLOAT   0           /SA ENTRY LEGALITY INDICATOR; 0 ILL, 1 LEGAL
1032      .ENDC
1033      00213 R 000000 A   OFLAG   0           /SPECIAL SQ FLAG
1034      00214 R 000000 A   M        0
1035      00215 R 000000 A   SUM      0
1036      00216 R 000000 A   TERM    0
1037      00217 R 000000 A   ACTEMP  0
1038      00220 R 000000 A   IPOINT  0           /INPUT CHARACTER POINTER
1039      00221 R 000000 A   IPLACE  0           /*12JAH14* CHAR POSITION WITHIN WORD
1040      00222 R 000000 A   OPOINT  0           /OUTPUT CHARACTER POINTER
1041      00223 R 000000 A   OPLACE  0           /*12JAH14* CHAR POSITION WITHIN WORD
1042      00224 R 000000 A   TEMP1   0
1043      00225 R 000000 A   POINT1  0
1044      00226 R 000000 A   PLACE1  0           /*12JAH14* CHAR POSITION WITHIN WORD
1045      00227 R 000000 A   TEMP2   0
1046      00230 R 000000 A   POINT2  0
1047      00231 R 000000 A   PLACE2  0           /*12JAH14* CHAR POSITION WITHIN WORD
1048      00232 R 000000 A   BSUM    0
1049      00233 R 000000 A   BTERM   0
1050      00234 R 000000 A   BVAL    0
1051      00235 R 000000 A   FORVAR  0           /FOR VARIABLE
1052      00236 R 000000 A   PROG    0           /CURRENT PROGRAM NAME
1053      00237 R 000000 A   PUSER   0           /USER UNDER WHICH THE CURRENT PROGRAM IS FILED
1054      00240 R 000000 A   ID      0           /SIGN IN PROGRAM
1055      00241 R 000000 A   USER   0           /SIGN IN USER
1056      00242 R 000000 A   BKCHAR  0
1057      00243 R 000000 A   T1      0           /POINTER TO "FOR" VARIABLE'S VALUE
1058      00244 R 000000 A   T2      0           /"FOR" INCREMENT
1059      00245 R 000000 A   T3      0           /"FOR" TERMINATOR
1060      00246 R 000000 A   IFSWT   0           /RESULT OF LAST "IF" 0=FALSE
1061      00247 R 000000 A   GUNIT   0           /UNIT OF "NEXT" ACCESS
1062      00250 R 000000 A   BUNIT   0           /UNIT # OF BLOCK IN PARTITION
1063      00251 R 000000 A   GBLKHI  0           /BLOCK ON THIS LEVEL WITH THE MOST ROOM
1064      00252 R 000000 A   GHI     0           /HOW MUCH ROOM IN GBLKHI
1065      63 00253 R 000000 A   GHEAD   0           /HEAD BLOCK OF CURRENT GLOBAL LEVEL
1066      00254 R 000000 A   GSYMBOL 0           /ADDR -5 WORD BLOCK OF CURRENT GLOBAL IN GTABLE
1067      00255 R 000000 A   GHDOLD  0           /HEAD BLOCK OF PREVIOUS GLOBAL LEVEL
1068      /.....DOWN TO HERE ARE "VARNUM" NUMBER OF REENTRANT VARIABLES
1069      000074 A   VARNUM=GHDOLD-PRNCDV+1
1070      .EJECT

```

253
155

76 = 62

```

1071 /
1072 /SUBROUTINES
1073 /FROM HERE DOWN TO .....
1074 00256 R 000000 A BOOLL 0
1075 00257 R 606313 R JMP XREL1 /RELATIONAL ENTRY TO XBOOL
1076 00260 R 000000 A BOOLE 0
1077 00261 R 606313 R JMP XREL1 /RELATIONAL ENTRY TO XBOOL
1078 00262 R 000000 A BOOLG 0
1079 00263 R 606313 R JMP XREL1 /RELATIONAL ENTRY TO XBOOL
1080 00264 R 000000 A DISLOP 0
1081 00265 R 634161 E JMP* XDISLOP /*12JAH12* CALL & OVERLAY COMMAND DISPATCHER
1082 00266 R 000000 A GASS 0
1083 00267 R 610113 R JMP XGASS /ASSEMBLE GLOBAL NAME
1084 00270 R 000000 A GSYM 0
1085 00271 R 610152 R JMP XGSYM
1086 00272 R 000000 A GLOBAL 0
1087 00273 R 610410 R JMP XGLOBAL
1088 00274 R 000000 A PTEMPX 0 /*12JAH17*
1089 00275 R 000000 A PBOOLX 0 /*12JAH17*
1090 00276 R 000000 A PRDCNTX 0 /*12JAH17*
1091 /..... DOWN TO HERE ARE
1092 / REENTRANT SUBROUTINE ENTRY POINTS
1093 000056 A SUBNUM=56 /*12JAH17*
1094 .IFPNZ DTA /*12JAH17*
1095 000057 A SUBNUM=SUBNUM+1 /*12JAH17*
1096 .ENDC /*12JAH17*
1097 .IFPNZ MODIFY /*12JAH17*
1098 000061 A SUBNUM=SUBNUM+2 /*12JAH17*
1099 .ENDC /*12JAH17*
1100 /
1101 000174 A SWAPSZ=REGS+VARNUM+3+SUBNUM /*12JAH12* TOTAL CELLS IN SWAP AREA
1102 /*12JAH12* 3 CELLS IN MUMPSB
1103 00277 R 777361 A USABLE 0-SWAPSZ-BLOCK-COMBUFF-STRACC-PERMSY
1104 00300 R 000000 A P
1105 / WHICH IS NUMBER OF WORDS AVAILABLE FOR PROGRAM AND USER
1106 / SYMBOL TABLES
1107 .EJECT

```

REMOVE
 JMP XREMOVE

```

1108 /STATUS TABLE - ONE WORD FOR EACH PARTITION
1109 /
1110 /BIT 0 1=INDEPENDENT, 0=DEPENDENT
1111 /BIT 1 NOT CURRENTLY USED
1112 /BIT 2 DITTO
1113 /BIT 3 IOINT
1114 /BIT 4 CURRENTLY UNUSED
1115 /BIT 5 DITTO
1116 /BIT 6 IF 1, MAGTAPE HANG
1117 /BIT 7 USED FOR %W VIOLATION
1118 /BIT 8 IF 1, JOB IN TRANSITION (SIGNING ON OR OFF)
1119 /
1120 /BIT 9 IF 1, JOB IN PRIORITY STATE
1121 /
1122 /FOLLOWING 8 BITS ARE "HANG BITS"
1123 /
1124 /BIT 10 NOT USED
1125 /BIT 11 IF 1, CRXX HANG
1126 /
1127 /BIT 12 IF 1, LP15 HANG
1128 /BIT 13 IF 1, DECTAPE HANG
1129 /BIT 14 IF 1, PAPER TAPE HANG
1130 /
1131 /BIT 15 IF 1, DC01 SCANNER HANG
1132 /BIT 16 IF 1, CLOCK HANG (E.G. HANG .05)
1133 /BIT 17 IF 1, RUNNABLE
1134 /
1135 /NOTE - IF ALL BITS ARE ZERO, PARTITION IS FREE
1136 /

```

```

1137 .REPT PRINUM
1138 STAT1 0

```

```

00301 R 000000 A
00302 R 000000 A *R
00303 R 000000 A *R
00304 R 000000 A *R
00305 R 000000 A *R
00306 R 000000 A *R
00307 R 000000 A *R
00310 R 000000 A *R
00311 R 000000 A *R
00312 R 000000 A *R
00313 R 000000 A *R
00314 R 000000 A *R
00315 R 000000 A *R
00316 R 000000 A *R
00317 R 000000 A *R
00320 R 000000 A *R
00321 R 000000 A *R
00322 R 000000 A *R
00323 R 000000 A *R
00324 R 000000 A *R
00325 R 000000 A *R
00326 R 000000 A *R

```

PAGE 35 MUMPS 100 PARTITION SWAPPING IMAGE

1139 00327 R 000000 A *R
00330 R 000330 R STATX .

1140 .TITLE INPUT/OUTPUT TABLES
 1141 /
 1142 /DEVICE OWNERSHIP TABLE
 1143 /
 1144 /CELLS ARE ON A ONE-TO-ONE CORRESPONDENCE WITH DEVICE NUMBER.
 1145 / IF CELL=0 , NOT IN USE
 1146 / IF CELL NON-ZERO , DEVICE IS IN USE, AND CONTENTS
 1147 / OF CELL = ADDRESS OF OWNER'S STATUS WORD

1148 00331 R 000331 R WHOTAB .
 1149 .REPT NUMAT
 1150 00332 R 000000 A TT 0

- 00333 R 000000 A *R
- 00334 R 000000 A *R
- 00335 R 000000 A *R
- 00336 R 000000 A *R
- 00337 R 000000 A *R
- 00340 R 000000 A *R
- 00341 R 000000 A *R
- 00342 R 000000 A *R
- 00343 R 000000 A *R
- 00344 R 000000 A *R
- 00345 R 000000 A *R
- 00346 R 000000 A *R
- 00347 R 000000 A *R
- 00350 R 000000 A *R
- 00351 R 000000 A *R
- 00352 R 000000 A *R
- 00353 R 000000 A *R
- 00354 R 000000 A *R
- 00355 R 000000 A *R
- 00356 R 000000 A *R
- 00357 R 000000 A *R
- 00360 R 000000 A *R
- 00361 R 000000 A *R
- 00362 R 000000 A *R
- 00363 R 000000 A *R
- 00364 R 000000 A *R
- 00365 R 000000 A *R
- 00366 R 000000 A *R
- 00367 R 000000 A *R
- 00370 R 000000 A *R
- 00371 R 000000 A *R
- 00372 R 000000 A *R
- 00373 R 000000 A *R
- 00374 R 000000 A *R
- 00375 R 000000 A *R
- 00376 R 000000 A *R
- 00377 R 000000 A *R
- 00400 R 000000 A *R
- 00401 R 000000 A *R
- 00402 R 000000 A *R

1152				.REPT	DTA	/NUMBER OF TRANSPORTS:
1153	00403	R	000000	A	DTWHO1	0
	00404	R	000000	A	*R	
1154				.ENDC		
1155	00405	R	000000	A	PTWHO	0
1156				.IFPNZ	LP15	/ONLY ONE PAPER TAPE STATION ALLOWED
1157					LPWHO	0
1158				.ENDC		
1159				.IFPNZ	LP15.2	
1160					LPWHO2	0
1161				.ENDC		
1162				.IFPNZ	CRXX	
1163					CDWHO	0
1164				.ENDC		
1165				.IFPNZ	MTA	
1166				.REPT	MTA	
1167	00406	R	000000	A	MTWHO1	0
1168				.ENDC		
1169				.IFPNZ	DUMMY	
1170				.REPT	DUMMY	
1171	00407	R	000000	A		0
	00410	R	000000	A	*R	
	00411	R	000000	A	*R	
	00412	R	000000	A	*R	
	00413	R	000000	A	*R	
1172				.ENDC		
1173				.EJECT		

1174				/OUTPUT DISPATCH TABLE
1175				.REPT NUMAT
1176	00414	R 634156	E	OUTTAB JMP* TPIN
	00415	R 634156	E *R	
	00416	R 634156	E *R	
	00417	R 634156	E *R	
	00420	R 634156	E *R	
	00421	R 634156	E *R	
	00422	R 634156	E *R	
	00423	R 634156	E *R	
	00424	R 634156	E *R	
	00425	R 634156	E *R	
	00426	R 634156	E *R	
	00427	R 634156	E *R	
	00430	R 634156	E *R	
	00431	R 634156	E *R	
	00432	R 634156	E *R	
	00433	R 634156	E *R	
	00434	R 634156	E *R	
	00435	R 634156	E *R	
	00436	R 634156	E *R	
	00437	R 634156	E *R	
	00440	R 634156	E *R	
	00441	R 634156	E *R	
	00442	R 634156	E *R	
	00443	R 634156	E *R	
	00444	R 634156	E *R	
	00445	R 634156	E *R	
	00446	R 634156	E *R	
	00447	R 634156	E *R	
	00450	R 634156	E *R	
	00451	R 634156	E *R	
	00452	R 634156	E *R	
	00453	R 634156	E *R	
	00454	R 634156	E *R	
	00455	R 634156	E *R	
	00456	R 634156	E *R	
	00457	R 634156	E *R	
	00460	R 634156	E *R	
	00461	R 634156	E *R	
	00462	R 634156	E *R	
	00463	R 634156	E *R	
	00464	R 634156	E *R	
1177				.IFPNZ DTA
1178				.REPT DTA
1179	00465	R 634065	E	JMP* DTOUT
	00466	R 634065	E *R	
1180				.ENDC
1181	00467	R 634143	E	JMP* PTPIN
1182				.IFPNZ LP15
1183				JMP* LPBIN
1184				.ENDC

1185		.IFPNZ	LP15.2	
1186		JMP*	LPBIN2	
1187		.ENDC		
1188		.IFPNZ	CRXX	
1189		ERR27		/"IOERR" WHEN ATTEMPTING OUTPUT TO CR
1190		.ENDC		
1191		.IFPNZ	MTA	
1192		.REPT	MTA	
1193	00470 R 634126 E	JMP*	MTPUT	
1194		.ENDC		
1195		.IFPNZ	DUMMY	
1196		.REPT	DUMMY	
1197	00471 R 000027 A	ERR27		/"IOERR" ON OUTPUTS TO DUMMIES
	00472 R 000027 A *R			
	00473 R 000027 A *R			
	00474 R 000027 A *R			
	00475 R 000027 A *R			
1198		.ENDC		
1199		.EJECT		

```

1200           /INPUT DISPATCH TABLE
1201           .REPT  NUMAT
1202           INTAB  JMP*  KBOUT
00476 R 634117 E
00477 R 634117 E *R
00500 R 634117 E *R
00501 R 634117 E *R
00502 R 634117 E *R
00503 R 634117 E *R
00504 R 634117 E *R
00505 R 634117 E *R
00506 R 634117 E *R
00507 R 634117 E *R
00510 R 634117 E *R
00511 R 634117 E *R
00512 R 634117 E *R
00513 R 634117 E *R
00514 R 634117 E *R
00515 R 634117 E *R
00516 R 634117 E *R
00517 R 634117 E *R
00520 R 634117 E *R
00521 R 634117 E *R
00522 R 634117 E *R
00523 R 634117 E *R
00524 R 634117 E *R
00525 R 634117 E *R
00526 R 634117 E *R
00527 R 634117 E *R
00530 R 634117 E *R
00531 R 634117 E *R
00532 R 634117 E *R
00533 R 634117 E *R
00534 R 634117 E *R
00535 R 634117 E *R
00536 R 634117 E *R
00537 R 634117 E *R
00540 R 634117 E *R
00541 R 634117 E *R
00542 R 634117 E *R
00543 R 634117 E *R
00544 R 634117 E *R
00545 R 634117 E *R
00546 R 634117 E *R

1203           .IFPNZ  DTA
1204           .REPT  DTA           /NUMBER OF DECTAPE TRANSPORTS
1205           JMP*  DTIN
00547 R 634063 E
00550 R 634063 E *R

1206           .ENDC
1207           JMP*  PTROUT
1208           .IFPNZ  LP15
1209           ERR27           /"IOERR" WHEN ATTEMPTING INPUT FROM LP
1210           .ENDC

```

1211		.IFPNZ	LP15.2	
1212		ERR27		
1213		.ENDC		
1214		.IFPNZ	CRXX	
1215		JMP*	CDBOUT	
1216		.ENDC		
1217		.IFPNZ	MTA	
1218		.REPT	MTA	
1219	00552 R 634125 E	JMP*	MTGET	
1220		.ENDC		
1221		.IFPNZ	DUMMY	
1222		.REPT	DUMMY	
1223	00553 R 000027 A	ERR27		/"IOERR" ON INPUTS FROM DUMMIES
	00554 R 000027 A *R			
	00555 R 000027 A *R			
	00556 R 000027 A *R			
	00557 R 000027 A *R			
1224		.ENDC		

```

1225 .TITLE SYSTEM TABLE AND PARAMETERS
1226 00560 R 000000 A INTAC 0 /AC FROM INTERRUPT
1227 00561 R 000000 A INTMQ 0 /MQ FROM INTERRUPT
1228 00562 R 000000 A INTX 0 /X REG FROM INTERRUPT
1229 00563 R 000000 A INTL 0 /LIMIT REG FROM INTERRUPT
1230 00564 R 000000 A IOIMP 0 /TEMP. VAR FOR IO ROUTINES WITH INT OFF--MUST BE IN PAGE 0!!
1231 00565 R 000000 A SCDUNT 0
1232 00566 R 000000 A STEMP 0 /TEMP VAR FOR SLPCHK ROUTINE
1233 00567 R 000000 A STEMP2 0
1234 00570 R 000000 A EMERG 0 /*12JAH10* EMERGENCY SWITCH (6) INDICATOR
1235 00571 R 000000 A SLPCNT 0 /COUNT OF SECONDS SINCE LAST /*12JAH17*
1236 /*%W/CLOCK HANG UPDATE /*12JAH17*
1237 00572 R 000000 A SLPTMP 0 /PTR TO PARTITION STATUS WORD /*12JAH17*
1238 /0=OFF, 1=ON /*12JAH17*
1239 /*12JAH10* 0=OFF, 1=ON
1240 00573 R 000000 A KEEP20 0 /TEMPORARY FOR R20 /*12JAH15*
1241 /
1242 00574 R 000000 A DDTSW 0 /CONSOLE TELLY SWITCH: 0=MONITOR, NON-0 DDT
1243 00575 R 000000 A MONTTY 0 /NUMBER OF TELLY BEING MONITORED *1000
1244 00576 R 000000 A CONSW 0 /CHARACTER STATUS SWITCH FOR CONSOLE TELLY
1245 00577 R 000000 A DSKACT 0 /BLOCK OWNERSHIP SWITCH
1246 00600 R 777766 A DSKCNT -DCOUNT /COUNT OF CONSECUTIVE DISK ACCESSES BY SINGLE USER
1247 00601 R 000000 A PRVDKA 0 /LAST OWNER OF DSKACT
1248 .IFPNZ API
1249 00602 R 740040 A PTTRAN XX
1250 .IFPNZ DTA
1251 00603 R 740040 A DTTRAN XX
1252 .ENDC
1253 .IFPNZ MTA
1254 00604 R 740040 A MTTRAN XX
1255 .ENDC
1256 .IFPNZ RF15
1257 RFTRAN XX
1258 .ENDC
1259 .IFPNZ RP /*12JAH8*
1260 00605 R 740040 A RPTRAN XX
1261 .ENDC
1262 .IFPNZ LP15
1263 LPTRAN XX
1264 .ENDC
1265 .IFPNZ LP15.2
1266 LP2TRN XX
1267 .ENDC
1268 .IFPNZ CRXX
1269 CDTRAN XX
1270 .ENDC
1271 .ENDC
1272 /
1273 /THE SYSTEM TABLE -- SYSTAB
1274 /
1275 /THIS TABLE IS POINTED TO BY LOCATION 5
1276 /THE TABLE CONTAINS BASIC SYSTEM INFO,STATUS INFO ABOUT

```

```

1277 /I/O DEVICES, AND POINTERS TO VARIOUS SYSTEM TABLES.
1278 /
1279 /
1280 000606 R SYSTAB=.
1281 /
1282 00606 R 000000 A .IORS 0 /PROCESSOR STATUS WORD
1283 00607 R 000000 A ACSWR 0 /CURRENT SETTING OF CONSOLE SW'S
1284 00610 R 000000 A ACSWRX 0 /XCOM THIS SPOT TO INTERNALLY CLEAR OUT AN AC SWITCH.
1285 00611 R 266161 A SYSID .SIXBT 'V11Aaa' /SYSTEM VERSION #
00612 R 010000 A

1286 /
1287 /ERROR STATUS FOR RP15 DISK SYSTEM UPDATE ON EVERY
1288 /ERROR DETECTED BY THE RP15 CONTROLLER
1289 /
1290 00613 R 000000 A RFSTAT 0 /STATUS REGISTER
1291 00614 R 000000 A RFADRO 0 /DISK ADDRESS -- AP0
1292 00615 R 000000 A RFADR1 0 /DISK ADDRESS -- AP1
1293 00616 R 000000 A RFERRS 0 /ERROR COUNT
1294 /
1295 /STATUS AND ERROR STATUS FOR RP15 DISK SYSTEM
1296 /
1297 00617 R 000000 A RPSTAT 0 /STATUS REG "A"
1298 /
1299 /FOLLOWING LOCATIONS ONLY UPDATED ON ERRORS
1300 /
1301 00620 R 000000 A RPDRV 0 /STATUS "A" (UNIT # IN 0-2)
1302 00621 R 000000 A RPERST 0 /STATUS "B"
1303 00622 R 000000 A RPADDR 0 /DISK ADDRESS CLY IN 0-7
1304 /HEAD IN 8-12 SEC IN 14-17
1305 00623 R 000000 A RPERRS 0 //ERROR COUNT
1306 /
1307 /DEC-TAPE ERROR STATUS
1308 /
1309 00624 R 000000 A DTERRA 0 /STATUS REG "A" (UNIT IN 0-2)
1310 00625 R 000000 A DTERRB 0 /STATUS REG "B"
1311 00626 R 000000 A DTERRS 0 /ERROR COUNT
1312 /
1313 /MAG-TAPE ERROR STATUS
1314 /
1315 00627 R 000000 A MTSTAT 0 /MAG TAPE STAUS /WITH UNIT # 15-17
1316 00630 R 000000 A MTERRS 0 /ERROR COUNT
1317 /
1318 /MAG-TAPE DRIVE STATUS UPDATED ON EVERY OPERATION TO THAT DRIVE
1319 /
1320 00631 R 000000 A MTUN1 0 /DRIVE 1
1321 00632 R 000000 A MTUN2 0
1322 00633 R 000000 A MTUN3 0
1323 00634 R 000000 A MTUN4 0
1324 00635 R 000000 A MTUN5 0
1325 00636 R 000000 A MTUN6 0
1326 00637 R 000000 A MTUN7 0
1327 00640 R 000000 A MTUN8 0

```

```

1328 /
1329 /STATUS WORDS FOR CARD READER, AND LINE PRINTERS
1330 /WHICH ARE UPDATE ON EVERY I/O OPERATION
1331 /
1332 00641 R 000000 A CDSTAT 0 /CARD READER
1333 00642 R 000000 A LPSTA 0 /LINE PRINTER #1
1334 00643 R 000000 A LPSTA2 0 /LINE PRINTER #2
1335 /
1336 /SYSTEM INFORMATION AND POINTERS
1337 /
1338 00644 R 000000 A NOFLAG 0 /COUNT OF ILLEGAL INTERRUPTS
1339 00645 R 000000 A DKCHK 0 /NUMBER OF CHKPT ERRORS ON RETRIES
1340 00646 R 000301 R JBSPTR STAT1 /PTR TO JOB STATUS TABLE
1341 00647 R 000027 A PRTNUM /# OF PARTITIONS IN SYSTEM
1342 00650 R 000331 R WHOPIR WHOTAB /PTR TO THE WHO TABLE
1343 00651 R 000063 A OUTTAB-WHOTAB /SIZE OF WHOTAB
1344 00652 R 000000 A SWAPSW 0 /0= INHIBIT SWAPPING;
1345 /+= SWAPPABLE;
1346 /-1=SLICE OVERFLOW, SWAP WHEN DONE
1347 00653 R 000000 A NBPNT 0 /NEXT BLOCK POINTER
1348 00654 R 000000 A PRTNMB 0 /NEXT PARTITION TO RUN
1349 00655 R 000000 A RUNNMB 0 /PARTITION NUMBER NOW RUNNING
1350 -00656 R 000000 A RUNDOWN 0 /COUNTER FOR "SWAPPED IN" TIME
1351 00657 R 000000 A PRTIDX 0 /LAST USER
1352 00660 R 000000 A PRTIN 0 /PARTITION SWAPPED IN
1353 00661 R 000000 A MONIT 0 /PRTBEG-1, OR 0 IF NO USER SWAPPED IN
1354 00662 R 000000 A ACSWRO 0 /CONTENTS OF AC SW. OR'ED WITH ACSWRX
1355 00663 R 777777 A HLTSWT -1 /SET TO ZERO BY AC SWITCHES WHEN DUMP TIME
1356 00664 R 000000 A HALTON HLOCK /BECOMES 1 WHEN HALTER ENTERED
1357 00665 R 777777 A USERS -1 /NUMBER OF USERS CURRENTLY ACTIVE
1358 /-1 ON ENTRY FROM MUPAK SO IT CAN'T CLOSE
1359 /IMMEDIATELY IF YOU FORGOT TO CLEAR THE SWITCHES
1360 00666 R 000000 A DTBUFF 0 /INITIALIZED TO ADDR OF 1ST DECTAPE BUFFER
1361 00667 R 000000 A MTBUFF 0 /INITIALIZED TO ADDR OF 1ST MAGTAPE BUFFER
1362 00670 R 000000 A OPNCNT 0 /NUMBER OF GLOBALS OPEN
1363 00671 R 000000 A GBAD 0 /NUMBER OF SYSTEM BOO-BOOS IN GARBAGE
1364 00672 R 000000 A GCHANP 0 /*12JAH12* PTR TO GCHAIN IN DABS
1365 00673 R 000000 A DEGRAD 0- /DATA BASE DEGARDED FLAG - SET NEGATIVE
1366 /ON ONE OF THE FOLLOWING:
1367 /AT SIGNOUT USER NOT FOUND IN THE DIRECTORY
1368 /UNRECOVERABLE DISK ERROR IN GARBAGE COLLECT
1369 /0 OFFSET IN GARBAGE CHAIN
1370 /MAPS PROBLEMS ENCOUNTERED IN ALLOCATION
1371 00674 R 000000 A MTEOM 0 /INITIALIZED TO POINT TO FLAG FOR MT EOMS
1372 00675 R 000155 R SWIM PC /PTR TO THE PARTITION SWAPPING IMAGE
1373 00676 R 000000 A DIRPIR 0 /*12JAH12* 1ST CORE BLOCK (FCB) #
1374 00677 R 002114 A PSIZE PRtsiz /DEFAULT PARTITION SIZE
1375 00700 R 001007 R PSTART INOTAB /PTR TO PARTITION STARTING ADDR TABLE
1376 00701 R 014142 E PLENG PASTAB /PTR TO PARTITION LENGTH TABLE
1377 00702 R 000132 R SWPPTR BUFF1 /PTR TO PARTITION SWAPPING AREA
1378 00703 R 000015 A SYSVNM SYSVAR /# OF SYSTEM VARIABLES
1379 00704 R 000000 A DSKBFF 0 /DISK BUFFER OF CURRENT USER

```

```

1380 00705 R 000000 A GBWD2 0 /POINTER TO 2ND WORD OF USER'S DISK BUFFER
1381 00706 R 000000 A DABSPT 0 /*12JAH12* PTR TO "DABS" IN MUBUFF
1382 00707 R 000000 A LNTBPT 0 /*12JAH12* PTR TO LINE TABLE
1383 00710 R 000000 A GBLTPT 0 /*12JAH12* PTR TO GLOBAL TABLE (5 WD/GLOBAL)
1384 00711 R 000074 A NMGBL NOG /# OF GLOBALS
1385 00712 R 000000 A NMRF RF15 /# OF RF15 PLAITERS
1386 00713 R 000003 A NMRRP RP /*12JAH8*
1387 00714 R 000000 A PRTPIR 0 /*12JAH12* PTR TO BEG. OF PARTITONS
1388 00715 R 000000 A CRFIL 0 /*12JAH12* TERM. FILLER CHAR TABLE PTR
1389 00716 R 777766 A DSKREP -DCOUNT /CONSECUTIVE DISK USE COUNT
1390 00717 R 000001 A SUBDSK SUBMERGE /SYS MANAGER DESIRES (1=SUBMERGED,0=REGULAR) DISK
1391 000720 R SUBQUES=.
1392 .IFPNZ SUBMERGE
1393 00720 R 740000 A NOP
1394 .ENDC
1395 .IFNOZ SUBMERGE
1396 SKP
1397 .ENDC
1398 00721 R 777775 A TIMESL -PERIOD /TIME SLICE LENGTH (IN CLOCK TICKS 2'S COMPL.)
1399 00722 R 000000 A PRGDRT 0 /0=GLOBALS ONLY SPAN BUFFER POOL,OTHER=GLOBALS,PROGRAMS,
1400 /DIRECTORIES,XCOM BLCOKS SPAN BUFFER POOL
1401 00723 R 000000 A BPWRIT PWRITE /0 OR NEG=BUFFER POOL IMMEDIATE WRITE TO DISK /*12JAH16*
1402 /+=BUFFER POOL DELAYED WRITE TO DISK /*12JAH6*
1403 00724 R 000001 A NMMT MTA /NUMBER OF MAGTAPES IN SYSTEM /*12JAH17*
1404 00725 R 000002 A NMDT DIA /NUMBER OF DECTAPES IN SYSTEM /*12JAH17*
1405 00726 R 000000 A NMLP LP15+LP15.2 /NUMBER OF LINE PRINTERS IN SYSTEM /*12JAH17*
1406 00727 R 000000 A NMCR CRXX /NUMBER OF CARD READERS IN SYSTEM /*12JAH17*
1407 00730 R 000005 A NMDC DC01 /NUMBER OF DC01'S IN SYSTEM /*12JAH17*
1408 00731 R 000005 A NMDUM DUMMY /NUMBER OF DUMMY DEVICES IN SYSTEM /*12JAH17*
1409 00732 R 777703 A MTOVFL -MTCOL-2 /MT INPUT CHAR LINE OVERFLOW LIMIT /*12JAH17*
1410 00733 R 014132 E STMUM2 MUMPS2 /STARTING ADDRESS OF MUMPS2 /*12JAH17*
1411 000734 R STMONEY=.
1412 .IFPNZ MONEY /*12JAH17*
1413 00734 R 014123 E MONEYM /STARTING ADDRESS OF MONEY /*12JAH17*
1414 .ENDC /*12JAH17*
1415 .IFNOZ MONEY /*12JAH17*
1416 0 /*12JAH17*
1417 .ENDC /*12JAH17*
1418 000735 R STMUDC=.
1419 .IFPNZ DC01 /*12JAH17*
1420 00735 R 014130 E MUDC /*12JAH17*
1421 .ENDC
1422 .IFNOZ DC01 /*12JAH17*
1423 0 /*12JAH17*
1424 .ENDC /*12JAH17*
1425 000736 R STMURP=.
1426 .IFPNZ RP /*12JAH17*
1427 00736 R 014135 E MURP /*12JAH17*
1428 .ENDC /*12JAH17*
1429 .IFNOZ RP /*12JAH17*
1430 0 /*12JAH17*
1431 .ENDC /*12JAH17*

```

1432		000737 R	STMURF=.		/*12JAH17*
1433			.IFPNZ RF15		/*12JAH17*
1434			MURF		/*12JAH17*
1435			.ENDC		/*12JAH17*
1436			.IFNOZ RF15		/*12JAH17*
1437	00737 R	000000 A	0		/*12JAH17*
1438			.ENDC		/*12JAH17*
1439		000740 R	STMUMT=.		/*12JAH17*
1440			.IFPNZ MTA		/*12JAH17*
1441	00740 R	014133 E	MUMT		/*12JAH17*
1442			.ENDC		/*12JAH17*
1443			.IFNOZ MTA		/*12JAH17*
1444			0		/*12JAH17*
1445			.ENDC		/*12JAH17*
1446		000741 R	STMULP=.		/*12JAH17*
1447			.IFPNZ LP15		/*12JAH17*
1448			MULP		/*12JAH17*
1449			.ENDC		/*12JAH17*
1450			.IFNOZ LP15		/*12JAH17*
1451	00741 R	000000 A	0		/*12JAH17*
1452			.ENDC		/*12JAH17*
1453		000742 R	STMULP2=.		/*12JAH17*
1454			.IFPNZ LP15.2		/*12JAH17*
1455			MULP2		/*12JAH17*
1456			.ENDC		/*12JAH17*
1457			.IFNOZ LP15.2		/*12JAH17*
1458	00742 R	000000 A	0		/*12JAH17*
1459			.ENDC		/*12JAH17*
1460	00743 R	014134 E	STMUPT MUPT		/*12JAH17*
1461		000744 R	STMUDI=.		/*12JAH17*
1462			.IFPNZ DTA		/*12JAH17*
1463	00744 R	014131 E	MUDI		/*12JAH17*
1464			.ENDC		/*12JAH17*
1465			.IFNOZ DTA		/*12JAH17*
1466			0		/*12JAH17*
1467			.ENDC		/*12JAH17*
1468		000745 R	STMUCR=.		/*12JAH17*
1469			.IFPNZ CRXX		/*12JAH17*
1470			MUCR		/*12JAH17*
1471			.ENDC		/*12JAH17*
1472			.IFNOZ CRXX		/*12JAH17*
1473	00745 R	000000 A	0		/*12JAH17*
1474			.ENDC		/*12JAH17*
1475	00746 R	014121 E	STMDMP MDUMP		/*12JAH17*
1476	00747 R	014127 E	STMUBF MUBUFF		/*12JAH17*
1477			/		
1478			.EJECT		

```

1479 /THE VARIABLE BLOCK (VBLOCK) CONSISTS OF PREDEFINED SYMBOLS WHICH
1480 /ARE PLACED DIRECTLY INTO THE SYMBOL TABLE AREA OF THE PARTITION
1481 /WHEN THE USER SIGNS ON.
1482 / ALL ENTRIES ARE INVARIANT EXCEPT THE DATE, WHICH IS
1483 /INITIALIZED BY MUPAK, AND DAILY UPDATED BY CLOCK ROUTINE.
1484 /NOTE: IF THE PARTITION IS ACTIVE WHILE MIDNIGHT OCCURS THE DATE
1485 / WITHIN THE PARTITION IS UPDATED WHEN SWAPPED IN.
1486 /
1487 /IMPORTANT!!!!!!!
1488 / THESE MUST APPEAR IN THE SAME ORDER AS PARTITION
1489 / ADDRESS POINTERS "AVAR" THROUGH "YVAR"
1490 /
1491 00750 R 000750 R VBLOCK .
1492 /SDATE
1493 00751 R 701604 A 701604 /(STRING - "SD" IN 5 BIT RIGHT ADJUSTED)
1494 00752 R 000000 A 0 /DATE IS FOUND IN "DABS"
1495 00753 R 000000 A 0
1496 00754 R 000000 A 0
1497 /%ADDRESS
1498 00755 R 501641 A 501641 /(NUMBER - "%A" IN 5 BIT, R.A.)
1499 00756 R 000000 A 000000
1500 /%KEY
1501 00757 R 501653 A 501653 /(NUMBER - "%K" IN 5 BIT, R.A.)
1502 00760 R 000000 A 0
1503 /%PAGE
1504 00761 R 501660 A 501660 /(NUMBER - "%P" IN 5 BIT, R.A.)
1505 00762 R 000000 A 000000 /0
1506 /%MARGIN (LEFT MARGIN)
1507 00763 R 501655 A 501655 /(NUMBER - "%M" IN 5 BIT, R.A.)
1508 00764 R 000000 A 000000
1509 /%FORMAT (RIGHT MARGIN)
1510 00765 R 501646 A 501646 /(NUMBER - "%F" IN 5 BIT, R.A.)
1511 00766 R 016660 A 016660 /76.00
1512 /%IO-DEVICE
1513 00767 R 501651 A 501651 /(NUMBER - "%I" IN 5 BIT, R.A.)
1514 00770 R 000000 A 000000
1515 /%WATCHDOG
1516 00771 R 501667 A 501667 /(NUMBER - "%W" IN 5 BIT, R.A.)
1517 00772 R 000074 A %W WTIMER
1518 /$LOCATION
1519 00773 R 501614 A 501614 /(NUMBER - "%L" IN 5 BIT, R.A.)
1520 00774 R 000000 A 000000
1521 /$STORAGE
1522 00775 R 501623 A 501623 /(NUMBER - "%S" IN 5 BIT, R.A.)
1523 00776 R 000000 A 000000
1524 /$TIME
1525 00777 R 501624 A 501624 /(NUMBER - "%T" IN 5 BIT, R.A.)
1526 01000 R 000000 A 0 /CLOCK IS FOUND IN "DABS"
1527 /$X-AXIS
1528 01001 R 501630 A 501630 /(NUMBER - "%X" IN 5 BIT, R.A.)
1529 01002 R 000000 A 000000
1530 /$Y-AXIS

```

```

1531      01003 R 501631 A          501631          / (NUMBER - "SY" IN 5 BIT, R.A.)
1532      01004 R 000000 A          000000
1533                                     .IFDEF KBBGFU
1534                                     /%BUTTON
1535                                     501642          / (NUMBER - "%B" IN 5 BIT R.A.)
1536                                     %B          000000          / (WHATEVER R.A. IS ??)
1537                                     .ENDC
1538                                     /%ERROR
1539      01005 R 501645 A          501645          / (NUMBER - "%E" IN 5 BIT, R.A.)
1540      01006 R 000000 A          %E          0
1541          001006 R          VEND=%E
1542      /
1543      /THE FOLLOWING CELL OF THE PARTITION HAS THE LABEL "SYMEND"
1544      /          WHICH IS COMPUTED DURING SWAPIN
1545      /
1546      ////// 0          /SYMEND POINTS HERE, PARTITION+(PRTSIZE-2)
1547      ////// THIS CELL (WITHIN THE PARTITION) IS USED TO HOLD THE TIME
1548      ////// HANG VALUE IN NEGATIVE SECONDS. AT THE SAME TIME THE
1549      ////// CLOCK HANG STATUS BIT IS SET.
1550      /
1551      /THE FOLLOWING CELL OF THE PARTITION HAS THE LABEL "PRIEND"
1552      /          WHICH IS COMPUTED DURING SWAPIN
1553      /
1554      ////// 0          THIS CELL, WITHIN THE PARTITION, IS USED
1555      ////// TO HOLD THE %W RUNDOWN VALUE IN NEGATIVE
1556      ////// SECONDS. AT THE SAME TIME THE DC01 HANG
1557      ////// BIT IS SET IN THE PARTITION STATUS WORD.
1558      ////// THE CELL MAY CONTAIN A ZERO, IN WHICH CASE
1559      ////// THERE IS NO TIMEOUT ON A READ
1560      /
1561          000041 A          PERMSY=VEND-VBLOCK+3          /NUMBER OF PERMANENT SYMBOL WORDS IN PARTITION
1562          .REPT          PRNUM
1563      01007 R 000000 A          INDTAB 0          /PARTITION STARTING ADDR.
01010 R 000000 A *R
01011 R 000000 A *R
01012 R 000000 A *R
01013 R 000000 A *R
01014 R 000000 A *R
01015 R 000000 A *R
01016 R 000000 A *R
01017 R 000000 A *R
01020 R 000000 A *R
01021 R 000000 A *R
01022 R 000000 A *R
01023 R 000000 A *R
01024 R 000000 A *R
01025 R 000000 A *R
01026 R 000000 A *R
01027 R 000000 A *R
01030 R 000000 A *R
01031 R 000000 A *R
01032 R 000000 A *R

```

	01033	R	000000	A	*R				
	01034	R	000000	A	*R				
	01035	R	000000	A	*R				
1564	01036	R	707764	A		GOERR	EBA		/*12JAH12*
1565	01037	R	634070	E			JMP*	ERROR0	/*12JAH12*
1566	01040	R	707764	A		GOBEG	EBA		/*12JAH12*
1567	01041	R	634041	E			JMP*	BEGIO	/*12JAH12*
1568	01042	R	707764	A		GODMP	EBA		/*12JAH12*
1569	01043	R	634056	E			JMP*	OMP	/*12JAH12*
1570	01044	R	707764	A		GOSNAF	EBA		/*12JAH12*
1571	01045	R	634154	E			JMP*	SNAFU	/*12JAH12*

```

1572          .TITLE INTERRUPT SKIP CHAIN
1573          001046 R   INTERU=.
1574          .IFPNZ  API
1575          01046 R 705522 A   INTOFF
1576          .ENDC
1577          01047 R 707764 A   EBA
1578          01050 R 040560 R   DAC      INTAC
1579          01051 R 641002 A   LACQ
1580          01052 R 040561 R   DAC      INTMO
1581          01053 R 724000 A   PXA
1582          01054 R 040562 R   DAC      INTX
1583          01055 R 730000 A   PLA
1584          01056 R 040563 R   DAC      INTL
1585          .IFNOZ  API
1586          /
1587          .IFPNZ  PFAIL
1588          PFSF                /POWER LOW?
1589          SKP
1590          JMP      PF          /YES, HURRY UP AND STOP
1591          .ENDC
1592          /
1593          .IFPNZ  CR03B
1594          /
1595          /CR03B CARD READER
1596          /
1597          CRSI                /CARD DONE, COL RDY, OR ERR.
1598          SKP!CLL
1599          JMP*      CDBIN
1600          .ENDC
1601          .IFPNZ  DTA
1602          /
1603          /DECIAPES
1604          /
1605          INTDI  DTEF
1606          SKP
1607          JMP*      DTENT1
1608          DTDF
1609          SKP!CLL
1610          JMP*      DTENT2
1611          DISKPC=.
1612          .ENDC
1613          .IFPNZ  RF15
1614          /
1615          /RF15/RS09 DECDISKS
1616          /
1617          DSSF                /RF JOB DONE OR ERROR
1618          SKP
1619          JMP*      RFFLAG
1620          .ENDC
1621          .IFPNZ  RP          /*12JAH8*
1622          DPSF                /RP JOB DONE OR ERROR (ALSO UNIT ATTN-- BUT SHOULDNT HAPPEN)
1623          /

```

```

1624      /RP MOVING HEAD DISKS *12JAH8*
1625      /
1626          SKP
1627          JMP*   RPFLAG
1628          .ENDC
1629          .IFPNZ MTA
1630      MTSF=707341
1631          MTSF
1632          SKP
1633          JMP*   IENTRY
1634          .ENDC
1635      /
1636      /CLOCK
1637      /
1638          CLSF
1639          SKP
1640          JMP   CLOK
1641      /
1642      /
1643          .IFPNZ DC01
1644      /
1645      /DC01ED LINE SCANNERS
1646      /
1647      INTSC  SSF
1648          JMP   .+6           /NOT SCANNER 0
1649          SRR
1650          AND   (7000
1651          JMS*  TTSET
1652          SRR
1653          JMP*  SCAN0
1654          .IFPNZ DC01-1
1655          SSF+40
1656          JMP   .+7           /NOT SCANNER 1
1657          SRR+40
1658          AND   (7000
1659          XOR   (10000
1660          JMS*  TTSET
1661          SRR+40
1662          JMP*  SCAN0
1663          .IFPNZ DC01-2
1664          SSF+100
1665          JMP   .+7           /NOT SCANNER 2
1666          SRR+100
1667          AND   (7000
1668          XOR   (20000
1669          JMS*  TTSET
1670          SRR+100
1671          JMP*  SCAN0
1672          .IFPNZ DC01-3
1673          SSF+140
1674          JMP   .+7           /NOT SCANNER 3
1675          SRR+140

```

```
1676      AND      (7000
1677      XOR      (30000
1678      JMS*     TTSET
1679      SRR+140
1680      JMP*     SCAN0
1681      .IFPNZ   DC01-4
1682      SSF+200
1683      JMP      .+7          /NOT SCANNER 4
1684      SRR+200
1685      AND      (7000
1686      XOR      (40000
1687      JMS*     TTSET
1688      SRR+200
1689      JMP*     SCAN0
1690      .IFPNZ   DC01-5
1691      SSF+240
1692      JMP      .+7          /NOT SCANNER 5
1693      SRR+240
1694      AND      (7000
1695      XOR      (50000
1696      JMS*     TTSET
1697      SRR+240
1698      JMP*     SCAN0
1699      .IFPNZ   DC01-6
1700      SSF+300
1701      JMP      .+7          /NOT SCANNER 6
1702      SRR+300
1703      AND      (7000
1704      XOR      (60000
1705      JMS*     TTSET
1706      SRR+300
1707      JMP*     SCAN0
1708      .IFPNZ   DC01-7
1709      SSF+340
1710      JMP      .+7          /NOT SCANNER 7
1711      SRR+340
1712      AND      (7000
1713      XOR      (70000
1714      JMS*     TTSET
1715      SRR+340
1716      JMP*     SCAN0
1717      .ENDC
1718      .ENDC
1719      .ENDC
1720      .ENDC
1721      .ENDC
1722      .ENDC
1723      .ENDC
1724      .ENDC
1725      /
1726      /
1727      .ENDC
```

```

1728 /
1729 /CONSOLE TELETYPE
1730 /
1731 01057 R 700301 A CTT KSF
1732 01060 R 741000 A SKP
1733 01061 R 601270 R JMP .DBKB
1734 01062 R 700401 A TSF
1735 01063 R 741000 A SKP
1736 01064 R 601303 R JMP .DBTP
1737 /
1738 .IFNOZ API
1739 /
1740 .IFPNZ LP15
1741 /
1742 /LP
1743 LPSF /LP DONE OR ERROR
1744 SKP!CLL
1745 JMP* LPOUT
1746 .ENDC
1747 /
1748 /
1749 /SECOND LINEPRINTER
1750 /
1751 .IFPNZ LP15.2
1752 LPSF.2
1753 SKP!CLL
1754 JMP* LPOUT2
1755 .ENDC
1756 /
1757 /CR15 CARD READER
1758 /
1759 .IFPNZ CR15
1760 CRSKP
1761 SKP!CLL
1762 JMP* CDBIN
1763 .ENDC
1764 /
1765 INTPTR RSF
1766 SKP!CLL
1767 JMP* PTRIN
1768 .ENDC
1769 /
1770 /PIP
1771 01065 R 700201 A PSF
1772 01066 R 745000 A SKP!CLL
1773 01067 R 634144 E JMP* PTPOUT
1774 /
1775 /NON-EXISTENT MEMORY CHECK *12JAH15*
1776 .IFPNZ XVM /*12JAH15*
1777 01070 R 701741 A MPSNE /*12JAH15* SKIP ON NEXM
1778 01071 R 745000 A SKP!CLL /*12JAH15* NO NEXM
1779 01072 R 601262 R JMP NEXMR /*12JAH15* GO GIVE ERROR

```

```

1780          .ENDC          /*12JAH15*
1781          /
1782          /
1783          /
1784          /
1785          01073 R 440644 R  NFLAG  ISZ      NOFLAG      /ILLEGAL INTERRUPT
1786          01074 R 740000 A      NOP          /IN CASE WE OVERFLOW.
1787          /
1788          /
1789          /COMMON INTERRUPT EXIT ROUTINE FOR ALL INTERRUPT SERVICE PROGRAMS
1790          01075 R 200563 R  EXIT      LAC      INTL
1791          01076 R 722000 A      PAL
1792          01077 R 200562 R      LAC      INTX
1793          01100 R 721000 A      PAX
1794          01101 R 200561 R      LAC      INTMQ
1795          01102 R 652000 A      LMQ
1796          01103 R 200560 R      LAC      INTAC
1797          .IFPNZ  API
1798          01104 R 700042 A      ION
1799          .ENDC
1800          01105 R 705521 A      INTON
1801          01106 R 703344 A      DBR
1802          01107 R 620000 A      JMP*    0
1803          /
1804          .IFPNZ  API
1805          .IFPNZ  DC01
1806          01110 R 000000 A  INTSCO  0
1807          01111 R 705522 A      INTOFF
1808          01112 R 707764 A      EBA
1809          01113 R 040560 R      DAC      INTAC
1810          01114 R 201110 R      LAC      INTSCO
1811          01115 R 040000 A      DAC      0
1812          01116 R 641002 A      LACQ
1813          01117 R 040561 R      DAC      INTMQ
1814          01120 R 724000 A      PXA
1815          01121 R 040562 R      DAC      INTX
1816          01122 R 730000 A      PLA
1817          01123 R 040563 R      DAC      INTL
1818          01124 R 704401 A      SSF
1819          01125 R 601073 R      JMP      NFLAG
1820          01126 R 704412 A      SRR
1821          01127 R 513636 R      AND      (7000
1822          01130 R 134157 E      JMS*    TTSET
1823          01131 R 704412 A      SRR
1824          01132 R 634153 E      JMP*    SCAN0
1825          .ENDC
1826          .IFPNZ  DC01-1
1827          01133 R 000000 A  INTSC1  0
1828          01134 R 705522 A      INTOFF
1829          01135 R 707764 A      EBA
1830          01136 R 040560 R      DAC      INTAC
1831          01137 R 201133 R      LAC      INTSC1

```

1832	01140	R	040000	A	DAC	0
1833	01141	R	641002	A	LACQ	
1834	01142	R	040561	R	DAC	INTMQ
1835	01143	R	724000	A	PXA	
1836	01144	R	040562	R	DAC	INTX
1837	01145	R	730000	A	PLA	
1838	01146	R	040563	R	DAC	INTL
1839	01147	R	704441	A	SSF+40	
1840	01150	R	601073	R	JMP	NFLAG
1841	01151	R	704452	A	SRR+40	
1842	01152	R	513636	R	AND	(7000
1843	01153	R	253637	R	XOR	(10000
1844	01154	R	134157	E	JMS*	TTSET
1845	01155	R	704452	A	SRR+40	
1846	01156	R	634153	E	JMP*	SCAN0
1847					.ENDC	
1848					.IFPNZ	DC01-2
1849	01157	R	000000	A	INTSC2	0
1850	01160	R	705522	A	INTOFF	
1851	01161	R	707764	A	EBA	
1852	01162	R	040560	R	DAC	INTAC
1853	01163	R	201157	R	LAC	INTSC2
1854	01164	R	040000	A	DAC	0
1855	01165	R	641002	A	LACQ	
1856	01166	R	040561	R	DAC	INTMQ
1857	01167	R	724000	A	PXA	
1858	01170	R	040562	R	DAC	INTX
1859	01171	R	730000	A	PLA	
1860	01172	R	040563	R	DAC	INTL
1861	01173	R	704501	A	SSF+100	
1862	01174	R	601073	R	JMP	NFLAG
1863	01175	R	704512	A	SRR+100	
1864	01176	R	513636	R	AND	(7000
1865	01177	R	253640	R	XOR	(20000
1866	01200	R	134157	E	JMS*	TTSET
1867	01201	R	704512	A	SRR+100	
1868	01202	R	634153	E	JMP*	SCAN0
1869					.ENDC	
1870					.IFPNZ	DC01-3
1871	01203	R	000000	A	INTSC3	0
1872	01204	R	705522	A	INTOFF	
1873	01205	R	707764	A	EBA	
1874	01206	R	040560	R	DAC	INTAC
1875	01207	R	201203	R	LAC	INTSC3
1876	01210	R	040000	A	DAC	0
1877	01211	R	641002	A	LACQ	
1878	01212	R	040561	R	DAC	INTMQ
1879	01213	R	724000	A	PXA	
1880	01214	R	040562	R	DAC	INTX
1881	01215	R	730000	A	PLA	
1882	01216	R	040563	R	DAC	INTL
1883	01217	R	704541	A	SSF+140	

1884	01220	R	601073	R	JMP	NFLAG
1885	01221	R	704552	A	SRR+140	
1886	01222	R	513636	R	AND	(7000
1887	01223	R	253641	R	XOR	(30000
1888	01224	R	134157	E	JMS*	TTSET
1889	01225	R	704552	A	SRR+140	
1890	01226	R	634153	E	JMP*	SCAN0
1891					.ENDC	
1892					.IFPNZ	DC01-4
1893	01227	R	000000	A	INTSC4	0
1894	01230	R	705522	A	INTOFF	
1895	01231	R	707764	A	EBA	
1896	01232	R	040560	R	DAC	INTAC
1897	01233	P	201227	R	LAC	INTSC4
1898	01234	R	040000	A	DAC	0
1899	01235	R	641002	A	LACQ	
1900	01236	R	040561	R	DAC	INTMQ
1901	01237	R	724000	A	PXA	
1902	01240	R	040562	R	DAC	INTX
1903	01241	R	730000	A	PLA	
1904	01242	R	040563	R	DAC	INTL
1905	01243	R	704601	A	SSF+200	
1906	01244	R	601073	R	JMP	NFLAG
1907	01245	R	704612	A	SRR+200	
1908	01246	R	513636	R	AND	(7000
1909	01247	R	253642	R	XOR	(40000
1910	01250	R	134157	E	JMS*	TTSET
1911	01251	R	704612	A	SRR+200	
1912	01252	R	634153	E	JMP*	SCAN0
1913					.ENDC	
1914					.IFPNZ	DC01-5
1915				INTSC5	0	
1916					INTOFF	
1917					EBA	
1918					DAC	INTAC
1919					LAC	INTSC5
1920					DAC	0
1921					LACQ	
1922					DAC	INTMQ
1923					PXA	
1924					DAC	INTX
1925					PLA	
1926					DAC	INTL
1927					SSF+240	
1928					JMP	NFLAG
1929					SRR+240	
1930					AND	(7000
1931					XOR	(50000
1932					JMS*	TTSET
1933					SRR+240	
1934					JMP*	SCAN0
1935					.ENDC	

1936			.IFPNZ	DC01-6
1937		INTSC6	0	
1938			INTOFF	
1939			EBA	
1940			DAC	INTAC
1941			LAC	INTSC6
1942			DAC	0
1943			LACQ	
1944			DAC	INTMQ
1945			PXA	
1946			DAC	INTX
1947			PLA	
1948			DAC	INTL
1949			SSF+300	
1950			JMP	NFLAG
1951			SRR+300	
1952			AND	(7000
1953			XOR	(60000
1954			JMS*	TTSET
1955			SRR+300	
1956			JMP*	SCAN0
1957			.ENDC	
1958			.IFPNZ	DC01-7
1959		INTSC7	0	
1960			INTOFF	
1961			EBA	
1962			DAC	INTAC
1963			LAC	INTSC7
1964			DAC	0
1965			LACQ	
1966			DAC	INTMQ
1967			PXA	
1968			DAC	INTX
1969			PLA	
1970			DAC	INTL
1971			SSF+340	
1972			JMP	NFLAG
1973			SRR+340	
1974			AND	(7000
1975			XOR	(70000
1976			JMS*	TTSET
1977			SRR+340	
1978			JMP*	SCAN0
1979			.ENDC	
1980		/		
1981	01253 R 000000 A	ILLINT	0	
1982	01254 R 705522 A		INTOFF	
1983	01255 R 440644 R		ISZ	NOFLAG
1984	01256 R 740000 A		NOP	
1985	01257 R 705521 A		INTON	
1986	01260 R 703344 A		DBR	
1987	01261 R 621253 R		JMP*	ILLINT

```

1988 /
1989 .ENDC
1990 /
1991 .IFPNZ PFAIL
1992 .IFPNZ API
1993 GOPF 0
1994 INTOFF
1995 EBA
1996 DAC INTAC
1997 LAC GOPF
1998 DAC 0
1999 LACQ
2000 DAC INTMQ
2001 PXA
2002 DAC INTX
2003 PLA
2004 DAC INTL
2005 PFSF /POWER LOW?
2006 JMP NFLAG
2007 .ENDC
2008 /POWER FAIL -- AWAIT THE INEVITABLE.
2009 /
2010 PF LAC (JMP RESTRI /SETUP LOC 0 SINCE POWER UP WILL START
2011 DAC 0 /EXECUTION THERE.
2012 JMP . /SIT AND WAIT
2013 /
2014 /
2015 /POWER FAIL -- ARISE THE PHOENIX
2016 /
2017 RESTRI EBA
2018 DZM INTAC /(INTAC USELESS TO US NOW)
2019 ISZ INTAC /DELAY TO MAKE SURE WE'RE REALLY UP.
2020 JMP .-1
2021 CAF /MAKE SURE SYSTEM IS VERY QUIET.
2022 .IFPNZ API
2023 ION
2024 .ENDC
2025 INTON
2026 JMP* HALTER /*12JAH12* TRY TO GET BACK TO MUPAK
2027 .ENDC
2028 /
2029 /PROCESS NON-EXISTENT MEMORY REFERENCE *12JAH15*
2030 /
2031 .IFPNZ XVM /*12JAH15*
2032 01262 R 701744 A NEXMR MPCNE /*12JAH15* CLEAR NEXM FLAG
2033 01263 R 703304 A DBK /*12JAH15*
2034 01264 R 200000 A LAC 0 /*12JAH15* FETCH INT. ADDR
2035 01265 R 705521 A INTON /*12JAH15*
2036 01266 R 040020 A DAC R20 /*12JAH15* SALT AWAY FOR PRINTOUT
2037 01267 R 601036 R JMP GOERR /*12JAH15* GO TO ERR ROUTINE
2038 .ENDC /*12JAH15*
2039 /

```

```

2040          / CODING TO MAKE CONSOLE TELLY LOOK LIKE A MUMPS TERMINAL
2041          /
2042          /KEYBOARD
2043          01270 R 700312 A      .DBKB   KRB
2044          01271 R 200574 R      LAC      DDTSW      /WHAT MODE?
2045          01272 R 740200 A      SZA
2046          01273 R 634113 E      JMP*    HKBENT     /*12JAH12* USE AS MINI DDT
2047          01274 R 200575 R      LAC      MONTTY     /WHAT TELLY BEING MONITORED?
2048          01275 R 740200 A      SZA      /NONE IF ZERO
2049          01276 R 601075 R      JMP      EXIT      /KB NOT USED DURING MONITOR
2050          01277 R 213643 R      LAC      (NUMTT*1000 /CONSOLE TELLY IS LAST TTY TERMINAL
2051          01300 R 134157 E      JMS*    TTSET
2052          01301 R 700312 A      KRB      /GET INPUT CHAR
2053          01302 R 634153 E      JMP*    SCANO
2054          /
2055          /PRINTER
2056          /
2057          01303 R 700402 A      .DBTP   TCF      /SHUT THE BEAST UP
2058          01304 R 200574 R      LAC      DDTSW     /WHAT MODE?
2059          01305 R 740200 A      SZA
2060          01306 R 634115 E      JMP*    HTPENT     /*12JAH12* USE AS MINI DDT
2061          01307 R 200575 R      LAC      MONTTY     /WHAT TELLY BEING MONITORED?
2062          01310 R 740200 A      SZA
2063          01311 R 601315 R      JMP      .DBTP1    /SOMEBODY BEING MONITORED
2064          01312 R 213643 R      LAC      (NUMTT*1000 /CONSOLE TTY IS LAST TERMINAL;
2065          01313 R 134157 E      JMS*    TTSET     /AS IF READ IN FROM DC01
2066          01314 R 634152 E      JMP*    SCANCT
2067          /
2068          /CONSOLE TELLY IS MONITORING SOME USER
2069          /
2070          /
2071          01315 R 200576 R      .DBTP1  LAC      CONSW     /WHAT IS STATUS OF LAST CHAR?
2072          01316 R 741100 A      SPA
2073          01317 R 751001 A      SKP!CMA!CLA /NEGATIVE - RESET TO 0, PERMISSION TO TLS
2074          01320 R 700406 A      TLS     /OUTPUT THIS CHAR
2075          01321 R 740001 A      CMA     /AND RECORD THAT IT IS IN TRANSIT
2076          01322 R 040576 R      DAC      CONSW     /EITHER 0 OR -1 AT THIS POINT
2077          01323 R 601075 R      JMP      EXIT

```

```

2078 .TITLE CLOCK INTERRUPT RESPONSE
2079 .IFPNZ API
2080 01324 R 000000 A GOCLOCK 0
2081 01325 R 705522 A INTOFF
2082 01326 R 707764 A EBA
2083 01327 R 040560 R DAC INTAC
2084 01330 R 201324 R LAC GOCLOCK
2085 01331 R 040000 A DAC 0
2086 01332 R 641002 A LACQ
2087 01333 R 040561 R DAC INTMQ
2088 01334 R 724000 A PXA
2089 01335 R 040562 R DAC INTX
2090 01336 R 730000 A PLA
2091 01337 R 040563 R DAC INTL
2092 01340 R 700001 A CLSF
2093 01341 R 601073 R JMP NFLAG
2094 .ENDC
2095 01342 R 700004 A CLOK CLOF
2096 01343 R 200007 A LAC R07 /GET CLOCK CELL
2097 01344 R 740030 A IAC
2098 01345 R 040564 R DAC IOTMP /KEEP CLOCK + 1 TICK.
2099 01346 R 754001 A CLC!CLL
2100 01347 R 040007 A DAC R07 /RESET CLOCK CELL FOR 1 TICK.
2101 01350 R 700044 A CLON
2102 01351 R 200656 R LAC RUNDOWN /GET COUNTER FOR SWAPPED-IN TIME.
2103 01352 R 740300 A SZA!SMA /ANY USER IN?
2104 01353 R 601361 R JMP CLOK1 /NO, MONITOR RUNNING
2105 01354 R 200564 R LAC IOTMP /YES.
2106 01355 R 300174 R ADD CLKCNT /UPDATE CPU COUNTER
2107 01356 R 040174 R DAC CLKCNT
2108 01357 R 745100 A SPA!CLL
2109 01360 R 140174 R DZM CLKCNT
2110 01361 R 200564 R CLOK1 LAC IOTMP
2111 01362 R 301531 R ADD SECOND /UPDATE ONE SECOND COUNTER
2112 01363 R 041531 R DAC SECOND
2113 01364 R 741100 A SPA /AT OR PAST END-OF-SECOND?
2114 01365 R 601456 R JMP CLOCK1 /NO.
2115 01366 R 723703 A AAC -CLKHZ-1
2116 01367 R 041531 R DAC SECOND /RESET END-OF-SECOND TO -CLKHZ+
2117 /NUMBER OF TICKS THIS SECOND
2118 01370 R 234047 E LAC* CLOCK /TIME OF DAY
2119 01371 R 740030 A IAC /PLUS 1 SECOND
2120 01372 R 553644 R SAD (250600 /DEC 86400(SECONDS PER DAY).
2121 01373 R 754002 A CLA!STL /RESET TIME-OF-DAY, IF MIDNIGHT
2122 01374 R 074047 E DAC* CLOCK
2123 01375 R 740200 A SZA /MIDNIGHT?
2124 01376 R 601425 R JMP CLOCK6 /NO /*12JAH17*
2125 .EJECT

```

```

2126 /MIDNIGHT
2127 01377 R 234053 E LAC* DATE.2 /YES, CHECK IF LAST DAY OF MONTH
2128 01400 R 652000 A LMQ
2129 01401 R 234052 E LAC* DATE.1
2130 01402 R 660606 A LLSS 6
2131 01403 R 513645 R AND (7777
2132 01404 R 574124 E SAD* MONTH
2133 01405 R 741000 A SKP
2134 01406 R 601416 R JMP CLOCK2 /NOT LAST DAY.
2135 01407 R 234052 E LAC* DATE.1 /LAST DAY
2136 01410 R 513646 R AND (777700
2137 01411 R 353647 R TAD (160 /ADD 1 TO MONTH
2138 01412 R 074052 E DAC* DATE.1
2139 01413 R 213650 R LAC (600000
2140 01414 R 074053 E DAC* DATE.2 /SET DAY TO ZERO
2141 01415 R 174124 E DZM* MONTH /CLEAR MONTH AS FLAG TO MUPAK
2142 01416 R 234053 E CLOCK2 LAC* DATE.2
2143 01417 R 553651 R SAD (710000
2144 01420 R 213652 R LAC (570000
2145 01421 R 313637 R ADD (10000
2146 01422 R 074053 E DAC* DATE.2 /SET DAY = DAY+1
2147 01423 R 553650 R SAD (600000
2148 01424 R 474052 E ISZ* DATE.1 /INC. FIRST DIGIT OF DATE IF NECESSARY
2149 01425 R 440571 R CLOCK6 ISZ SLPCNT /INDICATE A SECOND HAS PASSED /*12JAH17*
2150 /CHECK ON DESIRED USE OF CONSOLE TELLY
2151 01426 R 200607 R LAC ACSWR
2152 01427 R 513653 R AND (40 /40 - SWITCH 12
2153 01430 R 540574 R SAD DDTSW /ANY CHANGE IN ODT SWITCH?
2154 01431 R 601441 R JMP CLOCK5 /NO, SAME AS BEFORE.
2155 01432 R 741200 A SNA /LEAVING ODT?
2156 01433 R 700406 A TLS /YES, GENERATE FLAG
2157 01434 R 213654 R LAC (EXIT /YES, JUST CHANGED--INSURE AGAINST
2158 01435 R 074114 E DAC* HOUT /*12JAH12* TELEPRINTER INT CRASH IN ODT
2159 01436 R 200607 R LAC ACSWR /SETUP DDTSW.
2160 01437 R 513653 R AND (40
2161 01440 R 040574 R DAC DDTSW /NON-ZERO=MINI-ODT ; 0=MUMPS OR MONITOR
2162 01441 R 750200 A CLOCK5 SZA!CLA
2163 01442 R 601454 R JMP CLOCK4
2164 01443 R 200607 R LAC ACSWR
2165 01444 R 513655 R AND (37 /37 - GET TERMINAL NUMBER
2166 01445 R 652000 A LMQ
2167 01446 R 353656 R TAD (-NUMAT
2168 01447 R 755100 A SPA!CLA!CLL
2169 01450 R 641002 A LACQ
2170 01451 R 640711 A ALS 11 /POSITION FOR COMPARE IN DC01 ROUTINE
2171 01452 R 540575 R SAD MONTTY
2172 01453 R 601456 R JMP CLOCK1 /SAME AS LAST TIME
2173 01454 R 140576 R CLOCK4 DZM CONSW
2174 01455 R 040575 R DAC MONTTY
2175 01456 R 200610 R CLOCK1 LAC ACSWRX /UPDATE SYSTAB ENTRIES
2176 01457 R 740004 A OAS
2177 01460 R 040662 R DAC ACSWRO

```

```

2178 01461 R 750004 A LAS
2179 01462 R 040607 R DAC ACSWR
2180 01463 R 700314 A IORS
2181 01464 R 040606 R DAC .IORS
2182 01465 R 200656 R LAC RUNDOWN
2183 01466 R 741100 A SPA /EXIT, USER NOT SWAPPED IN.
2184 01467 R 440656 R ISZ RUNDOWN /BUMP SLICE COUNT AND EXIT IF TIME NOT UP
2185 01470 R 601075 R JMP EXIT
2186 01471 R 200652 R LAC SWAPSW /SLICE TIME IS UP.
2187 01472 R 750201 A SZA!CLC /CHECK FOR SWAP INHIBIT
2188 01473 R 601476 R JMP CLKHNG
2189 01474 R 040652 R DAC SWAPSW /SW WAS SET FOR NO SWAP;
2190 /RESET FOR "SWAP-WHEN-DONE"
2191 01475 R 601075 R JMP EXIT
2192 /
2193 703304 A DBK=703304
2194 01476 R 200601 R CLKHNG LAC PRVDKA /WAS CURRENT PART. LAST
2195 01477 R 540654 R SAD PRTNMB /ONE TO USE DISK?
2196 01500 R 741000 A SKP /YES
2197 01501 R 601504 R JMP NUSEDK /NO
2198 01502 R 200716 R LAC DSKREP
2199 01503 R 040600 R DAC DSKCNT /RESET MAX # OF CONSEC DISK ACCESSES
2200 001504 R NUSEDK=.
2201 .IFPNZ API
2202 01504 R 703304 A DBK /FREE UP API LEVEL
2203 .ENDC
2204 01505 R 200562 R LAC INTX
2205 01506 R 721000 A PAX
2206 01507 R 200563 R LAC INTL
2207 01510 R 722000 A PAL
2208 01511 R 200561 R LAC INTMQ
2209 01512 R 652000 A LMQ
2210 01513 R 200000 A LAC 0 /CHECK FOR PC AT HANG
2211 01514 R 513657 R AND (77777
2212 01515 R 553660 R SAD (HANG+1
2213 01516 R 601523 R JMP CLOCK3
2214 01517 R 200560 R LAC INTAC /SWAP FROM SLICE OVERFLOW
2215 01520 R 040156 R DAC AC
2216 01521 R 200000 A LAC 0
2217 01522 R 601545 R JMP HANG.4
2218 01523 R 200560 R CLOCK3 LAC INTAC /SWAP FROM JMS HANG
2219 .IFNOZ XVM /*12JAH15*
2220 JMP HANG.2
2221 .ENDC /*12JAH15*
2222 .IFPNZ XVM /*12JAH15*
2223 01524 R 040156 R DAC AC /SAVE ACCUMULATOR /*12JAH15*
2224 01525 R 200000 A LAC 0 /FETCH INT. ADDR /*12JAH15*
2225 01526 R 513661 R AND (700000 /EXTRACT LINK,BANK,USER STATUS /*12JAH15*
2226 01527 R 241533 R XOR HANG /INSERT RETURN ADDR /*12JAH15*
2227 01530 R 601545 R JMP HANG.4 /JOIN HANG ROUTINE /*12JAH15*
2228 .ENDC /*12JAH15*
2229 /

```

2230 01531 R 777704 A SECOND -CLKHZ /SIXTY TICKS OR WHAT HAVE YOU
2231 01532 R 000000 A CLKTMP 0

```

2232          .TITLE  MONITOR
2233          /HANG SUBROUTINE: SWAPOUT, STATUS CHECK, SWAPIN
2234          /
2235          /PRTNMB IS NEXT PARTITION TO RUN
2236          /PRTIDX IS LAST PARTITION
2237          /SCOUNT IS TEMPORARY POINTER
2238          /MONIT IS 0 IF NO LAST USER, ONE LESS THAN START OF PARTITION OTHERWISE
2239          /
2240          01533 R 740040 A   HANG   XX
2241          01534 R 705522 A           INTOFF
2242          01535 R 040156 R   HANG.2  DAC     AC
2243          01536 R 201533 R           LAC     HANG
2244          .IFPNZ  XVM
2245          01537 R 741400 A           SZL           /LINK SET?
2246          01540 R 253662 R           XOR     (400000 /YES
2247          707761 A   SBA=707761
2248          01541 R 707761 A           SBA           /BANK MODE?
2249          01542 R 741000 A           SKP           /NO
2250          01543 R 253663 R           XOR     (200000 /YES
2251          01544 R 253664 R           XOR     (100000 /ALWAYS IN USER MODE
2252          .ENDC
2253          01545 R 040155 R   HANG.4  DAC     PC           /SLICE OVERFLOW ENTRY
2254          01546 R 750030 A           CLA!IAC      /+1 TO RUNDOWN
2255          01547 R 040656 R           DAC     RUNDOWN
2256          01550 R 705521 A           INTDN
2257          01551 R 641002 A           LACQ
2258          01552 R 040157 R           DAC     MQ
2259          01553 R 724000 A           PXA
2260          01554 R 040160 R           DAC     XR
2261          01555 R 730000 A           PLA
2262          01556 R 040161 R           DAC     LR
2263          .IFPNZ  GGROW
2264          001557 R   STACH8=.
2265          .ENDC
2266          01557 R 200571 R   STACH   LAC     SLPCNT      /HAS A SECOND (OR MORE) ELAPSED /*12JAH17*
2267          01560 P 740200 A           SZA
2268          01561 R 603102 R           JMP     SLPCHK      /SINCE LAST UPDATE OR SLEEP HUNG PARTS. /*12JAH17*
2269          01562 R 140571 R   STACH0  DZM     SLPCNT      /YES, GO UPDATE SYMEND & PRTEND /*12JAH17*
2270          01563 R 200662 R           LAC     ACSWRO      /CLEAR COUNT /*12JAH17*
2271          01564 P 513650 R           AND     (600000     /CHECK FOR SYSTEM CLOSE /*12JAH17*
2272          01565 R 253650 R           XOR     (600000     /SWITCHES 0 AND 1
2273          01566 R 040663 R           DAC     HLTSWT
2274          01567 R 200657 R           LAC     PRTIDX
2275          01570 R 140654 R   STACH1  DZM     PRTNMB
2276          01571 R 740030 A   STACH4  IAC
2277          01572 R 540330 R           SAD     STATX
2278          01573 R 213665 R           LAC     (STAT1
2279          01574 R 040565 R           DAC     SCOUNT
2280          01575 R 750004 A           LAS
2281          01576 R 553666 R           SAD     (4000
2282          01577 R 601623 R           JMP     STACH7
2283          01600 R 220565 R   STACH9  LAC*   SCOUNT
2283          /*12JAH10* NO, CHECK STATUS.

```

```

2284 01601 R 253667 R XOR (1
2285 01602 R 513670 R AND (4377
2286 01603 R 744200 A SZA!CLL /PARTITION RUNNABLE?
2287 01604 R 601632 R JMP STACH3 /NOT RUNNABLE
2288 01605 R 200654 R LAC PRTNMB
2289 01606 R 740200 A SZA /FIRST RUNNABLE?
2290 01607 R 601612 R JMP .+3 /NO
2291 01610 R 200565 R LAC SCOUNT
2292 01611 R 040654 R DAC PRTNMB
2293 01612 R 777377 A LAW 17377 /CHECK IF PRIORITY RUN
2294 01613 R 705522 A INTOFF
2295 01614 R 520565 R AND* SCOUNT
2296 01615 R 560565 R SAD* SCOUNT
2297 01616 R 601632 R JMP STACH3 /NO
2298 01617 R 060565 R DAC* SCOUNT /YES, RESET PRIORITY BIT.
2299 01620 R 200565 R STACH2 LAC SCOUNT
2300 01621 R 040654 R DAC PRTNMB
2301 01622 R 601644 R JMP EXEC
2302 /
2303 01623 R 200577 R STACH7 LAC DSKACT /GLOBAL DATA /*12JAH17*
2304 01624 R 740200 A SZA /*12JAH10* SWITCH OWNED?
2305 01625 R 601600 R JMP STACH9 /*12JAH10* YES, NO EMERGENCY CLOSE NOW
2306 01626 P 440570 R ISZ EMERG /INDICATE SWITCH 6 IS ON /*12JAH17*
2307 01627 R 220565 R LAC* SCOUNT /*12JAH10* DO EMERGENCY CLOSE
2308 01630 R 744200 A SZA!CLL / IS PARTITION ACTIVE
2309 01631 R 601620 R JMP STACH2 /YES, RUN IT REGARDLESS OF STATUS
2310 /
2311 001632 R STACH3=.
2312 .IFPNZ API
2313 01632 R 703304 A DBK
2314 01633 R 700042 A ION
2315 .ENDC
2316 01634 R 705521 A INTON
2317 01635 R 200565 R LAC SCOUNT
2318 01636 R 540657 R SAD PRTIDX /FINISHED LOOP?
2319 01637 R 741000 A SKP /YES
2320 01640 R 601571 R JMP STACH4 /NO
2321 .IFPNZ GGROW
2322 01641 R 200654 R LAC PRTNMB /ANYONE RUNNABLE?
2323 01642 R 745200 A SNA!CLL
2324 .ENDC
2325 01643 R 634077 E JMP* GARB /*12JAH12* NO, LET'S DO SOME GARBAGE COLLECTION
2326 .IFZER GGROW
2327 STACH8 LAC PRTNMB /ANYONE RUNNABLE
2328 SNA!CLL
2329 JMP STACH /NO, LOOP SOME MORE
2330 .ENDC
2331 .EJECT

```

```

2332 /EXECUTIVE FOR 32K CORE OR LESS
2333 /
2334 001644 R EXEC=.
2335 .IFPNZ API
2336 01644 R 703304 A DBK
2337 .ENDC
2338 01645 R 705521 A INTON.
2339 01646 R 400720 R XCT SUBQUES /WHICH DISK CURRENTLY RUNNING?
2340 01647 R 741000 A SKP /SUBMERGED
2341 01650 R 601677 R JMP EXEC2 /SERIAL
2342 01651 R 200577 R LAC DSKACT
2343 01652 R 741200 A SNA /ANYONE USING DISK?
2344 01653 R 601672 R JMP EXEC1 /NO
2345 01654 R 540654 R SAD PRINMR /IS NEW PARTITION USING DISK?
2346 01655 R 601705 R JMP EXEC4 /YES
2347 01656 R 200654 R LAC PRINMR /RESET MONITOR LOOP
2348 01657 R 353671 R TAD (INDTAB-STAT1 /FETCH PTR TO PARTITION START
2349 01660 R 040565 R DAC SCOUNT
2350 01661 R 220565 R LAC* SCOUNT /GET PARTITION STARTING ADDRESS
2351 01662 R 040565 R DAC SCOUNT
2352 01663 R 220565 R LAC* SCOUNT /FETCH CONTENTS OF PC WORD
2353 01664 R 513672 R AND (177777 /STRIP OUT ADDRESS
2354 01665 R 553673 R SAD (PRODRT /IS THIS PARTITION WAITING FOR DISK?
2355 01666 R 741000 A SKP
2356 01667 R 601705 R JMP EXEC4 /NO, SWAP HIM IN
2357 01670 R 200654 R LAC PRINMR /YES, DON'T BOTHER SWAPPING HIM IN
2358 01671 R 601570 R JMP STACH1 /GO GET NEXT ELGIBLE JOB
2359 01672 R 200717 R EXEC1 LAC SUBDSK
2360 01673 R 740300 A SMA!SZA /WHICH DISK DESIRED?
2361 01674 R 601705 R JMP EXEC4 /SUBMERGED, SAME AS ONE RUNNING, PROCEED
2362 01675 R 213674 R LAC (SKP /YES, DO IT!
2363 01676 R 601704 R JMP EXEC3
2364 01677 R 200717 R EXEC2 LAC SUBDSK
2365 01700 R 741300 A SPA!SNA /WHICH DISK DESIRED?
2366 01701 R 601705 R JMP EXEC4 /SERIAL, SAME AS ONE RUNNING, PROCEED
2367 01702 R 140577 R DZM DSKACT /SUBMERGED, ZERO OWNERSHIP
2368 01703 R 213675 R LAC (NOP /SWITCH TO SUBMERGED
2369 01704 R 040720 R EXEC3 DAC SUBQUES
2370 001705 R EXEC4=.
2371 01705 R 200660 R LAC PRIN
2372 01706 R 540654 R SAD PRINMR
2373 01707 R 603000 R JMP SWPIN3 /PARTITION ALREADY SWAPPED IN
2374 01710 R 101725 R JMS SWPOUT /OUT WITH THE OLD
2375 01711 R 102327 R JMS SWPIN /IN WITH THE NEW
2376 01712 R 603000 R JMP SWPIN3
2377 /GENERAL ENTRY TO HANG A DEVICE
2378 /
2379 /SETS APPROPRIATE I/O HANG BIT IN JOB STATUS WORD
2380 /
2381 /CALLING SEQUENCE:
2382 /
2383 / LAC X /X IS BIT

```

2384		/	STL		STL FOR OUTPUT; CLL FOR INPUT
2385		/	JMP	DVHANG	
2386		/			
2387	01713 R 705522 A	DVHANG	INTOFF		
2388	01714 R 040564 R		DAC	IDTMP	
2389	01715 R 740001 A		CMA		/COMPLEMENT AND
2390	01716 R 520654 R		AND*	PRTNMB	/MASK WITH JOB STATUS WORD
2391	01717 R 240564 R		XOR	IDTMP	/SET BIT
2392	01720 R 060654 R		DAC*	PRTNMB	/REPLACE STATUS
2393	01721 R 101533 R	DVHNG1	JMS	HANG	/ENTRY FOR PAPER TAPE READER
2394	01722 R 741400 A		SZL		/INPUT OR OUTPUT?
2395	01723 R 603253 R		JMP	XOUT4A	/OUTPUT
2396	01724 R 603334 R		JMP	XPACK2	/INPUT
2397			.EJECT		

```

2398 /CONTEXT SWITCH OUT ROUTINE
2399 SWPOUT XX
2400 01725 R 740040 A LAC MONIT /ANY "OLD" USER?
2401 01726 R 200661 R SNA
2402 01727 R 741200 A JMP* SWPOUT /NO- SYSTEM WAS IDLE; NOTHING TO SWAPOUT
2403 01730 R 621725 R PAX /*12JAH17*
2404 01731 R 721000 A .DBREL ; DBA /*12JAH17*
    01732 R 707762 A LAC home
2405 /SWAP OUT OLD USER
2406 01733 R 200155 R LAC PC DBA
2407 01734 R 050001 A DAC 1,X 050171 DAC 2,X /*12JAH17*
2408 01735 R 200156 R LAC AC
2409 01736 R 050002 A DAC 2,X /*12JAH17*
2410 01737 R 200157 R LAC MQ
2411 01740 R 050003 A DAC 3,X /*12JAH17*
2412 01741 R 200160 R LAC XR
2413 01742 R 050004 A DAC 4,X /*12JAH17*
2414 01743 R 200161 R LAC LR
2415 01744 R 050005 A DAC 5,X /*12JAH17*
2416 01745 R 200010 A LAC R10
2417 01746 R 050006 A DAC 6,X /*12JAH17*
2418 01747 R 200013 A LAC R13
2419 01750 R 050007 A DAC 7,X /*12JAH17*
2420 01751 R 200014 A LAC R14
2421 01752 R 050010 A DAC 10,X /*12JAH17*
2422 01753 R 200015 A LAC R15
2423 01754 R 050011 A DAC 11,X /*12JAH17*
2424 01755 R 200016 A LAC R16
2425 01756 R 050012 A DAC 12,X /*12JAH17*
2426 01757 R 200017 A LAC R17
2427 01760 R 050013 A DAC 13,X /*12JAH17*
2428 01761 R 200020 A LAC R20
2429 01762 R 050014 A DAC 14,X /*12JAH17*
2430 01763 R 200162 R LAC PRNCDV /*12JAH17*
2431 01764 R 050015 A DAC 15,X /*12JAH17*
2432 01765 R 200163 R LAC INPUT /*12JAH17*
2433 01766 R 050016 A DAC 16,X /*12JAH17*
2434 01767 R 200164 R LAC OUTPUT /*12JAH17*
2435 01770 R 050017 A DAC 17,X /*12JAH17*
2436 01771 R 200165 R LAC STACK /*12JAH17*
2437 01772 R 050020 A DAC 20,X /*12JAH17*
2438 01773 R 200166 R LAC SPOINT /*12JAH17*
2439 01774 R 050021 A DAC 21,X /*12JAH17*
2440 01775 R 200167 R LAC END /*12JAH17*
2441 01776 R 050022 A DAC 22,X /*12JAH17*
2442 01777 R 200170 R LAC SYMBEG /*12JAH17*
2443 02000 R 050023 A DAC 23,X /*12JAH17*
2444 02001 R 200171 R LAC OBPNT /*12JAH17*
2445 02002 R 050024 A DAC 24,X /*12JAH17*
2446 02003 R 200172 R LAC WRISW /*12JAH17*
2447 02004 R 050025 A DAC 25,X /*12JAH17*
2448 02005 R 200173 R LAC SWPCNT /*12JAH17*

```

2449	02006	R	050026	A	DAC	26,X	/*12JAH17*
2450	02007	R	200174	R	LAC	CLKCNT	/*12JAH17*
2451	02010	R	050027	A	DAC	27,X	/*12JAH17*
2452	02011	R	200175	R	LAC	WRTCNT	/*12JAH17*
2453	02012	R	050030	A	DAC	30,X	/*12JAH17*
2454	02013	R	200176	R	LAC	RDCNT	/*12JAH17*
2455	02014	R	050031	A	DAC	31,X	/*12JAH17*
2456	02015	R	200177	R	LAC	SYM	/*12JAH17*
2457	02016	R	050032	A	DAC	32,X	/*12JAH17*
2458	02017	R	200200	R	LAC	SCRIP	/*12JAH17*
2459	02020	R	050033	A	DAC	33,X	/*12JAH17*
2460	02021	R	200201	R	LAC	SAVPRD	/*12JAH17*
2461	02022	R	050034	A	DAC	34,X	/*12JAH17*
2462	02023	R	200202	R	LAC	TEMP	/*12JAH17*
2463	02024	R	050035	A	DAC	35,X	/*12JAH17*
2464	02025	R	200203	R	LAC	FORTST	/*12JAH17*
2465	02026	R	050036	A	DAC	36,X	/*12JAH17*
2466	02027	R	200204	R	LAC	SYMS	/*12JAH17*
2467	02030	R	050037	A	DAC	37,X	/*12JAH17*
2468	02031	R	200205	R	LAC	SCRIPS	/*12JAH17*
2469	02032	R	050040	A	DAC	40,X	/*12JAH17*
2470	02033	R	200206	R	LAC	LOC	/*12JAH17*
2471	02034	R	050041	A	DAC	41,X	/*12JAH17*
2472	02035	R	200207	R	LAC	VAL	/*12JAH17*
2473	02036	R	050042	A	DAC	42,X	/*12JAH17*
2474			000042	A			/*12JAH17*
2475					.IFPNZ	MONEY	/*12JAH17*
2476			000043	A			/*12JAH17*
2477	02037	R	200210	R	LAC	MONYSW	/*12JAH17*
2478	02040	R	050043	A	DAC	Q,X	/*12JAH17*
2479					.ENDC		/*12JAH17*
2480					.IFPNZ	FMONEY	/*12JAH17*
2481			000044	A			/*12JAH17*
2482	02041	R	200211	R	LAC	VALF	/*12JAH17*
2483	02042	R	050044	A	DAC	Q,X	/*12JAH17*
2484			000045	A			/*12JAH17*
2485	02043	R	200212	R	LAC	FLOAT	/*12JAH17*
2486	02044	R	050045	A	DAC	Q,X	/*12JAH17*
2487					.ENDC		/*12JAH17*
2488	02045	R	200213	R	LAC	OFLAG	/*12JAH17*
2489	02046	R	050046	A	DAC	Q+1,X	/*12JAH17*
2490	02047	R	200214	R	LAC	M	/*12JAH17*
2491	02050	R	050047	A	DAC	Q+2,X	/*12JAH17*
2492	02051	R	200215	R	LAC	SUM	/*12JAH17*
2493	02052	R	050050	A	DAC	Q+3,X	/*12JAH17*
2494	02053	R	200216	R	LAC	TERM	/*12JAH17*
2495	02054	R	050051	A	DAC	Q+4,X	/*12JAH17*
2496	02055	R	200217	R	LAC	ACTEMP	/*12JAH17*
2497	02056	R	050052	A	DAC	Q+5,X	/*12JAH17*
2498	02057	R	200220	R	LAC	IPOINT	/*12JAH17*
2499	02060	R	050053	A	DAC	Q+6,X	/*12JAH17*
2500	02061	R	200221	R	LAC	IPLACE	/*12JAH17*

/*12JAH17*

2501	02062	R	050054	A	DAC	Q+7,X	/*12JAH17*
2502	02063	R	200222	R	LAC	OPOINT	/*12JAH17*
2503	02064	R	050055	A	DAC	Q+10,X	/*12JAH17*
2504	02065	R	200223	R	LAC	OPLACE	/*12JAH17*
2505	02066	R	050056	A	DAC	Q+11,X	/*12JAH17*
2506	02067	R	200224	R	LAC	TEMP1	/*12JAH17*
2507	02070	R	050057	A	DAC	Q+12,X	/*12JAH17*
2508	02071	R	200225	R	LAC	POINT1	/*12JAH17*
2509	02072	R	050060	A	DAC	Q+13,X	/*12JAH17*
2510	02073	R	200226	R	LAC	PLACE1	/*12JAH17*
2511	02074	R	050061	A	DAC	Q+14,X	/*12JAH17*
2512	02075	R	200227	R	LAC	TEMP2	/*12JAH17*
2513	02076	R	050062	A	DAC	Q+15,X	/*12JAH17*
2514	02077	R	200230	R	LAC	POINT2	/*12JAH17*
2515	02100	R	050063	A	DAC	Q+16,X	/*12JAH17*
2516	02101	R	200231	R	LAC	PLACE2	/*12JAH17*
2517	02102	R	050064	A	DAC	Q+17,X	/*12JAH17*
2518	02103	R	200232	R	LAC	BSUM	/*12JAH17*
2519	02104	R	050065	A	DAC	Q+20,X	/*12JAH17*
2520	02105	R	200233	R	LAC	BTERM	/*12JAH17*
2521	02106	R	050066	A	DAC	Q+21,X	/*12JAH17*
2522	02107	R	200234	R	LAC	BVAL	/*12JAH17*
2523	02110	R	050067	A	DAC	Q+22,X	/*12JAH17*
2524	02111	R	200235	R	LAC	FORVAR	/*12JAH17*
2525	02112	R	050070	A	DAC	Q+23,X	/*12JAH17*
2526	02113	R	200236	R	LAC	PROG	/*12JAH17*
2527	02114	R	050071	A	DAC	Q+24,X	/*12JAH17*
2528	02115	R	200237	R	LAC	PUSER	/*12JAH17*
2529	02116	R	050072	A	DAC	Q+25,X	/*12JAH17*
2530	02117	R	200240	R	LAC	ID	/*12JAH17*
2531	02120	R	050073	A	DAC	Q+26,X	/*12JAH17*
2532	02121	R	200241	R	LAC	USER	/*12JAH17*
2533	02122	R	050074	A	DAC	Q+27,X	/*12JAH17*
2534	02123	R	200242	R	LAC	BKCHAR	/*12JAH17*
2535	02124	R	050075	A	DAC	Q+30,X	/*12JAH17*
2536	02125	R	200243	R	LAC	T1	/*12JAH17*
2537	02126	R	050076	A	DAC	Q+31,X	/*12JAH17*
2538	02127	R	200244	R	LAC	T2	/*12JAH17*
2539	02130	R	050077	A	DAC	Q+32,X	/*12JAH17*
2540	02131	R	200245	R	LAC	T3	/*12JAH17*
2541	02132	R	050100	A	DAC	Q+33,X	/*12JAH17*
2542	02133	R	200246	R	LAC	IFSWT	/*12JAH17*
2543	02134	R	050101	A	DAC	Q+34,X	/*12JAH17*
2544	02135	R	200247	R	LAC	GUNIT	/*12JAH17*
2545	02136	R	050102	A	DAC	Q+35,X	/*12JAH17*
2546	02137	R	200250	R	LAC	BUNIT	/*12JAH17*
2547	02140	R	050103	A	DAC	Q+36,X	/*12JAH17*
2548	02141	R	200251	R	LAC	GBLKHT	/*12JAH17*
2549	02142	R	050104	A	DAC	Q+37,X	/*12JAH17*
2550	02143	R	200252	R	LAC	GHI	/*12JAH17*
2551	02144	R	050105	A	DAC	Q+40,X	/*12JAH17*
2552	02145	R	200253	R	LAC	GHEAD	/*12JAH17*

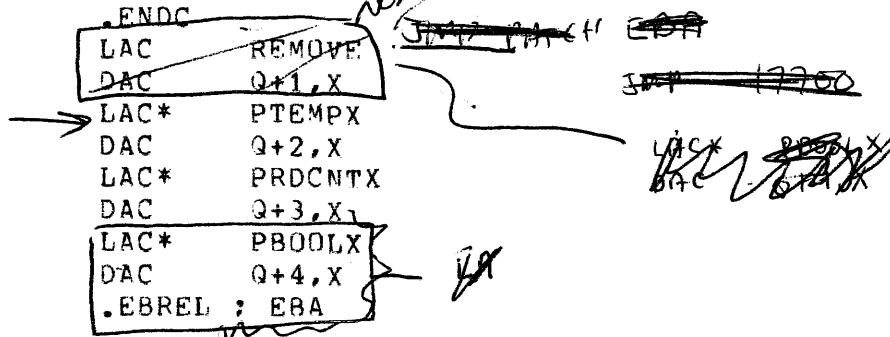
2553	02146	R	050106	A	DAC	Q+41,X	/*12JAH17*
2554	02147	R	200254	R	LAC	GSYMBOL	/*12JAH17*
2555	02150	R	050107	A	DAC	Q+42,X	/*12JAH17*
2556	02151	R	200255	R	LAC	GHDOLD	/*12JAH17*
2557	02152	R	050110	A	DAC	Q+43,X	/*12JAH17*
2558	02153	R	207041	R	LAC	WRPART	/*12JAH17*
2559	02154	R	050111	A	DAC	Q+44,X	/*12JAH17*
2560	02155	R	206621	R	LAC	CONT	/*12JAH17*
2561	02156	R	050112	A	DAC	Q+45,X	/*12JAH17*
2562	02157	R	206203	R	LAC	BOOL	/*12JAH17*
2563	02160	R	050113	A	DAC	Q+46,X	/*12JAH17*
2564	02161	R	200256	R	LAC	BOOLL	/*12JAH17*
2565	02162	R	050114	A	DAC	Q+47,X	/*12JAH17*
2566	02163	R	200260	R	LAC	BOOLE	/*12JAH17*
2567	02164	R	050115	A	DAC	Q+50,X	/*12JAH17*
2568	02165	R	200262	R	LAC	BOOLG	/*12JAH17*
2569	02166	R	050116	A	DAC	Q+51,X	/*12JAH17*
2570	02167	R	204770	R	LAC	CHAR1	/*12JAH17*
2571	02170	R	050117	A	DAC	Q+52,X	/*12JAH17*
2572	02171	R	205014	R	LAC	CHAR2	/*12JAH17*
2573	02172	R	050120	A	DAC	Q+53,X	/*12JAH17*
2574	02173	R	205201	R	LAC	CHECK	/*12JAH17*
2575	02174	R	050121	A	DAC	Q+54,X	/*12JAH17*
2576	02175	R	203535	R	LAC	DISPAT	/*12JAH17*
2577	02176	R	050122	A	DAC	Q+55,X	/*12JAH17*
2578	02177	R	200264	R	LAC	DISLOP	/*12JAH17*
2579	02200	R	050123	A	DAC	Q+56,X	/*12JAH17*
2580	02201	R	203641	R	LAC	PRODA	/*12JAH17*
2581	02202	R	050124	A	DAC	Q+57,X	/*12JAH17*
2582	02203	R	204340	R	LAC	PLACE	/*12JAH17*
2583	02204	R	050125	A	DAC	Q+60,X	/*12JAH17*
2584	02205	R	204645	R	LAC	OCHAR	/*12JAH17*
2585	02206	R	050126	A	DAC	Q+61,X	/*12JAH17*
2586	02207	R	204660	R	LAC	OCHART	/*12JAH17*
2587	02210	R	050127	A	DAC	Q+62,X	/*12JAH17*
2588	02211	R	204710	R	LAC	SUBOPT	/*12JAH17*
2589	02212	R	050130	A	DAC	Q+63,X	/*12JAH17*
2590	02213	R	205040	R	LAC	SAVE	/*12JAH17*
2591	02214	R	050131	A	DAC	Q+64,X	/*12JAH17*
2592	02215	R	205103	R	LAC	UNSAVF	/*12JAH17*
2593	02216	R	050132	A	DAC	Q+65,X	/*12JAH17*
2594	02217	R	205123	R	LAC	STEP	/*12JAH17*
2595	02220	R	050133	A	DAC	Q+66,X	/*12JAH17*
2596	02221	R	205151	R	LAC	PART	/*12JAH17*
2597	02222	R	050134	A	DAC	Q+67,X	/*12JAH17*
2598	02223	R	207163	R	LAC	EVAL	/*12JAH17*
2599	02224	R	050135	A	DAC	Q+70,X	/*12JAH17*
2600	02225	R	205434	R	LAC	SYMBOL	/*12JAH17*
2601	02226	R	050136	A	DAC	Q+71,X	/*12JAH17*
2602	02227	R	203724	R	LAC	SYNTAX	/*12JAH17*
2603	02230	R	050137	A	DAC	Q+72,X	/*12JAH17*
2604	02231	R	203744	R	LAC	STOR1	/*12JAH17*

2605	02232	R	050140	A	DAC	Q+73,X	/*12JAH17*
2606	02233	R	204034	R	LAC	STOR2	/*12JAH17*
2607	02234	R	050141	A	DAC	Q+74,X	/*12JAH17*
2608	02235	R	204103	R	LAC	DECIN	/*12JAH17*
2609	02236	R	050142	A	DAC	Q+75,X	/*12JAH17*
2610	02237	R	204255	R	LAC	DECDUT	/*12JAH17*
2611	02240	R	050143	A	DAC	Q+76,X	/*12JAH17*
2612	02241	R	204320	R	LAC	OCTOUT	/*12JAH17*
2613	02242	R	050144	A	DAC	Q+77,X	/*12JAH17*
2614	02243	R	206673	R	LAC	FORMAT	/*12JAH17*
2615	02244	R	050145	A	DAC	Q+100,X	/*12JAH17*
2616	02245	R	205304	R	LAC	ASSEM	/*12JAH17*
2617	02246	R	050146	A	DAC	Q+101,X	/*12JAH17*
2618	02247	R	200266	R	LAC	GASS	/*12JAH17*
2619	02250	R	050147	A	DAC	Q+102,X	/*12JAH17*
2620	02251	R	203174	R	LAC	OUT	/*12JAH17*
2621	02252	R	050150	A	DAC	Q+103,X	/*12JAH17*
2622	02253	R	204725	R	LAC	ICHARI	/*12JAH17*
2623	02254	R	050151	A	DAC	Q+104,X	/*12JAH17*
2624	02255	R	204235	R	LAC	DEC2	/*12JAH17*
2625	02256	R	050152	A	DAC	Q+105,X	/*12JAH17*
2626	02257	R	203325	R	LAC	PACKEM	/*12JAH17*
2627	02260	R	050153	A	DAC	Q+106,X	/*12JAH17*
2628	02261	R	207066	R	LAC	WRSTEP	/*12JAH17*
2629	02262	R	050154	A	DAC	Q+107,X	/*12JAH17*
2630	02263	R	203712	R	LAC	OTEXT	/*12JAH17*
2631	02264	R	050155	A	DAC	Q+110,X	/*12JAH17*
2632	02265	R	205363	R	LAC	SVE	/*12JAH17*
2633	02266	R	050156	A	DAC	Q+111,X	/*12JAH17*
2634	02267	R	207107	R	LAC	SETIO	/*12JAH17*
2635	02270	R	050157	A	DAC	Q+112,X	/*12JAH17*
2636			000157	A			
2637					Q=Q+112	.IFPNZ DTA	/*12JAH17*
2638			000160	A	Q=Q+1		/*12JAH17*
2639	02271	R	203701	R	LAC	LOKDT	/*12JAH17*
2640	02272	R	050160	A	DAC	Q,X	/*12JAH17*
2641						.ENDC	/*12JAH17*
2642	02273	R	205272	R	LAC	KEYCHK	/*12JAH17*
2643	02274	R	050161	A	DAC	Q+1,X	/*12JAH17*
2644	02275	R	205242	R	LAC	GETKEY	/*12JAH17*
2645	02276	R	050162	A	DAC	Q+2,X	/*12JAH17*
2646	02277	R	207021	R	LAC	TSTR	/*12JAH17*
2647	02300	R	050163	A	DAC	Q+3,X	/*12JAH17*
2648	02301	R	200270	R	LAC	GSYM	/*12JAH17*
2649	02302	R	050164	A	DAC	Q+4,X	/*12JAH17*
2650	02303	R	200272	R	LAC	GLOBAL	/*12JAH17*
2651	02304	R	050165	A	DAC	Q+5,X	/*12JAH17*
2652	02305	R	204357	R	LAC	PLAC1	/*12JAH17*
2653	02306	R	050166	A	DAC	Q+6,X	/*12JAH17*
2654			000166	A	Q=Q+6		/*12JAH17*
2655						.IFPNZ MODIFY	/*12JAH17*
2656			000167	A	Q=Q+1		/*12JAH17*

2657 02307 R 204616 R
 2658 02310 R 050167 A
 2659 02311 R 204573 R
 2660 02312 R 050170 A
 2661 000170 A
 2662
 2663 B 02313 R 210005 R
 2664 02314 R 050171 A
 2665 02315 R 220274 R
 2666 02316 R 050172 A
 2667 02317 R 220276 R
 2668 02320 R 050173 A
 2669 02321 R 220275 R
 2670 02322 R 050174 A
 2671
 02323 R 707764 A
 2672 02324 R 140661 R
 2673 02325 R 140660 R
 2674 02326 R 621725 R
 2675

LAC ERMDD
 DAC Q,X
 LAC MBUF2
 DAC Q+1,X
 .ENDC
 LAC REMOVE
 DAC Q+1,X
 LAC* PTEMPX
 DAC Q+2,X
 LAC* PRDCNTX
 DAC Q+3,X
 LAC* PBOOLX
 DAC Q+4,X
 .EBREL : EBA
 DZM MONIT
 DZM PRIN
 JMP* SWPOUT
 .EJECT

Q=Q+1



/*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*
 /*12JAH17*

17700 / LAC REMOVE
 DBA
 DAC Q+1, X
 JMP* +1
 Address of next instr

```

2676 /SWAP IN NEW USER
2677 SWPIN XX
2678 LAC PRTNMB
2679 DAC PRTIDX
2680 DAC PRTIN
2681 ADD (-STAT1
2682 DAC RUNNMB /SET PARTITION NO. NOW RUNNING.
2683 LAC (SYMBOT
2684 DAC R17
2685 LAW -SYSVAR
2686 DAC R20 /LENGTH OF SYST. VARIABLE LIST
2687 LAC PRTIDX
2688 TAD (-STAT1
2689 DAC PC /TEMP SAVE PART. # -1
2690 TAD (INDTAB
2691 DAC AC
2692 LAC* AC /PART. START. ADDR.
2693 AAC -1
2694 DAC MONIT /SAVE FOR SWAP OUT AND SET GARB SWITCH
2695 PAX
2696 /
2697 /COMPUTE SYSTEM PARAMETERS FOR THIS PARTITION.
2698 /
2699 ADD (SWAPSZ+1
2700 DAC DSKBFF /ADDRESS OF DISK BUFFER
2701 IAC
2702 DAC GBWD2 /"GLOBAL BUFFER WORD 2"
2703 ADD (BLOCK-1
2704 DAC BUFF1 /ADDRESS OF DIRECT COMMAND BUFFER
2705 ADD (COMBUFF
2706 DAC BUFF2 /ADDRESS OF STRING ACCUMULATOR
2707 ADD (STRACC
2708 DAC BUFF /START OF PROGRAM BUFFER
2709 TAD USABLE
2710 DAC SYMBOT /ADDRESS OF BOTTOM OF USER DEFINED
2711 /SYMBOL TABLE (EXCLUSIVE OF PERMANENT
2712 /SYMBOLS)
2713 LAC PC /PART. # -1
2714 TAD PASTAB
2715 DAC PC
2716 LAC* PC /PART. SIZE
2717 TAD SYMBOT
2718 DAC SYMBOT
2719 ADD (4 /4 CELLS FOR DATE
2720 SWPIN4 ADD (2 /SETUP SYSTEM VARIABLES
2721 DAC* R17
2722 ISZ R20 /-SYSVAR PREVIOUSLY PLACED IN R20
2723 JMP SWPIN4
2724 IAC
2725 DAC SYMEND /PRTBEG+PRTSIZE-2
2726 IAC
2727 DAC PRTEND /PRTBEG+PRTSIZE-1

```

EBREL
EBA
DBA
-DBREL

```

2728          /SWAP IN SWAPPING IMAGE.
2729          /
2730          .DBREL ; DBA                                     /*12JAH17*
02405 R 707762 A
2731          02406 R 210001 A          LAC          1,X                                     /*12JAH17*
2732          02407 R 040155 R          DAC          PC
2733          02410 R 210002 A          LAC          2,X                                     /*12JAH17*
2734          02411 R 040156 R          DAC          AC
2735          02412 R 210003 A          LAC          3,X                                     /*12JAH17*
2736          02413 R 040157 R          DAC          MQ
2737          02414 R 210004 A          LAC          4,X                                     /*12JAH17*
2738          02415 R 040160 R          DAC          XR
2739          02416 R 210005 A          LAC          5,X                                     /*12JAH17*
2740          02417 R 040161 R          DAC          LR
2741          02420 R 210006 A          LAC          6,X                                     /*12JAH17*
2742          02421 R 040010 A          DAC          R10
2743          02422 R 210007 A          LAC          7,X                                     /*12JAH17*
2744          02423 R 040013 A          DAC          R13
2745          02424 R 210010 A          LAC          10,X                                    /*12JAH17*
2746          02425 R 040014 A          DAC          R14
2747          02426 R 210011 A          LAC          11,X                                    /*12JAH17:
2748          02427 R 040015 A          DAC          R15
2749          02430 R 210012 A          LAC          12,X                                    /*12JAH17*
2750          02431 R 040016 A          DAC          R16
2751          02432 R 210013 A          LAC          13,X                                    /*12JAH17*
2752          02433 R 040017 A          DAC          R17
2753          02434 R 210014 A          LAC          14,X                                    /*12JAH17*
2754          02435 R 040020 A          DAC          R20
2755          02436 R 210015 A          LAC          15,X                                    /*12JAH17*
2756          02437 R 040162 R          DAC          PRNCDV                                /*12JAH17*
2757          02440 R 210016 A          LAC          16,X                                    /*12JAH17*
2758          02441 R 040163 R          DAC          INPUT                                /*12JAH17*
2759          02442 R 210017 A          LAC          17,X                                    /*12JAH17*
2760          02443 R 040164 R          DAC          OUTPUT                                /*12JAH17*
2761          02444 R 210020 A          LAC          20,X                                    /*12JAH17*
2762          02445 R 040165 R          DAC          STACK                                /*12JAH17*
2763          02446 R 210021 A          LAC          21,X                                    /*12JAH17*
2764          02447 R 040166 R          DAC          SPOINT                                /*12JAH17*
2765          02450 R 210022 A          LAC          22,X                                    /*12JAH17*
2766          02451 R 040167 R          DAC          END                                /*12JAH17*
2767          02452 R 210023 A          LAC          23,X                                    /*12JAH17*
2768          02453 R 040170 R          DAC          SYMBEG                                /*12JAH17*
2769          02454 R 210024 A          LAC          24,X                                    /*12JAH17*
2770          02455 R 040171 R          DAC          DBPNT                                /*12JAH17*
2771          02456 R 210025 A          LAC          25,X                                    /*12JAH17*
2772          02457 R 040172 R          DAC          WRTSW                                /*12JAH17*
2773          02460 R 210026 A          LAC          26,X                                    /*12JAH17*
2774          02461 R 040173 R          DAC          SWPCNT                                /*12JAH17*
2775          02462 R 210027 A          LAC          27,X                                    /*12JAH17*
2776          02463 R 040174 R          DAC          CLKCNT                                /*12JAH17*
2777          02464 R 210030 A          LAC          30,X                                    /*12JAH17*
2778          02465 R 040175 R          DAC          WRTCNT                                /*12JAH17*

```

2779	02466	R	210031	A	LAC	31,X	/*12JAH17*
2780	02467	R	040176	R	DAC	RDCNT	/*12JAH17*
2781	02470	R	210032	A	LAC	32,X	/*12JAH17*
2782	02471	R	040177	R	DAC	SYM	/*12JAH17*
2783	02472	R	210033	A	LAC	33,X	/*12JAH17:
2784	02473	R	040200	R	DAC	SCRIP	/*12JAH17*
2785	02474	R	210034	A	LAC	34,X	/*12JAH17:
2786	02475	R	040201	R	DAC	SAVPRD	/*12JAH17*
2787	02476	R	210035	A	LAC	35,X	/*12JAH17*
2788	02477	R	040202	R	DAC	TEMP	/*12JAH17*
2789	02500	R	210036	A	LAC	36,X	/*12JAH17*
2790	02501	R	040203	R	DAC	FORTST	/*12JAH17*
2791	02502	R	210037	A	LAC	37,X	/*12JAH17*
2792	02503	R	040204	R	DAC	SYMS	/*12JAH17*
2793	02504	R	210040	A	LAC	40,X	/*12JAH17*
2794	02505	R	040205	R	DAC	SCRIPS	/*12JAH17*
2795	02506	R	210041	A	LAC	41,X	/*12JAH17*
2796	02507	R	040206	R	DAC	LOC	/*12JAH17*
2797	02510	R	210042	A	LAC	42,X	/*12JAH17*
2798	02511	R	040207	R	DAC	VAL	/*12JAH17*
2799			000042	A		Q=42	/*12JAH17*
2800					.IFPNZ	MONEY	/*12JAH17*
2801			000043	A		Q=Q+1	/*12JAH17*
2802	02512	R	210043	A	LAC	Q,X	/*12JAH17*
2803	02513	R	040210	R	DAC	MONYSW	/*12JAH17*
2804					.ENDC		/*12JAH17*
2805					.IFPNZ	FMONEY	/*12JAH17*
2806			000044	A		Q=Q+1	/*12JAH17*
2807	02514	R	210044	A	LAC	Q,X	/*12JAH17*
2808	02515	R	040211	R	DAC	VALF	/*12JAH17*
2809			000045	A		Q=Q+1	/*12JAH17*
2810	02516	R	210045	A	LAC	Q,X	/*12JAH17*
2811	02517	R	040212	R	DAC	FLOAT	/*12JAH17*
2812					.ENDC		/*12JAH17*
2813	02520	R	210046	A	LAC	Q+1,X	/*12JAH17*
2814	02521	R	040213	R	DAC	OFLAG	/*12JAH17*
2815	02522	R	210047	A	LAC	Q+2,X	/*12JAH17*
2816	02523	R	040214	R	DAC	M	/*12JAH17*
2817	02524	R	210050	A	LAC	Q+3,X	/*12JAH17*
2818	02525	R	040215	R	DAC	SUM	/*12JAH17*
2819	02526	R	210051	A	LAC	Q+4,X	/*12JAH17*
2820	02527	R	040216	R	DAC	TERM	/*12JAH17*
2821	02530	R	210052	A	LAC	Q+5,X	/*12JAH17*
2822	02531	R	040217	R	DAC	ACTEMP	/*12JAH17*
2823	02532	R	210053	A	LAC	Q+6,X	/*12JAH17*
2824	02533	R	040220	R	DAC	IPOINT	/*12JAH17*
2825	02534	R	210054	A	LAC	Q+7,X	/*12JAH17*
2826	02535	R	040221	R	DAC	IPLACE	/*12JAH17*
2827	02536	R	210055	A	LAC	Q+10,X	/*12JAH17*
2828	02537	R	040222	R	DAC	OPOINT	/*12JAH17*
2829	02540	R	210056	A	LAC	Q+11,X	/*12JAH17*
2830	02541	R	040223	R	DAC	OPLACE	/*12JAH17*

2831	02542	R	210057	A	LAC	Q+12,X	/*12JAH17*
2832	02543	R	040224	R	DAC	TEMP1	/*12JAH17*
2833	02544	R	210060	A	LAC	Q+13,X	/*12JAH17*
2834	02545	R	040225	R	DAC	POINT1	/*12JAH17*
2835	02546	R	210061	A	LAC	Q+14,X	/*12JAH17*
2836	02547	R	040226	R	DAC	PLACE1	/*12JAH17*
2837	02550	R	210062	A	LAC	Q+15,X	/*12JAH17*
2838	02551	R	040227	R	DAC	TEMP2	/*12JAH17*
2839	02552	R	210063	A	LAC	Q+16,X	/*12JAH17*
2840	02553	R	040230	R	DAC	POINT2	/*12JAH17*
2841	02554	R	210064	A	LAC	Q+17,X	/*12JAH17*
2842	02555	R	040231	R	DAC	PLACE2	/*12JAH17*
2843	02556	R	210065	A	LAC	Q+20,X	/*12JAH17*
2844	02557	R	040232	R	DAC	BSUM	/*12JAH17*
2845	02560	R	210066	A	LAC	Q+21,X	/*12JAH17*
2846	02561	R	040233	R	DAC	BTERM	/*12JAH17*
2847	02562	R	210067	A	LAC	Q+22,X	/*12JAH17*
2848	02563	R	040234	R	DAC	BVAL	/*12JAH17*
2849	02564	R	210070	A	LAC	Q+23,X	/*12JAH17Z*
2850	02565	R	040235	R	DAC	FORVAR	/*12JAH17*
2851	02566	R	210071	A	LAC	Q+24,X	/*12JAH17*
2852	02567	R	040236	R	DAC	PROG	/*12JAH17*
2853	02570	R	210072	A	LAC	Q+25,X	/*12JAH17*
2854	02571	R	040237	R	DAC	PUSER	/*12JAH17*
2855	02572	R	210073	A	LAC	Q+26,X	/*12JAH17*
2856	02573	R	040240	R	DAC	ID	/*12JAH17*
2857	02574	R	210074	A	LAC	Q+27,X	/*12JAH17*
2858	02575	R	040241	R	DAC	USER	/*12JAH17*
2859	02576	R	210075	A	LAC	Q+30,X	/*12JAH17*
2860	02577	R	040242	R	DAC	BKCHAR	/*12JAH17*
2861	02600	R	210076	A	LAC	Q+31,X	/*12JAH17*
2862	02601	R	040243	R	DAC	T1	/*12JAH17*
2863	02602	R	210077	A	LAC	Q+32,X	/*12JAH17*
2864	02603	R	040244	R	DAC	T2	/*12JAH17*
2865	02604	R	210100	A	LAC	Q+33,X	/*12JAH17*
2866	02605	R	040245	R	DAC	T3	/*12JAH17*
2867	02606	R	210101	A	LAC	Q+34,X	/*12JAH17*
2868	02607	R	040246	R	DAC	IFSWT	/*12JAH17*
2869	02610	R	210102	A	LAC	Q+35,X	/*12JAH17*
2870	02611	R	040247	R	DAC	GUNIT	/*12JAH17*
2871	02612	R	210103	A	LAC	Q+36,X	/*12JAH17*
2872	02613	R	040250	R	DAC	BUNIT	/*12JAH17*
2873	02614	R	210104	A	LAC	Q+37,X	/*12JAH17*
2874	02615	R	040251	R	DAC	GBLKHI	/*12JAH17*
2875	02616	R	210105	A	LAC	Q+40,X	/*12JAH17*
2876	02617	R	040252	R	DAC	GHI	/*12JAH17*
2877	02620	R	210106	A	LAC	Q+41,X	/*12JAH17*
2878	02621	R	040253	R	DAC	GHEAD	/*12JAH17*
2879	02622	R	210107	A	LAC	Q+42,X	/*12JAH17*
2880	02623	R	040254	R	DAC	GSYMBOL	/*12JAH17*
2881	02624	R	210110	A	LAC	Q+43,X	/*12JAH17*
2882	02625	R	040255	R	DAC	GHDOLD	/*12JAH17*

2883	02626	R	210111	A	LAC	Q+44,X	/*12JAH17*
2884	02627	R	047041	R	DAC	WRPART	/*12JAH17*
2885	02630	R	210112	A	LAC	Q+45,X	/*12JAH17*
2886	02631	R	046621	R	DAC	CONT	
2887	02632	R	210113	A	LAC	Q+46,X	/*12JAH17*
2888	02633	R	046203	R	DAC	BOOL	/*12JAH17*
2889	02634	R	210114	A	LAC	Q+47,X	/*12JAH17*
2890	02635	R	040256	R	DAC	BOOLL	/*12JAH17*
2891	02636	R	210115	A	LAC	Q+50,X	/*12JAH17*
2892	02637	R	040260	R	DAC	BOOLE	/*12JAH17*
2893	02640	R	210116	A	LAC	Q+51,X	/*12JAH17*
2894	02641	R	040262	R	DAC	BOOLG	/*12JAH17*
2895	02642	R	210117	A	LAC	Q+52,X	/*12JAH17*
2896	02643	R	044770	R	DAC	CHAR1	/*12JAH17*
2897	02644	R	210120	A	LAC	Q+53,X	/*12JAH17*
2898	02645	R	045014	R	DAC	CHAR2	/*12JAH17*
2899	02646	R	210121	A	LAC	Q+54,X	/*12JAH17*
2900	02647	R	045201	R	DAC	CHECK	/*12JAH17*
2901	02650	R	210122	A	LAC	Q+55,X	/*12JAH17*
2902	02651	R	043535	R	DAC	DISPAT	/*12JAH17*
2903	02652	R	210123	A	LAC	Q+56,X	/*12JAH17*
2904	02653	R	040264	R	DAC	DISLOP	/*12JAH17*
2905	02654	R	210124	A	LAC	Q+57,X	/*12JAH17*
2906	02655	R	043641	R	DAC	PRODA	/*12JAH17*
2907	02656	R	210125	A	LAC	Q+60,X	/*12JAH17*
2908	02657	R	044340	R	DAC	PLACE	/*12JAH17*
2909	02660	R	210126	A	LAC	Q+61,X	/*12JAH17*
2910	02661	R	044645	R	DAC	OCHAR	/*12JAH17*
2911	02662	R	210127	A	LAC	Q+62,X	/*12JAH17*
2912	02663	R	044660	R	DAC	OCHARI	/*12JAH17*
2913	02664	R	210130	A	LAC	Q+63,X	/*12JAH17*
2914	02665	R	044710	R	DAC	SUBOPT	/*12JAH17*
2915	02666	R	210131	A	LAC	Q+64,X	/*12JAH17*
2916	02667	R	045040	R	DAC	SAVE	/*12JAH17*
2917	02670	R	210132	A	LAC	Q+65,X	/*12JAH17*
2918	02671	R	045103	R	DAC	UNSAVE	/*12JAH17*
2919	02672	R	210133	A	LAC	Q+66,X	/*12JAH17*
2920	02673	R	045123	R	DAC	STEP	/*12JAH17*
2921	02674	R	210134	A	LAC	Q+67,X	/*12JAH17*
2922	02675	R	045151	R	DAC	PART	/*12JAH17*
2923	02676	R	210135	A	LAC	Q+70,X	/*12JAH17*
2924	02677	R	047163	R	DAC	EVAL	/*12JAH17*
2925	02700	R	210136	A	LAC	Q+71,X	/*12JAH17*
2926	02701	R	045434	R	DAC	SYMBOL	/*12JAH17*
2927	02702	R	210137	A	LAC	Q+72,X	/*12JAH17*
2928	02703	R	043724	R	DAC	SYNTAX	/*12JAH17*
2929	02704	R	210140	A	LAC	Q+73,X	/*12JAH17*
2930	02705	R	043744	R	DAC	STOR1	/*12JAH17*
2931	02706	R	210141	A	LAC	Q+74,X	/*12JAH17*
2932	02707	R	044034	R	DAC	STOR2	/*12JAH17*
2933	02710	R	210142	A	LAC	Q+75,X	/*12JAH17*
2934	02711	R	044103	R	DAC	DECIN	/*12JAH17*

2935	02712	R	210143	A	LAC	Q+76,X	/*12JAH17*
2936	02713	R	044255	R	DAC	DECOU	/*12JAH17*
2937	02714	R	210144	A	LAC	Q+77,X	/*12JAH17*
2938	02715	R	044320	R	DAC	DCTOUT	/*12JAH17*
2939	02716	R	210145	A	LAC	Q+100,X	/*12JAH17*
2940	02717	R	046673	R	DAC	FORMAT	/*12JAH17*
2941	02720	R	210146	A	LAC	Q+101,X	/*12JAH17*
2942	02721	R	045304	R	DAC	ASSEM	/*12JAH17:
2943	02722	R	210147	A	LAC	Q+102,X	/*12JAH17*
2944	02723	R	040266	R	DAC	GASS	/*12JAH17*
2945	02724	R	210150	A	LAC	Q+103,X	/*12JAH17*
2946	02725	R	043174	R	DAC	OUT	/*12JAH17:
2947	02726	R	210151	A	LAC	Q+104,X	/*12JAH17*
2948	02727	R	044725	R	DAC	ICHARI	/*12JAH17:
2949	02730	R	210152	A	LAC	Q+105,X	/*12JAH17:
2950	02731	R	044235	R	DAC	DEC2	/*12JAH17:
2951	02732	R	210153	A	LAC	Q+106,X	/*12JAH17*
2952	02733	R	043325	R	DAC	PACKEM	/*12JAH17:
2953	02734	R	210154	A	LAC	Q+107,X	/*12JAH17*
2954	02735	R	047066	R	DAC	WRSTEP	/*12JAH17:
2955	02736	R	210155	A	LAC	Q+110,X	/*12JAH17*
2956	02737	R	043712	R	DAC	OTEXT	/*12JAH17:
2957	02740	R	210156	A	LAC	Q+111,X	/*12JAH17*
2958	02741	R	045363	R	DAC	SVE	/*12JAH17*
2959	02742	R	210157	A	LAC	Q+112,X	/*12JAH17*
2960	02743	R	047107	R	DAC	SETIO	/*12JAH17*
2961			000157	A		Q=Q+112	/*12JAH17*
2962					.IFPNZ	DTA	/*12JAH17*
2963			000160	A		Q=Q+1	/*12JAH17*
2964	02744	R	210160	A	LAC	Q,X	/*12JAH17*
2965	02745	R	043701	R	DAC	LOKDT	/*12JAH17*
2966					.ENDC		/*12JAH17*
2967	02746	R	210161	A	LAC	Q+1,X	/*12JAH17*
2968	02747	R	045272	R	DAC	KEYCHK	/*12JAH17*
2969	02750	R	210162	A	LAC	Q+2,X	/*12JAH17*
2970	02751	R	045242	R	DAC	GETKEY	/*12JAH17*
2971	02752	R	210163	A	LAC	Q+3,X	/*12JAH17*
2972	02753	R	047021	R	DAC	TSTR	/*12JAH17*
2973	02754	R	210164	A	LAC	Q+4,X	/*12JAH17*
2974	02755	R	040270	R	DAC	GSYM	/*12JAH17*
2975	02756	R	210165	A	LAC	Q+5,X	/*12JAH17*
2976	02757	R	040272	R	DAC	GLOBAL	/*12JAH17*
2977	02760	R	210166	A	LAC	Q+6,X	/*12JAH17*
2978	02761	R	044357	R	DAC	PLAC1	/*12JAH17*
2979			000166	A		Q=Q+6	/*12JAH17*
2980					.IFPNZ	MODIFY	/*12JAH17*
2981			000167	A		Q=Q+1	/*12JAH17*
2982	02762	R	210167	A	LAC	Q,X	/*12JAH17*
2983	02763	R	044616	R	DAC	ERMOD	/*12JAH17*
2984	02764	R	210170	A	LAC	Q+1,X	/*12JAH17*
2985	02765	R	044573	R	DAC	MBUF2	/*12JAH17*
2986			000170	A		Q=Q+1	/*12JAH17*

```

2987
2988      02766 R 210171 A
2989      B 02767 R 050005 R
2990      02770 R 210172 A
2991      02771 R 060274 R
2992      02772 R 210173 A
2993      02773 R 060276 R
2994      02774 R 210174 A
2995      02775 R 060275 R
2996
          02776 R 707764 A
          02777 R 622327 R
2997
2998
    
```

```

      .ENDC
      LAC Q+1,X
      DAC REMOVE
      LAC Q+2,X
      DAC* PTEMPX
      LAC Q+3,X
      DAC* PRDCNTX
      LAC Q+4,X
      DAC* PBOOLX
      .EBREL :
          JMP* SWPIN
          .EJECT
    
```

Handwritten annotations:
 LAC Q+4,X
 DAC PBOOLX
 MOVE up
 LAC Q+1,X
 EBA
 DAC REMOVE

```

/*12JAH17*
/*12JAH17*
/*12JAH17*
/*12JAH17*
/*12JAH17*
/*12JAH17*
/*12JAH17*
/*12JAH17*
/*12JAH17*
    
```

17677



*from P&P
Come here
just prior to
running new
partition*

```

2999          003000 R      SWPIN3=.          /*12JAH15*
3000          .IFPNZ  XVM          /*12JAH15*
3001      03000 R 701742 A      MPEU          /ENTER USER MODE /*12JAH15*
3002          .ENDC          /*12JAH15*
3003      03001 R 200157 R      LAC      MQ          /*12JAH15*
3004      03002 R 652000 A      LMQ          /RESTORE MQ.
3005      03003 R 200160 R      LAC      XR          /RESTORE X REG
3006      03004 R 721000 A      PAX
3007      03005 R 200161 R      LAC      LR          /RESTORE L REG
3008      03006 R 722000 A      PAL
3009      03007 R 200721 R      LAC      TIMESL
3010      03010 R 040656 R      DAC      RUNDOWN(-2) /SET SLICE
3011      03011 R 750030 A      CLA!IAC
3012      03012 R 040652 R      DAC      SWAPSW      /MAKE SWAPPABLE
3013      03013 R 440173 R      ISZ      SWPCNT
3014      03014 R 740000 A      NOP
3015      03015 R 234047 E      LAC*     CLOCK      /UPDATE TIME
3016      03016 R 060147 R      DAC*     TVAR
3017      03017 R 200136 R      LAC      AVAR
3018      03020 R 353706 R      TAD      (-5
3019      03021 R 040011 A      DAC      R11
3020      03022 R 234051 E      LAC*     DATE
3021      03023 R 060011 A      DAC*     R11
3022      03024 R 234052 E      LAC*     DATE.1
3023      03025 R 060011 A      DAC*     R11          /UPDATE DATE
3024      03026 R 234053 E      LAC*     DATE.2
3025      03027 R 060011 A      DAC*     R11
3026      03030 R 220154 R      LAC*     PRTEHD      /DID WE UNHANG BECAUSE OF %W VIOLATION
3027      03031 R 741300 A      SPA!SNA
3028      03032 R 603040 R      JMP      SWPIN0      /NO.
3029      03033 R 205103 R      LAC      UNSAVE      /YES, RESET INPUT BUFFER POINTER
3030      03034 R 040220 R      DAC      IPOINT
3031      03035 R 160220 R      DZM*     IPOINT      /MAKE SURE IT INDICATES NULL STRING.
3032      03036 R 160144 R      DZM*     WVAR          /SET %W TO INDICATE OVERFLOW.
3033      03037 R 603434 R      JMP      ALTMOD      /FAKE EOL (END OF LINE)
3034      03040 R 200577 R      SWPIN0 LAC      OSKACT      /IS THIS USER IN MIDDLE
3035      03041 R 540654 R      SAD      PRTNMB      /OF LOGICAL DISK ACTIVITY?
3036      03042 R 603062 R      JMP      SWPOK2      /YES, INHIBIT IOINT, EMER CLOSE TEMPORARILY
3037      03043 R 220654 R      LAC*     PRTNMB
3038      03044 R 513642 R      AND      (40000      /LOOK FOR IOINT
3039      03045 R 741200 A      SNA
3040      03046 R 603052 R      JMP      SWPOK      /NO IOINT -- GO CONTINUE
3041      03047 R 260654 R      XOR*     PRTNMB      /CLEAR IOINT BIT
3042      03050 R 060654 R      DAC*     PRTNMB
3043      03051 R 000023 A      ERR23      /GIVE HIM THE IOINT
3044          003052 R      SWPOK=.          /*12JAH15*
3045          .IFPNZ  XVM          /*12JAH15*
3046      03052 R 200020 A      LAC      R20          /SAVE LOCATION 20 WHICH IS /*12JAH15*
3047      03053 R 040573 R      DAC      KEEP20      /DESTROYED BY CAL /*12JAH15*
3048      03054 R 000000 A      CAL
3049      03055 R 200573 R      MEXEC LAC      KEEP20      /GO INTO EXEC MODE /*12JAH15*
3050      03056 R 040020 A      DAC      R20          /RESTORE LOCATION 20 /*12JAH15*
                          /TO WHAT IT WAS BEFORE CAL /*12JAH15*

```

input

-2

out

*SAD
JMP 3001*

as saved

```

3051          .ENDC                                     /*12JAH15*
3052      03057 R 200570 R          LAC      EMERG      /EMERGENCY CLOSE?      /*12JAH15*
3053      03060 R 740200 A          SZA      / *12JAH10* (SWITCH 6)
3054      03061 R 603067 R          JMP      SWPIN2     /YES.
3055      03062 R 203101 R      SWPDK2 LAC      IOU      /NO, MAKE OUT PRINT    /*12JAH17*
3056      03063 R 043175 R          DAC      OUT+1
3057      03064 R 200156 R      SWPIN1 LAC      AC        /RESTORE AC
3058      03065 R 707742 A          RES      /RESTORE LINK, BANK/PAGE MODE
3059      03066 R 620155 R          JMP*     PC        /EXIT
3060      /
3061      03067 R 213707 R      SWPIN2 LAC      (JMP* OUT   /MAKE "OUT" NON-PRINTING SO NO OUTPUT HANGS
3062      03070 R 043175 R          DAC      OUT+1
3063          .IFPNZ XVM                                     /*12JAH15*
3064      03071 R 701742 A          MPEU      /ENTER USER MODE      /*12JAH15*
3065          .ENDC                                     /*12JAH15*
3066      03072 R 777777 A          LAW      -1        /1 SEC CNTDOWN TO WATCHDOG FOR
3067      03073 R 060154 R          DAC*     PRTE     /NOISY D001S OR IDLE TERMINALS
3068      03074 R 220654 R          LAC*     PRINMB
3069      03075 R 513710 R          AND      (1000
3070      03076 R 744200 A          SZA!CLL   /IN HALTER?           /*12JAH15*
3071      03077 R 603064 R          JMP      SWPIN1     /YES                   /*12JAH15*
3072      03100 R 634111 E          JMP*     HALT1      /*12JAH12* NO
3073      03101 R 513702 R      IOU      AND      (77      /1ST INST OF SUBR "OUT" /*12JAH17*

```

```

3074          .TITLE  SLEEP HUNG PARTITION UPDATER
3075          /
3076          /THIS ROUTINE CHECK EACH PARTITION TO SEE IF IT IS CLOCK /*12JAH17*
3077          /HUNG (BIT 16 OF STATUS WORD SET) OR DC01 HUNG (BIT 15 OF /*12JAH17*
3078          /STATUS WORD SET) WITH NEGATIVE WATCHDOG COUNT. IT WILL. /*12JAH17*
3079          /INCREMENT EITHER THE CLOCK HUNG WORD (NEXT TO LAST WORD /*12JAH17*
3080          /OF PARTITION) OR %WATCHDOG COUNT (LAST WORD OF PARTITION) /*12JAH17*
3081          /AND ON OVERFLOW REMOVE THE APPROPRIATE HANG BIT /*12JAH17*
3082          /
3083          03102 R 213665 R SLPCHK LAC (STAT1 /START OF PARTITION STATUS WORDS /*12JAH17*
3084          03103 R 040572 R SLPCK1 DAC SLPTMP /MOVE TO NEXT PARTITION /*12JAH17*
3085          03104 R 220572 R LAC* SLPTMP /FETCH PARTITION STATUS WORD /*12JAH17*
3086          03105 R 742020 A RTR /MOVE CLOCK HANG BIT TO LINK /*12JAH17*
3087          03106 R 741420 A SZL!RAR /CLOCK HUNG? MOVE %W BIT TO LINK /*12JAH17*
3088          03107 R 603117 R JMP SLPHNG /YES, GO INCREMENT SYMEND /*12JAH17*
3089          03110 R 741400 A SZL /%WATCHDOG? /*12JAH17*
3090          03111 R 603134 R JMP WATHNG /YES, GO INCREMENT PRTEHD /*12JAH17*
3091          03112 R 200572 R SLPCK2 LAC SLPTMP /FETCH STATUS WORD PTR /*12JAH17*
3092          03113 R 740030 A IAC /ADVANCE TO NEXT STATUS WORD /*12JAH17*
3093          03114 R 540330 R SAD STATX /END OF TABLE? /*12JAH17*
3094          03115 R 601562 R JMP STACHO /YES /*12JAH17*
3095          03116 R 603103 R JMP SLPCK1 /NO, GO CHECK NEXT PARTITION /*12JAH17*
3096          03117 R 103155 R SLPHNG JMS SETPRT /SET UP PTR TO SYMEND /*12JAH17*
3097          03120 R 200571 R LAC SLPCNT /FETCH SECONDS SINCE LAST UPDATE /*12JAH17*
3098          03121 R 350000 A IAD 0,X /BUMP HANG /*12JAH17*
3099          03122 R 050000 A DAC 0,X /TIME LOCATION /*12JAH17*
3100          03123 R 707764 A EBA /NO MORE NEED OF XR /*12JAH17*
3101          03124 R 741100 A SPA /HANG TIME OVERFLOW? /*12JAH17*
3102          03125 R 603112 R JMP SLPCK2 /NO /*12JAH17*
3103          03126 R 777775 A LAW 17775 /YES /*12JAH17*
3104          03127 R 705522 A INTOFF /ALTERING STATUS WORD /*12JAH17*
3105          03130 R 520572 R AND* SLPTMP /CLEAR HANG TIME BIT /*12JAH17*
3106          03131 R 060572 R DAC* SLPTMP /INSERT NEW STATUS WORD /*12JAH17*
3107          03132 R 705521 A INTON /DONE /*12JAH17*
3108          03133 R 603112 R JMP SLPCK2 /GO CHECK NEXT PARTITION /*12JAH17*
3109          03134 R 103155 R WATHNG JMS SETPRT /SET UP PTR TO SYMEND /*12JAH17*
3110          03135 R 210001 A LAC 1,X /FETCH %WATCHDOG RUNDOWN /*12JAH17*
3111          03136 R 740100 A SMA /ANY RUNDOWN /*12JAH17*
3112          03137 R 603152 R JMP WATH1 /NO /*12JAH17*
3113          03140 R 340571 R IAD SLPCNT /ADD SECONDS SINCE LAST UPDATE /*12JAH17*
3114          03141 R 050001 A DAC 1,X /UPDATE SYMEND IN PARTITION /*12JAH17*
3115          03142 R 751130 A SPA!CLA!IAC /%W OVERFLOW? /*12JAH17*
3116          03143 R 603152 R JMP WATH1 /NO /*12JAH17*
3117          03144 R 050001 A DAC 1,X /YES, INSERT +1 TO SHOW OVERFLOW /*12JAH17*
3118          03145 R 203154 R LAC WMASK /SAVE I-MODE, IOINT, TRANS, RUN BITS /*12JAH17*
3119          03146 R 705522 A INTOFF /ALTERING STATUS WORD /*12JAH17*
3120          03147 R 520572 R AND* SLPTMP /CLEAR ALL HANG BITS /*12JAH17*
3121          03150 R 060572 R DAC* SLPTMP /UPDATE STATUS WORD /*12JAH17*
3122          03151 R 705521 A INTON /COME ALIVE AGAIN /*12JAH17*
3123          03152 R 707764 A WATH1 EBA / /*12JAH17*
3124          03153 R 603112 R JMP SLPCK2 /CHECK NEXT PARTITION /*12JAH17*
3125          03154 R 441001 A WMASK 441001 /MASK OF I-MODE, IOINT, TRANS, & RUN /*12JAH17*

```

3126				/				
3127	03155	R	000000	A	SETPRT	0	/SET UP PTR TO SYMEND	/*12JAH17*
3128	03156	R	200572	R	LAC	SLPTMP	/FETCH STATUS WORD PTR	/*12JAH17*
3129	03157	R	353676	R	TAD	(-STAT1	/START OF STATUS WORD TABLE	/*12JAH17*
3130	03160	R	040567	R	DAC	STEMP2	/PARTITION NUMBER	/*12JAH17*
3131	03161	R	353700	R	TAD	(INDTAB	/PART. STARTING ADDR TABLE	/*12JAH17*
3132	03162	R	040566	R	DAC	STEMP	/PTR TO STARTING ADDR OF PART	/*12JAH17*
3133	03163	R	200567	R	LAC	STEMP2	/PARTITION NUMBER	/*12JAH17*
3134	03164	R	354142	E	TAD	PASTAB	/PART. SIZE TABLE	/*12JAH17*
3135	03165	R	040567	R	DAC	STEMP2	/PTR TO PART. SIZE	/*12JAH17*
3136	03166	R	220567	R	LAC*	STEMP2	/PART. SIZE	/*12JAH17*
3137	03167	R	360566	R	TAD*	STEMP	/PART. START	/*12JAH17*
3138	03170	R	723776	A	AAC	-2	/SYMEND = NEXT TO LAST LOC OF PART	/*12JAH17*
3139	03171	R	721000	A	PAX		/XR TO HOLD SYMEND ADDR	/*12JAH17*
3140	03172	R	707762	A	DBA		/	/*12JAH17*
3141	03173	R	623155	R	JMP*	SETPRT	/RETURN	/*12JAH17*

```

3142          .TITLE  INPUT/OUTPUT SUBROUTINES
3143          /
3144          /OUTPUT SUBROUTINE
3145          /
3146          03174 R 000000 A      OUT      0                      /*12JAH17*
3147          03175 R 513702 R      XOUT    AND      (77          /MASK THE 6-BIT CHAR IN AC AND
3148          03176 R 652000 A              LMQ          /SAVE IN MQ.
3149          03177 R 200164 R              LAC      OUTPUT    /CHECK DEVICE TYPE
3150          03200 R 543324 R              SAD      .JCFN1     /*12JAH12*
3151          03201 R 603322 R              JMP      XOUT10    /THIS IS SC FUNCTION
3152          .IFPNZ  DTA
3153          03202 R 543203 R              SAD      DTJMPO
3154          03203 R 634066 E      DTJMPO  JMP*     DTOUT1    /WE'RE PUTTING STUFF TO DT
3155          .ENDC
3156          03204 R 353711 R              TAD      (-JMP-OUTTAB-NUMAT
3157          03205 R 745100 A              SPA!CLL
3158          03206 R 603214 R              JMP      XOUT1    /TTY - GO SEE IF LEADING SPACES NEEDED
3159          .IFPNZ  DTA
3160          03207 R 353712 R              TAD      (-DTA
3161          03210 R 745100 A              SPA!CLL
3162          03211 R 600164 R              JMP      OUTPUT    /DECTAPE
3163          .ENDC
3164          03212 R 741200 A              SNA
3165          03213 R 603320 R              JMP      XOUT9    /PT?
3166          03214 R 641002 A      XOUT1   LACQ
3167          03215 R 144710 R              DZM      SUBOPT    /O THE SPACING COUNTER
3168          03216 R 044770 R              DAC      CHAR1
3169          03217 R 553653 R              SAD      (40
3170          03220 R 603267 R              JMP      XOUT6    /SPACE - CHECK RIGHT MARGIN
3171          03221 R 745202 A              SNA!STL
3172          03222 R 603277 R              JMP      XOUT7    /CR
3173          03223 R 553713 R              SAD      (34
3174          03224 R 603314 R              JMP      XOUT8    /FF
3175          /
3176          03225 R 220141 R      XOUT2   LAC*     MVAR
3177          03226 R 745101 A              SPA!CMA!CLL
3178          03227 R 750001 A              CLC
3179          03230 R 740030 A              IAC
3180          03231 R 360150 R              TAD*     XVAR
3181          03232 R 044710 R              DAC      SUBOPT    /SPACE COUNT * 144
3182          03233 R 220150 R      XOUT4   LAC*     XVAR    /INCREMENT SX
3183          03234 R 723144 A              AAC      144
3184          03235 R 060150 R              DAC*     XVAR
3185          03236 R 204770 R      XOUT5   LAC      CHAR1    /OUTPUT CHAR...
3186          03237 R 744400 A              SNL!CLL
3187          03240 R 213653 R              LAC      (40      /...OR SPACE
3188          03241 R 745200 A      XOUT5A  SNA!CLL    /EOM?
3189          03242 R 760015 A              LAW      15      /CONVERT TO CR
3190          03243 R 553713 R              SAD      (34
3191          03244 R 760014 A              LAW      14      /FF
3192          03245 R 745100 A              SPA!CLL
3193          03246 R 603251 R              JMP      .+3

```

```

3194 03247 R 253653 R XOUT5B XOR (40 /CONVERT TO 7-BIT.
3195 03250 R 313653 R ADD (40
3196 03251 R 513714 R AND (177 /RUBOUT
3197 03252 R 652000 A LMQ
3198 03253 R 200020 A XOUT4A LAC R20 /GO TO IO ROUTINE WITH DEVICE #
3199 03254 R 600164 R JMP OUTPUT
3200 /
3201 03255 R 744000 A XOUT3 CLL / TT RETURN HERE
3202 03256 R 220142 R LAC* FVAR /IGNORE SPACE COUNT IF %F=0
3203 03257 R 741200 A SNA
3204 03260 R 623174 R XOUTDT JMP* OUT
3205 03261 R 204710 R LAC SUBOPT
3206 03262 R 353715 R TAD (144
3207 03263 R 044710 R DAC SUBOPT /INCREMENT SPACE COUNTER
3208 03264 R 740300 A SZA!SMA
3209 03265 R 623174 R JMP* OUT /FINISHED
3210 03266 R 603233 R JMP XOUT4 /LINK TRANSMITS WHETHER SPACE OR CHAR WANTED
3211 /
3212 03267 R 220142 R XOUT6 LAC* FVAR /SPACES COME HERE
3213 03270 R 741201 A CMA!SNA
3214 03271 R 603225 R JMP XOUT2 /IGNORE %F IF IT IS 0
3215 03272 R 320150 R ADD* XVAR
3216 03273 R 313715 R ADD (144 /RIGHT MARGIN CHECK
3217 03274 R 741100 A SPA /FULL LINE?
3218 03275 R 603225 R JMP XOUT2 /NO
3219 03276 R 144770 R DZM CHAR1 /YES - CHANGE TO CR
3220 /
3221 03277 R 213715 R XOUT7 LAC (144 /CR
3222 03300 R 060150 R DAC* XVAR
3223 03301 R 320151 R ADD* YVAR
3224 03302 R 060151 R DAC* YVAR
3225 03303 R 220140 R LAC* PVAR /GET PAGE LENGTH.
3226 03304 R 745202 A SNA!STL
3227 03305 R 603236 R JMP XOUT5 /%P=0 - OUTPUT CR
3228 03306 R 740001 A CMA
3229 03307 R 320151 R ADD* YVAR
3230 03310 R 745102 A SPA!STL
3231 03311 R 603236 R JMP XOUT5 /$Y<%P - OUTPUT CR
3232 03312 R 213713 R LAC (34
3233 03313 R 044770 R DAC CHAR1 /CHANGE CR TO FF
3234 /
3235 03314 R 213715 R XOUT8 LAC (144 /FF
3236 03315 R 060151 R DAC* YVAR /RESET LINE NO. ($Y)
3237 03316 R 060150 R DAC* XVAR /RESET CHAR. POSITION ($X)
3238 03317 R 603236 R JMP XOUT5
3239 /
3240 03320 R 641002 A XOUT9 LACQ /PAPER TAPE
3241 03321 R 603241 R JMP XOUT5A
3242 /
3243 03322 R 641002 A XOUT10 LACQ /SCHAR FUNCTION
3244 03323 R 600164 R JMP OUTPUT
3245 03324 R 634046 E .JCFN1 JMP* CFUN1 /*12JAH12*

```

PAGE 87

MUMPS 100

INPUT/OUTPUT SUBROUTINES

3246

.EJECT

```

3247 /INPUT SUBROUTINE
3248 /
3249 03325 R 000000 A PACKEM 0 /*12JAH17*
3250 03326 R 045103 R DAC UNSAVE /ADDR. OF 1ST WORD IN BUFFER /*12JAH17*
3251 03327 R 540133 R SAD BUFF2 /STRING BUFFER?
3252 03330 R 723000 A AAC STRACC-COMBUFF /YES
3253 03331 R 723030 A AAC COMBUFF-1
3254 03332 R 045040 R DAC SAVE /ADDR. OF LAST WD. IN BUFFER.
3255 03333 R 650000 A CLQ /MAKE SURE THERE IS NO CONFUSION IN MQ
3256 03334 R 200020 A XPACK2 LAC R20 /20 CONTAINS DEVICE NUMBER
3257 03335 R 600163 R JMP INPUT /SETIO STORES "JMP INTAB+DEV"
3258 /
3259 /JMP TO KEY FROM EACH INPUT I/O PACKAGE
3260 /
3261 03336 R 553714 R KEY SAD (177 /RUBOUT
3262 03337 R 603441 R JMP RUB
3263 03340 R 553701 R KEY.2 SAD (175 /ALTMOD,ESC,EOM
3264 03341 R 603434 R JMP ALTMOD
3265 03342 R 553716 R SAD (15 /CR
3266 03343 R 603430 R JMP KEYCR
3267 03344 R 553717 R SAD (134 /BACKSLASH
3268 03345 R 603405 R JMP DEL
3269 03346 R 553720 R SAD (14 /FORMFEED
3270 03347 R 213717 R LAC (134
3271 03350 R 553721 R SAD (23 /X-OFF
3272 03351 R 603334 R JMP XPACK2 /IGNORE X-OFF
3273 03352 R 353722 R TAD (-33 /TRANSLATE CONTROL LETTER
3274 03353 R 741100 A SPA /TO PRINTABLE LETTER
3275 03354 R 353723 R TAD (100
3276 03355 R 353724 R TAD (33
3277 03356 R 353725 R TAD (777640 /AC>137
3278 03357 R 744100 A SMA!CLL
3279 03360 R 603334 R JMP XPACK2 /CHAR. NOT IN CHAR SET.
3280 03361 R 353723 R TAD (100 /AC<40
3281 03362 R 253653 R XOR (40 /AC=100
3282 03363 R 745300 A SPA!SNA!CLL
3283 03364 R 603334 R JMP XPACK2
3284 03365 R 104725 R IN CALL ICHARI
3285 03366 R 200020 A LAC R20 /CHECK DEVICE TYPE
3286 03367 R 353726 R TAD (-NUMAT-1
3287 03370 R 744100 A SMA!CLL
3288 03371 R 603375 R JMP IN2 /DEVICE NOT MARGIN SPACING
3289 03372 R 220150 R LAC* XVAR
3290 03373 R 313715 R ADD (144 /INCREMENT $X
3291 03374 R 060150 R DAC* XVAR
3292 03375 R 200220 R IN2 LAC IPOINT
3293 03376 R 545040 R SAD SAVE
3294 03377 R 745000 A SKP!CLL /TOO MANY
3295 03400 R 603334 R JMP XPACK2 /GET NEXT CHAR
3296 03401 R 200221 R LAC IPLACE /*12JAH14* POS WITHIN WORD
3297 03402 R 553727 R SAD (XCI3 /*12JAH14* LAST CHAR POS FILLED?
3298 03403 R 745000 A SKP!CLL /*12JAH14* YES

```

```

3299      03404 R 603334 R      JMP      XPACK2      /*12JAH14* NO
3300      03405 R 200220 R      DEL      LAC      IPOINT
3301      03406 R 545103 R      SAD      UNSAVE      /BUFFER EMPTY?
3302      03407 R 741000 A      SKP
3303      03410 R 603414 R      JMP      DEL2      /*12JAH14* FIRST WORD OF BUFFER
3304      03411 R 200221 R      LAC      IPLACE      /*12JAH14* NOT FIRST WORD
3305      03412 R 553730 R      SAD      (XCI1      /*12JAH14* CHAR POS
3306      03413 R 603334 R      JMP      XPACK2      /*12JAH14* FIRST ONE?
3307      03414 R 777777 A      DEL2     LAW      -1      /*12JAH14* YES, CAN'T BACK UP
3308      03415 R 340221 R      TAD      IPLACE      /*12JAH14* IS CHAR PTR
3309      03416 R 553731 R      SAD      (XCI1-1      /*12JAH14* POINTING TO
3310      03417 R 603421 R      JMP      DEL3      /*12JAH14* 1ST CHAR OF WORD
3311      03420 R 603425 R      JMP      DEL4      /*12JAH14* YES
3312      03421 R 777777 A      DEL3     LAW      -1      /*12JAH14* NO
3313      03422 R 340220 R      TAD      IPOINT      /*12JAH14* MOVE WORD
3314      03423 R 040220 R      DAC      IPOINT      /*12JAH14* POINTER
3315      03424 R 213727 R      LAC      (XCI3      /*12JAH14* BACK ONE LOCATION
3316      03425 R 040221 R      DEL4     DAC      IPLACE      /*12JAH14* POINT TO 3RD CHAR IN WRD
3317      03426 R 603334 R      JMP      XPACK2
3318      .IFPNZ DTA
3319      03427 R 634064 E      DTJMPI  JMP*     DTIN1
3320      .ENDC
3321      .EJECT

```

```

3322
3323      03430 R 213715 R      /
3324      03431 R 060150 R      KEYCR  LAC      (144
3325      03432 R 320151 R      DAC*   XVAR      /RESET SX
3326      03433 R 060151 R      ADD*   YVAR
3327      03433 R 060151 R      DAC*   YVAR      /INCREMENT SY
3328      03434 R 754000 A      /
3329      03435 R 060154 R      ALTMOD CLA!CLL
3330      03436 R 104725 R      DAC*   PRTEHD /CLEAR %W COUNTDOWN FOR CLOCK
3331      03437 R 443325 R      CALL   ICHARI /PACK EOM INTO BUFFER
3332      03440 R 623325 R      ISZ    PACKEM /BUMP RETURN ADDR. FOR
3333      03440 R 623325 R      JMP*   PACKEM /NORMAL RETURN.
3334      03441 R 200020 A      /
3335      03442 R 723727 A      RUB   LAC      R20      /IS CURRENT          /*12JAH17*
3336      03443 R 740300 A      AAC    -DC01*10-1 /DEVICE A           /*12JAH17*
3337      03444 R 603334 R      SMA!SZA /TERMINAL?        /*12JAH17*
3338      03445 R 760036 A      JMP    XPACK2 /NO, TREAT CODE 177 AS ILLEGAL /*12JAH17*
3339      03446 R 103174 R      LAW    36      /ECHO FOR LINE DELETE /*12JAH17*
3340      03447 R 760025 A      CALL   OUT
3341      03450 R 103174 R      LAW    25      /
3342      03451 R 760000 A      CALL   OUT
3343      03452 R 103174 R      LAW    00      /CRLF
3344      03453 R 213715 R      CALL   OUT
3345      03454 R 060150 R      LAC    (144
3346      03455 R 360151 R      DAC*   XVAR      /SET CHAR POSN = 1
3347      03456 R 060151 R      TAD*   YVAR
3348      03457 R 623325 R      DAC*   YVAR      /AND ADD ONE TO LINE COUNT
3348      03457 R 623325 R      JMP*   PACKEM /LINE RUBOUT RETURN.

```

```

3349          .TITLE REENTRANT MAIN PROGRAM
3350          /INTERPRETER ENTRY FOR DIRECT MODE
3351          /
3352          03460 R 160145 R BEGIN  DZM*   LVAR           /SET STEP = 0
3353          03461 R 140652 R BEG1   DZM     SWAPSW
3354          03462 R 200162 R        LAC     PRNCDV  /LOCK ON TO PRNC
3355          03463 R 300331 R        ADD     WHOTAB  /DEVICE FOR THIS JOB
3356          03464 R 043174 R        DAC     OUT
3357          03465 R 223174 R        LAC*    OUT     /GET CURRENT OWNER
3358          03466 R 540654 R        SAD     PRINMB  /SAME ONE
3359          03467 R 603476 R        JMP     BEG3    /YES--OK GO ON
3360          03470 R 741200 A        SNA     /NO--IS IT OWNED
3361          03471 R 603474 R        JMP     BEG2
3362          03472 R 101533 R        JMS     HANG    /YES--GO AWAY FOR AWHILE
3363          03473 R 603461 R        JMP     BEG1    /TRY AGAIN
3364          /
3365          03474 R 200654 R BEG2   LAC     PRINMB  /LOCK DEVICE
3366          03475 R 063174 R        DAC*    OUT
3367          03476 R 760000 A BEG3   LAW     00           /CR-LF
3368          03477 R 103174 R        CALL    OUT
3369          03500 R 760076 A BEGIN1  LAW     76           /LEFT CARET CUE
3370          03501 R 103174 R        CALL    OUT
3371          .IFDEF KBBGFU
3372          DZM*   BVAR
3373          .ENDC
3374          03502 R 213730 R        LAC     (XCI1   /*12JAH14* CHAR POS =
3375          03503 R 040221 R        DAC     IPLACE  /*12JAH12* MIDDLE
3376          03504 R 200132 R        LAC     BUFF1
3377          03505 R 040220 R        DAC     IPOINT  /SET UP POINTER AND
3378          03506 R 103325 R        CALL    PACKEM  /FILL DIRECT COMMAND BUFFER.
3379          03507 R 603500 R        JMP     BEGIN1  /LINE RUBOUT, REPEAT CUE.
3380          03510 R 200134 R        LAC     BUFF
3381          03511 R 040206 R        DAC     LOC     /BEGINNING OF STEP BUFFER.
3382          03512 R 200132 R        LAC     BUFF1  /POINTS TO FIRST LOC OF DIR CMND BUFF.
3383          03513 R 040222 R        DAC     OPOINT  /SET UP POINTER TO
3384          03514 R 213732 R        LAC     (XCO1   /*12JAH14* CHAR POS =
3385          03515 R 040223 R        DAC     OPLACE  /*12JAH14* MIDDLE
3386          03516 R 104660 R        CALL    OCHARI  /INDEX AND LOAD A CHAR.
3387          03517 R 103724 R        CALL    SYNTAX  /TEST CHAR. TYPE
3388          03520 R 603527 R        JMP     DIRECT  /LETTER - GO PROCESS DIRECT CMND.
3389          03521 R 603525 R        JMP     INDIR   /NUMBER - GO PROCESS INDIRECT CMND.
3390          03522 R 553733 R        SAD     (73     /PUNCT, IS IT A ;
3391          03523 R 603460 R        JMP     BEGIN   /YES, COMMENT, IGNORE AND GET NEXT LINE
3392          03524 R 000003 A        ERR3

```

```

3393          .TITLE COMMAND DISPATCHER
3394          /
3395          /HANDLE INDIRECT COMMAND
3396          /
3397          03525 R 104340 R  INDIR  CALL  PLACE          /STORE STEP IN PROGRAM BUFFER.
3398          03526 R 603460 R          JMP    BEGIN          /GET NEXT COMMAND.
3399          /
3400          /HANDLE DIRECT COMMAND
3401          /
3402          03527 R 760000 A  DIRECT  LAW    00
3403          03530 R 103174 R          CALL  OUT          /OUTPUT CR LF
3404          03531 R 104710 R          CALL  SUBOPT        /DECREMENT OPOINT
3405          03532 R 103535 R          CALL  DISPAT        /GO TO COMMAND DISPATCHER
3406          03533 R 000004 A          ERR4          /DIRECT MODE QUIT EXIT- "CMMND"
3407          03534 R 603460 R          JMP    BEGIN          /CONTINUE WITH NEXT DIRECT CMND.
3408          /
3409          /DISPATCH ON FIRST LETTER OF COMMAND WORD
3410          /
3411          03535 R 000000 A  DISPAT  0                                /*12JAH17*
3412          03536 R 140203 R          DZM    FORTST          /SET "NOT IN FOR"                                /*12JAH17*
3413          03537 R 104660 R  FORXIT  CALL  OCHARI          /GET NEXT CHAR.
3414          03540 R 103724 R          CALL  SYNTAX        /TEST CHAR. TYPE
3415          03541 R 603550 R          JMP    DISPT3        /LETTER
3416          03542 R 000003 A          ERR3          /NUMBER- "SYNTAX"
3417          03543 R 553733 R          SAD    (73          /PUNCTUATION, IS IT A ;
3418          03544 R 603602 R          JMP    DTATST        /YES,STEP COMPLETED IF NOT IN FOR
3419          03545 R 553653 R          SAD    (40          /NO, IS IT A SPACE?
3420          03546 R 603537 R          JMP    FORXIT        /YES, GO GET NEXT CHAR
3421          03547 R 000003 A          ERR3          /NO, "SYNTAX"
3422          03550 R 313734 R  DISPT3  ADD    (JMP DTAB-1        /STORE JMP TO DTAB
3423          03551 R 047163 R          DAC    EVAL          /FOR COMMAND DISPATCH.
3424          03552 R 104660 R  DISPT1  CALL  OCHARI          /GET NEXT CHAR.
3425          03553 R 745200 A          SNA!CLL        /EOM?
3426          03554 R 407163 R          XCT    EVAL          /YES,DISPATCH TO CMND PROC.
3427          03555 R 553733 R          SAD    (73
3428          03556 R 407163 R          XCT    EVAL          /COMMENT FOUND
3429          03557 R 553653 R          SAD    (40          /NO,IS IT A SPACE?
3430          03560 R 741000 A          SKP
3431          03561 R 603552 R          JMP    DISPT1        /NO SPACE FOUND YET
3432          03562 R 144645 R          DZM    OCHAR          /SAVE 1 AS THE LINK(POSSIBLE OPERAND)
3433          03563 R 444645 R  DISPT2  ISZ    OCHAR
3434          03564 R 104660 R          CALL  OCHARI          /GET NEXT CHAR
3435          03565 R 553653 R          SAD    (40          /IS IT A SPACE?
3436          03566 R 603563 R          JMP    DISPT2        /YES, ANOTHER SPACE, PASS OVER IT
3437          03567 R 104710 R          CALL  SUBOPT        /DECREMENT OPOINT AND
3438          03570 R 204645 R          LAC    OCHAR          /GET A CHARACTER
3439          03571 R 740020 A          RAR
3440          03572 R 407163 R          XCT    EVAL          /SET UP LINK FOR "POSSIBLE OPERAND FLAG"
3441          .EJECT          /DISPATCH TO COMMAND PROCESSING.

```

```

3442          /RETURN FROM COMMAND PROCESSING.
3443          /
3444          03573 R 104645 R   DTABLE CALL   OCHAR       /GET A CHAR.
3445          03574 R 553653 R   DTAB1  SAD     (40        /*12JAH12* IS IT A SPACE?
3446          03575 R 603537 R           JMP     FORXIT    /YES, GO GET NEXT CHAR.
3447          03576 R 553733 R           SAD     (73        /NO, IS IT A ;
3448          03577 R 603602 R           JMP     DTATST    /YES,COMMENT, TREAT AS END OF LINE
3449          03600 R 744200 A           SZA!CLL          /NO, IS IT EOM
3450          03601 R 000003 A           ERR3           /NO, "SYNTAX"
3451          03602 R 200203 R   DTATST  LAC     FORTST
3452          03603 R 744200 A           SZA!CLL          /ARE WE IN FOR LOOP?
3453          03604 R 612077 R           JMP     FORSTP   /YES, GO TO NEXT ITERATION
3454          03605 R 443535 R           ISZ     DISPAT  /NO, BUMP FOR NORMAL RETURN.
3455          03606 R 623535 R           JMP*   DISPAT   /RETURN, STEP COMPLETED.
3456          .EJECT

```

```

3457          /DISPATCH TABLE FOR COMMANDS
3458          /
3459          03607 R 611477 R   DTAB   JMP   ASK
3460          03610 R 611536 R           JMP   BREAK
3461          03611 R 634043 E           JMP*  CALLP   /*12JAH12* CALL OR CLJSE
3462          03612 R 611553 R           JMP   DO
3463          03613 R 611677 R           JMP   ERASE
3464          03614 R 611772 R           JMP   FOR     /FOR OR FILE
3465          03615 R 612266 R           JMP   GO      /GO OR GOTO
3466          03616 R 634107 E           JMP*  HALT    /*12JAH12* HALT OR HANG
3467          03617 R 612347 R           JMP   IF
3468          03620 R 612366 R           JMP   JOIN
3469          03621 R 612506 R           JMP   KILL
3470          03622 R 634120 E           JMP*  LOCK    /*12JAH12*
3471          .IFPNZ  MODIFY
3472          03623 R 612533 R           JMP   MODFY
3473          .ENDC
3474          .IFNOZ  MODIFY
3475          ERR4
3476          .ENDC
3477          03624 R 000004 A           ERR4    /"CMMND"
3478          03625 R 607677 R           JMP   OPEN   /OPEN OR OVERLAY
3479          03626 R 613020 R           JMP   PRINT
3480          03627 R 613054 R           JMP   QUIT
3481          03630 R 613073 R           JMP   READ
3482          03631 R 613213 R           JMP   SET
3483          03632 R 613250 R           JMP   TYPE
3484          03633 R 634160 E           JMP*  UNLOCK /*12JAH12*
3485          03634 R 000004 A           ERR4    /"CMMND"
3486          03635 R 613417 R           JMP   WRITE
3487          03636 R 613467 R           JMP   XCOM   /SPECIAL X COMMAND FOR READING DISC BLOCKS
3488          03637 R 000004 A           ERR4    /"CMMND"
3489          03640 R 000004 A           ERR4    /"CMMND"

```

```

3490          .TITLE  SUBROUTINES FOR INTERPRETER
3491          /
3492          /XPRODA - GET OWNERSHIP OF DISK & CORE BLOCK SWITCH
3493          /      CALLING SEQUENCE:
3494          /      CALL PRODA
3495          /      ANY INSTRUCTION
3496          /      PRODA RETURNS HERE
3497          /
3498          03641 R 000000 A  PRODA      0                                /*12JAH17*
3499          03642 R 705522 A  XPRODA    INTOFF          /PROHIBIT CHANGE OF SUBQUES
3500          03643 R 400720 R          XCT      SUBQUES      /WHICH DISK IS RUNNING?
3501          03644 R 603650 R          JMP      PROD1        /SUBMERGED
3502          03645 R 140652 R          DZM      SWAPSW       /SERIAL, PROHIBIT INTERRUPTION
3503          03646 R 705521 A          INTON
3504          03647 R 623641 R          JMP*     PRODA
3505          03650 R 040201 R  PROD1    DAC      SAVPRD      /SAVE AC
3506          03651 R 200577 R          LAC      DSKACT      /DISK IN USE?
3507          03652 R 741200 A          SNA
3508          03653 R 603661 R          JMP      PROD2        /NO, IT'S ALL OURS
3509          03654 R 540654 R          SAD      PRINMB      /DO WE ALREADY OWN IT?
3510          03655 R 603666 R          JMP      PROD3        /YES
3511          03656 R 101533 R  PROD15   JMS      HANG        /NO, WAIT YOUR TURN
3512          03657 R 200201 R  PRODRT   LAC      SAVPRD      /RESTORE AC
3513          03660 R 603642 R          JMP      XPRODA      /TRY AGAIN
3514          03661 R 440600 R  PROD2    ISZ      DSKCNT      /INCREMENT CONSEC DISK ACCUESSES
3515          03662 R 603666 R          JMP      PROD3        /NO OVERFLOW
3516          03663 R 200716 R          LAC      DSKREP      /DONE YOUR SHARE!
3517          03664 R 040600 R          DAC      DSKCNT      /RESET MAX ACCESSES CNTR
3518          03665 R 603656 R          JMP      PROD15     /GIVE UP TIME SLICE
3519          03666 R 200654 R  PROD3    LAC      PRINMB
3520          03667 R 040577 R          DAC      DSKACT      /WE NOW OWN DISK
3521          03670 R 040601 R          DAC      PRVDKA     /PREV OWNER SWITCH
3522          03671 R 140652 R          DZM      SWAPSW       /INHIBIT SWAP TILL DISK I/O BEGUN
3523          03672 R 705521 A          INTON
3524          03673 R 200201 R          LAC      SAVPRD
3525          03674 R 623641 R          JMP*     PRODA
3526          .EJECT

```

```

3527             .IFPNZ DTA
3528             /
3529             /SUBROUTINE TO LOCK ONTO THE DECTAPE CONTROLLER FOR THIS PARTITION
3530             /
3531             /CALL LOCK DT WITH ION AND SWAP ENABLED
3532             /
3533             03675 R 101533 R   LOKDT1 JMS      HANG           /MAY AS WELL HANG FOR AWHILE
3534             03676 R 540654 R           SAD      PRINMB
3535             03677 R 000003 A           ERR3           /PARTITION ALREADY LOCKED TO DT ROUTINES
3536                                     /"SYNTAX" ERROR
3537             03700 R 603702 R           JMP      XLOKDT           /*12JAH17*
3538             /ENTRY POINT FOR LOKDT
3539             03701 R 000000 A   LOKDT  0           /*12JAH17*
3540             03702 R 705522 A   XLOKDT INTOFF
3541             03703 R 234062 E           LAC*     DTACT           /IS DT CONTROLLER IN USE?(MOVE,SEARCH, ETC.)
3542             03704 R 744200 A           SZA!CLL
3543             03705 R 603675 R           JMP      LOKDT1          /...YES
3544             03706 R 200654 R           LAC      PRINMB
3545             03707 R 074062 E           DAC*     DTACT           /DT CONTROLLER NOW OWNED BY THIS PARTITION
3546             03710 R 705521 A           INTON
3547             03711 R 623701 R           JMP*     LOKDT
3548             .ENDC
3549             .EJECT

```

```

3550 /
3551 /XOTEXT - OUTPUT CHARACTERS UP TO EOM
3552 /
3553 03712 R 000000 A OTEXT 0 /*12JAH17*
3554 03713 R 513735 R AND (377777 /TRIM MESSAGE ADDR /*12JAH17*
3555 03714 R 040222 R DAC OPOINT /GETTING CHAR.
3556 03715 R 213736 R LAC (XCO3 /*12JAH14* INIT
3557 03716 R 040223 R DAC OPLACE /*12JAH14* CHAR POS
3558 03717 R 104660 R CALL OCHARI /INDEX & LOAD CHAR USING OPOINT.
3559 03720 R 741200 A SNA /EOM?
3560 03721 R 623712 R JMP* OTEXT /YES, FINISHED.
3561 03722 R 103174 R CALL OUT /NO, OUTPUT CHAR AND
3562 03723 R 603717 R JMP .-4 /CONTINUE.
3563 .EJECT

```

```

3564 /
3565 /XSYNTAX - CHECK CHARACTER TYPE
3566 /
3567 /ENTER WITH CHARACTER TO CHECKED IN THE AC
3568 /EXITS: (WITH AC RESTORED)
3569 / R1 - CHARACTER IS ALPHABETIC
3570 / R2 - CHARACTER IS NUMERIC (DOES NOT INCLUDE DECIMAL POINT OR SIGNS)
3571 / R3 - CHARACTER IS PUNCTUATION (INCLUDES EDM (00) AND SPACE (40))
3572 /
3573 /
3574 03724 R 000000 A SYNTAX 0 /*12JAH17*
3575 03725 R 652000 A LMO /SAVE CHAR TEMPORARILY /*12JAH17*
3576 03726 R 745200 A SNA!CLL
3577 03727 R 603733 R JMP XSYN1 /*12JAH12* EDM: CONSIDERED TO BE PUNC
3578 03730 R 313722 R ADD (777745 /-33
3579 03731 R 741100 A SPA
3580 03732 R 603742 R JMP RESTORE / LETTER
3581 03733 R 443724 R XSYN1 ISZ SYNTAX /*12JAH12*
3582 03734 R 313737 R ADD (777752 /-26
3583 03735 R 741100 A SPA
3584 03736 R 603741 R JMP XSYN2 /*12JAH12* PUNCTUATION
3585 03737 R 313740 R ADD (777765 /-13
3586 03740 R 740100 A SMA / NUMBER
3587 03741 R 443724 R XSYN2 ISZ SYNTAX /*12JAH12*
3588 03742 R 641002 A RESTORE LACQ / RESTORE AC AND
3589 03743 R 623724 R JMP* SYNTAX /RETURN.
3590 .EJECT

```

```

3591          /XSTOR1,XSTOR2 - CHECK STORAGE AVAILABLE WITHIN USER'S PARTITION
3592          /
3593          /
3594          /
3595          /XSTOR1 - PROGRAM LENGTH CHANGE
3596          /      ENTER WITH NEW VALUE FOR "END" IN THE AC
3597          /
3598          03744 R 000000 A      STOR1      0                      /*12JAH17*
3599          03745 R 652000 A                      LMQ                      / <AC> NEW VALUE FOR END      /*12JAH17*
3600          03746 R 740031 A                      CMA!IAC
3601          03747 R 340165 R                      TAD      STACK
3602
3603          03750 R 741100 A                      SPA                      /NUMBER OF WORDS BETWEEN "END" AND START OF STACK
3604          03751 R 603766 R                      JMP      MSTACK          /WILL NEW "END" CAUSE STACK TO BE MOVED?
3605          03752 R 340170 R      XRET1     TAD      SYMBEG          /YES, CHANGE LOCATION OF STACK
3606          03753 R 740031 A                      CMA!IAC
3607          03754 R 340166 R                      TAD      SPOINT          /ADD # OF WORDS BET. END OF STACK & "SYMBEG"
3608          03755 R 740031 A                      CMA!IAC
3609          03756 R 740030 A                      IAC                      /TOTAL FREE WORDS
3610          03757 R 060146 R                      DAC*     SVAR            / CONVERT FROM WORDS TO CHARACTERS
3611          03760 R 744010 A                      RCL
3612          03761 R 320146 R                      ADD*     SVAR            / 3 * (NO. OF WDS)
3613          03762 R 060146 R                      DAC*     SVAR            / SET UP NEW $$
3614          03763 R 641002 A                      LACQ
3615          03764 R 040167 R                      DAC      END            /UPDATE VALUE FOR "END"
3616          03765 R 623744 R                      JMP*     STOR1          / EXIT
3617          03766 R 740001 A      MSTACK   CMA
3618          03767 R 040266 R                      DAC      GASS            /SAVE LENGTH OF STACK OFFSET
3619          03770 R 340166 R                      TAD      SPOINT
3620          03771 R 740031 A                      CMA!IAC
3621          03772 R 340170 R                      TAD      SYMBEG
3622          03773 R 741100 A                      SPA                      /ENOUGH ROOM?
3623          03774 R 604026 R                      JMP      XSTO11          /NOT ENOUGH ROOM FOR INCREASE /*12JAH17*
3624          03775 R 641002 A                      LACQ
3625          03776 R 040165 R                      DAC      STACK          /NEW "END" BECOMES NEW "STACK"
3626          03777 R 777777 A                      LAW      -1
3627          04000 R 340166 R                      TAD      SPOINT          /FETCH OLD END OF STACK
3628          04001 R 046621 R                      DAC      CONT
3629          04002 R 340266 R                      TAD      GASS            /SUBTRACT OFFSET
3630          04003 R 740030 A                      IAC
3631          04004 R 040266 R                      DAC      GASS            /NEW END OF STACK
3632          04005 R 740030 A                      IAC
3633          04006 R 040166 R                      DAC      SPOINT
3634          04007 R 353741 R                      TAD      (-1
3635          04010 R 540165 R                      SAD      STACK          /ZERO LENGTH STACK?
3636          04011 R 604023 R                      JMP      NOSTAC          /YES
3637          04012 R 226621 R      MSTKR    LAC*     CONT          /TRANSFER STACK
3638          04013 R 060266 R                      DAC*     GASS
3639          04014 R 777777 A                      LAW      -1
3640          04015 R 340266 R                      TAD      GASS
3641          04016 R 040266 R                      DAC      GASS
3642          04017 R 777777 A                      LAW      -1

```

```

3643      04020 R 346621 R      TAD      CONT
3644      04021 R 046621 R      DAC      CONT
3645      04022 R 540167 R      SAD      END          /END OF TRANSFER?
3646      04023 R 751000 A      NOSTAC  SKP!CLA      /YES
3647      04024 R 604012 R      JMP      MSTKR      /NO
3648      04025 R 603752 R      JMP      XRET1
3649      04026 R 200134 R      XSTO11 LAC      BUFF      /CLEAR          /*12JAH17*
3650      04027 R 740030 A      IAC      /OUT          /*12JAH17*
3651      04030 R 040167 R      DAC      END          /PROG          /*12JAH17*
3652      04031 R 160134 R      DZM*     BUFF      /BUFFER        /*12JAH17*
3653      04032 R 160167 R      DZM*     END          /TO EMPTY     /*12JAH17*
3654      04033 R 000001 A      ERR1     /"STORE" ERROR /*12JAH17*
3655      /
3656      /
3657      /XSTOR2 - SYMBOL TABLE LENGTH CHANGE
3658      /      ENTER WITH NEW VALUE FOR "SYMBEG" IN THE AC
3659      /
3660      04034 R 000000 A      STOR2    0          /*12JAH17*
3661      04035 R 652000 A      LMQ      / <AC> NEW VALUE FOR SYMBEG /*12JAH17*
3662      04036 R 740001 A      CMA
3663      04037 R 340166 R      TAD      SPOINT
3664      04040 R 740331 A      SMA!SZA!CMA!IAC  /CHANGE OF STACK LOCATION NECESSARY?
3665      04041 R 604055 R      JMP      NSTACK      /YES
3666      04042 R 340165 R      XRET2    TAD      STACK
3667      04043 R 740031 A      CMA!IAC
3668      04044 R 340167 R      TAD      END          /ADD FREE WORDS FROM END TO STACK
3669      04045 R 740031 A      CMA!IAC
3670      04046 R 060146 R      DAC*     SVAR      / CONVERT FROM WORDS TO CHARACTERS
3671      04047 R 744010 A      RCL
3672      04050 R 320146 R      ADD*     SVAR      / 3 * (NO. OF WDS)
3673      04051 R 060146 R      DAC*     SVAR      / SET UP NEW SS
3674      04052 R 641002 A      LACQ
3675      04053 R 040170 R      DAC      SYMBEG     /UPDATE VALUE FOR "SYMBEG"
3676      04054 R 624034 R      JMP*     STOR2      / EXIT
3677      04055 R 040266 R      NSTACK   DAC      GASS      /NEG. LENGTH OF OFFSET
3678      04056 R 340166 R      TAD      SPOINT
3679      04057 R 040166 R      DAC      SPOINT     /NEW END OF STACK
3680      04060 R 200165 R      LAC      STACK
3681      04061 R 046621 R      DAC      CONT
3682      04062 R 340266 R      TAD      GASS
3683      04063 R 040165 R      DAC      STACK     /NEW START OF STACK
3684      04064 R 040266 R      DAC      GASS
3685      04065 R 740031 A      CMA!IAC
3686      04066 R 340167 R      TAD      END
3687      04067 R 740300 A      SMA!SZA      /PARTITION BIG ENOUGH?
3688      04070 R 000001 A      ERR1     /NO, "STORE": NOT ENOUGH ROOM FOR INCREASE
3689      04071 R 226621 R      NSTKR    LAC*     CONT      /TRANSFER STACK
3690      04072 R 060266 R      DAC*     GASS
3691      04073 R 446621 R      ISZ      CONT
3692      04074 R 440266 R      ISZ      GASS
3693      04075 R 777777 A      LAW      -1
3694      04076 R 346621 R      TAD      CONT

```

3695	04077 R 540170 R	SAD	SYMBEG	/END OF TRANSFER?
3696	04100 R 751000 A	SKP!CLA		/YES
3697	04101 R 604071 R	JMP	NSIKR	/NO
3698	04102 R 604042 R	JMP	XRET2	
3699		.EJECT		

```

3700      /XDECIN - CONVERT NUMERIC STRING TO BINARY
3701      /
3702      /      ENTER WITH FIRST CHAR OF STRING IN AC AND OPOINT TO BE INDEXED TO NEXT CHAR.
3703      /      ANY ACCEPTABLE LEADING SIGNS MUST BE HANDLED BY CALLING ROUTINE BEFORE CALL DECIN
3704      /      THERE ARE TWO EXITS FROM DECIN: RETURN TO CALL DECIN+1 IF ALPHAS OTHER THAN ONE
3705      /      DECIMAL POINT ARE ENCOUNTERED, IF RESULT IS > 1310.71, IF RESULT IS < .01; SYNTAX
3706      /      IS SET TO THE APPROPRIATE CAL. (THIS IS SO THAT $V CAN EASILY RETURN ZERO FOR
3707      /      SUCH ERRORS INSTEAD OF KLUDGING UP THE ERROR PROCESSOR).
3708      /      RETURN TO CALL DECIN+2 WHEN PUNCTUATION OTHER THAN DECIMAL POINT ENCOUNTERED
3709      /      IMPLYING END OF INPUT STRING ENCOUNTERED. CALLING ROUTINE MUST
3710      /      CHECK FOR ACCEPTABLE PUNCTUATION AT THIS POINT.
3711      /
3712      /
3713      04103 R 000000 A      DECIN      0                                /*12JAH17*
3714      04104 R 043724 R      DAC        SYNTAX      /SAVE FIRST CHAR      /*12JAH17*
3715      04105 R 777773 A      XDECM5    LAW        -5                                /SETUP COUNTER FOR LEGAL # OF DIGITS
3716      04106 R 040214 R      DAC        M                                /BEFORE THE DECIMAL POINT
3717      04107 R 140217 R      DZM        ACTEMP      /ZERO THE "AC"
3718      04110 R 203724 R      LAC        SYNTAX
3719      04111 R 741000 A      SKP
3720      04112 R 104660 R      XDEC00    CALL      OCHAR      /GET NEXT CHAR.
3721      04113 R 103724 R      CALL      SYNTAX      /WHAT TYPE IS IT?
3722      04114 R 604164 R      JMP        DECER3     /A -- "HARD" SYNTAX ERROR
3723      04115 R 604121 R      JMP        DECNUM     /N -- PROCESS THE NUMBER
3724      04116 R 553742 R      SAD        (56        /P -- DECIMAL POINT?
3725      04117 R 604151 R      JMP        DECPT     /YES, PROCESS IT.
3726      04120 R 604206 R      JMP        XDOUT     /NO, GO FINISH UP NUMBER AND GO AWAY.
3727      04121 R 513743 R      DECNUM    AND        (17        /STRIP OFF THE ASCII BITS.
3728      04122 R 043724 R      DAC        SYNTAX      /SAVE FOR NOW
3729      04123 R 200214 R      LAC        M
3730      04124 R 553705 R      SAD        (2          /USED UP ALL OUR DECIMAL PLACES?
3731      04125 R 604156 R      JMP        DECPT0     /YES, ALLOW ONLY TRAILING ZERO.
3732      04126 R 343724 R      TAD        SYNTAX      /LEADING ZERO (I.E. CHAR=0 & M=-5)?
3733      04127 R 544105 R      SAD        XDECM5
3734      04130 R 604112 R      JMP        XDEC00     /YES, IGNORE IT.
3735      04131 R 440214 R      ISZ        M          /WILL THIS EXCEED # OF LEGAL DIGITS IN INT. PART?
3736      04132 R 741000 A      SKP
3737      04133 R 604203 R      JMP        DECER6     /YES, "HARD" MAXIM ERROR.
3738      04134 R 200217 R      LAC        ACTEMP      /NO CONTINUE FORMING NUMBER.
3739      04135 R 653122 A      MUL;      12
3740      04136 R 000012 A
3740      04137 R 744200 A      SZA!CLL      /OVERFLO?
3741      04140 R 604167 R      JMP        DECER4     /YES - MAXIM
3742      04141 R 641002 A      LACQ
3743      04142 R 343724 R      TAD        SYNTAX
3744      04143 R 741400 A      SZL          /OVERFLO?
3745      04144 R 604173 R      JMP        DECER5     /YES - MAXIM
3746      04145 R 741100 A      SPA          /OR MAYBE A LITTLE OVERFLO?
3747      04146 R 604201 R      JMP        DECR55     /YUP - JUST A LITTLE
3748      04147 R 040217 R      DAC        ACTEMP
3749      04150 R 604112 R      JMP        XDEC00
3750      04151 R 200214 R      DECPT    LAC        M          /TOO MANY DECIMAL POINTS?

```

```

3751      04152 R 740100 A          SMA
3752      04153 R 604164 R          JMP      DECER3      /YES, "HARD" SYNTAX ERROR.
3753      04154 R 140214 R          DZM      M            /NO, SET DECIMAL PLACE.
3754      04155 R 604112 R          JMP      XDEC00
3755      04156 R 203724 R          DECPI0  LAC      SYNTAX
3756      04157 R 751200 A          SNA!CLA          /TRAILING ZERO?
3757      04160 R 604112 R          JMP      XDEC00      /YES, GO ON
3758      04161 R 723007 A          AAC      7            /NO, "HARD" MINIM ERROR
3759      04162 R 043724 R          DAC      SYNTAX
3760      04163 R 604207 R          JMP      XDOUTE      /TO ERROR EXIT
3761      04164 R 213744 R          DECER3  LAC      (3          /"SYNTAX" ERROR ENCOUNTERED -- SETUP RETURN.
3762      04165 R 043724 R          DAC      SYNTAX
3763      04166 R 604207 R          JMP      XDOUTE      /TO ERROR EXIT
3764      04167 R 200217 R          DECER4  LAC      ACTEMP      /GET UNSCALED RESULT
3765      04170 R 653323 A          IDIV;   144         /TRUNCATE 2 DIGITS
          04171 R 000144 A
3766      04172 R 604176 R          JMP      DECR53
3767      04173 R 200217 R          DECER5  LAC      ACTEMP      /GET RESULT BEFORE SCALING
3768      04174 R 653323 A          IDIV;   12          /TRUNCATE 1 BIT
          04175 R 000012 A
3769      04176 R 641002 A          DECR53  LACQ
3770      04177 R 040217 R          DAC      ACTEMP      /STORE RESULT FOR SCALING
3771      04200 R 751000 A          CLA!SKP          /M=0 FOR MAXIMUM SCALING
3772      04201 R 750030 A          DECR55  CLA!IAC      /M=1 FOR 1 PLACE SCALING
3773      04202 R 040214 R          DAC      M
3774      04203 R 213745 R          DECER6  LAC      (6          /"MAXIM" ERROR
3775      04204 R 043724 R          DAC      SYNTAX
3776      04205 R 604207 R          JMP      XDOUTE      /TO ERROR EXIT
3777      04206 R 143724 R          XDOUT   DZM      SYNTAX      /NOTE THAT ALL IS GOOD - SO FAR
3778      04207 R 200217 R          XDOUTE  LAC      ACTEMP
3779      04210 R 652000 A          LMQ
3780      04211 R 200214 R          LAC      M
3781      04212 R 553705 R          SAD      (2          /M=2 (I.E. FILLED IN BOTH DEC. PLACES)?
3782      04213 R 604227 R          JMP      XDOUT2      /YES, NO MULTIPLY NEEDED.
3783      04214 R 740300 A          SZA!SMA          /M=1 (I.E. FILLED IN ONLY 1 DEC. PLACE)?
3784      04215 R 604222 R          JMP      XDOUT1      /YES, MULTIPLY BY 10 (MQ LOADED)
3785      04216 R 641122 A          MUL-LMQ+EAE      /NO DEC. PLACES FILLED IN, MULTIPLY BY 100
3786      04217 R 000012 A          12
3787      04220 R 740200 A          SZA          /OVERFLO?
3788      04221 R 604167 R          JMP      DECER4      /YES - MAXIM
3789      04222 R 641122 A          XDOUT1  MUL-LMQ+EAE
3790      04223 R 000012 A          12
3791      04224 R 740200 A          SZA          /OVERFLO?
3792      04225 R 604173 R          JMP      DECER5      /YES - MAXIM
3793      04226 R 641002 A          LACQ          /GET ANSWER
3794      04227 R 751100 A          XDOUT2  SPA!CLA      /WITHIN MUMPS RANGE?
3795      04230 R 604173 R          JMP      DECER5      /NO - MAXIM
3796      04231 R 543724 R          SAD      SYNTAX      /ALL OK?
3797      04232 R 444103 R          ISZ      DECIN      /YES, BUMP RETURN
3798      04233 R 641002 A          LACQ          /GET ANSWER
3799      04234 R 624103 R          JMP*     DECIN
3800      .EJECT

```

```

3801          /XDEC2 - OUTPUT A DECIMAL CHARACTER
3802          /
3803          /      ENTER WITH DIGIT'S VALUE IN THE MQ
3804          /      "REMOVE (TEMP STORAGE)" = 1 IF ZERO VALUES
3805          /      ARE TO BE OUTPUT.
3806          /
3807          /      (IF CALLED FROM DECOUT, MQ CONTAINS RMNDR FROM IDIV)
3808          /
3809          /
3810          04235 R 000000 A      DEC2      0                                /*12JAH17*
3811          04236 R 040217 R      DAC      ACTEMP          /SAVE THE AC                                /*12JAH17*
3812          04237 R 210005 R      LAC      REMOVE          /GET SWITCH (REMOVE USED AS TEMP STORAGE)
3813          04240 R 740010 A      RAL
3814          04241 R 641002 A      LACQ
3815          04242 R 740201 A      SZA!CMA
3816          04243 R 604246 R      JMP      XDEC2A          /*12JAH12*
3817          04244 R 744400 A      SNL!CLL          / ZERO VALUE. TYPE IT?
3818          04245 R 604253 R      JMP      XDEC2B          /*12JAH12* NO , EXIT
3819          04246 R 050005 R      XDEC2A  DAC      REMOVE          /*12JAH12* YES, AND SET SWITCH
3820          04247 R 641002 A      LACQ
3821          04250 R 313746 R      ADD      (60            /CONVERT TO 6-BIT
3822          04251 R 744000 A      CLL
3823          04252 R 103174 R      CALL     OUT            /OUTPUT THE CHARACTER.
3824          04253 R 200217 R      XDEC2B  LAC      ACTEMP          /*12JAH12*
3825          04254 R 624235 R      JMP*    DEC2            / EXIT WITH THE AC RESTORED
3826          .EJECT

```

```

3827          /XDECOUT - CONVERT BINARY TO NUMERIC STRING
3828          /
3829          /      ENTER WITH BINARY NUMBER IN THE AC
3830          /
3831          04255 R 000000 A      DECOUT  0                                /*12JAH17*
3832          04256 R 664000 A      GSM          /SIGN INTO LINK                                /*12JAH17*
3833          04257 R 040217 R      DAC      ACTEMP      /SAVE ABS VALUE.
3834          04260 R 760055 A      LAW      55          /-
3835          04261 R 745400 A      SZL!CLL      /IS NUM NEG?
3836          04262 R 103174 R      CALL      OUT      /YES, TYPE MINUS SIGN
3837          04263 R 200217 R      LAC      ACTEMP
3838          04264 R 150005 R      DZM      REMOVE      / TYPE-ZEROS-SWITCH: IF NEGATIVE TYPE ZERO
3839          .DEC          /
3840          04265 R 653323 A      IDIV;    100000      ELSE IGNORE ZEROS
          04266 R 303240 A
3841          04267 R 104235 R      CALL      DEC2      /THOUSANDS CHAR POSITION.
3842          04270 R 653323 A      IDIV;    10000
          04271 R 023420 A
3843          04272 R 104235 R      CALL      DEC2      /HUNDREDS CHAR POSITION.
3844          04273 R 653323 A      IDIV;    1000
          04274 R 001750 A
3845          04275 R 104235 R      CALL      DEC2      /TENS CHAR POSITION.
3846          04276 R 777777 A      LAW      -1
3847          04277 R 050005 R      DAC      REMOVE      / SET SWITCH TO TYPE ZEROS
3848          04300 R 200217 R      LAC      ACTEMP
3849          04301 R 653323 A      IDIV;    100
          04302 R 000144 A
3850          .OCT
3851          04303 R 104235 R      CALL      DEC2      /UNITS CHAR POSITION.
3852          04304 R 200217 R      LAC      ACTEMP
3853          04305 R 741200 A      SNA          /IS NUMBER AN INTEGER?
3854          04306 R 624255 R      JMP*     DECOUT     /YES, EXIT.
3855          04307 R 760056 A      LAW      56          /NO.
3856          04310 R 103174 R      CALL      OUT      / TYPE DECIMAL POINT
3857          04311 R 200217 R      LAC      ACTEMP
3858          .DEC
3859          04312 R 653323 A      IDIV;    10
          04313 R 000012 A
3860          .OCT
3861          04314 R 104235 R      CALL      DEC2      /TENTH'S CHAR POSITION.
3862          04315 R 653000 A      LMQ!CLAC
3863          04316 R 104235 R      CALL      DEC2      /HUNDREDTH'S CHAR POSITION.
3864          04317 R 624255 R      JMP*     DECOUT     /EXIT.
3865          .EJECT

```

```

3866 /
3867 /XOCTOUT - CONVERT BINARY TO OCTAL STRING
3868 /
3869 / ENTER WITH BINARY NUMBER IN AC
3870 /
3871 04320 R 000000 A OCTOUT 0 /*12JAH17*
3872 04321 R 040217 R DAC ACTEMP /SAVE NUMBER /*12JAH17*
3873 04322 R 777772 A LAW -6 /SIX OCTAL DIGIITS = 1 WORD
3874 04323 R 044255 R DAC DECOU
3875 04324 R 200217 R LAC ACTEMP
3876 04325 R 652000 A OCT2 LMQ
3877 04326 R 750000 A CLA
3878 04327 R 640603 A LLS 3 /PICK UP NEXT OCTAL DIGIT
3879 04330 R 723060 A AAC 60
3880 04331 R 103174 R CALL OUT
3881 04332 R 200217 R LAC ACTEMP /RETRIEVE WHAT'S LEFT OF OCTAL DIGITS
3882 04333 R 640603 A LLS 3 /GET RID OF DIGIT JUST PRINTED
3883 04334 R 040217 R DAC ACTEMP
3884 04335 R 444255 R ISZ DECOU /ALL DIGITS PRINTED?
3885 04336 R 604325 R JMP OCT2 /NO
3886 04337 R 624320 R JMP* OCTOUT /YES, EXIT
3887 /
3888 .EJECT

```

```

3889      /XPLACE - PLACE A STEP INTO PROGRAM STEP BUFFER
3890      /
3891      /      (STEPS ARE HELD IN NUMERICAL ORDER WITH THE STEP NUMBER IN
3892      /      BINARY - 1 WORD - FOLLOWED BY THE REMAINDER OF THE STEP IN
3893      /      STRING FORMAT)
3894      /
3895      /
3896      04340 R 000000 A      PLACE      0                                /*12JAH17*
3897      04341 R 104645 R      CALL      OCHAR      /GET FIRST CHAR      /*12JAH17*
3898      04342 R 104103 R      CALL      DECIN      /CONVERT TO NUMERIC
3899      04343 R 603724 R      JMP      SYNTAX      /ALL ERRORS FOUND BY DECIN ARE ERRORS HERE.
3900      04344 R 040207 R      DAC      VAL
3901      04345 R 653323 A      IDIV;     144
3902      04346 R 000144 A
3903      04347 R 741200 A      SNA
3904      04350 R 000014 A      ERR14      / "FRACT": STEP NUMBERS MAY NOT BE INTEGERS
3905      04351 R 104645 R      CALL      OCHAR
3906      04352 R 553653 R      SAD      (40
3907      04353 R 745000 A      SKP!CLL
3908      04354 R 000003 A      ERR3
3909      04355 R 104357 R      CALL      PLAC1      /"SYNTAX": A SPACE MUST FOLLOW THE STEP NUMBER
3910      04356 R 624340 R      JMP*     PLACE      /COMPLETE PLACEMENT
3911      /
3912      /XPLAC1 - COMPLETE PLACING A STEP INTO PROGRAM BUFFER
3913      /
3914      /      USED BY MODIFY COMMAND AS WELL AS XPLACE
3915      /
3916      04357 R 000000 A      PLAC1     0                                /*12JAH17*
3917      04360 R 200223 R      LAC      OPLACE      /POS PTR (POINTING TO SPACE) /*12JAH17*
3918      04361 R 553732 R      SAD      (XC01      /*12JAH14* TO 2ND CHAR?
3919      04362 R 604374 R      JMP      XPLAC3      /*12JAH14* YES
3920      04363 R 553747 R      SAD      (XC02      /*12JAH14* TO 2ND CHAR?
3921      04364 R 604371 R      JMP      XPLAC2      /*12JAH14* YES
3922      04365 R 200221 R      LAC      IPLACE      /*12JAH14* NO, 1ST CHAR (POINTING TO EOM)
3923      04366 R 553727 R      SAD      (XC13      /*12JAH14* LAST PTR TO 1ST CHAR
3924      04367 R 604374 R      JMP      XPLAC3      /*12JAH14* YES
3925      04370 R 604373 R      JMP      XPLAC4      /*12JAH14* NO
3926      04371 R 200221 R      XPLAC2   LAC      IPLACE      /*12JAH14* LAST PTR
3927      04372 R 553730 R      SAD      (XC11      /*12JAH14* TO 2ND CHAR?
3928      04373 R 751030 A      XPLAC4   CLA!IAC!SKP      /*12JAH14* YES
3929      04374 R 750000 A      XPLAC3   CLA
3930      04375 R 340222 R      TAD      OPOINT      /*12JAH14* UPDATE WORD
3931      04376 R 740031 A      CMA!IAC      /*12JAH14* PTR
3932      04377 R 340220 R      TAD      IPOINT      /*12JAH14*
3933      04400 R 513702 R      AND      (77
3934      04401 R 553702 R      SAD      (77
3935      04402 R 750000 A      CLA
3936      04403 R 044645 R      DAC      OCHAR      / LENGTH OF NEW STEP
3937      04404 R 744001 A      CMA!CLL
3938      04405 R 044725 R      DAC      ICHARI
3939      04406 R 220145 R      LAC*     LVAR      / GET STEP NUMBER OF STEP BEING EXECUTED

```

3940	04407 R 540207 R		SAD VAL	/ (0 IF IN DIRECT MODE)
3941	04410 R 000015 A		ERR15	/ "STEPS": THE STEP BEING EXECUTED CAN'T
3942				/ BE REPLACED
3943	04411 R 200134 R		LAC BUFF	/ NO, START AT THE BEGINNING OF THE STEP BUFFER
3944	04412 R 040014 A		DAC R14	/ YES
3945	04413 R 140652 R		DZM SWAPSW	/INHIBIT SWAP DURING PROG UPDATE
3946	04414 R 220014 A	XPLAC5	LAC* R14	
3947	04415 R 653606 A		LMQ!CLAC!LLS+6	/OFFSET OF THIS STEP
3948	04416 R 300014 A		ADD R14	
3949	04417 R 040014 A		DAC R14	/ POINTING TO NEXT STEP NUMBER NOW
3950	04420 R 540167 R		SAD END	/ END OF STEP BUFFER?
3951	04421 R 604516 R		JMP APPEND	/ YES, APPEND NEW STEP TO END OF BUFFER
3952	04422 R 220014 A		LAC* R14	/ NO, GET NEXT STEP'S NUMBER
3953	04423 R 740001 A		CMA	
3954	04424 R 300207 R		ADD VAL	
3955	04425 R 740101 A		SMA!CMA	/ IS OLD STEP'S NUMBER (* 14) = OR >
3956				/NEW ONE (VAL)?
3957	04426 R 604414 R		JMP XPLAC5	/NO KEEP LOOKING
3958	04427 R 744200 A		SZA!CLL	/ YES
3959	04430 R 604535 R		JMP INSERT	/ OLD>NEW: GO INSERT NEW STEP
3960	04431 R 200014 A		LAC R14	/ OLD=NEW: REPLACE OLD STEP WITH NEW
3961	04432 R 040015 A		DAC R15	
3962	04433 R 220015 A		LAC* R15	
3963	04434 R 653606 A		LMQ!CLAC!LLS+6	/ <AC> BECOMES OFFSET OF OLD STEP
3964	04435 R 300015 A		ADD R15	
3965	04436 R 540167 R		SAD END	
3966	04437 R 604527 R		JMP PLACE8	/REPLACE LAST STEP IN STEP BUFFER
3967	04440 R 040015 A		DAC R15	/<15>=<14>+1+OFFSET
3968	04441 R 740031 A		CMA!IAC	
3969	04442 R 300014 A		ADD R14	/ FORMS NEGATIVE LENGTH OF OLD STEP
3970	04443 R 304645 R		ADD OCHAR	/ FORMS DIFFERENCE IN LENGTH BETWEEN
3971				/OLD AND NEW
3972	04444 R 740100 A		SMA	/ IS OLD STEP LONGER THAN NEW?
3973	04445 R 604542 R		JMP PLACE6	/ NO; GO MAKE ROOM FOR NEW STEP
3974	04446 R 204645 R	PLACE7	LAC OCHAR	/ YES; PLACE NEW STEP INTO BUFFER
3975	04447 R 444725 R		ISZ ICHARI	/ <ICHARI>=NEGATIVE LENGTH OF NEW STEP
3976	04450 R 604453 R		JMP PLAC3A	/*12JAH12*
3977	04451 R 604473 R		JMP PLACE5	/ONE CHARACTER STEP
3978	04452 R 104660 R	PLACE3	CALL OCHARI	/ PLACE NEW STEP INTO STEP BUFFER
3979	04453 R 640706 A	PLAC3A	ALS 6	/*12JAH12*
3980	04454 R 044645 R		DAC OCHAR	
3981	04455 R 104660 R		CALL OCHARI	
3982	04456 R 244645 R		XOR OCHAR	
3983	04457 R 640706 A		ALS 6	
3984	04460 R 044645 R		DAC OCHAR	
3985	04461 R 104660 R		CALL OCHARI	
3986	04462 R 244645 R		XOR OCHAR	
3987	04463 R 060014 A		DAC* R14	
3988	04464 R 444725 R		ISZ ICHARI	/*12JAH14* LAST WORD?
3989	04465 R 604452 R		JMP PLACE3	/NO
3990	04466 R 104660 R		CALL OCHARI	/YES
3991	04467 R 741200 A		SNA	/ ANY MORE CHARACTERS?

3992	04470	R	604476	R		JMP	PLACE4	/ NO, ITS AN EDM
3993	04471	R	640706	A		ALS	6	/YES
3994	04472	R	044645	R		DAC	OCHAR	
3995	04473	R	104660	R	PLACE5	CALL	OCHARI	
3996	04474	R	244645	R		XOR	OCHAR	
3997	04475	R	640706	A		ALS	6	
3998	04476	R	060014	A	PLACE4	DAC*	R14	
3999	04477	R	200015	A		LAC	R15	
4000	04500	R	540014	A		SAD	R14	/EXACT FIT?
4001	04501	R	604512	R		JMP	PLAC4B	/*12JAH12* YES
4002	04502	R	540167	R		SAD	END	/NO
4003	04503	R	604510	R		JMP	PLAC4A	/*12JAH12* MOVE UP
4004	04504	R	220015	A		LAC*	R15	
4005	04505	R	060014	A		DAC*	R14	
4006	04506	R	200015	A		LAC	R15	
4007	04507	R	604502	R		JMP	.-5	
4008	04510	R	200014	A	PLAC4A	LAC	R14	/*12JAH12*
4009	04511	R	103744	R		CALL	STOR1	
4010	04512	R	440652	R	PLAC4B	ISZ	SWAPSW	/*12JAH12* TIME SLICE UP?
4011	04513	R	624357	R		JMP*	PLAC1	/NO
4012	04514	R	101533	R		JMS	HANG	/YES, LET SOMEONE ELSE RUN
4013	04515	R	624357	R		JMP*	PLAC1	
4014	04516	R	200014	A	APPEND	LAC	R14	/APPEND
4015	04517	R	304645	R		ADD	OCHAR	
4016	04520	R	723002	A		AAC	2	
4017	04521	R	040015	A		DAC	R15	
4018	04522	R	103744	R		CALL	STOR1	
4019	04523	R	200207	R		LAC	VAL	
4020	04524	R	060014	A		DAC*	R14	
4021	04525	R	604446	R		JMP	PLACE7	
4022	04526	R	060014	A		DAC*	R14	
4023	04527	R	200014	A	PLACE8	LAC	R14	
4024	04530	R	304645	R		ADD	OCHAR	
4025	04531	R	740030	A		IAC		
4026	04532	R	040015	A		DAC	R15	
4027	04533	R	103744	R		CALL	STOR1	
4028	04534	R	604446	R		JMP	PLACE7	
4029	04535	R	200014	A	INSERT	LAC	R14	/INSERT
4030	04536	R	040015	A		DAC	R15	
4031	04537	R	204645	R		LAC	OCHAR	
4032	04540	R	313705	R		ADD	(2	
4033	04541	R	741000	A		SKP		
4034	04542	R	440015	A	PLACE6	ISZ	R15	
4035	04543	R	044660	R		DAC	OCHARI	
4036	04544	R	200167	R		LAC	END	
4037	04545	R	045103	R		DAC	UNSAVE	
4038	04546	R	304660	R		ADD	OCHARI	
4039	04547	R	103744	R		CALL	STOR1	/ RTN WITH <AC>=END
4040	04550	R	045040	R	MOVEDN	DAC	SAVE	
4041	04551	R	225103	R		LAC*	UNSAVE	
4042	04552	R	065040	R		DAC*	SAVE	
4043	04553	R	777776	A		LAW	-2	

4044	04554	R	305103	R	ADD	UNSAVE		
4045	04555	R	045103	R	DAC	UNSAVE		
4046	04556	R	540015	A	SAD	R15	/15=OCTAL STEP #	
4047	04557	R	604562	R	JMP	MOVD1	/*12JAH12*	
4048	04560	R	304660	R	ADD	OCHARI		
4049	04561	R	604550	R	JMP	MOVEDN		
4050	04562	R	304660	R	MOVD1	ADD	OCHARI	/*12JAH12*
4051	04563	R	045040	R	DAC	SAVE		
4052	04564	R	313712	R	ADD	(777776	/-2, REALLY -1 (WHAT CLUTZ WROTE THIS?)	
4053	04565	R	040015	A	DAC	R15		
4054	04566	R	225103	R	LAC*	UNSAVE		
4055	04567	R	065040	R	DAC*	SAVE		
4056	04570	R	200207	R	LAC	VAL		
4057	04571	R	065103	R	DAC*	UNSAVE		
4058	04572	R	604446	R	JMP	PLACE7		
4059					.EJECT			

```

4060          .IFPNZ  MODIFY
4061          /
4062          /
4063          /XMBUF2 - MOVE STEP FROM PARTITION TO STRING BUFFER
4064          /
4065          /
4066          04573 R 000000 A      MBUF2  0                      /*12JAH17*
4067          04574 R 040017 A      DAC    R17                  /START OF OLD LOC      /*12JAH17*
4068          04575 R 220017 A      LAC*   R17
4069          04576 R 653606 A      LMQ!CLAC!LLS+6
4070          04577 R 040214 R      DAC    M                    /STEP COUNT
4071          04600 R 340017 A      TAD    R17
4072          04601 R 040014 A      DAC    R14                  /END OF OLD LOC
4073          04602 R 200017 A      LAC    R17
4074          04603 R 723776 A      AAC    -2
4075          04604 R 040017 A      DAC    R17                  /SET UP AUTO-INDEX
4076          04605 R 200133 R      LAC    BUFF2                 /START OF STRING BUFFER
4077          04606 R 723777 A      AAC    -1
4078          04607 R 040015 A      DAC    R15
4079          04610 R 220017 A      MSBUF  LAC*   R17             /MOVE WORD
4080          04611 R 060015 A      DAC*   R15
4081          04612 R 200017 A      LAC    R17
4082          04613 R 540014 A      SAD    R14                  /DONE?
4083          04614 R 624573 R      JMP*   MBUF2                 /YES
4084          04615 R 604610 R      JMP    MSBUF                 /NO
4085          /
4086          /
4087          /XERMOD - ERASE STEP FROM PARTITION
4088          /
4089          /
4090          04616 R 000000 A      ERMOD  0                      /*12JAH17*
4091          04617 R 040207 R      DAC    VAL                    /SAVE STEP #          /*12JAH17*
4092          04620 R 105123 R      CALL   STEP                  /FIND IT
4093          04621 R 777777 A      LAW    -1
4094          04622 R 340016 A      TAD    R16
4095          04623 R 040014 A      DAC    R14                  /START OF STEP
4096          04624 R 220016 A      LAC*   R16
4097          04625 R 653606 A      LMQ!CLAC!LLS+6              /STEP COUNT
4098          04626 R 340016 A      TAD    R16
4099          04627 R 040015 A      DAC    R15                  /END OF STEP
4100          04630 R 140652 R      DZM    SWAPSW                /INHIBIT SWAP DURING PRG UPDATE
4101          04631 R 540167 R      ER55  SAD    END              /DONE?
4102          04632 R 604637 R      JMP    ER6                  /YES
4103          04633 R 220015 A      LAC*   R15                  /MOVE NEXT
4104          04634 R 060014 A      DAC*   R14                  /WORD UP IN PARTITION
4105          04635 R 200015 A      LAC    R15
4106          04636 R 604631 R      JMP    ER55
4107          04637 R 200014 A      ER6   LAC    R14              /RESET "END"
4108          04640 R 103744 R      CALL   STOR1
4109          04641 R 440652 R      ISZ    SWAPSW                /TIME SLICE UP?
4110          04642 R 624616 R      JMP*   ERMOD                 /NO
4111          04643 R 101533 R      JMS    HANG                  /YES, LET SOMEONE ELSE RUN

```

```
4112 04644 R 624616 R      JMP*   ERMOD
4113                                     /
4114                                     .ENDC
4115                                     .EJECT
```

```

4116          /XCHAR - LOAD A CHARACTER USING OPOINT & OPLACE
4117          /
4118          / OPOINT CONTAINS PTR TO WORD CONTAINING CHARACTER
4119          / OPLACE CONTAINS (XC01, (XC02, OR (XC03 DEPENDING ON WHETHER THE
4120          / CHARACTER JUST EXTRACTED WAS THE FIRST, SECOND OR THIRD RESPECTIVELY
4121          / IN THE WORD POINTED TO BY OPOINT
4122          /
4123          04645 R 000000 A      OCHAR      0                      /*12JAH17*
4124          04646 R 220222 R      LAC*      OPOINT          /FETCH ALL 3 CHARS OF WORD      /*12JAH17*
4125          04647 R 744000 A      CLL                      /*12JAH14* LINK MUST BE CLEAR UPON EXIT
4126          04650 R 420223 R      XCT*      OPLACE          /*12JAH14* PICK & EXTRACT CHAR
4127          04651 R 624645 R      JMP*      OCHAR          /*12JAH14* 3RD CHAR
4128          04652 R 742020 A      RTR ; RTR ; RTR          /*12JAH14* 2ND CHAR
4129          04653 R 742020 A
4130          04654 R 742020 A
4129          04655 R 513702 R      AND        (77          /*12JAH14* REMOVE EXTRANEIOUS BITS
4130          04656 R 744000 A      CLL                      /*12JAH14* LINK MUST BE CLEAR UPON EXIT
4131          04657 R 624645 R      JMP*      OCHAR          /*12JAH14*
4132          /
4133          /
4134          /
4135          /XCHARI - INDEX AND LOAD A CHARACTER USING OPOINT AND OPLACE
4136          /
4137          04660 R 000000 A      OCHARI     0                      /*12JAH17*
4138          04661 R 440223 R      ISZ        OPLACE          /ADVANCE TO NEXT CHAR POS      /*12JAH17*
4139          04662 R 220222 R      LAC*      OPOINT          /*12JAH14* FETCH ALL 3 CHARS OF WORD
4140          04663 R 744000 A      CLL                      /*12JAH14* LINK MUST BE CLEAR UPON EXIT
4141          04664 R 420223 R      XCT*      OPLACE          /*12JAH14* SELECT CHAR POS & EXTRACT CHAR
4142          04665 R 624660 R      JMP*      OCHARI         /*12JAH14* 3RD CHAR
4143          04666 R 747420 A      SZL!RTR!CLL          /*12JAH14* MIDDLE CHAR?
4144          04667 R 604675 R      JMP        XCH2          /*12JAH14* NO, ADVANCE OPOINT
4145          04670 R 742020 A      RTR ; RTR          /*12JAH14* YES
4146          04671 R 742020 A
4146          04672 R 513702 R      AND        (77          /*12JAH14* REMOVE EXTRANEIOUS BITS
4147          04673 R 744000 A      CLL                      /*12JAH14* LINK MUST BE CLEAR UPON EXIT
4148          04674 R 624660 R      JMP*      OCHARI         /*12JAH14*
4149          04675 R 440222 R      XCH2     ISZ        OPOINT          /*12JAH14* ADVANCE TO NEW SORD
4150          04676 R 213732 R      LAC        (XC01          /*12JAH14* POSITION TO
4151          04677 R 040223 R      DAC        OPLACE          /*12JAH14* FIRST CHAR
4152          04700 R 220222 R      LAC*      OPOINT          /*12JAH14* FETCH NEW WORD
4153          04701 R 653606 A      CLAC!LMQ!LLS+6          /*12JAH14* SHIFT IN LEFTMOST CHAR
4154          04702 R 744000 A      CLL                      /*12JAH14*
4155          04703 R 624660 R      JMP*      OCHARI         /*12JAH14*
4156          /
4157          04704 R 653606 A      XC01     CLAC!LMQ!LLS+6          /*12JAH14* EXTRACT 1ST CHAR
4158          04705 R 745000 A      XC02     SKP!CLL          /*12JAH14* EXTRACT 2ND CHAR
4159          04706 R 513702 R      XC03     AND        (77          /*12JAH14* EXTRACT 3RD CHAR
4160          04707 R 745002 A      XC04     SKP!STL          /*12JAH14* ADVANCE TO 1ST CHAR OF NEXT WORD
4161          /
4162          .EJECT

```

```
4163 /
4164 /XSOPT - BACKSPACE ONE CHARACTER
4165 /
4166 04710 R 000000 A SUBOPT 0 /*12JAH17*
4167 04711 R 777777 A LAW -1 /MOVE /*12JAH17*
4168 04712 R 340223 R TAD OPLACE /*12JAH14* CHAR POSIION
4169 04713 R 553750 R SAD (XC01-1 /*12JAH14* BACK 1. STILL IN SAME WORD?
4170 04714 R 604717 R JMP XSOP1 /*12JAH14* NO
4171 04715 R 040223 R DAC OPLACE /*12JAH14* YES
4172 04716 R 624710 R JMP* SUBOPT /*12JAH14*
4173 04717 R 777777 A XSOP1 LAW -1 /*12JAH14* MOVE WORD
4174 04720 R 340222 R TAD OPOINT /*12JAH14* PTR BACK
4175 04721 R 040222 R DAC OPOINT /*12JAH14* ONE
4176 04722 R 213736 R LAC (XC03 /*12JAH14* POINT TO
4177 04723 R 040223 R DAC OPLACE /*12JAH14* 3RD CHAR
4178 04724 R 624710 R JMP* SUBOPT /*12JAH14*
4179 /
4180 .EJECT
```

```

4181 /
4182 /XICHARI - INDEX AND DEPOSIT A CHARACTER USING IPOINT AND IPLACE
4183 /
4184 / IPOINT CONTAINS PTR TO WORD TO CONTAIN CHARACTERS
4185 / IPLACE CONTAINS (XCI1, (XCI2, (XCI3 DEPENDING ON WHETHER THE NEXT
4186 / CHAR TO BE DEPOSITED SHOULD GO INTO THE 1ST , 2ND, OR 3RD CHAR
4187 / RESPECTIVELY OF THE WORD.
4188 /
4189 04725 R 000000 A ICHARI 0 /*12JAH17*
4190 04726 R 620221 R JMP* IPLACE /SELECT CHAR POSITION /*12JAH17*
4191 04727 R 746020 A IENTO CLL!RTR ; RTR ; RTR ; RAR /*12JAH14* POS CHAR IN LEFTMOST FIELD
04730 R 742020 A
04731 R 742020 A
04732 R 740020 A
4192 04733 R 652000 A LMQ /*12JAH14* STATCH IT AWAY
4193 04734 R 213730 R LAC (XCI1 /*12JAH14* SET UP
4194 04735 R 040221 R DAC IPLACE /*12JAH14* NEXT CHAR POS
4195 04736 R 440220 R ISZ IPOINT /*12JAH14* ADVANCE TO NEXT WORD
4196 04737 R 220220 R LAC* IPOINT /*12JAH14* FETCH CHARS
4197 04740 R 513645 R AND (7777 /*12JAH14* SAVE REST OF WORD
4198 04741 R 640002 A OMQ /*12JAH14* INSERT NEW CHAR
4199 04742 R 060220 R DAC* IPOINT /*12JAH14* RETURN TO TARGET WORD
4200 04743 R 624725 R JMP* ICHARI /*12JAH14*
4201 04744 R 440221 R IENT1 ISZ IPLACE /*12JAH14* ADVANCE CHAR POS
4202 04745 R 746010 A CLL!RTL ; RTL ; RTL /*12JAH14* POS CHAR IN MIDDLE FIELD
04746 R 742010 A
04747 R 742010 A
4203 04750 R 652000 A LMQ /*12JAH14* STATCH IT AWAY
4204 04751 R 770077 A LAW -7701 /*12JAH14* SAVE OLD 1ST &
4205 04752 R 520220 R AND* IPOINT /*12JAH14* 3RD CHARS
4206 04753 R 640002 A OMQ /*12JAH14* INSERT NEW CHAR
4207 04754 R 060220 R DAC* IPOINT /*12JAH14* RETURN TO TARGET WORD
4208 04755 R 624725 R JMP* ICHARI /*12JAH14*
4209 04756 R 440221 R IENT2 ISZ IPLACE /*12JAH14* ADVANCE CHAR POS
4210 04757 R 652000 A LMQ /*12JAH14* STATCH IT AWAY
4211 04760 R 777700 A LAW -100 /*12JAH14* SAVE OLD 1ST &
4212 04761 R 520220 R AND* IPOINT /*12JAH14* 2ND CHARS
4213 04762 R 640002 A OMQ /*12JAH14* INSERT NEW CHAR
4214 04763 R 060220 R DAC* IPOINT /*12JAH14* RETURN TO TARGET WORD
4215 04764 R 624725 R JMP* ICHART /*12JAH14*
4216 /
4217 04765 R 604744 R XCI1 JMP IENT1 /*12JAH14* 2ND CHAR
4218 04766 R 604756 R XCI2 JMP IENT2 /*12JAH14* 3RD CHAR
4219 04767 R 604727 R XCI3 JMP IENTO /*12JAH14* 1ST CHAR
4220 /
4221 .EJECT

```

```

4222 /
4223 /XCHAR1 - INDEX AND LOAD A CHARACTER USING POINT1 AND PLACE1
4224 / (SIMILAR TO XCHAR1)
4225 /
4226 / CALL CHAR1
4227 / (RETURN IF CHAR SAME AS TEMP1)
4228 / (RETURN OTHERWISE)
4229 /
4230 04770 R 000000 A CHAR1 0 /*12JAH17*
4231 04771 R 440226 R ISZ PLACE1 /ADVANCE TO NEXT CHAR POS /*12JAH17*
4232 04772 R 220225 R LAC* POINT1 /*12JAH14* FETCH ALL 3 CHARS
4233 04773 R 420226 R XCT* PLACE1 /*12JAH14* SELECT CHAR POS & EXTRACT CHAR
4234 04774 R 605010 R JMP XCH1.2 /*12JAH14* CHECK FOR TEMP1
4235 04775 R 747420 A SZL!RTR!CLL /*12JAH14* MIDDLE CHAR?
4236 04776 R 605003 R JMP XCH1.1 /*12JAH14* NO, ADVANCE POINT1
4237 04777 R 742020 A RTR ; RTR /*12JAH14* YES
05000 R 742020 A
4238 05001 R 513702 R AND (77 /*12JAH14* REMOVE EXTRANEIOUS BITS
4239 05002 R 605010 R JMP XCH1.2 /*12JAH14* CHECK FOR TEMP1
4240 05003 R 440225 R XCH1.1 ISZ POINT1 /*12JAH14* ADVANCE TO NEW WORD
4241 05004 R 213732 R LAC (XCO1 /*12JAH14* POSITION TO
4242 05005 R 040226 R DAC PLACE1 /*12JAH14* FIRST CHAR
4243 05006 R 220225 R LAC* POINT1 /*12JAH14* FETCH NEW WORD
4244 05007 R 653606 A CLAC!LMQ!LLS+6 /*12JAH14* SHIFT IN LEFTMOST CHAR
4245 05010 R 540224 R XCH1.2 SAD TEMP1 /*12JAH14*
4246 05011 R 624770 R JMP* CHAR1 /*12JAH14* SAME AS TEMP1
4247 05012 R 444770 R ISZ CHAR1 /*12JAH14*
4248 05013 R 624770 R JMP* CHAR1 /*12JAH14* CHAR NOT THE SAME
4249 /
4250 .EJECT

```

```

4251 /
4252 /XCHAR2 - INDEX AND LOAD A CHARACTER USING POINT2 AND PLACE2
4253 / (SIMILAR TO XCHAR1)
4254 /
4255 / CALL CHAR2
4256 / (RETURN IF CHAR SAME AS TEMP2)
4257 / (RETURN IF CHAR OTHERWISE)
4258 /
4259 05014 R 000000 A CHAR2 0 /*12JAH17*
4260 05015 R 440231 R ISZ PLACE2 /ADVANCE TO NEXT CHAR POS /*12JAH17*
4261 05016 R 220230 R LAC* POINT2 /*12JAH14* FETCH ALL 3 CHARS
4262 05017 R 420231 R XCT* PLACE2 /*12JAH14* SELECT CHAR POS & EXTRACT CHAR
4263 05020 R 605034 R JMP XCH2.2 /*12JAH14* CHECK FOR TEMP2
4264 05021 R 747420 A SZL!RTR!CLL /*12JAH14* MIDDLE CHAR?
4265 05022 R 605027 R JMP XCH2.1 /*12JAH14* NO, ADVANCE POINT2
4266 05023 R 742020 A RTR ; RTR /*12JAH14* YES
4267 05024 R 742020 A
4267 05025 R 513702 R AND (77 /*12JAH14* REMOVE EXTRANEIOUS BITS
4268 05026 R 605034 R JMP XCH2.2 /*12JAH14* CHECK FOR TEMP2
4269 05027 R 440230 R XCH2.1 ISZ POINT2 /*12JAH14* ADVANCE TO NEW SORD
4270 05030 R 213732 R LAC (XC01 /*12JAH14* POSITION TO
4271 05031 R 040231 R DAC PLACE2 /*12JAH14* FIRST CHAR
4272 05032 R 220230 R LAC* POINT2 /*12JAH14* FETCH NEW DWRD
4273 05033 R 653606 A CLAC!LMQ!LLS+6 /*12JAH14* SIFT IN LEFTMOST CHAR
4274 05034 R 540227 R XCH2.2 SAD TEMP2 /*12JAH14*
4275 05035 R 625014 R JMP* CHAR2 /*12JAH14* SAME AS TEMP2
4276 05036 R 445014 R ISZ CHAR2 /*12JAH14*
4277 05037 R 625014 R JMP* CHAR2 /*12JAH14* CHAR NOT THE SAME
4278 /
4279 .EJECT

```

```

4280          /XSAVE - PUSH ITEM ON THE STACK
4281          /
4282          /CALLING SEQUENCE:::
4283          /
4284          /      LAW      -N      ///COUNT OF ITEMS TO GO ONTO STACK
4285          /      CALL    SAVE
4286          /      LAC     THIS
4287          /      LAC*   THAT
4288          /      ... OR ANY INSTRUCTION THAT PUTS A VALUE INTO THE ACCUM
4289          /      LAC     N-TH ITEM
4290          /      ...RETURNS TO THIS LOCATION...
4291          /
4292          05040 R 000000 A      SAVE      0                          /*12JAH17*
4293          05041 R 045103 R      DAC        UNSAVE          /SAVE ITEM COUNT      /*12JAH17*
4294          05042 R 777777 A      SBACK     LAW          -1
4295          05043 R 340166 R      TAD        SPOINT          /GET STACK POINTER AND
4296          05044 R 540170 R      SAD        SYMBEG          /CHECK FOR END OF STACK BUFFER.
4297          05045 R 605060 R      JMP        EXTST
4298          05046 R 425040 R      S2BACK    XCT*       SAVE          /GET ITEM FOR STACK.
4299          05047 R 445040 R      ISZ        SAVE          /BUMP FOR NEXT ITEM OR RETURN.
4300          05050 R 060166 R      DAC*       SPOINT          /SAVE ITEM ON STACK AND
4301          05051 R 440166 R      ISZ        SPOINT          /BUMP STACK PTR.
4302          05052 R 777775 A      LAW        -3            /3 LESS CHARACTERS
4303          05053 R 360146 R      TAD*       SVAR          /AVAILABLE FOR USER'S
4304          05054 R 060146 R      DAC*       SVAR          /PARTITION
4305          05055 R 445103 R      ISZ        UNSAVE          /BUMP ITEM COUNT.
4306          05056 R 605042 R      JMP        SBACK          /CONTINUE.
4307          05057 R 625040 R      JMP*       SAVE          /FINISHED, RETURN.
4308          05060 R 200165 R      EXTST     LAC        STACK
4309          05061 R 540167 R      SAD        END          /OUT OF SPACE?
4310          05062 R 000016 A      ERR16     /YES, "STACK" ERROR
4311          05063 R 046621 R      DAC        CONT          /NO
4312          05064 R 353741 R      TAD        (-1
4313          05065 R 040165 R      DAC        STACK          /NEW START OF STACK
4314          05066 R 040266 R      DAC        GASS
4315          05067 R 226621 R      REPTRA   LAC*       CONT          /TRANSFER STACK 1 LOCATION UP
4316          05070 R 060266 R      DAC*       GASS
4317          05071 R 446621 R      ISZ        CONT
4318          05072 R 440266 R      ISZ        GASS
4319          05073 R 777777 A      LAW        -1
4320          05074 R 346621 R      TAD        CONT
4321          05075 R 540170 R      SAD        SYMBEG          /END OF TRANSFER?
4322          05076 R 751001 A      SKP!CLA!CMA      /YES
4323          05077 R 605067 R      JMP        REPTRA      /NO
4324          05100 R 340166 R      TAD        SPOINT
4325          05101 R 040166 R      DAC        SPOINT          /NEW END OF STACK
4326          05102 R 605046 R      JMP        S2BACK
4327          /
4328          /
4329          /
4330          /
4331          /

```

```

4332          /XUNSAVE - POP ITEM FROM THE STACK
4333          /
4334          /CALLING SEQUENCE:
4335          /
4336          /      LAW      -N      ///COUNT OF ITEMS TO COME OFF STACK
4337          /      CALL     UNSAVE
4338          /      FOLLOWED BY ANY EXECUTABLE INSTRUCTION, INCLUDING NOP
4339          /      BUT GENERALLY AN INSTRUCTION TO MOVE VALUE FROM ACCUM TO MEMORY
4340          /
4341          05103 R 000000 A      UNSAVE 0                               /*12JAH17*
4342          05104 R 045040 R      DAC     SAVE      /SAVE ITEM COUNT      /*12JAH17*
4343          05105 R 777776 A      UBACK  LAW      -2
4344          05106 R 300166 R      ADD     SPOINT   /DECREMENT STACK POINTER.
4345          05107 R 540165 R      SAD     STACK    /CHECK FOR BEGINNING OF STACK BUFFER.
4346          05110 R 000016 A      ERR16          /"STACK" ERROR - STACK UNDERFLOW,
4347          /                               /WHICH IS A SYSTEM PROBLEM
4348          05111 R 040166 R      DAC     SPOINT
4349          05112 R 220146 R      LAC*   SVAR     /3 MORE CHARACTERS
4350          05113 R 723003 A      AAC     3       /AVAILABLE FOR USER'S
4351          05114 R 060146 R      DAC*   SVAR     /PARTITION
4352          05115 R 220166 R      LAC*   SPOINT   /GET ITEM FROM STACK AND
4353          05116 R 425103 R      XCT*   UNSAVE   /SAVE FOR CALLER
4354          05117 R 445103 R      ISZ    UNSAVE   /BUMP FOR NEXT ITEM OR RETURN
4355          05120 R 445040 R      ISZ    SAVE     /BUMP ITEM COUNT
4356          05121 R 605105 R      JMP    UBACK   /CONTINUE.
4357          05122 R 625103 R      JMP*   UNSAVE   /FINISHED, RETURN.
4358          .EJECT

```

```

4359          /XSTEP - FIND A STEP
4360          /
4361          05123 R 000000 A      STEP      0                               /*12JAH17*
4362          05124 R 200206 R      LAC      LOC      /PTR TO FIRST LOC OF      /*12JAH17*
4363          05125 R 040016 A      DAC      R16      /STEP BEING EXECUTED.
4364          05126 R 220145 R      LAC*     LVAR     /NO OF STEP BEING EXECUTED.
4365          05127 R 540207 R      SAD      VAL      /NO OF STEP TO BE FOUND.
4366          05130 R 625123 R      JMP*     STEP     /SAME, STEP FOUND, RETURN.
4367          05131 R 740001 A      CMA
4368          05132 R 300207 R      ADD      VAL      /VAL-LVAR
4369          05133 R 740100 A      SMA
4370          05134 R 605137 R      JMP      STEP1    /STEP TO BE FOUND ABOVE CURRENT STEP?
4371          05135 R 200134 R      LAC      BUFF     /BELOW, START SEARCH FROM CURRENT STEP.
4372          05136 R 040016 A      DAC      R16      /ABOVE, START SEARCH AT TOP.
4373          05137 R 220016 A      STEP1    LAC*     R16
4374          05140 R 653606 A      LMQ!CLAC!LLS+6 /GET STEP OFFSET AND
4375          05141 R 300016 A      ADD      R16      /ADD TO SEARCH PTR.
4376          05142 R 540167 R      SAD      END      /END OF STEP BUFFER?
4377          05143 R 000015 A      ERR15
4378          05144 R 040016 A      DAC      R16      /YES, "STEPS"
4379          05145 R 220016 A      LAC*     R16      /NO.
4380          05146 R 540207 R      SAD      VAL      /GET NO. OF NEXT STEP.
4381          05147 R 625123 R      JMP*     STEP     /IS THIS THE STEP WE'RE LOOKING FOR?
4382          05150 R 605137 R      JMP      STEP1    /YES, RETURN.
4383          .EJECT                               /NO, CONTINUE SEARCH.

```

```

4384          /XPART - FIND A PART
4385          /
4386          05151 R 000000 A      PART      0                               /*12JAH17*
4387          05152 R 220145 R      LAC*      LVAR          /NO OF STEP BEING EXECUTED /*12JAH17*
4388          05153 R 740001 A      CMA
4389          05154 R 300207 R      ADD        VAL          /VAL-LVAR
4390          05155 R 740010 A      RAL
4391          05156 R 200206 R      LAC        LOC          /POINTER TO CURRENT STEP
4392          05157 R 745400 A      SZL!CLL          /REF PART ABOVE OR BELOW?
4393          05160 R 200134 R      LAC        BUFF          /ABOVE, START AT TOP.
4394          05161 R 040016 A      DAC        R16
4395          05162 R 641002 A      LACQ
4396          05163 R 040207 R      DAC        VAL          /INTEGER PORTION (PART NO) VAL/100
4397          05164 R 220016 A      XPART1 LAC*      R16
4398          05165 R 653606 A      LMQ!CLAC!LLS+6      /GET STEP OFFSET AND
4399          05166 R 300016 A      ADD        R16          /ADD TO SEARCH PTR.
4400          05167 R 540167 R      SAD        END          /END OF STEP BUFFER?
4401          05170 R 000015 A      ERR15
4402          05171 R 040016 A      DAC        R16          /NO.
4403          05172 R 220016 A      LAC*      R16          /GET NO OF NEXT STEP.
4404          05173 R 653323 A      IDIV;      144
4405          05175 R 641002 A      LACQ
4406          05176 R 540207 R      SAD        VAL          /IS THIS THE PART WE'RE LOOKING FOR?
4407          05177 R 625151 R      JMP*      PART          /YES, RETURN.
4408          05200 R 605164 R      JMP        XPART1        /NO, CONTINUE SEARCH.
4409          .EJECT

```

```

4410          /XCHECK - HANDLE CHANGES IN THE STEP BUFFER DURING EXECUTION
4411          /
4412          05201 R 000000 A      CHECK    0                               /*12JAH17*
4413          05202 R 220145 R      LAC*    LVAR          /GET NO OF STEP BEING EXECUTED /*12JAH17*
4414          05203 R 741200 A      SNA      /DIRECT MODE CMMND?
4415          05204 R 605234 R      JMP     CHECK2        /YES.
4416          05205 R 777777 A      LAW     -1           /NO.
4417          05206 R 340206 R      TAD     LOC
4418          05207 R 040207 R      DAC     VAL
4419          05210 R 220207 R      LAC*    VAL
4420          05211 R 513702 R      AND     (77
4421          05212 R 744200 A      SZA!CLL          /DOES PREV STEP HAVE TRAILING EOM?
4422          05213 R 605217 R      JMP     CHECK1        /NO.
4423          05214 R 220206 R      LAC*    LOC          /DOES STEP BUFFER PTR
4424          05215 R 560145 R      SAD*    LVAR          /POINT TO STEP BEING EXECUTED?
4425          05216 R 605234 R      JMP     CHECK2        /YES.
4426          05217 R 220145 R      CHECK1  LAC*    LVAR          /GET NO OF CURRENT STEP.
4427          05220 R 040207 R      DAC     VAL          /HOLD FOR SEARCH.
4428          05221 R 460145 R      ISZ*    LVAR          /CHANGE TO AVOID IMMEDIATE EXIT FROM XSTEP.
4429          05222 R 105123 R      CALL    SIEP         /CALLED TO UPDATE PTR INTO STEP BUFFER.
4430          05223 R 060145 R      DAC*    LVAR          /RESET STEP NO.
4431          05224 R 200206 R      LAC     LOC
4432          05225 R 740001 A      CMA
4433          05226 R 300016 A      ADD     R16
4434          05227 R 300222 R      ADD     OPOINT        /OPOINT + OFFSET OF CHANGE =
4435          05230 R 040222 R      DAC     OPOINT        /NEW VALUE FOR OPOINT.
4436          05231 R 200016 A      LAC     R16
4437          05232 R 040206 R      DAC     LOC          /NEW VALUE FOR PTR INTO STEP BUFFER.
4438          05233 R 625201 R      JMP*    CHECK        /PTRS UPDATED, RETURN.
4439          05234 R 200206 R      CHECK2  LAC     LOC
4440          05235 R 740001 A      CMA
4441          05236 R 300167 R      ADD     END          /DOES STEP BUFFER PTR
4442          05237 R 745100 A      SPA!CLL          /POINT PAST END OF STEP BUFFER?
4443          05240 R 605217 R      JMP     CHECK1        /YES, GO UPDATE PTRS.
4444          05241 R 625201 R      JMP*    CHECK        /NO, RETURN.
4445          .EJECT

```

```

4446          /XGETKEY - EVALUATES KEY ENCLOSED BY SQUARE BRACKETS
4447          /
4448          /CALLING SEQUENCE:
4449          /
4450          /      CALL      GETKEY
4451          /      RETURNS HERE ON SYNTAX ERROR
4452          /      RETURNS HER IF SYNTAX OK
4453          /      (NEXT CHAR IN AC, AND AVAILABLE VIA OCHAR)
4454          /RETURNS WITH USER SUPPLIED KEY IN VAL
4455          /
4456          05242 R 000000 A  GETKEY  0                                /*12JAH17*
4457          /##### SAVE ON STACK #####
4458          05243 R 777772 A          LAW      -6                                /*12JAH17*
4459          05244 R 105040 R          CALL     SAVE
4460          05245 R 207163 R          LAC      EVAL
4461          05246 R 200242 R          LAC      BKCHAR
4462          05247 R 200216 R          LAC      TERM
4463          05250 R 200215 R          LAC      SUM
4464          05251 R 200202 R          LAC      TEMP
4465          05252 R 200214 R          LAC      M
4466          /#####          #####
4467          05253 R 744000 A          CLL              /NUMERIC RESULT ONLY
4468          05254 R 107163 R          CALL     EVAL
4469          /##### POPSTACK #####
4470          05255 R 777772 A          LAW      -6
4471          05256 R 105103 R          CALL     UNSAVE
4472          05257 R 040214 R          DAC      M
4473          05260 R 040202 R          DAC      TEMP
4474          05261 R 040215 R          DAC      SUM
4475          05262 R 040216 R          DAC      TERM
4476          05263 R 040242 R          DAC      BKCHAR
4477          05264 R 047163 R          DAC      EVAL
4478          /#####          #####
4479          05265 R 104645 R          CALL     OCHAR
4480          05266 R 553751 R          SAD      (35          /RIGHT BRACKET
4481          05267 R 445242 R          ISZ     GETKEY          /SYNTAX OK
4482          05270 R 104660 R          CALL     OCHARI          /SET UP FOR FUTURE "CALL OCHAR"
4483          05271 R 625242 R          JMP*    GETKEY
4484          .EJECT

```

```

4485          /XKEYCHK - VALIDATES USER SUPPLIED KEY
4486          /
4487          /CALLING SEQUENCE:
4488          /          CALL    KEYCHK
4489          /          LAC(*)  REQUIRED KEY
4490          /          SAD(*)  USER SUPPLIED KEY
4491          /
4492          /RETURNS WITH ACCUM=0 IF KEYS MATCH
4493          /          AND WITH ACCUM>0 IN KEYS DO NOT MATCH
4494          /          LINK SET IF TRAP REQUIRED
4495          /
4496          05272 R 000000 A    KEYCHK  0                                /*12JAH17*
4497          05273 R 425272 R    XCT*   KEYCHK          /FETCH REQUIRED KEY          /*12JAH17*
4498          05274 R 445272 R    ISZ    KEYCHK
4499          05275 R 741200 A    SNA
4500          05276 R 605302 R    JMP    KEYCH2        /REQUIRED KEY ZERO, PASS ANY USER KEY
4501          05277 R 513735 R    AND    (377777       /GET POSITIVE KEY
4502          05300 R 425272 R    XCT*   KEYCHK        /EXECUTE THE SAD COMPARE
4503          05301 R 750000 A    CLA
4504          05302 R 445272 R    KEYCH2 ISZ    KEYCHK        /HOP OVER THE SAD
4505          05303 R 625272 R    JMP*   KEYCHK
4506          /
4507          /IF CALLER DECIDES NON-MATCH IS ERROR, COMES HERE...
4508          /
4509          .EJECT

```

```

4510          /XASSEM - ASSEMBLE A VARIABLE NAME
4511          /
4512          05304 R 000000 A      ASSEM  0                               /*12JAH17*
4513          05305 R 777775 A      LAW    -3                               /*12JAH17*
4514          05306 R 040214 R      DAC    M      /SET TO GET UP TO 3 CHARS
4515          05307 R 104660 R      CALL   OCHARI  /GET FIRST CHAR AND
4516          05310 R 103724 R      CALL   SYNTAX  /CHECK CHAR TYPE.
4517          05311 R 605326 R      JMP    ASSEM5  /A
4518          05312 R 000003 A      ERR3   /N- "SYNTAX"
4519          05313 R 553752 R      SAD    (45    /P - IS IT A  %
4520          05314 R 605361 R      JMP    ASSEM4  /YES.
4521          05315 R 000003 A      ERR3   /NO, "SYNTAX"
4522          /
4523          05316 R 440214 R      ASSEM1 ISZ    M      /THREE 5-BIT CHARS PACKED YET?
4524          05317 R 605323 R      JMP    ASEM15  /*12JAH12* NO, PACK THIS ONE
4525          05320 R 777777 A      LAW    -1      /YES, IGNORE THIS AND
4526          05321 R 040214 R      DAC    M      /ALL OTHER ALPHAS UP TO PUNCT.
4527          05322 R 605327 R      JMP    ASSEM2
4528          05323 R 640505 A      ASEM15 LRS    5      /*12JAH12*
4529          05324 R 200204 R      LAC    SYMS
4530          05325 R 640605 A      LLS    5
4531          05326 R 040204 R      ASSEM5 DAC    SYMS  /SAVE ACCUM. CHARS.
4532          05327 R 104660 R      ASSEM2 CALL   OCHARI  /GET NEXT CHAR AND
4533          05330 R 103724 R      CALL   SYNTAX  /CHECK TYPE.
4534          05331 R 605316 R      JMP    ASSEM1  /A
4535          05332 R 000003 A      ERR3   /N- "SYNTAX"
4536          05333 R 553753 R      SAD    (50    /P - IS IT A  (
4537          05334 R 605337 R      JMP    .+3    /YES, GO EVAL SUBSCR.
4538          05335 R 140205 R      DZM    SCRIPS  /NO.
4539          05336 R 625304 R      JMP*   ASSEM  /RETURN.
4540          .IFPNZ FMONEY
4541          /##### SAVE ON STACK #####
4542          05337 R 777777 A      LAW    -1
4543          05340 R 105040 R      CALL   SAVE
4544          05341 R 200211 R      LAC    VALF
4545          .ENDC
4546          05342 R 744000 A      CLL
4547          05343 R 107163 R      CALL   EVAL    /L CLEAR FOR NUM RESULT ONLY.
4548          05344 R 040205 R      DAC    SCRIPS  /SAVE VALUE OF SUBSCR.
4549          05345 R 513661 R      AND    (700000
4550          05346 R 740200 A      SZA
4551          05347 R 000005 A      ERR5   /IS IT POS?
4552          .IFPNZ FMONEY
4553          /##### POP STACK #####
4554          05350 R 777777 A      LAW    -1
4555          05351 R 105103 R      CALL   UNSAVE
4556          05352 R 040211 R      DAC    VALF
4557          .ENDC
4558          05353 R 104645 R      CALL   OCHAR  /GET CHAR AFTER SUBSCR.
4559          05354 R 553754 R      SAD    (51    /IS IT A  )
4560          05355 R 741000 A      SKP
4561          05356 R 000003 A      ERR3   /YES.
4561          05356 R 000003 A      ERR3   /NO, "SYNTAX"

```

4562	05357 R 104660 R		CALL	OCHARI	/INDEX TO GET NEXT CHAR AND
4563	05360 R 625304 R		JMP*	ASSEM	/RETURN.
4564		/			
4565	05361 R 213751 R	ASSEM4	LAC	(35	/CHANGE % TO] AND
4566	05362 R 605326 R		JMP	ASSEM5	/CONTINUE.
4567			.EJECT		

```

4568          /XSVE - HANDLE A STRING VALUED EXPRESSION
4569          /
4570          05363 R 000000 A      SVE      0                                /*12JAH17*
4571          05364 R 104660 R      CALL     OCHARI          /GET NEXT CHAR          /*12JAH17*
4572          05365 R 553755 R      SAD      (42             /IS IT A "
4573          05366 R 605421 R      JMP      SVE1           /YES, GO PROC. TEXT LITERAL.
4574          05367 R 104710 R      CALL     SUBOPT         /NO, BACKUP OPOINT.
4575          /##### SAVE ON STACK #####
4576          05370 R 777776 A      LAW      -2
4577          05371 R 105040 R      CALL     SAVE
4578          05372 R 200204 R      LAC      SYMS
4579          05373 R 200205 R      LAC      SCRIPS
4580          /#####
4581          05374 R 105304 R      CALL     ASSEM          /ASSEMBLE THE VARIABLE NAME.
4582          05375 R 200204 R      LAC      SYMS
4583          05376 R 040177 R      DAC      SYM             /SAVE NAME.
4584          05377 R 200205 R      LAC      SCRIPS
4585          05400 R 040200 R      DAC      SCRIP          /SAVE SUBSCR.
4586          05401 R 754000 A      CLL!CLA
4587          05402 R 105434 R      CALL     SYMBOL         /SEARCH SYMBOL TABLE.
4588          05403 R 641002 A      LACQ
4589          05404 R 741200 A      SNA
4590          05405 R 000010 A      ERR10
4591          05406 R 744400 A      SNL!CLL
4592          05407 R 000011 A      ERR11
4593          /##### POPSTACK #####
4594          05410 R 777776 A      LAW      -2            / STRING
4595          05411 R 105103 R      CALL     UNSAVE
4596          05412 R 040205 R      DAC      SCRIPS
4597          05413 R 040204 R      DAC      SYMS
4598          /#####
4599          05414 R 213732 R      LAC      (XC01          /*12JAH14* POS
4600          05415 R 040016 A      DAC      R16           /*12JAH14* CHAR
4601          05416 R 750000 A      CLA
4602          05417 R 440017 A      ISZ      R17           /POINTS TO STR.
4603          05420 R 625363 R      JMP*     SVE           /RETURN.
4604          /
4605          /TEXT LITERAL
4606          /
4607          05421 R 200222 R      SVE1     LAC      OPOINT
4608          05422 R 040017 A      DAC      R17           /POINTS TO TEXT LIT.
4609          05423 R 200223 R      LAC      OPLACE        /*12JAH14* SAVE
4610          05424 R 040016 A      DAC      R16           /*12JAH14* CHAR POS
4611          05425 R 104660 R      CALL     OCHARI
4612          05426 R 553755 R      SAD      (42             /IGNORE CHARS UP TO NEXT "
4613          05427 R 741000 A      SKP
4614          05430 R 605425 R      JMP      .-3
4615          05431 R 104660 R      CALL     OCHART        /INDEX TO CHAR AFTER "
4616          05432 R 213755 R      LAC      (42             / " IN AC FOR
4617          05433 R 625363 R      JMP*     SVE           /RETURN.
4618          .EJECT

```

```

4619          /SYMBOL STUFF: GET, PUT, KILL
4620          / IN SYMBOL TABLE:
4621          /   00X UNDEFINED OR FLOATING POINT
4622          /   01X ARRAY   (NEG SUBSCR IF STR)
4623          /   10X NUMERIC VARIABLE
4624          /   11X STRING VARIABLE
4625          / SYMBOL:
4626          /   0X0 GET
4627          /   1X0 PUT
4628          /   0X1 KILL
4629          / CALLS:
4630          /   GET  CLL!CLA
4631          /         CALL  SYMBOL
4632          /   PUT  STL!CLA
4633          /         CALL  SYMBOL
4634          /   KILL STL
4635          /         LAC  E00000
4636          /         CALL  SYMBOL
4637          / RETURN:
4638          /   GOOD SYMBOL IN MQ
4639          /         LAC*  R17 GETS VALUE OR STRING
4640          /         IF "GOT STRING", LINK=1.  ELSE LINK=0
4641          /   BAD  MQ AND LINK CLEAR
4642          /
4643          /
4644          05434 R 000000 A  SYMBOL 0                                     /*12JAH17*
4645          .IFPNZ  MONEY
4646          05435 R 400210 R  XCT  MONYSW                               /IGNORE DZM SWAPSW IF ALREADY IN SM
4647          05436 R 741000 A  SKP
4648          .ENDC
4649          05437 R 140652 R  DZM  SWAPSW                               /DON'T ALLOW USER TO BREAK
4650          05440 R 245434 R  XOR  SYMBOL
4651          .IFPNZ  XVM                                               /*12JAH15*
4652          05441 R 741400 A  SZL                                       /LINK SET?                               /*12JAH15*
4653          05442 R 253662 R  XOR  (400000                             /YES, SET BIT IS NECC IN USER MODE       /*12JAH15*
4654          .ENDC                                               /*12JAH15*
4655          05443 R 045434 R  DAC  SYMBOL
4656          05444 R 200177 R  LAC  SYM
4657          05445 R 060153 R  DAC* SYMEND
4658          05446 R 660703 A  ALSS  3
4659          05447 R 045040 R  DAC  SAVE
4660          05450 R 200170 R  LAC  SYMBEG
4661          05451 R 040017 A  BACK DAC  R17                               / LOOK FOR SYMBOL
4662          05452 R 220017 A  BACK1 LAC* R17
4663          05453 R 652703 A  LMQ!ALS 3
4664          05454 R 545040 R  SAD  SAVE
4665          05455 R 605501 R  JMP  EQUAL1                               / FOUND SYMBOL
4666          05456 R 641002 A  LACQ
4667          05457 R 740110 A  SMA!RAL
4668          05460 R 605465 R  JMP  OVER1                               /UNDEF OR ARRAY OR FLJATING POINT
4669          05461 R 745100 A  SPA!CLL                               /NUM OR STR?
4670          05462 R 605472 R  JMP  OVER2                               /STR

```

```

4671      05463 R 440017 A      ISZ      R17      /NUM, POINT AROUND IT.
4672      05464 R 605452 R      JMP      BACK1     /GO CHECK NEXT SYMBOL.
4673
4674      005465 R      /
4675      OVER1=.
4676      05465 R 740100 A      .IFPNZ  FMONEY
4677      05466 R 605476 R      SMA      /F.P. OR ARRAY?
4678      .ENDC
4679      05467 R 220017 A      JMP      OVER3     /F.P.
4680      05470 R 300017 A      LAC*     R17      /GET ARRAY OFFSET
4681      05471 R 605451 R      ADD      R17      /TO POINT AROUND ARRAY.
4682      05472 R 220017 A      OVER2   LAC*     R17      /GO CHECK NEXT SYMBOL.
4683      05473 R 653606 A      LMQ!CLAC!LLS+6 /GET STR OFFSET
4684      05474 R 300017 A      ADD      R17      /TO POINT AROUND STR.
4685      05475 R 605451 R      JMP      BACK     /GO CHECK NEXT SYMBOL.
4686      .IFPNZ  FMONEY
4687      05476 R 200017 A      OVER3   LAC      R17
4688      05477 R 723004 A      AAC      4        /FOUR WORDS IN F.P. VAR
4689      05500 R 605451 R      JMP      BACK
4690      .ENDC
4691      /
4692      05501 R 641002 A      EQUAL1 LACQ
4693      05502 R 040016 A      DAC      R16
4694      05503 R 200017 A      LAC      R17
4695      05504 R 540153 R      SAD      SYMEND   /END OF SYMBOL TABLE?
4696      05505 R 605547 R      JMP      NOSYM   /YES, SYMBOL NOT DEFINED
4697      05506 R 200200 R      LAC      SCRIP   /NO.
4698      05507 R 741200 A      SNA
4699      05510 R 605707 R      JMP      FNDSYM  /DOES SYMBOL HAVE SUBSCRIPTS?
4700      05511 R 641002 A      LACQ
4701      05512 R 741110 A      SPA!RAL /NO, FOUND VARIABLE
4702      05513 R 000017 A      ERR17   /*12JAH6*
4703
4704      05514 R 740100 A      SMA      /LOOKING FOR ARRAY, BUT FOUND VARIABLE
4705      05515 R 000017 A      ERR17   /- "SYMBO"
4706      05516 R 220017 A      LAC*     R17      /F.P. OR UNDEFINED? *12JAH6*
4707      05517 R 300017 A      ADD      R17      /YES, "SYMBO" *12JAH6*
4708      05520 R 045103 R      DAC      UNSAVE  / POINTER TO NEXT SYMBOL
4709      05521 R 200017 A      LAC      R17
4710      05522 R 045040 R      DAC      SAVE    / POINTER TO ARRAY'S OFFSET
4711      /
4712      /LOOKING FOR AND FOUND ARRAY
4713      /
4714      05523 R 220017 A      ARRAY1  LAC*     R17      / GET SUBSCRIPT
4715      05524 R 745100 A      SPA!CLL /STR OR NUM ELEMENT?
4716      05525 R 605535 R      JMP      ARRAY2  /STR
4717      05526 R 540200 R      SAD      SCRIP   /NUM, IS IT ELEM WE'RE LOOKING FOR?
4718      05527 R 606076 R      JMP      FNDELE  /YES, ARRAY ELEMENT FOUND (NUMERIC, LINK=0)
4719      05530 R 440017 A      ISZ      R17      /UPDATE SEARCH PTR.
4720      05531 R 200017 A      LAC      R17
4721      05532 R 545103 R      SAD      UNSAVE  /END OF ARRAY REACHED?
4722      05533 R 606022 R      JMP      NOARR   /YES, ARRAY ELEMENT NOT FOUND

```

```

4723      05534 R 605523 R          JMP      ARRAY1          /NO, CONTINUE.
4724
4725      /
4726      /ELEMENT FOUND IS STR ELEMENT
4727      /
4727      05535 R 744003 A      ARRAY2  CMA!STL
4728      05536 R 540200 R          SAD      SCRIP          /IS IT ELEM WE'RE LOOKING FOR?
4729      05537 R 606076 R          JMP      FNDELE        /YES, ARRAY ELEMENT FOUND (STRING, LINK=1)
4730      05540 R 220017 A          LAC*    R17           /NO, GET STRING'S OFFSET AND
4731      05541 R 653606 A          LMQ!CLAC!LLS+6      /ADD TO SEARCH PTR
4732      05542 R 300017 A          ADD      R17           /TO GET NEXT ELEMENT.
4733      05543 R 545103 R          SAD      UNSAVE        /END OF ARRAY REACHED?
4734      05544 R 606022 R          JMP      NOARR         /YES, ARRAY ELEMENT NOT FOUND.
4735      05545 R 040017 A          DAC      R17           /NO, CONTINUE.
4736      05546 R 605523 R          JMP      ARRAY1
4737
4738      /
4739      /SYMBOL NOT DEFINED
4740
4740      05547 R 205434 R      NOSYM   LAC      SYMBOL
4741      05550 R 650000 A          CLQ
4742      05551 R 744100 A          SMA!CLL          / PUT?
4743      05552 R 606166 R          JMP      SYMEX        / NO; UNDEFINED, EXIT WITH MQ AND LINK =0
4744      05553 R 200200 R          LAC      SCRIP        / YES
4745      05554 R 750230 A          SZA!CLA!IAC      /ARRAY ELEMENT?
4746      05555 R 605621 R          JMP      APPARR      /YES, GO PUT NEW ARRAY ELEMENT
4747      .IFPNZ  FMONEY
4748      05556 R 540211 R          SAD      VALF
4749      05557 R 605641 R          JMP      APPFP       /FLOATING POINT
4750      .ENDC
4751      05560 R 777777 A          LAW      -1
4752      05561 R 540207 R          SAD      VAL          /IS VARIABLE NUM OR STR?
4753      05562 R 605575 R          JMP      APPSTR      /STR
4754
4755      /
4756      /PUT NEW NUMERIC VARIABLE
4757
4757      05563 R 777775 A      APPNUM  LAW      -3          /BACK UP TO
4758      05564 R 300170 R          ADD      SYMBEG      /NEW VALUE FOR START
4759      05565 R 040017 A          DAC      R17         /OF SYMBOL TABLE.
4760      05566 R 104034 R          CALL    STOR2        /CHECK FOR SYMBOL TABLE OVERFLOW.
4761      05567 R 200177 R          LAC      SYM
4762      05570 R 253662 R          XOR      (400000    /INDICATES NUM VARIABLE.
4763      05571 R 060017 A      DEFNUM  DAC*    R17        /STORE SYM NAME (OR SUBSCR IF HERE FROM NUMDEF)
4764      05572 R 200207 R          LAC      VAL          /AND VALUE.
4765      05573 R 060017 A          DAC*    R17
4766      05574 R 606166 R          JMP      SYMEX        / EXIT FROM PUT NEW NUMERIC VARIABLE
4767
4768      /
4769      /PUT NEW STR VARIABLE
4770
4770      05575 R 777776 A      APPSTR  LAW      -2
4771      05576 R 300133 R          ADD      BUFF2
4772      05577 R 040015 A          DAC      R15         /PTR TO STR AC
4773      05600 R 220133 R          LAC*    BUFF2
4774      05601 R 653606 A          LMQ!CLAC!LLS+6

```

```

4775      05602 R 040220 R      DAC      IPOINT      /STR OFFSET
4776      05603 R 740030 A      IAC
4777      05604 R 740001 A      CMA
4778      05605 R 044103 R      DAC      DECIN      /COUNT FOR STORING SIR
4779      05606 R 723777 A      AAC      -1      /SETUP NEW VALUE
4780      05607 R 300170 R      ADD      SYMBEG      /FOR START OF
4781      05610 R 040017 A      DAC      R17      /SYMBOL TABLE.
4782      05611 R 104034 R      CALL     STOR2      /CHECK FOR SYM TBL OVERFLOW.
4783      05612 R 200177 R      LAC      SYM
4784      05613 R 253650 R      XOR      (600000      /INDICATES STR VARIABLE.
4795      05614 R 060017 A      DEFSIR  DAC*      R17
4786      05615 R 220015 A      LAC*     R15      /GET ANOTHER WD OF SIR.
4787      05616 R 444103 R      ISZ      DECIN      /FINISHED?
4788      05617 R 605614 R      JMP      .-3      /NO, CONTINUE STORING.
4789      05620 R 606166 R      JMP      SYMEX      /YES, EXIT FROM PUT NEW STRING VARIABLE
4790
4791      /
4792      /PUT NEW ARRAY ELEMENT
4793      /
4793      005621 R      APPARR=.
4794      .IFPNZ  FMONEY
4795      05621 R 540211 R      SAD      VALF
4796      05622 R 000017 A      ERR17
4797      .ENDC
4798      05623 R 777777 A      LAW      -1
4799      05624 R 540207 R      SAD      VAL
4800      05625 R 605661 R      JMP      ARRSTR      /NEW ELEMENT IS A STRING
4801      05626 R 777773 A      LAW      -5      / NEW ELEMENT IS NUMERIC.
4802      05627 R 300170 R      ADD      SYMBEG
4803      05630 R 040017 A      DAC      R17
4804      05631 R 104034 R      CALL     STOR2      /CHECK FOR SYM TBL OVERFLOW.
4805      05632 R 200177 R      LAC      SYM
4806      05633 R 253663 R      XOR      (200000
4807      05634 R 060017 A      DAC*     R17      / ENTER SYMBOL IN SYMBOL TABLE
4808      05635 R 213705 R      LAC      (2
4809      05636 R 060017 A      DAC*     R17      / ENTER NEW ARRAY'S OFFSET
4810      05637 R 200200 R      NUMDEF  LAC      SCRIP
4811      05640 R 605571 R      JMP      DEFNUM      /ENTER NEW SUBSCR VALUE.
4812
4813      /
4813      .IFPNZ  FMONEY
4814
4815      /PUT NEW FLOATING POINT VARIABLE
4816
4817      APPFP  LAW      -6      /COUNT FOR STORING F.P.
4818      05642 R 300170 R      ADD      SYMBEG      /SET UP NEW VALUE FOR START
4819      05643 R 040017 A      DAC      R17      /OF SYMBOL TABLE
4820      05644 R 104034 R      CALL     STOR2      /CHECK IF OUT OF SPACE
4821      05645 R 200177 R      LAC      SYM      /00 IN BITS 0,1 INDICATE F.P.
4822      05646 R 060017 A      DAC*     R17
4823      05647 R 777776 A      LAW      -2
4824      05650 R 300133 R      ADD      BUFF2
4825      05651 R 040015 A      DAC      R15
4826      05652 R 777774 A      LAW      -4

```

```

4827      05653 R 044103 R      DAC      DECIN
4828      05654 R 220015 A      LAC*     R15
4829      05655 R 060017 A      DAC*     R17
4830      05656 R 444103 R      ISZ      DECIN
4831      05657 R 605654 R      JMP      .-3
4832      05660 R 606166 R      JMP      SYMEX
4833
4834
4835      05661 R 777776 A      /
4836      05662 R 300133 R      ARRSTR  LAW      -2
4837      05663 R 040015 A      ADD      BUFF2
4838      05664 R 220133 R      DAC      R15      / POINTER TO BEGINNING OF STRING AC
4839      05665 R 653606 A      LAC*     BUFF2
4840      05666 R 040220 R      LMO!CLAC!LLS+6
4841      05667 R 740030 A      DAC      IPOINT  / OFFSET
4842      05670 R 740001 A      IAC
4843      05671 R 044103 R      CMA
4844      05672 R 723775 A      DAC      DECIN  /COUNT FOR STORING STR.
4845      05673 R 300170 R      AAC      -3
4846      05674 R 040017 A      ADD      SYMBEG
4847      05675 R 104034 R      DAC      R17
4848      05676 R 200177 R      CALL    STOR2   /CHECK FOR SYM TBL OVERFLOW.
4849      05677 R 253663 R      LAC      SYM
4850      05700 R 060017 A      XOR      (200000 /INDICATES ARRAY ELEMENT.
4851      05701 R 200220 R      DAC*     R17   /STORE SYMBOL NAME
4852      05702 R 313705 R      LAC      IPOINT
4853      05703 R 060017 A      ADD      (2
4854      05704 R 200200 R      DAC*     R17   / ENTER NEW ARRAY'S OFFSET
4855      05705 R 740001 A      STRDEF  LAC      SCRIP
4856      05706 R 605614 R      CMA
4857
4858      /
4859      /FOUND A SYMBOL
4860      05707 R 641002 A      /
4861      05710 R 742110 A      FNDSYM  LACQ     /GET SYM TBL ENTRY FOR FOUND SYM
4862      05711 R 605740 R      SMA!RTL
4863      05712 R 205434 R      JMP      FNDARR  / IT'S AN ARRAY (CAN ONLY BE KILLED)
4864      05713 R 513756 R      LAC      SYMBOL  / LOOKING FOR AND FOUND VARIABLE
4865      05714 R 741200 A      AND      (500000
4866      05715 R 606166 R      SNA
4867      05716 R 740400 A      JMP      SYMEX   / GOT VAR; EXIT, LK=1 IF STR, 0 IF NUM
4868      05717 R 606152 R      SNL
4869      05720 R 220017 A      JMP      NUMVAR  /PUT OR KILL: IS IT STR OR NUM?
4870      05721 R 653606 A      LAC*     R17   / FOUND NUM VAR
4871      05722 R 605747 R      LMO!CLAC!LLS+6 / FOUND STR VAR; KILL IT, CHECK FOR PUT OR KILL
4872
4873
4874      05723 R 205434 R      /
4875      05724 R 513756 R      FLOAT1 .IFPNZ  FMONEY
4876      05725 R 740200 A      LAC      SYMBOL  /FLOATING POINT
4877      05726 R 605735 R      AND      (500000
4878      05727 R 200211 R      SZA
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000

```

```

4879      05730 R 741200 A          SNA                /LEGAL GET?
4880      05731 R 000017 A          ERR17              /NO
4881      05732 R 754003 A          CLC!STL          /SIMULATE STRING RETURN FOR XSVE'S BENEFIT
4882      05733 R 140211 R          DZM             VALF          /FLAG TO INDICATE F.P. VALUE RETURNED
4883      05734 R 606166 R          JMP             SYMEX
4884      05735 R 440017 A          FLOAT2 ISZ      R17          /GO KILL IT
4885      05736 R 213757 R          LAC             (5
4886      05737 R 605750 R          JMP             DELET2
4887
4888
4889
4890      005740 R          /
4891      05740 R 740400 A          FNDARR=.      .IFPNZ  FMONEY
4892      05741 R 605723 R          SNL                /FLOATING POINT OR ARRAY?
4893      .ENDC
4894      05742 R 205434 R          JMP             FLOAT1     /FLOATING POINT
4895      05743 R 513664 R          .ENDC
4896      05744 R 745200 A          LAC             SYMBOL     /ENTIRE ARRAY TO BE KILLED
4897      05745 R 000017 A          AND             (100000
4898      05746 R 220017 A          SNA!CLL          / KILL?
4899      05747 R 313705 R          ERR17           / NO- "SYMB0"
4900      05750 R 044103 R          DELETE LAC*      R17      / YES; ARRAY'S OFFSET
4901      05751 R 200016 A          DELET1 ADD       (2       / ENTER HERE TO DELETE STRING VARIABLE
4902      05752 R 513664 R          DELET2 DAC       DECIN    / ENTER HERE TO DELETE NUMERIC OR FLOATING POINT
4903      05753 R 740200 A          /              / (VARIABLES AND ELEMENTS)
4904      05754 R 000017 A          DELET3 LAC       R16
4905      05755 R 777775 A          AND             (100000
4906      05756 R 300017 A          SZA
4907
4908      ERR17          / TRIED TO KILL A SPECIAL VARIABLE- "SYMB0"
4909      LAW           -3       / ENTER HERE TO DELETE STRING ELEMENT
4910      ADD          R17
4911
4912      /
4913      /MOVE EVERYTHING IN SYMBOL TABLE DOWN
4914      /((OVERLAY AREA OCCUPIED BY DELETED ENTRY)
4915
4916      /
4917      DELMOV  SAD      SYMBEG
4918      JMP      DELEND  /FINISHED.
4919      DAC      UNSAVE  /FETCH PTR
4920      ADD      DECIN   /PLUS OFFSET
4921      DAC      SAVE    /EQUALS "STORE" PTR.
4922      LAC*     UNSAVE  /FETCH WORD AND
4923      DAC*     SAVE    /STORE IT.
4924      LAW      -2
4925      ADD      UNSAVE  /DECREMENT "FETCH" PTR.
4926      JMP      DELMOV
4927
4928      /
4929      DELEND  ADD      DECIN  /NEW VALUE FOR SYMBEG.
4930      CALL    STOR2   /UPDATE SYMBEG & SVAR.
4931      LAC     SYMBOL
4932      SMA
4933      JMP     SYMEX   /WAS CALL FOR PUT OR KILL?
4934      LAC     R16
4935      .IFPNZ  FMONEY
4936      SPA!RAL /DATA TYPE?

```

```

4931      06000 R 750000 A      CLA          /STRING OR NUMBER
4932      06001 R 751130 A      SPA!CLA!IAC
4933      .ENDC
4934      .IFNOZ  FMONEY
4935      SMA
4936      .ENDC
4937      06002 R 606011 R      JMP      CHKARR          /PUT NEW ARRAY ELEM.
4938      .IFPNZ  FMONEY
4939      06003 R 540211 R      SAD      VALF          /FLOATING PT.?
4940      06004 R 605641 R      JMP      APPFP          /YES
4941      .ENDC
4942      06005 R 777777 A      LAW      -1          /NO
4943      06006 R 540207 R      SAD      VAL
4944      06007 R 605575 R      JMP      APPSTR          / PUT NEW STR VAR
4945      06010 R 605563 R      JMP      APPNUM          / PUT NEW NUM VAR
4946
4947      006011 R      /
4948      CHKARR=.
4949      .IFPNZ  FMONEY
4950      06011 R 540211 R      SAD      VALF          /FLOATING POINT?
4951      06012 R 000017 A      ERR17          /ERROR, F.P. CANNOT GO INTO ARRAY ELEMENT
4952      .ENDC
4953      06013 R 207066 R      LAC      WRSTEP          /PTR TO ARRAY'S OFFSET BEFORE MOVE,
4954      06014 R 304103 R      ADD      DECIN          /PLUS OFFSET OF DELETED ELEM,
4955      06015 R 045040 R      DAC      SAVE          /POINTS TO PRESENT LOC OF ARRAYS OFFSET.
4956      06016 R 750001 A      CLC
4957      06017 R 540207 R      SAD      VAL
4958      06020 R 606046 R      JMP      INSSTR          / PUT NEW STR ELE
4959      06021 R 606030 R      JMP      INSNUM          / PUT NEW NUM ELE
4960
4961      06022 R 205434 R      /
4962      06023 R 650000 A      NOARR  LAC      SYMBOL          / LOOKING FOR AND FOUND ARRAY, BUT DIDN'T
4963      06024 R 754101 A      CLQ          / FIND SPECIFIED ELEMENT
4964      06025 R 606166 R      SMA!CLC!CLL          / PUT?
4965      06026 R 540207 R      JMP      SYMEX          / NO, EXIT, UNDEF ARRAY ELEM ON GET OR KILL
4966      06027 R 606046 R      SAD      VAL          / YES, STR OR NUM?
4967      JMP      INSSTR          /STR
4968
4969      /
4970      /PUT NEW NUMERIC ARRAY ELEMENT
4971      /
4972      INSNUM  LAC      SYMBEG          / PUT NEW NUMERIC ARRAY ELEMENT
4973      DAC      R14
4974      AAC      -2
4975      DAC      R17          /NEW VALUE FOR SYMBEG
4976      CALL    STOR2          /UPDATE SYMBEG & CHK TBL OVERFLOW.
4977      LAC*    SAVE
4978      ADD     (2          /GOT TO BE ONE'S COMPLEMENT!!!
4979      DAC*    SAVE          /INCREASE ARRAY'S OFFSET
4980      LAC*    R14          /MAKE ROOM FOR NEW NUM ELEMENT.
4981      DAC*    R17
4982      LAC     R14
4983      SAD     SAVE          /FINISHED?
4984      JMP     NUMDEF          /YES, GO INSERT NEW NUM ELEM.
4985      JMP     .-5          /NO, CONTINUE.

```

```

4983 /
4984 /PUT NEW STR ARRAY ELEMENT
4985 /
4986 06046 R 777776 A INSSIR LAC -2
4987 06047 R 300133 R ADD BUFF2 /POINTS TO BEGINNING OF STR AC
4988 06050 R 040015 A DAC R15
4989 06051 R 220133 R LAC* BUFF2
4990 06052 R 653606 A LMQ!CLAC!LLS+6
4991 06053 R 723002 A AAC 2
4992 06054 R 040220 R DAC IPOINT /STR OFFSET
4993 06055 R 740001 A CMA
4994 06056 R 044103 R DAC DECIN /COUNT FOR STORING STR.
4995 06057 R 200170 R LAC SYMBEG
4996 06060 R 040014 A DAC R14
4997 06061 R 304103 R ADD DECIN
4998 06062 R 040017 A DAC R17 /NEW VALUE FOR SYMBEG.
4999 06063 R 104034 R CALL STOR2 /UPDATE SYMBEG & CK TBL OVERFLOW.
5000 06064 R 225040 R LAC* SAVE
5001 06065 R 300220 R ADD IPOINT
5002 06066 R 065040 R DAC* SAVE /INCREASE ARRAY'S OFFSET
5003 06067 R 444103 R ISZ DECIN
5004 06070 R 220014 A LAC* R14 /MAKE ROOM FOR NEW STR ELEMENT.
5005 06071 R 060017 A DAC* R17
5006 06072 R 200014 A LAC R14
5007 06073 R 545040 R SAD SAVE /FINISHED?
5008 06074 R 605704 R JMP STRDEF /YES, GO INSERT NEW STR ELEM.
5009 06075 R 606070 R JMP .-5 /NO, CONTINUE.
5010 /
5011 06076 R 205434 R FNDELE LAC SYMBOL / LOOKING FOR AND FOUND ARRAY ELEMENT,
5012 06077 R 513756 R AND (500000) / STRING IF LINK=1, NUMERIC IF 0
5013 06100 R 741200 A SNA
5014 06101 R 606166 R JMP SYMEX / GOT ELEMENT; EXIT, LINK=1 FOR STRING
5015 06102 R 740400 A SNL
5016 06103 R 606137 R JMP NUMELE / FOUND NUMERIC ELEMENT
5017 006104 R DELELEM=.
5018 .IFPNZ FMONEY
5019 06104 R 750130 A SMA!CLA!IAC /PUT?
5020 06105 R 606110 R JMP DELELO /NO
5021 06106 R 540211 R SAD VALF /YES, F.P.?
5022 06107 R 000017 A ERR17 /ARRAY CANNOT CONTAIN F.P.!!!!
5023 .ENDC
5024 06110 R 220017 A DELELO LAC* R17 /FOUND STRING ELEMENT; KILL IT, CHECK
5025 /FOR PUT OR KILL
5026 06111 R 653606 A LMQ!CLAC!LLS+6
5027 06112 R 723002 A AAC 2
5028 06113 R 044103 R DELEL2 DAC DECIN
5029 06114 R 740001 A CMA
5030 06115 R 325040 R ADD* SAVE
5031 06116 R 741100 A SPA /IS THIS ONLY ELEM IN ARRAY?
5032 06117 R 606124 R JMP ONLYONE /YES.
5033 06120 R 065040 R DELEL1 DAC* SAVE /NO, SAVE ARRAY'S NEW OFFSET
5034 06121 R 205040 R LAC SAVE /AND

```

```

5035      06122 R 047066 R      DAC      WRSTEP      /A PTR TO IT.
5036      06123 R 605751 R      JMP      DELET3
5037
5038      06124 R 047066 R      ONLYONE DAC      WRSTEP
5039      06125 R 205434 R      LAC      SYMBOL
5040      06126 R 742010 A      RTL
5041      06127 R 740010 A      RAL
5042      06130 R 207066 R      LAC      WRSTEP
5043      06131 R 740400 A      SNL      /PUT OR KILL?
5044      06132 R 606120 R      JMP      DELEL1  /PUT, JK FOR PUT
5045      06133 R 777774 A      LAW      -4      / KILL ARRAY WITH ONLY ONE ELEMENT
5046      06134 R 300017 A      ADD      R17
5047      06135 R 040017 A      DAC      R17
5048      06136 R 605746 R      JMP      DELETE  / GO KILL ENTIRE ARRAY
5049
5050      006137 R      /
5051      NUMELE=.
5052      06137 R 751130 A      .IFPNZ  FMONEY
5053      CLA!IAC!SPA
5054      .ENDC
5055      .IFNOZ  FMONEY
5056      SPA!CLC
5057      .ENDC
5057      06140 R 606144 R      JMP      DELNL1
5058      06141 R 440017 A      DELNELE ISZ      R17      / FOUND NUM ELE; KILLIT
5059      06142 R 213705 R      LAC      (2
5060      06143 R 606113 R      JMP      DELEL2
5061      006144 R      DELNL1=.
5062      .IFPNZ  FMONEY
5063      06144 R 540211 R      SAD      VALF
5064      06145 R 000017 A      ERR17
5065      .ENDC
5066      06146 R 777777 A      LAW      -1
5067      06147 R 540207 R      SAD      VAL
5068      06150 R 606141 R      JMP      DELNELE  / FOUND NUM ELE; PUT STR ELE
5069      06151 R 606164 R      JMP      PUTNUM   / FOUND NUM ELE; PUT NUM ELE
5070
5071      /
5072      /NUM VAR FOUND: PUT OR KILL
5073
5073      06152 R 751101 A      NUMVAR  SPA!CLC      /PUT OR KILL?
5074      06153 R 606157 R      JMP      NUMV1      /PUT.
5075      06154 R 440017 A      DELNVAR ISZ      R17      / FOUND NUM VAR; KILL IT
5076      06155 R 213705 R      LAC      (2
5077      06156 R 605750 R      JMP      DELET2
5078      06157 R 540207 R      NUMV1   SAD      VAL
5079      06160 R 606154 R      JMP      DELNVAR   / FOUND NUM VAR; PUT STR VAR
5080      .IFPNZ  FMONEY
5081      06161 R 750030 A      CLA!IAC
5082      06162 R 540211 R      SAD      VALF
5083      06163 R 606154 R      JMP      DELNVAR   /FOUND NUM VAR; PUT F.P. VAR.
5084      .ENDC
5085      06164 R 200207 R      PUTNUM  LAC      VAL      / FOUND NUM VAR; PUT NUM VAR
5086      06165 R 060017 A      DAC*    R17

```

```
5087          006166 R    SYMEX=.
5088          .IFPNZ MONEY
5089          06166 R 400210 R    XCT    MONYSW    /IGNORE SWAPSW CODING IF IN $M
5090          06167 R 606173 R    JMP    ESYMBOL
5091          .ENDC
5092          06170 R 440652 R    ISZ    SWAPSW    /TIME SLICE UP?
5093          06171 R 741000 A    SKP
5094          06172 R 101533 R    JMS    HANG    /YES,ALLOW OTHERS TO RUN
5095          06173 R 047041 R    ESYMBOL DAC    WRPART    /PRESERVE AC STATUS
5096          06174 R 205434 R    LAC    SYMBOL
5097          06175 R 513657 R    AND    (77777    /STRIP OFF IBT 2 TO ALLOW
5098          06176 R 045434 R    DAC    SYMBOL    /64K PARTITON ECO TO RUN
5099          06177 R 207041 R    LAC    WRPART    /RETRIEVE FOR $H
5100          06200 R 625434 R    JMP*   SYMBOL
5101          .EJECT

/*12JAH15*
```

```

5102          /XBOOL - BOOLEAN EVALUATOR SUBROUTINE
5103          /
5104          /      CALLS TO BOOLL, BOOLG, AND BOOLE ARE RECURSIVE
5105          /      CALLS TO XREL1 TO PROCESS NUMERIC RELATIONAL
5106          /      OPERATORS.
5107          /
5108          06201 R 440234 R      UNOT      ISZ      BVAL      /UNARY NOT
5109          06202 R 606211 R      JMP      XBOOL3
5110          /
5111          /ENTRY TO BOOL
5112          /
5113          06203 R 000000 A      BOOL      0                      /*12JAH17*
5114          06204 R 213662 R      LAC      (400000                /*12JAH17*
5115          06205 R 040234 R      DAC      BVAL
5116          06206 R 140232 R      DZM      BSUM
5117          06207 R 750030 A      XBOOL1   CLA!IAC
5118          06210 R 040233 R      XBOOL2   DAC      BTERM
5119          06211 R 104660 R      XBOOL3   CALL     OCHARI      /GET NEXT CHAR.
5120          06212 R 553760 R      SAD      (47      /'
5121          06213 R 606201 R      JMP      UNOT      /UNARY NOT.
5122          06214 R 553753 R      SAD      (50      / (
5123          06215 R 606375 R      JMP      LPAREN     /SUBEXPRESSION.
5124          06216 R 104710 R      CALL     SUBOPT     /DECREMENT OPOINT.
5125          06217 R 744002 A      STL
5126          06220 R 107163 R      CALL     EVAL      /GO EVALUATE EXP ON THE LEFT
5127          06221 R 553741 R      SAD      (777777      /WAS EXP AN SVE?
5128          06222 R 606264 R      JMP      ITEXT1     /YES, GO PROCESS.
5129          06223 R 043174 R      DAC      OUT      /NO, HOLD NUM RESULT.
5130          06224 R 140256 R      DZM      BOOLL
5131          06225 R 140260 R      DZM      BOOLE
5132          06226 R 140262 R      DZM      BOOLG
5133          06227 R 104645 R      CALL     OCHAR      /GET CHAR.
5134          06230 R 553761 R      SAD      (74      /<
5135          06231 R 100256 R      CALL     BOOLL     /PROCESS (<) OPERATOR.
5136          06232 R 553762 R      SAD      (75      /=
5137          06233 R 100260 R      CALL     BOOLE     /PROCESS (=) OPERATOR.
5138          06234 R 553763 R      SAD      (76      />
5139          06235 R 100262 R      CALL     BOOLG     /PROCESS (>) OPERATOR.
5140          06236 R 553754 R      SAD      (51      / )
5141          06237 R 606242 R      JMP      .+3      / YES, GO EVAL SUBSCR
5142          06240 R 200207 R      BOOLNM   LAC      VAL      /NUMBER ONLY, RETURN IT
5143          06241 R 606341 R      JMP      REL2+1
5144          06242 R 200234 R      LAC      BVAL
5145          06243 R 744020 A      RCR
5146          06244 R 741600 A      SZL!SNA
5147          06245 R 606240 R      JMP      BOOLNM
5148          /##### POPSTACK #####
5149          06246 R 777774 A      LAW      -4
5150          06247 R 105103 R      CALL     UNSAVE
5151          06250 R 046203 R      DAC      BOOL
5152          06251 R 040233 R      DAC      BTERM
5153          06252 R 040232 R      DAC      BSUM

```

```

5154      06253 R 040234 R      DAC      BVAL
5155      /#####          #####
5156      06254 R 140242 R      DZM      BKCHAR      /SETUP VARIABLES FOR
5157      06255 R 140215 R      DZM      SUM          /RETURN TO EVAL.
5158      06256 R 777777 A      LAW      -1
5159      06257 R 040216 R      DAC      TERM
5160      06260 R 104660 R      CALL     OCHARI      /INCREMENT OPOINT, IGNORE CHAR.
5161      06261 R 200207 R      LAC      VAL
5162      06262 R 140207 R      DZM      VAL
5163      06263 R 607332 R      JMP      OP1          /EXIT TO EVAL.
5164      /
5165      /RELATION BETWEEN STRINGS
5166      /
5167      06264 R 200133 R      ITEXT1  LAC      BUFF2      /SETUP FOR CALLS TO CHAR1:
5168      06265 R 040225 R      DAC      POINT1      / POINTS TO STR AC.
5169      06266 R 213732 R      LAC      (XC01      /*12JAH14* CHAR
5170      06267 R 040226 R      DAC      PLACE1      /*12JAH14* POS
5171      06270 R 140224 R      DZM      TEMP1      / ENABLES EOM ESCAPE.
5172      06271 R 104645 R      CALL     OCHAR      /GET A CMND CHAR.
5173      06272 R 553764 R      SAD      (72      /:
5174      06273 R 606460 R      JMP      DEFIN      / GO TO SYNTAX VERIFIER
5175      06274 R 043174 R      DAC      OUT          /SAVE CHAR.
5176      06275 R 105363 R      CALL     SVE          /FIND SVE IN SYMB TBL AND CHECK IT.
5177      06276 R 040227 R      DAC      TEMP2      /ENABLE EOM - OR (") - ESC FROM CHAR2.
5178      06277 R 200017 A      LAC      R17
5179      06300 R 040230 R      DAC      POINT2      /POINTS TO BEG. OF SVE.
5180      06301 R 200016 A      LAC      R16          /*12JAH14* RESTORE
5181      06302 R 040231 R      DAC      PLACE2      /*12JAH14* CHAR POS
5182      06303 R 203174 R      LAC      OUT
5183      06304 R 553724 R      SAD      (33          /LEFT BRACKET
5184      06305 R 606452 R      JMP      CONTAINS      /PROCESS "CONTAINS" OPERATOR.
5185      06306 R 553762 R      SAD      (75          /=
5186      06307 R 606423 R      JMP      EQUAL          /PROCESS "STR EQUALS" OPERATOR.
5187      06310 R 553751 R      SAD      (35          /RIGHT BRACKET
5188      06311 R 606436 R      JMP      FOLLOWS      /PROCESS "FOLLOWS" OPERATOR.
5189      06312 R 000003 A      ERR3          /"SYNTAX"
5190      /
5191      /RELATION BETWEEN NUMERICS
5192      /
5193      06313 R 104660 R      XREL1  CALL     OCHARI      /GET NEXT CHAR.
5194      06314 R 553761 R      SAD      (74          /<
5195      06315 R 100256 R      CALL     BOOLL      /PROCESS "LESS THAN"
5196      06316 R 553762 R      SAD      (75          /=
5197      06317 R 100260 R      CALL     BOOLE      /PROCESS "EQUAL"
5198      06320 R 553763 R      SAD      (76          />
5199      06321 R 100262 R      CALL     BOOLG      /PROCESS "GREATER THAN"
5200      06322 R 104710 R      CALL     SUBOPT      /BACKUP OPOINT AND
5201      06323 R 107163 R      CALL     EVAL          /GO TO EVALUATOR.
5202      06324 R 652000 A      LMQ          /SAVE RES IN MQ.
5203      06325 R 243174 R      XOR      OUT
5204      06326 R 741210 A      SNA!RAL
5205      06327 R 606340 R      JMP      REL2          / LEFT=RIGHT

```

```

5206      06330 R 641002 A          LACQ          /NOT EQUAL.
5207      06331 R 740401 A          SNL!CMA
5208      06332 R 303174 R          ADD          OUT
5209      06333 R 740010 A          RAL
5210      06334 R 200256 R          LAC          BOOLL
5211      06335 R 744400 A          SNL!CLL
5212      06336 R 200262 R          LAC          BOOLG      /((LEFT EXP) > (RIGHT EXP))
5213      06337 R 741000 A          SKP          /((LEFT EXP) < (RIGHT EXP))
5214      06340 R 200260 R          REL2 LAC          BOOLE
5215      06341 R 741200 A          SNA
5216      06342 R 751000 A          LOSE CLA!SKP
5217      06343 R 750030 A          WIN  CLA!IAC
5218      06344 R 240234 R          XOR          BVAL
5219      06345 R 513667 R          AND          (1
5220      06346 R 040234 R          DAC          BVAL
5221      06347 R 104645 R          CALL         OCHAR      /GET A CHAR.
5222      06350 R 553765 R          SAD          (46      /&
5223      06351 R 606363 R          JMP          BAND      /PROCESS BOOLEAN AND.
5224      06352 R 553766 R          SAD          (41      /!
5225      06353 R 606367 R          JMP          BOR      /PROCESS BOOLEAN OR.
5226      06354 R 553653 R          SAD          (40      /SPACE
5227      06355 R 606416 R          JMP          BEOM
5228      06356 R 553754 R          SAD          (51      / )
5229      06357 R 606416 R          JMP          BEOM
5230      06360 R 553767 R          SAD          (54      /,
5231      06361 R 606416 R          JMP          BEOM
5232      06362 R 000003 A          ERR3      /"SYNTAX"
5233
5234      /
5235      /HANDLE AND OPERATOR
5236      /
5236      06363 R 200234 R          BAND LAC          BVAL
5237      06364 R 500233 R          AND          BTERM
5238      06365 R 140234 R          DZM          BVAL
5239      06366 R 606210 R          JMP          XBOOL2
5240
5241      /
5242      /HANDLE OR OPERATOR
5243      /
5243      06367 R 200234 R          BOR LAC          BVAL
5244      06370 R 500233 R          AND          BTERM
5245      06371 R 740200 A          SZA
5246      06372 R 040232 R          DAC          BSUM
5247      06373 R 140234 R          DZM          BVAL
5248      06374 R 606207 R          JMP          XBOOL1
5249
5250      /
5251      /HANDLE SUBEXPRESSIONS
5252      /
5252      /##### SAVE ON STACK #####
5253      06375 R 777774 A          LPAREN LAW          -4
5254      06376 R 105040 R          CALL         SAVE
5255      06377 R 200234 R          LAC          HVAL
5256      06400 R 200232 R          LAC          BSUM
5257      06401 R 200233 R          LAC          BTERM

```

```

5258      06402 R 206203 R      LAC      BOOL
5259      /#####          #####          /RECURSIVE CALL FOR
5260      06403 R 106203 R      CALL     BOOL          /BOOLEAN SUBEXPRESSION.
5261      06404 R 040234 R      DAC      BVAL
5262      /#####          #####
5263      06405 R 777774 A      LAW      -4
5264      06406 R 105103 R      CALL     UNSAVE
5265      06407 R 046203 R      DAC      BOOL
5266      06410 R 040233 R      DAC      BTERM
5267      06411 R 040232 R      DAC      BSUM
5268      06412 R 240234 R      XOR      BVAL          /CARRY THROUGH BVAL FROM SUBEXP.
5269      /#####          #####
5270      06413 R 040234 R      DAC      BVAL
5271      06414 R 104660 R      CALL     OCHARI          /INDEX OPOINT.
5272      06415 R 606347 R      JMP      WIN+4
5273      /
5274      /ALL DONE WITH BOOLEAN
5275      /
5276      06416 R 200234 R      BEOM    LAC      BVAL
5277      06417 R 500233 R      AND      BTERM
5278      06420 R 741200 A      SNA
5279      06421 R 200232 R      LAC      BSUM
5280      06422 R 626203 R      JMP*    BOOL
5281      /
5282      /STRING EQUALS
5283      /
5284      /      (DOES STR1 EQUAL STR2?)
5285      /
5286      06423 R 104770 R      EQUAL   CALL     CHAR1          /GET CHAR FROM STR1
5287      06424 R 606433 R      JMP      EQUAL2          /R1:EOM FOUND IN STR1.
5288      06425 R 044770 R      DAC      CHAR1          /R2: SAVE CHAR.
5289      06426 R 105014 R      CALL     CHAR2          /GET CHAR FROM STR2.
5290      06427 R 606342 R      JMP      LOSE          /R1:EOM OCCURS IN STR2 BEFORE STR1.
5291      06430 R 544770 R      SAD     CHAR1          /R2: STR2 CHAR = STR1 CHAR?
5292      06431 R 606423 R      JMP      EQUAL          /YES, CONTINUE.
5293      06432 R 606342 R      JMP      LOSE          /NO, YOU LOSE.
5294      06433 R 105014 R      EQUAL2  CALL     CHAR2          /GET NEXT CHAR IN STR2.
5295      06434 R 606343 R      JMP      WIN           /EOM IS NEXT CHAR IN STR2 ALSO.
5296      06435 R 606342 R      JMP      LOSE          /EOM NOT NEXT IN STR2, YOU LOSE.
5297      /
5298      /STRING FOLLOWS
5299      /
5300      /      (DOES STR1 FOLLOW STR2 IN 6-BIT COLLATING SEQUENCE?)
5301      /
5302      06436 R 104770 R      FOLLOWS CALL     CHAR1          /GET CHAR FROM STR1.
5303      06437 R 606342 R      JMP      LOSE          /R1: EOM IS LOSER.
5304      06440 R 044770 R      DAC      CHAR1          /R2: SAVE CHAR.
5305      06441 R 105014 R      CALL     CHAR2          /GET CHAR FROM STR2.
5306      06442 R 606343 R      JMP      WIN           /R1: EOM IS WINNER.
5307      06443 R 740001 A      CMA
5308      06444 R 304770 R      ADD     CHAR1          /CHAR1-CHAR2
5309      06445 R 740101 A      SMA!CMA

```

```

5310      06446 R 606343 R      JMP      WIN      /CHAR1 FOLLOWS, YOU WIN.
5311      06447 R 740200 A      SZA      /CHAR1=CHAR2?
5312      06450 R 606342 R      JMP      LOSE     /NO, YOU LOSE.
5313      06451 R 606436 R      JMP      FOLLOWS  /YES, CONTINUE.
5314      /
5315      /STRING CONTAINS
5316      /
5317      /      (DOES STR1 CONTAIN STR2?)
5318      /
5319      06452 R 140207 R      CONTAIN DZM      VAL      /SETUP TO START SEARCH AT BEG. OF STR1.
5320      06453 R 106621 R      CALL     CONT     /RE-ENTRANT SUBR. FOR "CONTAINS"
5321      06454 R 200207 R      LAC      VAL
5322      06455 R 740200 A      SZA
5323      06456 R 606343 R      JMP      WIN      /FOUND.
5324      06457 R 606342 R      JMP      LOSE     /NOT FOUND.
5325      /
5326      /SYNTAX VERIFIER (PATTERN VERIFICATION)
5327      /
5328      06460 R 200222 R      DEFIN   LAC      OPOINT
5329      06461 R 040230 R      DAC      POINT2  /SETUP POINTER FOR CHAR2
5330      06462 R 200223 R      LAC      OPLACE  /*12JAH14* SET UP
5331      06463 R 040231 R      DAC      PLACE2  /*12JAH14* CHAR POS
5332      06464 R 140227 R      DZM      TEMP2   /SETUP EOM RETURN FROM CHAR2.
5333      06465 R 105014 R      DEFO    CALL     CHAR2  /GET A PATTERN CHAR.
5334      06466 R 000003 A      ERR3
5335      06467 R 103724 R      CALL     SYNTAX  /TEST CHAR TYPE.
5336      06470 R 606570 R      JMP      DEF10   /A
5337      06471 R 606522 R      JMP      DEF2    /N
5338      06472 R 553755 R      SAD      (42     /P - IS IT A "
5339      06473 R 606504 R      JMP      DEF1    /YES, GO PROCESS.
5340      06474 R 104770 R      CALL     CHAR1   /NO, GET A CHAR FROM THE SVE.
5341      06475 R 741000 A      SKP
5342      06476 R 606563 R      JMP      DEF6    /EOM, END OF SVE - WIN
5343      06477 R 200230 R      LAC      POINT2  /SETUP FOR LOSE.
5344      06500 R 040222 R      DAC      OPOINT  /SETUP FOR WIN.
5345      06501 R 200231 R      LAC      PLACE2  /*12JAH14* SET UP
5346      06502 R 040223 R      DAC      OPLACE  /*12JAH14* CHAR POS
5347      06503 R 606343 R      JMP      WIN
5348      06504 R 105014 R      DEF1    CALL     CHAR2  /GET A PATTERN CHAR.
5349      06505 R 000003 A      ERR3
5350      06506 R 553755 R      SAD      (42     /IS IT ANOTHER "
5351      06507 R 606465 R      JMP      DEF0    /YES, FINISHED LITERAL MATCH.
5352      06510 R 045014 R      DAC      CHAR2   /NO, SAVE THE CHAR.
5353      06511 R 104770 R      CALL     CHAR1   /GET A CHAR FROM SVE.
5354      06512 R 606515 R      JMP      DEF3    /EOM.
5355      06513 R 545014 R      SAD      CHAR2   /SAME AS LITERAL CHAR?
5356      06514 R 606504 R      JMP      DEF1    /YES, CONTINUE.
5357      /
5358      /INDEX POINT2 PAST CLOSING QUOTES.
5359      /
5360      06515 R 105014 R      DEF3    CALL     CHAR2  /GET A PATTERN CHAR.
5361      06516 R 000003 A      ERR3

```

```

5362      06517 R 553755 R          SAD      (42          /IS IT A  "
5363      06520 R 606554 R          JMP      DEF5          /YES.
5364      06521 R 606515 R          JMP      DEF3          /NO
5365
5366      /
5367      /PASS SPECIFIED NO OF CHARS OF TYPE INDICATED.
5368      /
5368      06522 R 513743 R          DEF2     AND      (17
5369      06523 R 740001 A          CMA
5370      06524 R 044645 R          DAC      OCHAR          /MINUS PATTERN REPEAT COUNT.
5371      06525 R 105014 R          CALL     CHAR2         /GET A PATTERN CHAR.
5372      06526 R 000003 A          ERR3
5373      06527 R 103724 R          CALL     SYNTAX        /IS NEXT CHAR NUMERIC ALSO? (NOT ALLOWING
5374      06530 R 606533 R          JMP      DEF21         />1 DIGIT SPECIFICATION) NO--ALPHA--OK
5375      06531 R 000003 A          ERR3
5376      06532 R 000003 A          ERR3
5377      06533 R 045014 R          DEF21   DAC      CHAR2         /NUMERIC--SYNTAX
5378      06534 R 444645 R          DEF4     ISZ      OCHAR         /PUNCTUATION--SYNTAX
5379      06535 R 741000 A          SKP
5380      06536 R 606465 R          JMP      DEF0          /SAVE PATT VERIF CODE
5381      06537 R 104770 R          CALL     CHAR1         /FINISHED WITH THIS PATTERN TYPE?
5382      06540 P 606554 R          JMP      DEF5          /YES, GET NEXT TYPE.
5383      06541 R 103724 R          CALL     SYNTAX        /NO, GET CHAR FROM SVE.
5384      06542 R 606550 R          JMP      .+6           /R1: EOM IN SVE.
5385      06543 R 606546 R          JMP      .+3           /R2: TEST CHAR TYPE.
5386      06544 R 760020 A          LAW      20           /A - 01 MASK
5387      06545 R 606551 R          JMP      .+4           /N - 16 MASK
5388      06546 R 760016 A          LAW      16           /P - 20 MASK
5389      06547 R 606551 R          JMP      .+2
5390      06550 R 760001 A          LAW      01
5391      06551 R 505014 R          AND      CHAR2         /MASK PATTERN TYPE SPECIFICATION.
5392      06552 R 740200 A          SZA
5393      06553 R 606534 R          JMP      DEF4          /FAIL IF ZERO.
5394      /
5395      /INDEX POINT2 TO FIRST PUNCT THAT IS NOT (") OR WITHIN (")
5396      /((THERE MAY BE A (!) WITH ANOTHER PATTERN MATCH)
5397      /
5398      06554 R 105014 R          DEF5     CALL     CHAR2         /GET A PATTERN CHAR.
5399      06555 R 000003 A          ERR3
5400      06556 R 103724 R          CALL     SYNTAX        /"SYNTAX" - EOM
5401      06557 R 606554 R          JMP      DEF5          /TEST CHAR TYPE.
5402      06560 R 606554 R          JMP      DEF5          /A
5403      06561 R 553755 R          SAD      (42          /N
5404      06562 R 606515 R          JMP      DEF3          /P - IS IT A  "
5405      06563 R 200230 R          DEF6     LAC      POINT2        /YES.
5406      06564 R 040222 R          DAC      OPOINT        /SETUP FOR LOSE.
5407      06565 R 200231 R          LAC      PLACE2        /*12JAH14* SET UP
5408      06566 R 040223 R          DAC      OPLACE        /*12JAH14* CHAR POS
5409      06567 R 606342 R          JMP      LOSE
5410      /
5411      /PASS UNSPECIFIED NO. OF CHARS. OF TYPE INDICATED.
5412      /
5413      06570 R 045014 R          DEF10    DAC      CHAR2         /SAVE PATTERN CHAR.

```

5414	06571	R	104770	R	DEF12	CALL	CHAR1	/*12JAH12* GET CHAR FROM SVE
5415	06572	R	606606	R		JMP	DEF11	/R1; EDM IN SVE
5416	06573	R	103724	R		CALL	SYNTAX	/R2: TEST CHAR TYPE
5417	06574	R	606602	R		JMP	+.6	/ALPHA - 01 MASK
5418	06575	R	606600	R		JMP	+.3	/NUM - 16 MASK
5419	06576	R	760020	A		LAW	20	/PUNCT - 20 MASK
5420	06577	R	606603	R		JMP	+.4	
5421	06600	R	760016	A		LAW	16	
5422	06601	R	606603	R		JMP	+.2	
5423	06602	R	760001	A		LAW	01	
5424	06603	R	505014	R		AND	CHAR2	/MASK PATTERN TYPE SPECIFICATION.
5425	06604	R	740200	A		SZA		/FAIL IF ZERO.
5426	06605	R	606571	R		JMP	DEF12	/*12JAH12* PASS, CONTINUE (NO COUNT)
5427	06606	R	777777	A	DEF11	LAW	-1	/*12JAH14* STEP BACK
5428	06607	R	340226	R		TAD	PLACE1	/*12JAH14* ONE CHAR
5429	06610	R	553750	R		SAD	(XC01-1	/*12JAH14* OUT OF CURRENT WORD?
5430	06611	R	606613	R		JMP	DEF11A	/*12JAH14* YES
5431	06612	R	606617	R		JMP	DEF11B	/*12JAH14* NO
5432	06613	R	777777	A	DEF11A	LAW	-1	/*12JAH14* MOVE PTR
5433	06614	R	340225	R		TAD	POINT1	/*12JAH14* WORD BACK ONE
5434	06615	R	040225	R		DAC	POINT1	/*12JAH14*
5435	06616	R	213736	R		LAC	(XC03	/*12JAH14* POINT TO POS 3
5436	06617	R	040226	R	DEF11B	DAC	PLACE1	/*12JAH14* SAVE POS
5437	06620	R	606465	R		JMP	DEFO	/PATTERN SPEC FOLLOWS.
5438						.EJECT		

```

5439          /XCONT - CONTAINS SUBROUTINE
5440          /
5441          /      (IS STR2 CONTAINED WITHIN STR1?)
5442          /      (USED BY $F AND "CONTAINS OPERATOR" PROCESSING)
5443          /
5444          /
5445          06621 R 000000 A      CONT      0                                /*12JAH17*
5446          06622 R 200230 R          LAC      POINT2                                /*12JAH17*
5447          06623 R 045040 R          DAC      SAVE                                /POINTS TO BEG. OF STR2.
5448          06624 R 200231 R          LAC      PLACE2 /*12JAH14* SAVE CHAR
5449          06625 R 040272 R          DAC      GLOBAL /*12JAH14* POS
5450          06626 R 440207 R          ISZ      VAL
5451          06627 R 105014 R          CONT1    CALL     CHAR2      /GET CHAR FROM STR2.
5452          06630 R 626621 R          JMP*     CONT      /R1: END OF STR2
5453          06631 R 045014 R          DAC      CHAR2      /R2: SAVE STR2 CHAR.
5454          06632 R 104770 R          CONT2    CALL     CHAR1      /GET CHAR FROM STR1.
5455          06633 R 606671 R          JMP      CONT4      /R1: END OF STR1 - LOSE.
5456          06634 R 440207 R          ISZ      VAL      /R2:
5457          06635 R 545014 R          SAD      CHAR2      /STR1 CHAR = STR2 CHAR?
5458          06636 R 741000 A          SKP
5459          06637 R 606632 R          JMP      CONT2      /NO, KEEP LOOKING.
5460          06640 R 200225 R          LAC      POINT1      /YES.
5461          06641 R 045103 R          DAC      UNSAVE
5462          06642 R 200226 R          LAC      PLACE1      /*12JAH14* SAVE CHAR
5463          06643 R 040270 R          DAC      GSYM      /*12JAH14* POS
5464          06644 R 200207 R          LAC      VAL
5465          06645 R 045201 R          DAC      CHECK
5466          06646 R 105014 R          CONT3    CALL     CHAR2      /GET CHAR FROM STR2
5467          06647 R 626621 R          JMP*     CONT      /R1: END OF STR2 -WIN
5468          06650 R 045014 R          DAC      CHAR2      /R2: SAVE STR2 CHAR
5469          06651 R 104770 R          CALL     CHAR1      /GET CHAR FROM STR1
5470          06652 R 606671 R          JMP      CONT4      /R1: END OF STR1 -LOSE.
5471          06653 R 440207 R          ISZ      VAL      /R2:
5472          06654 R 545014 R          SAD      CHAR2      /STR1 CHAR = STR2 CHAR?
5473          06655 R 606646 R          JMP      CONT3      /YES, KEEP LOOKING.
5474          06656 R 205201 R          LAC      CHECK      /NO.
5475          06657 R 040207 R          DAC      VAL
5476          06660 R 200270 R          LAC      GSYM      /*12JAH14* RESOTRE
5477          06661 R 040226 R          DAC      PLACE1      /*12JAH14* CHAR
5478          06662 R 200272 R          LAC      GLOBAL      /*12JAH14* POSITION
5479          06663 R 040231 R          DAC      PLACE2      /*12JAH14* INDICATORS
5480          06664 R 205103 R          LAC      UNSAVE      /RESTORE POINTERS.
5481          06665 R 040225 R          DAC      POINT1
5482          06666 R 205040 R          LAC      SAVE
5483          06667 R 040230 R          DAC      POINT2
5484          06670 R 606627 R          JMP      CONT1      /START LOOKING AGAIN.
5485          06671 R 140207 R          CONT4    DZM      VAL
5486          06672 R 626621 R          JMP*     CONT      /LOSE EXIT.
5487          .EJECT

```

```

5488          /FORMAT SUBROUTINE
5489          /
5490          /SETTING OF LINK AT EXIT: LINK SET = ' FORMAT
5491          /          LINK CLEAR = OTHER
5492          /
5493          06673 R 000000 A   FORMAT 0                                /*12JAH17*
5494          06674 R 104660 R   XFORMAT CALL   OCHART
5495          06675 R 744002 A   XF0      STL
5496          06676 R 553766 R           SAD      (41          /!
5497          06677 R 606763 R           JMP      LINE
5498          06700 R 553755 R           SAD      (42          /"
5499          06701 R 606747 R           JMP      QUOTE
5500          06702 R 553770 R           SAD      (43          /#
5501          06703 R 606764 R           JMP      PAGE
5502          06704 R 553702 R           SAD      (77          /?
5503          06705 R 606713 R           JMP      QUEST
5504          06706 R 553760 R           SAD      (47          /'
5505          06707 R 626673 R           JMP*     FORMAT
5506          06710 R 104710 R   XF1      CALL   SUBOPT
5507          06711 R 744000 A           CLL
5508          06712 R 626673 R           JMP*     FORMAT
5509          /QUESTION MARK AS TABBING MECHANISM
5510          06713 R 744000 A   QUEST    CLL
5511          06714 R 107163 R           CALL   EVAL
5512          06715 R 741100 A           SPA
5513          06716 R 000005 A           ERR5          /"MINUS"
5514          06717 R 220150 R           LAC*     XVAR
5515          06720 R 740031 A           TCA
5516          06721 R 360141 R           TAD*     MVAR          /%M-SX
5517          06722 R 740100 A           SMA
5518          06723 R 741000 A           SKP          /USE VAL-%M
5519          06724 R 606727 R           JMP      .+3          /USE VAL-SX
5520          06725 R 220141 R           LAC*     MVAR
5521          06726 R 741000 A           SKP
5522          06727 R 220150 R           LAC*     XVAR
5523          06730 R 740031 A           TCA
5524          06731 R 340207 R           TAD      VAL
5525          06732 R 744000 A           CLL
5526          06733 R 657323 A           IDIVS; 144
5527          06734 R 000144 A
5527          06735 R 641002 A           LACQ
5528          06736 R 740031 A           TCA
5529          06737 R 740100 A           SMA
5530          06740 R 606777 R           JMP      NEXT          /NO ACTION NECESSARY
5531          06741 R 040207 R           DAC      VAL
5532          06742 R 760040 A           LAW      40          /SPACE
5533          06743 R 103174 R           CALL   OUT
5534          06744 R 440207 R           ISZ     VAL
5535          06745 R 606742 R           JMP      .-3
5536          06746 R 606777 R           JMP      NEXT
5537          /FREE TEXT
5538          06747 R 104660 R   QUOTE    CALL   OCHART

```

```

5539      06750 R 741200 A          SNA
5540      06751 R 000003 A          ERR3          /"SYNTAX"
5541      06752 R 553755 R          SAD           (42          /CLOSING QUOTE?
5542      06753 R 606756 R          JMP           QUOTE1
5543      06754 R 103174 R          CALL          OUT
5544      06755 R 606747 R          JMP           QUOTE
5545
5546      06756 R 104660 R          /
5547      06757 R 744000 A          QUOTE1 CALL    OCHARI
5548      06760 R 553742 R          CLL
5549      06761 R 626673 R          SAD           (56          /.
5550      06762 R 607000 R          JMP*          FORMAT
5551      06762 R 607000 R          JMP           NEXT+1
5552
5553      06763 R 751000 A          /
5554      06764 R 760034 A          /LINE FEED AND PAGE FEED
5555      06765 R 103174 R          LINE        CLA!SKP
5556      06766 R 104660 R          PAGE        LAW          34
5557      06767 R 607000 R          CALL          OUT
5558      06767 R 607000 R          CALL          OCHARI
5559      06770 R 200204 R          JMP           NEXT+1
5560      06771 R 040177 R          /PUT AWAY THE VARIABLE
5561      06772 R 200205 R          FORM2      LAC          SYMS
5562      06773 R 040200 R          DAC          SYM
5563      06773 R 040200 R          LAC          SCRIPS
5564      06774 R 140211 R          DAC          SCRIP
5565      06774 R 140211 R          .IFPNZ      FMONEY          /*12JAH6*
5566      06775 R 754002 A          DZM          VALF          /INHIBIT TYPE-OUT OF F.P. VAR  *12JAH6*
5567      06776 R 105434 R          .ENDC          /*12JAH6*
5568      06777 R 104645 R          STL!CLA
5569      07000 R 553767 R          CALL          SYMBOL          / PUT
5570      07001 R 606674 R          NEXT       CALL          OCHAR
5571      07002 R 553653 R          SAD          (54          /IF A COMMA, GET NEXT CHAR
5572      07003 R 607010 R          JMP          XFORMAT          /AND CHECK IT FOR FORMAT CTL
5573      07004 R 553733 R          SAD          (40          /SPACE, END OF COMMAND
5574      07005 R 607010 R          JMP          NEXT2
5575      07006 R 740200 A          SAD          (73          /: END OF COMMAND
5576      07007 R 606675 R          JMP          XFO          /EOM, END OF COMMAND
5577      07010 R 652000 A          NEXT2      LMQ
5578      07011 R 200162 R          NEXT1     LAC          PRNCDV          /RESTORE IO TO PRINCIPLE DEVICE
5579      07012 R 040020 A          DAC          R20
5580      07013 R 353771 R          TAD          (JMP OUTTAB-1
5581      07014 R 040164 R          DAC          OUTPUT
5582      07015 R 353772 R          TAD          (DEV
5583      07016 R 040163 R          DAC          INPUT
5584      07017 R 641002 A          LACQ
5585      07020 R 603574 R          JMP          DTABLE+1
5586
5587      07021 R 000000 A          /
5588      07022 R 200017 A          TSTR       0          /*12JAH17*
5589      07023 R 740030 A          LAC          R17          /TYPE STRING VARIABLE          /*12JAH17*
5590      07024 R 040225 R          IAC
5590      07024 R 040225 R          DAC          POINT1          / SET UP CHARACTER POINTER

```

```

5591      07025 R 213732 R          LAC      (XC01      /*12JAH14* INIT
5592      07026 R 040226 R          DAC      PLACE1     /*12JAH14* CHAR POS
5593      07027 R 140224 R          DZM      TEMP1      / TERMINATOR FOR STRING
5594      07030 R 104770 R      XSTR1  CALL      CHAR1     /*12JAH12* GET A CHARACTER
5595      07031 R 607034 R          JMP      XTSTR2    /*12JAH12* R1: TERMINATOR
5596      07032 R 103174 R          CALL     OUT       / R2: TYPE CHARACTER IN AC
5597      07033 R 607030 R          JMP      XSTR1     /*12JAH12*
5598      07034 R 220017 A      XTSTR2  LAC*     R17     /*12JAH12* GET OFFSET
5599      07035 R 653606 A          LMQ!CLAC!LLS+6
5600      07036 R 300017 A          ADD      R17      / POINTING TO NEXT VARIABLE
5601      07037 R 040017 A          DAC      R17
5602      07040 R 627021 R          JMP*     TSTR
5603
5604      07041 R 000000 A      /SUBROUTINE TO PRINT PARTS
5605      07042 R 760000 A      WRPART  0
5606      07043 R 103174 R      XWRPART LAW      00
5607      07044 R 200016 A          CALL     OUT
5608      07045 R 045151 R          LAC      R16
5609      07046 R 225151 R          DAC      PART
5610      07047 R 107066 R          LAC*     PART
5611      07050 R 220016 A          CALL     WRSTEP
5612      07051 R 653606 A          LAC*     R16
5613      07052 R 300016 A          LMQ!CLAC!LLS+6
5614      07053 R 540167 R          ADD      R16
5615      07054 R 613453 R          SAD      END
5616      07055 R 040016 A          JMP      WRITE4  /IN CASE OF WRITE ALL
5617      07056 R 220016 A          DAC      R16
5618      07057 R 653323 A          LAC*     R16
5619      07060 R 000144 A          IDIV;    144
5619      07061 R 641002 A          LACQ
5620      07062 R 540207 R          SAD      VAL
5621      07063 R 607044 R          JMP      XWRPART+2
5622      07064 R 040207 R          DAC      VAL
5623      07065 R 627041 R          JMP*     WRPART
5624
5625      07066 R 000000 A      /SUBROUTINE TO PRINT STEPS
5626      07067 R 104255 R      WRSTEP  0
5627      07070 R 760040 A          CALL     DECOU
5628      07071 R 103174 R          LAW      40
5629      07072 R 200016 A          CALL     OUT
5630      07073 R 740030 A          LAC      R16
5631      07074 R 040230 R          IAC
5632      07075 R 213732 R          DAC      POINT2
5633      07076 R 040231 R          LAC      (XC01     /*12JAH14* INIT
5634      07077 R 140227 R          DAC      PLACE2    /*12JAH14* CHAR POS PTR
5635      07100 R 105014 R          DZM      TEMP2
5636      07101 R 607104 R          CALL     CHAR2
5637      07102 R 103174 R          JMP      .+3
5638      07103 R 607100 R          CALL     OUT
5639      07104 R 760000 A          JMP      .-3
5640      07105 R 103174 R          LAW      00
5641      07106 R 627066 R          CALL     OUT
5641      07106 R 627066 R          JMP*     WRSTEP

```

```

5642                /SUBROUTINE TO SET UP I-O DEVICES
5643      07107 R 000000 A      SETIO      0                      /*12JAH17*
5644                .OCT                      /*12JAH17*
5645                .IFPNZ  XVM                /*12JAH15*
5646      07110 R 207107 R      LAC      SETIO      /SET UP LINK          /*12JAH15*
5647      07111 R 741400 A      SZL                      /STATUS IN          /*12JAH15*
5648      07112 R 253662 R      XOR      (400000      /BIT 0 OF           /*12JAH15*
5649      07113 R 047107 R      DAC      SETIO      /RETURN ADDR WORD   /*12JAH15*
5650                .ENDC                      /*12JAH15*
5651      07114 R 220143 R      LAC*     IVAR      /IS %I NEG?         /*12JAH15*
5652      07115 R 745100 A      SPA!CLL      /IS %I NEG?
5653      07116 R 000005 A      ERR5                /YES, "MINUS"
5654      07117 R 653323 A      IDIV;     144
5655      07121 R 641002 A      LACQ
5656      07122 R 353773 R      TAD      (-DEV-1      /IS DEVICE NUMBER TOO LARGE?
5657      07123 R 744100 A      SMA!CLL
5658      07124 R 000006 A      ERR6                /YES, "MAXIM"
5659      07125 R 641002 A      LACQ                /NO.
5660      07126 R 741200 A      SNA                /%I = PRINCIPAL I/O DEVICE?
5661      07127 R 200162 R      LAC      PRNC DV      /YES
5662      07130 R 040020 A      DAC      R20          /SETUP DEVICE NO.
5663      07131 R 313771 R      ADD      (JMP OUTTAB-1
5664      07132 R 040164 R      DAC      OUTPUT      /STORE "JMP OUTTAB+DEV" FOR OUTPUT.
5665      07133 R 313772 R      ADD      (DEV
5666      07134 R 040163 R      DAC      INPUT      /STORE "JMP INTAB+DEV" FOR INPUT.
5667      07135 R 353774 R      TAD      (-JMP-INTAB
5668      07136 R 353775 R      TAD      (WHOTAB+1
5669      07137 R 043174 R      DAC      OUT
5670      07140 R 223174 R      XSET0    LAC*     OUT      /GET OWNER FOR THIS DEVICE
5671      07141 R 540654 R      SAD      PRTNMB      /OURS?
5672      07142 R 607160 R      JMP      XSET1      /YES!!!
5673      07143 R 200020 A      LAC      R20
5674      07144 R 540162 R      SAD      PRNC DV      /PRINCIPAL DEVICE?
5675      07145 R 741000 A      SKP                      /YES--TRY TO LOCK TO IT
5676      07146 R 000031 A      ERR31           /NO--"IOLOK"
5677      07147 R 741000 A      SKP
5678      07150 R 101533 R      XSET01    JMS      HANG      /WAIT FOR PRINCIPAL DEVICE
5679      07151 R 705522 A      INTOFF
5680      07152 R 223174 R      LAC*     OUT
5681      07153 R 740200 A      SZA                /DEVICE AVAILABLE?
5682      07154 R 607150 R      JMP      XSET01      /NO--WAIT
5683      07155 R 200654 R      LAC      PRTNMB      /YES--TAKE IT
5684      07156 R 063174 R      DAC*     OUT
5685      07157 R 705521 A      INTON
5686      07160 R 207107 R      XSET1    LAC      SETIO
5687      07161 R 740010 A      RAL                      /RESTORE LINK AND
5688      07162 R 627107 R      JMP*     SETIO      /RETURN.

```

```

5689          .TITLE  EVALUATOR
5690          /EVALUATOR FOR NUMERIC AND STRING EXPRESSIONS
5691          /SET LINK ALLOWS NUMERIC OR STRING RESULT.
5692          /CLEAR LINK ALLOWS ONLY NUMERIC RESULT.
5693          /
5694          07163 R 000000 A      EVAL      0                               /*12JAH17*
5695          .IFPNZ  .-10000      /*12JAH17*
5696          ***ERROR-SWAPPING IMAGE RETURN ADDRESSES MUST BE RELOCATED
5697          ***BELOW 10000 AS IN GLOBAL, GSYM, ETC.
5698          .END
5699          .ENDC                               /*12JAH17*
5700          .IFPNZ  FMONEY
5701          07164 R 140211 R      DZM      VALF          /DON'T ALLOW F.P. ARGUMENTS
5702          .ENDC
5703          07165 R 754400 A      SNL!CLL!CLA      /NUMERIC RESULT ONLY?
5704          07166 R 607174 R      JMP      EVAL0      /YES
5705          07167 R 213730 R      LAC      (XCI1      /*12JAH14* INIT
5706          07170 R 040221 R      DAC      IPLACE     /*12JAH14* CHAR POS PTR
5707          07171 R 200133 R      LAC      BUFF2      /NO
5708          07172 R 040220 R      DAC      IPOINT     /SET UP BUFFER POINTER FOR STORING
5709          .ENDC                               /POSSIBLE STRING
5710          07173 R 723030 A      AAC      STRACC-1
5711          07174 R 040202 R      EVAL0    DAC      TEMP      /SET=0 FOR NUM ONLY;ELSE BUFF2+STRACC-1+400000
5712          07175 R 140215 R      DZM      SUM
5713          07176 R 751000 A      PLUS     CLA!SKP      / 000000 FOR +
5714          07177 R 213663 R      MINUS    LAC      (200000 / 200000 FOR -
5715          07200 R 040242 R      DAC      BKCHAR     /SET "SIGN" BIT (1) FOR TERM EVALUATION
5716          07201 R 777777 A      LAW      -1
5717          07202 R 040216 R      DAC      TERM      /-1 INDICATES * OR / NOT ENCOUNTERED.
5718          07203 R 140207 R      SCAN     DZM      VAL
5719          07204 R 104660 R      SCAN00  CALL     OCHARI   /*12JAH12* GET NEXT CHAR
5720          07205 R 553776 R      SAD      (55        /-
5721          07206 R 607231 R      JMP      UMINUS   /UNARY MINUS
5722          07207 R 553777 R      SAD      (53        /+
5723          07210 R 607233 R      JMP      UPLUS    /UNARY PLUS
5724          07211 R 103724 R      SCAN1   CALL     SYNTAX   /TEST CHAR TYPE.
5725          07212 R 607261 R      JMP      LETTER   /A
5726          07213 R 607306 R      JMP      DIGIT   /N
5727          07214 R 553753 R      SAD      (50        / (
5728          07215 R 607453 R      JMP      SUBEXP   /SUBEXPRESSION
5729          07216 R 554000 R      SAD      (44        /$
5730          07217 R 607236 R      JMP      DOLLAR   /FUNCTION OR SYST VARIABLE.
5731          07220 R 553752 R      SAD      (45        /%
5732          07221 R 607260 R      JMP      PERCENT   /SYST VARIABLE.
5733          07222 R 553755 R      SAD      (42        /"
5734          07223 R 607432 R      JMP      STRO     /TEXT LITERAL.
5735          07224 R 554001 R      SAD      (36        /^
5736          07225 R 610065 R      JMP      GGET    /GLOBAL VARIABLE.
5737          07226 R 553742 R      SAD      (56        /.
5738          07227 R 607306 R      JMP      DIGIT
5739          07230 R 000003 A      ERR3     /"SYNTAX"
5740          /

```

```

5741          /UNARY PLUS OR MINUS
5742          /
5743          07231 R 777777 A   UMINUS  LAW    -1
5744          07232 R 040207 R   DAC      VAL
5745          07233 R 140202 R   UPLUS   DZM     TEMP
5746          07234 R 104660 R   CALL    OCHARI  /GET NEXT CHAR. AND
5747          07235 R 607211 R   JMP     SCAN1  /CONTINUE EVALUATION.
5748          /DOLLAR SIGN, EITHER FUNCTION OR SYSTEM VARIABLE
5749          /
5750          07236 R 104660 R   DOLLAR  CALL    OCHARI  /GET NEXT CHAR.
5751          07237 R 103724 R   CALL    SYNTAX /MUST BE ALPHABETIC
5752          07240 R 607243 R   JMP     .+3    /A
5753          07241 R 000003 A   ERR3    /"SYNTAX"
5754          07242 R 000003 A   ERR3    /"SYNTAX"
5755          07243 R 214002 R   LAC     (1600 /DOLLAR (TRANSFORMED) SHIFTED BY 5 LEFT
5756          07244 R 640002 A   OMQ    /SYNTAX DOES LMQ
5757          07245 R 040177 R   DAC     SYM
5758          07246 R 104660 R   CALL    OCHARI  /GET REMAINING CHARS IN SNAME,
5759          07247 R 040217 R   DAC     ACTEMP  /*12JAH13* SAVE 2ND CHAR OF FCN NAME
5760          07250 R 741000 A   SKP    /*12JAH13* (IN CASE $J)
5761          07251 R 104660 R   CALL    OCHARI  /*12JAH13* GET MORE CHARS IN FCN NAME
5762          07252 R 103724 R   CALL    SYNTAX /CHECKING FOR ALPHA'S.
5763          07253 R 607251 R   JMP     .-2    /A
5764          07254 R 000003 A   ERR3    /N- "SYNTAX"
5765          07255 R 553753 R   SAD     (50   /PUNCT, IS IT (
5766          07256 R 634075 E   JMP*    FUNCT  /YES, CONTINUE PROCESSING FUNCTION.
5767          07257 R 607272 R   JMP     LETT2  /NO, POSSIBLY SYSTEM VARIABLE.
5768          /
5769          /PERCENT, TRANSFORM TO CODE 35
5770          /
5771          07260 R 213751 R   PERCENT LAC    (35   /BRACKET
5772          /
5773          /ALPHABETIC CHARACTERS
5774          /
5775          07261 R 040177 R   LETTER  DAC     SYM
5776          07262 R 777775 A   LAW     -3
5777          07263 R 040214 R   LETT0   DAC     M          /*12JAH12*
5778          07264 R 104660 R   LETT3   CALL    OCHARI  /*12JAH12* GET NEXT CHAR.
5779          07265 R 103724 R   CALL    SYNTAX /TEST CHAR TYPE.
5780          07266 R 607274 R   JMP     LETT1  /A
5781          07267 R 000003 A   ERR3    /N- "SYNTAX"
5782          07270 R 553753 R   SAD     (50   /PUNCT, IS IT (
5783          07271 R 607467 R   JMP     SUBSCR /YES, LOOK FOR SUBSCRIPT.
5784          07272 R 140200 R   LETT2   DZM     SCRIP  /NO, SET FOR "NO SUBSCRIPT" AND
5785          07273 R 607322 R   JMP     LOOKUP /GO SEARCH SYMBOL TABLE.
5786          /
5787          /PACK UP TO 3 ALPHA'S (5-BIT); IGNORE ALL ALPHA'S AFTER THIRD.
5788          /
5789          07274 R 043724 R   LETT1   DAC     SYNTAX
5790          07275 R 777777 A   LAW     -1
5791          07276 R 440214 R   ISZ    M          /3 CHARS YET?
5792          07277 R 745000 A   SKP!CLL /NO, CONTINUE

```

5793	07300 R 607263 R	JMP	LETT0	/*12JAH12* YES
5794	07301 R 200177 R	LAC	SYM	
5795	07302 R 640705 A	ALS	5	
5796	07303 R 243724 R	XOR	SYNTAX	
5797	07304 R 040177 R	DAC	SYM	
5798	07305 R 607264 R	JMP	LETT3	/*12JAH12*
5799		.EJECT		

```

5800          /NUMERICS
5801          /
5802          07306 R 104103 R  DIGIT  CALL  DECIN  /CONVERT TO NUMERIC
5803          07307 R 603724 R          JMP  SYNTAX /RETURN HERE IF ERROR ENCOUNTERED -- SYNTAX SETUP FOR PROPER CAL
5804          07310 R 607332 R          JMP  OP1   /GO ON TO REST OF EVALUATION.
5805          /
5806          /
5807          /
5808          /MULTIPLICATION
5809          /
5810          07311 R 200242 R  STAR   LAC   BKCHAR /BKCHAR (0) IS MUL-DIV SW.
5811          07312 R 740100 A          SMA   /FORCE TO 0 FOR MUL.
5812          07313 R 607203 R          JMP   SCAN /GO GET NEXT TERM.
5813          07314 R 607317 R          JMP   SLASH1 /*12JAH12*
5814          /
5815          /DIVISION
5816          /
5817          07315 R 200242 R  SLASH  LAC   BKCHAR
5818          07316 R 740100 A          SMA   /FORCE TO 1 FOR DIV.
5819          07317 R 253662 R  SLASH1 XOR   (400000 /*12JAH12*
5820          07320 R 040242 R          DAC   BKCHAR
5821          07321 R 607203 R          JMP   SCAN /GO GET NEXT TERM.
5822          /
5823          /SYMBOL IN THE TABLE?
5824          /
5825          07322 R 754000 A  LOOKUP CLL!CLA /CONDITION FOR SYMBOL "GET"
5826          07323 R 105434 R          CALL  SYMBOL /GET THE SYMBOL
5827          07324 R 641002 A          LACQ  /((RETURNED IN MQ)
5828          07325 R 741200 A  LOKUP1 SNA   /IS SYMBOL DEFINED?
5829          07326 R 000010 A          ERR10 /NO, "UNDEF"
5830          07327 R 745400 A          SZL!CLL /YES, IS IT STR OR NUM?
5831          07330 R 607525 R          JMP   STR1 /STR, GO PROCESS.
5832          07331 R 220017 A          LAC*  R17  /NUM, GET VALUE AND CONTINUE.
5833          07332 R 740200 A  OP1   SZA
5834          07333 R 240207 R          XOR   VAL
5835          07334 R 040207 R          DAC   VAL
5836          07335 R 140202 R          DZM   TEMP
5837          07336 R 777777 A          LAW   -1
5838          07337 R 540216 R          SAD   TERM /WAS A * OR / ENCOUNTERED AFTER LAST TERM?
5839          07340 R 607377 R          JMP   OP2 /NO, BYPASS MUL-DIV.
5840          07341 R 200242 R          LAC   BKCHAR /YES, CHECK MUL-DIV SWITCH (BIT 0)
5841          07342 R 705522 A          INTOFF
5842          07343 R 745100 A          SPA!CLL
5843          07344 R 607354 R          JMP   DIV0 /DIVIDE.
5844          07345 R 200207 R  MULO  LAC   VAL /MULTIPLY.
5845          07346 R 664000 A          GSM
5846          07347 R 047352 R          DAC   MUL2
5847          07350 R 200216 R          LAC   TERM
5848          07351 R 657122 A          MULS
5849          07352 R 000000 A  MUL2  0
5850          07353 R 607366 R          JMP   MUL3
5851          07354 R 200207 R  DIV0  LAC   VAL

```

```

5852      07355 R 664000 A          GSM
5853      07356 R 047364 R          DAC      DIV2
5854      07357 R 200216 R          LAC      TERM
5855      07360 R 657122 A          MULS;   144          /TERM * 100 / VAL
          07361 R 000144 A
5856      07362 R 740000 A          NOP
5857      07363 R 644323 A          DIVS
5858      07364 R 000000 A          DIV2    0
5859      07365 R 607370 R          JMP      DIV3
5860      07366 R 644323 A          MUL3    DIVS;   144          /TERM * VAL / 100
          07367 R 000144 A
5861      07370 R 705521 A          DIV3    INTON
5862      07371 R 741410 A          SZL!RAL /RESULT TOO LARGE?
5863      07372 R 000006 A          ERR6    /YES, "MAXIM"
5864      07373 R 641002 A          LACQ    /NO.
5865      07374 R 553741 R          SAD     (777777 /-0?
5866      07375 R 750000 A          CLA     /YES, CHANGE TO +0
5867      07376 R 741000 A          SKP
5868      07377 R 200207 R          OP2    LAC      VAL
5869      07400 R 040216 R          DAC     TERM          /SAVE TERM AND
5870      07401 R 104645 R          CALL   OCHAR        /GET A CHAR.
5871      07402 R 554003 R          SAD     (52          /*
5872      07403 R 607311 R          JMP     STAR        /SETUP FOR MULTIPLICATION.
5873      07404 R 554004 R          SAD     (57          //
5874      07405 R 607315 R          JMP     SLASH       /SETUP FOR DIVISION.
5875      07406 R 044645 R          DAC     OCHAR
5876      07407 R 200242 R          LAC     BKCHAR
5877      07410 R 742010 A          RTL     /"SIGN" BIT FOR THIS TERM INTO LINK.
5878      07411 R 200216 R          LAC     TERM
5879      07412 R 745400 A          SZL!CLL /COMPLEMENT IF TERM NEG.
5880      07413 R 740001 A          CMA
5881      07414 R 300215 R          ADD     SUM          /ACCUMULATE SUM.
5882      07415 R 553741 R          SAD     (777777 /IF -0, CHANGE TO +0
5883      07416 R 740001 A          CMA
5884      07417 R 040215 R          DAC     SUM
5885      07420 R 745400 A          SZL!CLL /NUMBER TOO LARGE?
5886      07421 R 000006 A          ERR6    /YES, "MAXIM"
5887      07422 R 204645 R          LAC     OCHAR        /NO, IF LAST CHAR + OR -, RECYCLE.
5888      07423 R 553777 R          SAD     (53          /+
5889      07424 R 607176 R          JMP     PLUS
5890      07425 R 553776 R          SAD     (55          /-
5891      07426 R 607177 R          JMP     MINUS
5892      07427 R 200215 R          LAC     SUM          /HOLD RESULT OF EVALUATION
5893      07430 R 040207 R          DAC     VAL          /IN "VAL" AND IN AC FOR
5894      07431 R 627163 R          JMP*    EVAL        /RETURN.
5895      /
5896      /QUOTES
5897      /
5898      07432 R 200202 R          STRO   LAC     TEMP
5899      07433 R 741200 A          SNA
5900      07434 R 000011 A          ERR11
5901      07435 R 104660 R          STRO.1 CALL   OCHARI /*12JAH12* NO, GET NEXT CHAR

```



```

5954      07510 R 741000 A      SKP
5955      07511 R 000003 A      ERR3      /NO, "SYNTAX"
5956      /#####      POPSTACK      #####
5957      07512 R 777771 A      LAW      -7      /YES
5958      07513 R 105103 R      CALL      UNSAVE      /POP STACK
5959      07514 R 040177 R      DAC      SYM
5960      07515 R 040202 R      DAC      TEMP
5961      07516 R 040207 R      DAC      VAL
5962      07517 R 040215 R      DAC      SUM
5963      07520 R 040216 R      DAC      TERM
5964      07521 R 040242 R      DAC      BKCHAR
5965      07522 R 047163 R      DAC      EVAL
5966      /#####      #####
5967      07523 R 104660 R      CALL      OCHARI
5968      07524 R 607322 R      JMP      LOOKUP      /GO LOOKUP SUBSCRIBTED VARIABLE
5969      /STRING VARIABLE
5970      /
5971      07525 R 200202 R      STR1     LAC      TEMP
5972      07526 R 741200 A      SNA
5973      07527 R 000011 A      ERR11
5974      07530 R 200017 A      LAC      R17
5975      07531 R 740030 A      IAC
5976      07532 R 040230 R      DAC      POINT2      /POINTER FOR CHAR2
5977      07533 R 213732 R      LAC      (XCO1      /*12JAH14* INIT
5978      07534 R 040231 R      DAC      PLACE2      /*12JAH14* CHAR POS PTR
5979      07535 R 140227 R      DZM      TEMP2
5980      07536 R 200133 R      LAC      BUFF2      /START OF STRING BUFFER      /*12JAH17*
5981      07537 R 721000 A      PAX
5982      07540 R 540220 R      SAD      IPOINT      /SAVE FOR TRANSFER      /*12JAH17*
5983      07541 R 741000 A      SKP      /NOW AT BEGINNING?      /*12JAH17*
5984      07542 R 607612 R      JMP      STR1.1      /YES      /*12JAH17*
5985      07543 R 200221 R      LAC      STR1.1      /NO, DO TRANSFER AS NDRMAL      /*12JAH17*
5986      07544 R 553730 R      LAC      IPLACE      /NOW AT 1ST CHAR?      /*12JAH17*
5987      07545 R 741000 A      SAD      (XCI1      /      /*12JAH17*
5988      07546 R 607612 R      SKP      /YES      /*12JAH17*
5989      07547 R 220230 R      JMP      STR1.1      /NO, DO TRANSFER AS NDRMAL      /*12JAH17*
5990      07550 R 653606 A      LAC*     POINT2      /FETCH 1ST WORD OF STR      /*12JAH17*
5991      07551 R 723747 A      CLAC!LMO!LLS+6 /CALC OFFSET      /*12JAH17*
5992      07552 R 740100 A      AAC      -STRACC      /MAX LENGTH      /*12JAH17*
5993      07553 R 000006 A      SMA
5994      07554 R 723031 A      ERR6
5995      07555 R 340133 R      AAC      STRACC      /TOO LONG?      /*12JAH17*
5996      07556 R 740030 A      TAD      BUFF2      /YES, MAXIM ERROR      /*12JAH17*
5997      07557 R 722000 A      IAC      /RESTORE OFFSET      /*12JAH17*
5998      07560 R 220017 A      PAL
5999      07561 R 741200 A      LAC*     R17      /ADD TO START OF STR ACC      /*12JAH17*
6000      07562 R 607625 R      IAC      /PLUS ONE      /*12JAH17*
6001      .DBREL ; DBA      PAL
6002      07563 R 707762 A      LAC*     R17      /LIMIT OF TRANSFER      /*12JAH17*
6003      07564 R 741000 A      SKP      /FETCH FIRST WORD OF STRING      /*12JAH17*
6004      07565 R 220017 A      LAC*     R17      /NULL STRING?      /*12JAH17*
6004      07566 R 050000 A      DAC      0,X      /YES, ALL DONE      /*12JAH17*
6004      /MUST BE IN PAGE 0      /*12JAH17*
6004      /1ST TIME WORD ALREADY PICKED UP /*12JAH17*
6004      /FETCH STR WORD      /*12JAH17*
6004      /PUT IN STR ACC      /*12JAH17*

```

6005	07567 R 725001 A		AXS	1	/DONE?	/*12JAH17*
6006	07570 R 607565 R		JMP	.-3	/NO	/*12JAH17*
6007			.EBREL ;	EBA	/YES, RESTORE TO BANK MODE	/*12JAH17*
	07571 R 707764 A					
6008	07572 R 740200 A		SZA		/LAST WORD ZERO?	/*12JAH17*
6009	07573 R 607601 R		JMP	STR1.2	/NO	/*12JAH17*
6010	07574 R 213727 R		LAC	(XCI3	/YES, SET	/*12JAH17*
6011	07575 R 040221 R		DAC	IPLACE	/CHAR POSITION	/*12JAH17*
6012	07576 R 724000 A		PXA		/FETCH END OF STR IN ACC	/*12JAH17*
6013	07577 R 723776 A		AAC	-2	/ADJUST	/*12JAH17*
6014	07600 R 607610 R		JMP	STR1.4	/FINISH UP	/*12JAH17*
6015	07601 R 513645 R	STR1.2	AND	(007777	/LOOK AT 1ST CHAR ONLY	/*12JAH17*
6016	07602 R 740200 A		SZA		/ONLY 1 CHAR?	/*12JAH17*
6017	07603 R 213667 R		LAC	(XCI2-XCI1	/NO	/*12JAH17*
6018	07604 R 353730 R		TAD	(XCI1	/YES	/*12JAH17*
6019	07605 R 040221 R		DAC	IPLACE	/SAVE CHAR POS	/*12JAH17*
6020	07606 R 724000 A		PXA		/FETCH END OF STR IN ACC	/*12JAH17*
6021	07607 R 723777 A		AAC	-1	/ADJUST	/*12JAH17*
6022	07610 R 040220 R	STR1.4	DAC	IPOINT	/SAVE WORD POS	/*12JAH17*
6023	07611 R 607625 R		JMP	STR3		/*12JAH17*
6024	007612 R	STR1.1=.				/*12JAH17*
6025	07612 R 105014 R	STR2	CALL	CHAR2	/GET A CHAR.	
6026	07613 R 607625 R		JMP	STR3	/EOM RETURN FROM CHAR2	
6027	07614 R 104725 R		CALL	ICHARI	/STORE CHAR.	
6028	07615 R 200220 R		LAC	IPOINT		
6029	07616 R 540202 R		SAD	TEMP	/BUFFER FILLED?	
6030	07617 R 741000 A		SKP		/*12JAH14* LAST WORD OF BUFFER	
6031	07620 R 607612 R		JMP	STR2	/*12JAH14* NOT LAST WORD	
6032	07621 R 200221 R		LAC	IPLACE	/*12JAH14* CHAR POS	
6033	07622 R 553727 R		SAD	(XCI3	/*12JAH14* LAST ONE?	
6034	07623 R 000006 A		ERR6		/*12JAH14* YES, "MAXIM"	
6035	07624 R 607612 R		JMP	STR2	/NO, CONTINUE.	
6036		/				
6037		/STRING OPERATORS?				
6038		/				
6039	07625 R 104645 R	STR3	CALL	OCHAR	/GET A CHAR.	
6040	07626 R 741000 A		SKP		/OR	
6041	07627 R 104660 R	STR3.1	CALL	OCHARI	/INDEX AND GET A CHAR.	
6042	07630 R 553742 R		SAD	(56	/. (CONCATENATION OPERATOR)?	
6043	07631 R 607204 R		JMP	SCAN00	/*12JAH12* YES, GO GET NEXT STRING	
6044	07632 R 750000 A		CLA		/NO.	
6045	07633 R 104725 R		CALL	ICHARI		
6046	07634 R 200221 R		LAC	IPLACE	/*12JAH14* CHAR POS PTR	
6047	07635 R 553730 R		SAD	(XCI1	/*12JAH14* FIN 1 CHAR?	
6048	07636 R 607645 R		JMP	STR3.2	/*12JAH14* YES	
6049	07637 R 553727 R		SAD	(XCI3	/*12JAH14* FIN 3 CHARS?	
6050	07640 R 607646 R		JMP	STR3.3	/*12JAH14* YES	
6051	07641 R 220220 R		LAC*	IPOINT	/*12JAH14* NO	
6052	07642 R 514005 R		AND	(770000	/*12JAH14* TRIM OFF LAST SIX BITS	
6053	07643 R 060220 R		DAC*	IPOINT	/*12JAH14* FROM LAST WORD OF STRING	
6054	07644 R 741000 A		SKP		/*12JAH14*	
6055	07645 R 160220 R	STR3.2	DZM*	IPOINT	/*12JAH14* CLEAR FINAL WORD	

```

6056      07646 R 200133 R   STR3.3  LAC      BUFF2      /*12JAH14*
6057      07647 R 740031 A          CMA!IAC      /*12JAH14*
6058      07650 R 340220 R          TAD      IPOINT    /*12JAH14* NO OF LOCS USED IN BUFF2.
6059      07651 R 513702 R          AND      (77
6060      07652 R 650506 A          CLQ!LRS 6
6061      07653 R 220133 R          LAC*     BUFF2
6062      07654 R 513645 R          AND      (7777
6063      07655 R 640002 A          OMO      /OFFSET INTO HI 6 BITS.
6064      07656 R 060133 R          DAC*     BUFF2
6065      07657 R 754001 A   MSTR    CLA!CMA!CLL  /HOLD 777777 (STR IND)
6066      07660 R 040207 R          DAC      VAL      /IN "VAL" AND IN AC FOR
6067      07661 R 627163 R          JMP*     EVAL     /RETURN.
6068      /
6069      /RETURN HERE FROM STR FUNCT PROCESSING.
6070      /
6071      007662 R   STR4=.
6072      .IFPNZ  MONEY
6073      07662 R 213674 R   LAC      (SKP
6074      07663 R 040210 R   DAC      MONYSW   /RESTORE SWITCH TO NON-SM MODE
6075      .ENDC
6076      07664 R 104645 R   CALL     OCHAR    /GET A CHAR
6077      07665 R 553754 R   SAD      (51     /MUST BE A )
6078      07666 R 751001 A   SKP!CLA!CMA     /-1 FOR POPPING STACK.
6079      /#####  POPSTACK      #####
6080      07667 R 000003 A   ERR3     /"SYNTAX"
6081      07670 R 105103 R   CALL     UNSAVE
6082      07671 R 047163 R   DAC      EVAL
6083      /#####  #####
6084      .IFPNZ  FMONEY
6085      07672 R 200211 R   LAC      VALF
6086      07673 R 741200 A   SNA      /FLOATING POINT VALUE RETURNED?
6087      .ENDC
6088      07674 R 607627 R   JMP      STR3.1   /NO, CONTINUE STRING PROCESSING
6089      .IFPNZ  FMONEY
6090      07675 R 104660 R   CALL     OCHARI   /YES, ADVANCE
6091      07676 R 607657 R   JMP      MSTR     /EXIT FROM EVAL
6092      .ENDC

```

```

6093          .TITLE GLOBAL VARIABLE ROUTINES
6094          /
6095          /STRUCTURE OF CORE DIRECTORY FOR GLOBALS
6096          /
6097          /WORD 0          000 (15 BIT GLOBAL NAME IN 5 BIT R.A.)
6098          /WORD 1          (18 BIT BLOCK POINTER TO FIRST BLOCK OF LEVEL 0)
6099          /WORD 2          UUU (15 BIT OPENED PARTITION NUMBER) UUU=UNIT
6100          /WORD 3 X (17 BIT READ KEY) X=1, TRAP ON KEY VIOLATION
6101          /WORD 4 Y (17 BIT OPEN KEY) Y=1, TRAP ON KEY VIOLATION
6102          /
6103          /GLOBAL OPEN AND CLOSE
6104          /
6105          07677 R 744400 A OPEN SNL!CLL
6106          07700 R 000004 A ERR4          /"CMMND"
6107          07701 R 104660 R CALL OCHART
6108          07702 R 554001 R SAD (36          /^
6109          07703 R 607707 R JMP OPEN0
6110          07704 R 104710 R OVERLAY CALL SUBOPT
6111          07705 R 754000 A CLA!CLL
6112          07706 R 634141 E JMP* OLAY          /*12JAH12*
6113          /
6114          07707 R 200222 R OPEN0 LAC OPOINT          /OPEN
6115          07710 R 040016 A DAC R16
6116          07711 R 200223 R LAC OPLACE          /*12JAH14* SAVE POS
6117          07712 R 040015 A DAC R15          /*12JAH14* PTR
6118          07713 R 140254 R DZM GSYMBOL          /NO NAKEDS AFTER OPEN
6119          07714 R 103641 R OPEN1 CALL PRODA          /GET DISK
6120          07715 R 100266 R OPEN5 CALL GASS          /ASSEMBLE GLOBAL NAME
6121          07716 R 000003 A ERR3          /"SYNTAX" - NAKED NOT ALLOWED
6122          07717 R 440177 R ISZ SYM
6123          07720 R 440177 R ISZ SYM
6124          07721 R 220177 R LAC* SYM
6125          07722 R 513702 R AND (77          /GET OWNER
6126          07723 R 745200 A SNA!CLL
6127          07724 R 607747 R JMP OPEN2          /NOBODY OWNS IT
6128          07725 R 540655 R SAD RUNNMB
6129          07726 R 607742 R JMP OPEN4          /THIS USER OWNS IT, BUT TEST KEY ANYWAY
6130          07727 R 110005 R CALL REMOVE          /SOMEONE ELSE OWNS IT - CLOSE...
6131          07730 R 200172 R LAC WRTSW          /WRITE OUT PARTITION BUFFER
6132          07731 R 740200 A SZA
6133          07732 R 134057 E JMS* DMPOLD          /*12JAH12*
6134          07733 R 140577 R DZM DSKACT          /FREE DISK
6135          07734 R 101533 R JMS HANG
6136          07735 R 200016 A LAC R16
6137          07736 R 040222 R DAC OPOINT          /RESTORE POINTER...
6138          07737 R 200015 A LAC R15          /*12JAH14* RESTORE
6139          07740 R 040223 R DAC OPLACE          /*12JAH14* CHAR POS PTR
6140          07741 R 607714 R JMP OPEN1          /...AND TRY AGAIN
6141          /
6142          07742 R 260177 R OPEN4 XOR* SYM          /CLOSE UNTIL HE PROVIDES CORRECT KEY
6143          07743 R 060177 R DAC* SYM
6144          07744 R 777777 A LAW -1

```

6145	07745 R 340670 R		TAD	OPNCNT	
6146	07746 R 040670 R		DAC	OPNCNT	/DECREMENT OPEN COUNTER
6147	07747 R 440177 R	OPEN2	ISZ	SYM	
6148	07750 R 440177 R		ISZ	SYM	/POINTER TO OPEN KEY
6149	07751 R 220137 R		LAC*	KVAR	
6150	07752 R 040207 R		DAC	VAL	
6151	07753 R 104645 R		CALL	OCHAR	
6152	07754 R 553724 R		SAD	(33	/LEFT BRACKET
6153	07755 R 741000 A		SKP		
6154	07756 R 607761 R		JMP	+.3	
6155	07757 R 105242 R		CALL	GETKEY	
6156	07760 R 000003 A		ERR3		/"SYNTAX"
6157	07761 R 105272 R		CALL	KEYCHK	
6158	07762 R 220177 R		LAC*	SYM	
6159	07763 R 540207 R		SAD	VAL	
6160	07764 R 740200 A		SZA		
6161	07765 R 000032 A		ERR32		/KEY DOES NOT MATCH
6162	07766 R 200177 R		LAC	SYM	
6163	07767 R 723776 A		AAC	-2	
6164	07770 R 040177 R		DAC	SYM	/POINTER TO WORD 2 (OPEN OWNER)
6165	07771 R 220177 R		LAC*	SYM	
6166	07772 R 240655 R		XOR	RUNNMB	
6167	07773 R 060177 R		DAC*	SYM	/OPENED TO THIS PARTITION
6168	07774 R 440670 R		ISZ	OPNCNT	
6169	07775 R 104645 R	OPEN3	CALL	OCHAR	
6170	07776 R 553767 R		SAD	(54	/. COMMA MEANS MORE TO OPEN
6171	07777 R 741000 A		SKP		
6172	10000 R 610040 R		JMP	CLOSE0	/DONE
6173	10001 R 104660 R		CALL	OCHAR1	
6174	10002 R 554001 R		SAD	(36	/^
6175	10003 R 607715 R		JMP	OPENS	
6176	10004 R 000013 A		ERR13		/NOT UP ARROW, "GLOBE" ERROR
6177			.EJECT		

```

6178          /REMOVE CLOSES ALL GLOBALS OWNED BY PRESENT USER.
6179          /
6180          10005 R 000000 A REMOVE 0 /*12JAH17*
6181          10006 R 214104 E XREMOVE LAC GTABLE /*12JAH17*
6182          10007 R 040177 R DAC SYM
6183          10010 R 220177 R REMOVE1 LAC* SYM
6184          10011 R 553741 R SAD (777777 /END OF TABLE MARKER
6185          10012 R 630005 R JMP* REMOVE /DONE
6186          10013 R 440177 R RMOV ISZ SYM
6187          10014 R 440177 R ISZ SYM /POINTER TO WORD 2 (OPEN OWNER)
6188          10015 R 220177 R LAC* SYM
6189          10016 R 513702 R AND (77
6190          10017 R 540655 R SAD RUNNMB
6191          10020 R 610027 R JMP REMOVE3 /RUNNMB MATCHES - THIS GUY OWNS IT
6192          10021 R 440177 R REMOVE2 ISZ SYM
6193          10022 R 440177 R ISZ SYM
6194          10023 R 440177 R ISZ SYM /GO TO NEXT ENTRY
6195          10024 R 741400 A SZL /SINGLE GLOBAL CLOSE?
6196          10025 R 610055 R JMP CLOSE3 /YES - LEAVE
6197          10026 R 610010 R JMP REMOVE1
6198          10027 R 220177 R REMOVE3 LAC* SYM
6199          10030 R 514006 R AND (377700 /MASK OFF ID#
6200          10031 R 060177 R DAC* SYM
6201          10032 R 200670 R LAC OPNCNT /DECREMENT OPEN COUNTER.
6202          10033 R 723777 A AAC -1
6203          10034 R 040670 R DAC OPNCNT
6204          10035 R 610021 R JMP REMOVE2
6205          /
6206          10036 R 103641 R CLOSE CALL PRODA /GET DISK, CLOSE USER'S GLOBALS
6207          10037 R 110005 R CALL REMOVE
6208          10040 R 200172 R CLOSE0 LAC WRTSW
6209          10041 R 744200 A SZA!CLL
6210          10042 R 134057 E JMS* DMPOLD /*12JAH12* WRITE OUT GLOBAL BUFF IF NECC
6211          10043 R 140577 R DZM DSKACT /FREE DISK
6212          10044 R 440652 R ISZ SWAPSW /TIME SLICE UP?
6213          10045 R 603573 R JMP DTABLE /NO
6214          10046 R 101533 R JMS HANG /YES, LET OTHERS RUN
6215          10047 R 603573 R JMP DTABLE
6216          10050 R 103641 R CLOSE1 CALL PRODA /GET DISK
6217          10051 R 100266 R CLOSE2 CALL GASS /FIND GLOBAL
6218          10052 R 000003 A ERR3 /NAKED OR NOT THERE
6219          10053 R 744002 A STL /FLAG SINGLE GLOBAL CLOSE
6220          10054 R 610013 R JMP RMOV /GO TAKE IT AWAY
6221          10055 R 104645 R CLOSE3 CALL OCHAR /WHAT'S NEXT
6222          10056 R 553767 R SAD (54 /COMMA?
6223          10057 R 741000 A SKP /MORE TO COME
6224          10060 R 610040 R JMP CLOSE0 /NO - DONE
6225          10061 R 104660 R CALL UCHAR /GET NEXT CHAR.
6226          10062 R 554001 R SAD (36 /^?
6227          10063 R 610051 R JMP CLOSE2 /CONTINUE
6228          10064 R 000013 A ERR13 /NOT ^, "GLOBE" ERROR
6229          .EJECT

```

```

6230          /DEFINITIONS
6231          /
6232          /
6233          /
6234          /GLOBAL GET. COMES HERE FROM EVAL
6235          /
6236          10065 R 100270 R  GGET  CALL  GSYM
6237          10066 R 754000 A          CLL!CLA
6238          10067 R 100272 R          CALL  GLOBAL
6239          10070 R 641002 A          LACQ
6240          10071 R 644000 A          ABS
6241          10072 R 607325 R          JMP    LOKUP1          /RETURN TO EVAL
6242          /
6243          /
6244          /GLOBAL PUT. COMES HERE FROM SET
6245          /
6246          10073 R 100270 R  GPUT  CALL  GSYM
6247          10074 R 553762 R          SAD    (75          /=
6248          10075 R 755002 A          SKP!CLA!STL
6249          10076 R 000003 A          ERR3          /"SYNTAX" - NUMERIC CHARACTER
6250          10077 R 107163 R          CALL  EVAL          /PUNCTUATION CHAR
6251          10100 R 754002 A          STL!CLA
6252          10101 R 740010 A          RAL
6253          10102 R 100272 R          CALL  GLOBAL          /PUT
6254          10103 R 613244 R          JMP    SET2          /JUMP BACK TO SET
6255          /
6256          /
6257          /GLOBAL KILL. COMES HERE FROM KILL.
6258          /
6259          10104 R 100270 R  GKILL  CALL  GSYM
6260          010105 R          GKILL2=.
6261          10105 R 750001 A          CLC
6262          10106 R 100272 R          CALL  GLOBAL
6263          10107 R 612527 R          JMP    KILL15          /*12JAH12*
6264          10110 R 750001 A          GKILL3  CLC
6265          10111 R 100272 R          CALL  GLOBAL          /UNSUBSCRIPTED KILL
6266          010112 R          GKILL4=.          /INDICATOR TO XGLOBAL OF COMP. KILL
6267          10112 R 612527 R          JMP    KILL15          /*12JAH12*
6268          .EJECT

```

```

6269          /XGASS - ASSEMBLE A GLOBAL NAME, FIND IT
6270          /
6271          /CALLING SEQUENCE:
6272          /      CALL      GASS
6273          /      RETURNS HERE IF POSSIBLE NAKED
6274          /      RETURNS HERE IF FOUND - SYM HOLDS POINTER TO WORD0 OF GLOBAL
6275          /
6276          /
6277          10113 R 777775 A  XGASS  LAW      -3          /CHAR COUNT IS 3
6278          10114 R 040214 R      DAC      M
6279          10115 R 104660 R      CALL     OCHARI
6280          10116 R 103724 R      CALL     SYNTAX
6281          10117 R 610132 R      JMP      GASS5
6282          10120 R 000003 A      ERR3
6283          10121 R 620266 R      JMP*    GASS          /"SYNTAX"
6284          10122 R 440214 R  GASS1  ISZ      M          /POSSIBLE NAKED
6285          10123 R 610127 R      JMP     GASS15       /BUMP CHAR COUNT
6286          10124 R 777777 A      LAW     -1          /*12JAH12*
6287          10125 R 040214 R      DAC     M          /EXCESS CHARS
6288          10126 R 610133 R      JMP     GASS2
6289          10127 R 640505 A  GASS15  LRS      5          /*12JAH12*
6290          10130 R 200204 R      LAC     SYMS        /RUNNING NAME
6291          10131 R 640605 A      LLS     5
6292          10132 R 040204 R  GASS5  DAC     SYMS
6293          10133 R 104660 R  GASS2  CALL     OCHARI
6294          10134 R 103724 R      CALL     SYNTAX
6295          10135 R 610122 R      JMP     GASS1
6296          10136 R 000003 A      ERR3          /"SYNTAX"
6297          /BEGIN SEARCH
6298          /      SYMS HOLDS 5 BIT R.A. GLOBAL NAME
6299          10137 R 440266 R      ISZ     GASS
6300          10140 R 214104 E      LAC     GTABLE       /GET GLOBAL DIRECTORY POINTER
6301          10141 R 040177 R  GASS6  DAC     SYM
6302          10142 R 220177 R      LAC*   SYM
6303          10143 R 553741 R      SAD     (777777)   /END OF TABLE MARKER
6304          10144 R 000010 A      ERR10          /"UNDEF" - NOT IN TABLE
6305          10145 R 540204 R      SAD     SYMS
6306          10146 R 620266 R      JMP*   GASS          /GOT IT!
6307          10147 R 200177 R      LAC     SYM
6308          10150 R 723005 A      AAC     5          /ADVANCE TO NEXT ENTRY
6309          10151 R 610141 R      JMP     GASS6
6310          .EJECT

```

```

6311          /GSYM PACKS SUBSCRIPTS INTO USER'S STACK.
6312          /
6313          10152 R 200165 R  XGSYM  LAC    STACK
6314          10153 R 740031 A          TCA
6315          10154 R 340166 R          TAD    SPOINT
6316          10155 R 047107 R          DAC    SETIO    /SAVE STACK POINTER OFFSET
6317          10156 R 140177 R          DZM    SYM
6318          10157 R 140213 R          DZM    OFLAG    /ZERO FIRST SUBSCRIPT INDICATOR *12JAH9*
6319          10160 R 100266 R          CALL   GASS
6320          10161 R 610375 R          JMP    GSYM11   /NAKED *12JAH9*
6321          /##### SAVE ON STACK #####
6322          10162 R 777777 A          LAW    -1
6323          10163 R 105040 R          CALL   SAVE
6324          10164 R 200207 R          LAC    VAL
6325          /#####
6326          10165 R 220137 R          LAC*   KVAR    /IN CASE %K NEEDED
6327          10166 R 040207 R          DAC    VAL
6328          10167 R 200177 R          LAC    SYM
6329          10170 R 723002 A          AAC    2
6330          10171 R 040214 R          DAC    M        /POINTER TO WORD 2 - OWNERSHIP
6331          10172 R 220214 R          LAC*   M        /PICK UP OWNER
6332          10173 R 440214 R          ISZ    M
6333          10174 R 440214 R          ISZ    M        /POINTER TO OPEN KEY
6334          10175 R 513702 R          AND    (77
6335          10176 R 540655 R          SAD    RUNNMB
6336          10177 R 610350 R          JMP    GSYM3    /ALREADY OWNED BY THIS GUY - NO
6337                                     /KEY CHECKING NECESSARY UNLESS HE INSISTS
6338          10200 R 104645 R          CALL   OCHAR
6339          10201 R 553724 R          SAD    (33     /LEFT BRACKET
6340          10202 R 741000 A          SKP
6341          10203 R 610214 R          JMP    GSYM21   /*12JAH9*
6342          10204 R 777777 A          GSYM2  LAW    -1    /SAVE SUBSCRIPT *12JAH9*
6343          10205 R 105040 R          CALL   SAVE    /INDICATOR *12JAH9*
6344          10206 R 200213 R          LAC    OFLAG   /WORD *12JAH9*
6345          10207 R 105242 R          CALL   GETKEY  /PUTS KEY IN VAL *12JAH9*
6346          10210 R 000003 A          ERR3
6347          10211 R 777777 A          LAW    -1    /RESTORE *12JAH9*
6348          10212 R 105103 R          CALL   UNSAVE  /SUBSCRIPT *12JAH9*
6349          10213 R 040213 R          DAC    OFLAG   /INDICATOR *12JAH9*
6350          /TRY OPEN KEY
6351          10214 R 105272 R          GSYM21 CALL   KEYCHK
6352          10215 R 220214 R          LAC*   M        /REQUIRED KEY
6353          10216 R 540207 R          SAD    VAL     /USER SUPPLIED KEY
6354          10217 R 741200 A          SNA
6355          10220 R 610231 R          JMP    GSYM0    /*12JAH12* OK - MATCHES OPEN KEY
6356          /TRY READ KEY
6357          10221 R 200214 R          LAC    M
6358          10222 R 723777 A          AAC    -1
6359          10223 R 040214 R          DAC    M        /POINTER TO READ KEY
6360          10224 R 105272 R          CALL   KEYCHK
6361          10225 R 220214 R          LAC*   M        /REQUIRED KEY
6362          10226 R 540207 R          SAD    VAL     /USER SUPPLIED KEY

```

```

6363      10227 R 740200 A          SZA
6364      10230 R 000032 A          ERR32          /KEYS DO NOT MATCH
6365                                     /#####POPSTACK #####
6366      10231 R 777777 A          GSYMO   LAW    -1          /*12JAH12*
6367      10232 R 105103 R          CALL    UNSAVE
6368      10233 R 040207 R          DAC     VAL
6369                                     /#####          #####
6370      10234 R 104645 R          CALL    OCHAR          / CHAR
6371      10235 R 553753 R          GSYM1   SAD    (50          /C
6372      10236 R 751001 A          SKP!CLC          /CLC PLACES -1 INTO ACCUM FOR SAVE COUNT
6373      10237 R 610354 R          JMP     GSYM4
6374                                     /##### SAVE ON STACK #####
6375      10240 R 105040 R          CALL    SAVE
6376      10241 R 200177 R          LAC     SYM
6377                                     /#####          #####
6378      /
6379      /
6380      /##### SAVE ON STACK #####
6381      10242 R 777767 A          GSCRIP  LAW    -11          /*12JAH9*
6382      10243 R 105040 R          CALL    SAVE
6383      10244 R 207163 R          LAC     EVAL
6384      10245 R 200242 R          LAC     BKCHAR
6385      10246 R 200216 R          LAC     TERM
6386      10247 R 200215 R          LAC     SUM
6387      10250 R 200207 R          LAC     VAL
6388      10251 R 200202 R          LAC     TEMP
6389      10252 R 207107 R          LAC     SETIO
6390      10253 R 200270 R          LAC     GSYM
6391      10254 R 200213 R          LAC     OFLAG          /*12JAH9*
6392      10255 R 744000 A          CLL
6393      10256 R 107163 R          CALL    EVAL          /EVALUATE SUBSCRIPT (WHICH MAY USE GLOBALS)
6394      10257 R 040200 R          DAC     SCRIP
6395      10260 R 777767 A          LAW    -11          /*12JAH9*
6396      10261 R 105103 R          CALL    UNSAVE          /*12JAH9*
6397      10262 R 040213 R          DAC     OFLAG          /*12JAH9*
6398      10263 R 040270 R          DAC     GSYM          /*12JAH9*
6399      10264 R 047107 R          DAC     SETIO          /*12JAH9*
6400      10265 R 040202 R          DAC     TEMP          /*12JAH9*
6401      10266 R 040207 R          DAC     VAL          /*12JAH9*
6402      10267 R 040215 R          DAC     SUM          /*12JAH9*
6403      10270 R 040216 R          DAC     TERM          /*12JAH9*
6404      10271 R 040242 R          DAC     BKCHAR          /*12JAH9*
6405      10272 R 047163 R          DAC     EVAL          /*12JAH9*
6406      /
6407      10273 R 200200 R          LAC     SCRIP          /FETCH SUBSCRIPT *12JAH9*
6408      10274 R 513661 R          AND    (700000          /<0 OR > 327.67 ?
6409      10275 R 741200 A          SNA
6410      10276 R 610332 R          JMP     GScri1          /NO, PROCEED AS NORMAL
6411      10277 R 104645 R          CALL    OCHAR          /LAST *12JAH9*
6412      10300 R 553754 R          SAD    (51          /SUBSCRIPT ? ) *12JAH9*
6413      10301 R 741000 A          SKP
6414      10302 R 000006 A          ERR6          /NO, RANGE ERROR *12JAH9*

```

```

6415      10303 R 200270 R      LAC      GSYM      /WHERE DID WE ENTER XGSYM FROM? *12JAH9*
6416      10304 R 513657 R      AND      (77777      /MASK ADDRESS BITS *12JAH9*
6417      10305 R 554112 E      SAD      HFGL01      /*12JAH12* FROM SHIGH
6418      10306 R 610330 R      JMP      GSYM10      /YES *12JAH9*
6419      10307 R 554146 E      SAD      QF7      /*12JAH12* FROM SQUERY?
6420      10310 R 751001 A      SKP!CLC      /YES *12JAH9*
6421      10311 R 000006 A      ERR6      /NO, RANGE ERROR *12JAH9*
6422      10312 R 105040 R      CALL     SAVE      /STACK SUBSCRIPT START PTR INFO *12JAH9*
6423      10313 R 207107 R      LAC      SETIO      /*12JAH9*
6424      10314 R 104660 R      CALL     OCHARI     /FETCH NEXT CHAR *12JAH9*
6425      10315 R 200213 R      LAC      OFLAG      /*12JAH9*
6426      10316 R 741300 A      SPA!SNA      /FIRST SUBSCRIPT? *12JAH9*
6427      10317 R 634136 E      JMP*     QF2      /*12JAH12* YES, NOT NAKED
6428      10320 R 513672 R      AND      (177777      /NAKED FIRST SUBSCRIPT? *12JAH9*
6429      10321 R 740200 A      SZA      /*12JAH9*
6430      10322 R 634140 E      JMP*     QF5      /*12JAH12* NO, JOIN SO
6431      10323 R 777776 A      LAW      -2      /YES *12JAH9*
6432      10324 R 105103 R      CALL     UNSAVE     /*12JAH9*
6433      10325 R 047107 R      DAC      SETIO      /*12JAH9*
6434      10326 R 040177 R      DAC      SYM      /*12JAH9*
6435      10327 R 634147 E      JMP*     QF8      /*12JAH12* RETURN TO $Q
6436      10330 R 777777 A      GSYM10  LAW      -1      /SET TO NEGATIVE SUBSCRIPT *12JAH9*
6437      10331 R 040200 R      DAC      SCRIP      /FOR $HIGH *12JAH9*
6438      /
6439      /
6440      /##### SAVE ON STACK #####
6441      10332 R 777777 A      GSCR11  LAW      -1      /*12JAH9*
6442      10333 R 105040 R      CALL     SAVE
6443      10334 R 200200 R      LAC      SCRIP      /PUT SUBSCRIPT INTO STACK
6444      /#####
6445      10335 R 440213 R      ISZ      OFLAG      /INCR # OF SUBSCRIPTS *12JAH9*
6446      10336 R 104645 R      CALL     OCHAR
6447      10337 R 553767 R      SAD      (54      /,
6448      10340 R 610242 R      JMP      GSCRIP      /MORE
6449      10341 R 553754 R      SAD      (51      /)
6450      10342 R 751001 A      SKP!CLC      /CLC PLACES -1 INTO ACCUM FOR SAVE
6451      10343 R 000003 A      ERR3      /"SYNTAX"
6452      /##### SAVE ON STACK #####
6453      10344 R 105040 R      CALL     SAVE
6454      10345 R 207107 R      LAC      SETIO      /OLD STACK POINTER
6455      10346 R 104660 R      CALL     OCHARI     /RETURN WITH NEXT CHAR
6456      10347 R 620270 R      JMP*     GSYM      /DONE
6457      /
6458      10350 R 104645 R      GSYM3   CALL     OCHAR
6459      10351 R 553724 R      SAD      (33      /LEFT BRACKET
6460      10352 R 610204 R      JMP      GSYM2      /HE INSISTS ON A KEY CHECK
6461      10353 R 610231 R      JMP      GSYM0      /*12JAH12* NO KEY GIVEN, BUT HE'S OK
6462      10354 R 553754 R      GSYM4   SAD      (51      /)
6463      10355 R 610363 R      JMP      GSYM5
6464      10356 R 200270 R      LAC      GSYM      /UNSUBSCRIPTED KILL ^GLD
6465      10357 R 513657 R      AND      (77777      /*12JAH12*
6466      10360 R 554007 R      SAD      (GKILL2

```

6467	10361	R	610402	R		JMP	GSYM6	/YES
6468	10362	R	000003	A		ERR3		/NO, "SYNTAX"
6469	10363	R	200270	R	GSYM5	LAC	GSYM	/FIRST LEVEL SO CALL?
6470	10364	R	513657	R		AND	(77777	/*12JAH12*
6471	10365	R	554140	E		SAD	OF5	/*12JAH12*
6472	10366	R	741000	A		SKP		
6473	10367	R	000003	A		ERR3		/NO, "SYNTAX"
6474	10370	R	777776	A		LAW	-2	/YES
6475	10371	R	105040	R		CALL	SAVE	
6476	10372	R	200177	R		LAC	SYM	
6477	10373	R	207107	R		LAC	SETIO	
6478	10374	R	634136	E		JMP*	OF2	/*12JAH12* GO FETCH 1ST GLOBAL BLK
6479	10375	R	652000	A	GSYM11	LMQ		/SAVE CHAR *12JAH9*
6480	10376	R	213663	R		LAC	(200000	/INDICATE NAKED *12JAH9*
6481	10377	R	040213	R		DAC	OFLAG	/*12JAH9*
6482	10400	R	641002	A		LACQ		/RETRIEVE CHAR *12JAH9*
6483	10401	R	610235	R		JMP	GSYM1	/*12JAH9*
6484	10402	R	777775	A	GSYM6	LAW	-3	/STACK PSEUDO SUBSCRIPT
6485	10403	R	105040	R		CALL	SAVE	
6486	10404	R	200177	R		LAC	SYM	
6487	10405	R	750000	A		CLA		/FAKE SUBSCRIPT
6488	10406	R	207107	R		LAC	SETIO	
6489	10407	R	610110	R		JMP	GKILL3	/ENTER GLOBAL ROUTINES
6490						.EJECT		

```

6491          /GLOBAL SUBROUTINE
6492          /
6493          /GET, PUT, AND KILL FUNCTION
6494          /
6495          / GLFNCT SAVES FUNCTION
6496          /     GET - GLFNCT=0
6497          /     PUT - GLFNCT=1
6498          /     JOIN PUT - GLFNCT=2
6499          /     KILL - GLFNCT=77777 (OR ANYTHING NEG.)
6500          /
6501          / CALLS:
6502          /   GET CLA
6503          /   CALL   GLOBAL
6504          /   PUT AC=400000 (200000 IF JOIN PUT)
6505          /   CALL   GLOBAL
6506          /   KILL CLC
6507          /   CALL   GLOBAL
6508          /
6509          / RETURNS:
6510          /   GET GOOD  SUBSCRIPT IN MQ
6511          /             LINK SET IF STRING
6512          /             LAC* R17 GETS DATA
6513          /             LAC* 15 GETS SUBSCRIPT
6514          /   BAD     CLEAR LINK
6515          /             MQ = 0 IF NODE NOT DEFINED
6516          /             MQ = -0 IF POINTER (BUT NO DATA) IS THERE
6517          /   PUT SUB IN MQ
6518          /             LAC* 15 GETS SUBSCRIPT
6519          /             NEW DATA AT BOTTOM OF BLOCK
6520          /   KILL    CLEAR MQ IF THERE WERE LOWER LEVELS
6521          /             SUB IN MQ IF NO LOWER LEVELS
6522          /
6523          /
6524          10410 R 103641 R  XGLOBAL CALL   PRODA          /GET DISK
6525          .IFPNZ  FMONEY
6526          10411 R 553662 R  SAD      (400000      /PUT?
6527          10412 R 610564 R  JMP      FPERRC      /YES
6528          10413 R 553663 R  SAD      (200000      /PUT?
6529          10414 R 610564 R  JMP      FPERRC      /YES
6530          .ENDC
6531          10415 R 051327 R  GBACK1  DAC      GLFNCT
6532          10416 R 777776 A  LAW      -2
6533          10417 R 340166 R  TAD      SPOINT          /COMPLEMENTS LINK
6534          10420 R 051321 R  DAC      GDONE          /POINTER TO LAST SUB
6535          10421 R 040166 R  DAC      SPOINT
6536          10422 R 440166 R  ISZ      SPOINT
6537          10423 R 220166 R  LAC*     SPOINT
6538          10424 R 340165 R  TAD      STACK          /ADD OFFSET
6539          10425 R 040166 R  DAC      SPOINT          /RESTORE STACK POINTER
6540          10426 R 051326 R  DAC      GINDEX         /POINTS TO GLOBAL 'NAME' IN STACK
6541          10427 R 200166 R  LAC      SPOINT          /CALCULATE *12-3*
6542          10430 R 740031 A  TCA      /EFFECTIVE *12-3*

```

6543	10431	R	351321	R	TAD	GDONE	/POP OF STACK *12-3*
6544	10432	R	723002	A	AAC	2	/*12-3*
6545	10433	R	040017	A	DAC	R17	/R17 USED AS TEMP STORAGE *12-3*
6546	10434	R	744010	A	RCL		/MULTIPLY BY *12-3*
6547	10435	R	300017	A	ADD	R17	/3 CHARS PER WORD *12-3*
6548	10436	R	360146	R	TAD*	SVAR	/ALTER %S *12-3*
6549	10437	R	060146	R	DAC*	SVAR	/ACCORDINGLY *12-3*
6550	10440	R	231326	R	LAC*	GINDEX	/GET NAME
6551	10441	R	740200	A	SZA		/NAKED?
6552	10442	R	040254	R	DAC	GSYMBOL	/NO
6553	10443	R	200254	R	LAC	GSYMBOL	/IS THERE A NAME
6554	10444	R	741200	A	SNA		
6555	10445	R	000013	A	ERR13		/NO, "GLOBE"
6556	10446	R	740030	A	IAC		/INCREMENT POINTER INTO GLOBAL TABLE
6557	10447	R	040017	A	DAC	R17	
6558	10450	R	211327	R	LAC	GLFNCT	/FETCH INPUT CODE *12-3*
6559	10451	R	745200	A	SNA!CLL		/CLEAR LINK = GET *12-3*
6560	10452	R	744002	A	STL		/*12-3*
6561	10453	R	220017	A	LAC*	R17	/GET WORD 3 OF THE GLOBAL
6562	10454	R	513702	R	AND	(77	/OWNERSHIP BITS
6563	10455	R	240655	R	XOR	RUNNMB	
6564	10456	R	744600	A	SZA!SNL!CLL		/LINK CLEAR IF KILL OR PUT
6565	10457	R	000020	A	ERR20		/"CLOSE"
6566	10460	R	231326	R	LAC*	GINDEX	/GET NAME
6567	10461	R	745200	A	SNA!CLL		/NAKED ACCESS ?
6568	10462	R	610600	R	JMP	GNOSYM	/YES
6569	10463	R	040017	A	DAC	R17	/NO
6570	10464	R	040255	R	DAC	GHDOLD	/SET UP POINTERS
6571	10465	R	220017	A	LAC*	R17	
6572	10466	R	040253	R	DAC	GHEAD	
6573	10467	R	220017	A	LAC*	R17	
6574	10470	R	653603	A	LMQ!CLAC!LLS+3		
6575	10471	R	040247	R	DAC	GUNIT	
6576	10472	R	200253	R	LAC	GHEAD	
6577	10473	R	040253	R	GBACK2	DAC	GHEAD
6578	10474	R	451326	R	ISZ	GINDEX	/NEXT SUBSCRIPT
6579	10475	R	140252	R	GBACK3	DZM	GHI
6580	10476	R	652000	A	LMQ		
6581	10477	R	200272	R	LAC	GLOBAL	
6582	10500	R	513657	R	AND	(77777	/*12JAH12* FIRST LEVEL \$O CALL?
6583	10501	R	554137	E	SAD	OF3	/*12JAH12*
6584	10502	R	620272	R	JMP*	GLOBAL	/YES, EXIT
6585	10503	R	641002	A	LACQ		/NO
6586	10504	R	111331	R	GBACK4	JMS	GDISK
6587	10505	R	540704	R	GBACK5	SAD	DSKBF
6588	10506	R	610572	R	JMP	GOOFF	/CHECK FOR OFFSET=0 (GEND IN AC)
6589	10507	R	200272	R	LAC	GLOBAL	/0 OFFSET, GO SEE WHAT FUNCTION
6590	10510	R	513657	R	AND	(77777	/UNSUBSCRIPTED KILL?
6591	10511	R	554010	R	SAD	(GKILL4	/*12JAH12*
6592	10512	R	741000	A	SKP		/YES
6593	10513	R	610535	R	JMP	BACK5A	/NO, PROCESS AS NORMAL
6594	10514	R	111426	R	BACK5B	JMS	GLSET
							/RESET R17 ON XFER TO HERE

```

6595      10515 R 200017 A          LAC      R17
6596      10516 R 551322 R          SAD      GEND      /END OF VALID DATA IN BLOCK
6597      10517 R 610525 R          JMP      BACK5C   /YES
6598      10520 R 220017 A          LAC*     R17      /FETCH SUBSCRIPT
6599      10521 R 513657 R          AND      (77777
6600      10522 R 071326 R          DAC*     GINDEX    /SPECIFY IT AS SUBSCRIPT TO LOOK FOR
6601      10523 R 111426 R          JMS      GLSET    /RESET R17
6602      10524 R 610540 R          JMP      GBACK6   /RETURN TO NORMAL PROCESSING
6603      10525 R 220704 R          BACK5C  LAC*     DSKBFF   /ANY CHAINED BLOCKS?
6604      10526 R 741200 A          SNA
6605      10527 R 611311 R          JMP      GEXIT    /NO, DONE
6606      10530 R 074105 E          DAC*     GTMBUF   /YES, PREPARE TO DETACH ALL THOSE THAT FOLLOW
6607      10531 R 040172 R          DAC      WRTSW    /MAKE SURE THIS BLOCK IS UPDATED
6608      10532 R 160704 R          DZM*    DSKBFF   /NOW THIS IS ONLY BLK AT LEVEL
6609      10533 R 750000 A          CLA
6610      10534 R 611225 R          JMP      GTHRED   /GO DELETE REST OF LEVEL
6611      10535 R 231326 R          BACK5A  LAC*     GINDEX    /GET SUBSCRIPT OUT OF STACK
6612      10536 R 745100 A          SPA!CLL
6613      10537 R 610603 R          JMP      GHIGHO   /SH COMPLEMENTS LAST SUBSCRIPT IN STACK *12JAH9*
6614      /
6615      10540 R 200017 A          GBACK6  LAC      R17
6616      10541 R 551322 R          GBACK7  SAD      GEND      /END OF VALID DATA IN BLOCK ?
6617      10542 R 610651 R          JMP      GCONT    /END OF DATA, NO MATCH IN PRESENT BLOCK
6618      10543 R 220017 A          LAC*     R17      /GET SUBSCRIPT FROM BLOCK
6619      10544 R 652000 A          LMQ
6620      10545 R 513657 R          AND      (77777
6621      10546 R 571326 R          SAD*     GINDEX    /COMPARE
6622      10547 R 610700 R          JMP      GNEXT    /MATCH IN PRESENT BLOCK
6623      10550 R 641002 A          LACQ
6624      10551 R 745110 A          SPA!RCL   /GET BACK SUBSCRIPT WITH PNS BITS...
6625      10552 R 440017 A          ISZ      R17      /...AND SKIP OVER POINTER AND/OR VALUE
6626      10553 R 745110 A          SPA!RCL
6627      10554 R 440017 A          ISZ      R17
6628      10555 R 744100 A          SMA!CLL   /STRING PRESENT IF MINUS
6629      10556 R 610540 R          JMP      GBACK6
6630      10557 R 220017 A          LAC*     R17
6631      10560 R 653606 A          LMQ!CLAC!LLS+6 /ADD OFFSET TO POINTER
6632      10561 R 300017 A          ADD      R17
6633      10562 R 040017 A          DAC      R17
6634      10563 R 610541 R          JMP      GBACK7   /GO LOOK AT NEXT SUB
6635      /
6636      .IFPNZ  FMONEY
6637      10564 R 652000 A          FPERRC  LMQ
6638      10565 R 750030 A          CLA!IAC
6639      10566 R 540211 R          SAD      VALF    /FLOATING POINT?
6640      10567 R 000017 A          ERR17   /YES, F.P. CANNOT BE PUT IN GLOBAL
6641      10570 R 641002 A          LACQ
6642      10571 R 610415 R          JMP      GBACK1
6643      .ENDC
6644      /
6645      /BLOCK READ HAS 0 OFFSET (FROM FREE LIST)
6646      /

```

```

6647      10572 R 140172 R      GOOFF  DZM      WRTSW      /DON'T WRITE BLOCK
6648      10573 R 211327 R      LAC      GLFNCT    /CHECK FUNCTION
6649      10574 R 740200 A      SZA
6650      10575 R 000021 A      ERR21      /PUT OR KILL - "IODSK"
6651      10576 R 160704 R      DZM*      DSKBFF      /ZORCH CONTINUATION POINTER
6652      10577 R 610651 R      JMP      GCONT      /TREAT AS LAST BLOCK IN CHAIN
6653      /
6654      /
6655      /NAKED REFERENCE
6656      /
6657      10600 R 451326 R      GNOSYM  ISZ      GINDEX
6658      10601 R 111426 R      JMS      GLSET
6659      10602 R 610505 R      JMP      GBACK5
6660      /
6661      /
6662      /SEARCH BLOCK IN CORE FOR NEXT HIGHER SUBSCRIPT (SH)
6663      /
6664      10603 R 231326 R      GHIGH0  LAC*      GINDEX      /FETCH LAST SUBSCRIPT? *12JAH9*
6665      10604 R 553662 R      SAD      (400000    /SH WNAT VERY FIRST SUBSCRIPT (EVEN IF ZERO)? *12JAH9*
6666      10605 R 750000 A      CLA
6667      10606 R 040016 A      DAC      R16      /YES *12JAH9*
6668      10607 R 200017 A      GHIGH   LAC      R17      /*12JAH9*
6669      10610 R 551322 R      GHIGH1  SAD      GEND
6670      10611 R 610651 R      JMP      GCONT
6671      10612 R 220017 A      LAC*      R17
6672      10613 R 652000 A      LMQ
6673      10614 R 513657 R      AND      (77777
6674      10615 R 340016 A      TAD      R16      /*12JAH9*
6675      10616 R 740101 A      SMA!CMA
6676      10617 R 610634 R      JMP      GHIGH3
6677      10620 R 641002 A      GHIGH2  LACQ      /SKIP TO NEXT SUBSCRIPT
6678      10621 R 745110 A      SPA!RCL
6679      10622 R 440017 A      ISZ      R17
6680      10623 R 745110 A      SPA!RCL
6681      10624 R 440017 A      ISZ      R17
6682      10625 R 744100 A      SMA!CLL
6683      10626 R 610607 R      JMP      GHIGH
6684      10627 R 220017 A      LAC*      R17
6685      10630 R 653606 A      LMQ!CLAC!LLS+6
6686      10631 R 300017 A      ADD      R17
6687      10632 R 040017 A      DAC      R17
6688      10633 R 610610 R      JMP      GHIGH1
6689      /
6690      10634 R 553741 R      GHIGH3  SAD      (-1      /SUBSCRIPT EXACTLY .01 HIGHER? *12JAH9*
6691      10635 R 610645 R      JMP      GHIGH4
6692      10636 R 340016 A      TAD      R16      /*12JAH9*
6693      10637 R 300217 R      ADD      ACTEMP
6694      10640 P 741101 A      SPA!CMA
6695      10641 R 610620 R      JMP      GHIGH2
6696      10642 R 300217 R      ADD      ACTEMP
6697      10643 R 040217 R      DAC      ACTEMP      /NEXT HIGHEST SO FAR
6698      10644 R 610620 R      JMP      GHIGH2

```

```

6699      10645 R 200016 A    GHIGH4  LAC      R16          /*12JAH9*
6700      10646 R 740031 A          CMA!IAC
6701      10647 R 040217 R          DAC      ACTEMP
6702      10650 R 611304 R          JMP      GXTBAD
6703      /
6704      /
6705      /HAVEN'T FOUND SUBSCRIPT YET - CONTINUE ALONG CHAIN IF POSSIBLE
6706      /NAKED REFERENCES SEARCH FIRST THE BLOCK IN CORE, THEN
6707      /FOLLOWS THE CHAIN. THIS TAKES ADVANTAGE OF THE CHAIN SPACING.
6708      /
6709      10651 R 220704 R    GCONT   LAC*     DSKBFF
6710      10652 R 751201 A          SNA!CLA!CMA      /IS THERE A CONTINUATION BLOCK?
6711      10653 R 610672 R          JMP      NKDCHK   /NO, REQUESTED SUBSCRIPT HAS NOT BEEN FOUND. CHK IF NAKED
6712      /
6713      10654 R 251322 R          XOR      GEND     /YES - GET -GEND IN AC
6714      10655 R 300704 R          ADD      DSKBFF   /NOW HAVE -(WORDS USED IN BLOCK)
6715      10656 R 652004 A          LMQ!CMQ
6716      10657 R 300252 R          ADD      GHI      /COMPARE TO LEAST USED SO FAR
6717      10660 R 751100 A          SPA!CLA
6718      10661 R 200252 R          LAC      GHI      /GHI=0 MEANS ALWAYS WANT THIS BLOCK
6719      10662 R 740200 A          SZA
6720      10663 R 610670 R          JMP      GCONT1   /*12JAH12*
6721      10664 R 641002 A          LACQ
6722      10665 R 040252 R          DAC      GHI      /THIS BLOCK HAS MORE SPACE
6723      10666 R 200171 R          LAC      QBPNT   /.. .SO REMEMBER HOW MUCH...
6724      10667 R 040251 R          DAC      GBLKHI  /...AND WHICH BLOCK IT IS
6725      10670 R 220704 R    GCONT1  LAC*     DSKBFF   /*12JAH12* GO GET CONT BLK
6726      10671 R 610504 R          JMP      GBACK4
6727      /
6728      10672 R 220166 R    NKDCHK  LAC*     SPOINT   /CHK FOR NAKED ACCESS
6729      10673 R 740201 A          SZA!CMA
6730      10674 R 611130 R          JMP      GUND5    /NOT NAKED SO DIDN'T FIND THE SUBSCRIPT
6731      10675 R 060166 R          DAC*     SPOINT   /WAS NAKED ACCESS, SET NAME TO -1 SO THAT NEXT TIME THRU EXITS
6732      10676 R 200253 R          LAC      GHEAD   /GET HEAD BLOCK OF THIS LEVEL
6733      10677 R 610475 R          JMP      GBACK3   /AND BEGIN SEARCH AT TDP
6734      /
6735      /FOUND SUBSCRIPT AT THIS LEVEL.
6736      /
6737      10700 R 211326 R    GNEXT   LAC      GINDEX
6738      10701 R 551321 R          SAD      GDONE   /MORE SUBS?
6739      10702 R 610712 R          JMP      GFOUND   /NO MORE SUBSCRIPTS IN THE STACK
6740      10703 R 641002 A          LACQ
6741      10704 R 740100 A          SMA
6742      10705 R 611142 R          JMP      GUND6    /POINTER PRESENT?
6743      10706 R 200253 R          LAC      GHEAD   /NO POINTER TO NEXT LEVEL
6744      10707 R 040255 R          DAC      GHDOLD   /SAVE HEAD ADDRESS AND DROP A LEVEL
6745      10710 R 220017 A          LAC*     R17
6746      10711 R 610473 R          JMP      GBACK2
6747      .EJECT

```

```

6748          /FOUND FINAL SUBSCRIPT. ACTION:
6749          / GET DONE, EXIT WITH MQ, R15, AND R17 SET UP.
6750          / PUT SAVE POINTER (IF IT EXISTS) AND SUBSCRIPT (IN STACK).
6751          / DELETE OLD DATA BY COPYING REST OF BLOCK UP OVER IT. FIND
6752          / SPACE (NOT NEC. IN THIS BLOCK) AND COPY IN SUB, POINTER,
6753          / AND NEW DATA (FROM VAL OR BUFF2).
6754          / KILL SAME AS PUT, BUT GOES TO GTHRED INSTEAD OF ADDING DATA.
6755          /
6756          10712 R 211327 R GFOUND LAC GLFNCT
6757          10713 R 755201 A          SNA!CLL!CLC
6758          10714 R 744002 A          STL /YES
6759          10715 R 340017 A          TAD R17 /COMPLIMENTS LINK
6760          10716 R 040016 A          DAC R16 /DECREMENT POINTER AND SAVE
6761          10717 R 040015 A          DAC R15
6762          10720 R 641002 A          LACQ
6763          10721 R 740400 A          SNL
6764          10722 R 611075 R          JMP GFND7 /GET - DONE
6765          10723 R 514011 R          AND (477777 /SUBSCRIPT WITH P BIT
6766          10724 R 071326 R          DAC* GINDEX /FORGET NS BITS
6767          10725 R 751100 A          SPA!CLA /SAVE 0...
6768          10726 R 220017 A          LAC* R17 /...OR POINTER...
6769          10727 R 074105 E          DAC* GTMBUF /...IN FIRST WORD OF GTMBUF
6770          10730 R 641002 A          LACQ /WHOLE SUBSCRIPT AGAIN
6771          10731 R 742010 A          RTL /N BIT IN LINK
6772          10732 R 741410 A          SZL!RAL
6773          10733 P 611001 R          JMP GFND0A /*12JAH11* IS NUMERIC
6774          10734 R 744400 A          SNL!CLL
6775          10735 R 610773 R          JMP GFND0 /*12JAH11* POINTER ONLY
6776          10736 R 220017 A          LAC* R17 /S - GET OFFSET
6777          10737 R 653606 A          LMQ!CLAC!LLS+6
6778          10740 R 051325 R          DAC GNOWDS /SAVE OFFSET
6779          10741 R 211327 R          LAC GLFNCT
6780          10742 R 751101 A          SPA!CLA!CMA /KILL?
6781          10743 R 610770 R          JMP GFNDA /YES, NO SHORT CUT
6782          10744 R 540207 R          SAD VAL /IS REPLACEMENT STR OR NUM?
6783          10745 R 610747 R          JMP GFNDB /STRING
6784          10746 R 610770 R          JMP GFNDA /INSERT NUMERIC VALUE
6785          10747 R 220133 R          GFNDB LAC* BUFF2 /FETCH REPLACEMENT
6786          10750 R 653606 A          LMQ!CLAC!LLS+6 /OFFSET
6787          10751 P 551325 R          SAD GNOWDS /SAME AS OLD OFFSET?
6788          10752 R 741001 A          SKP!CMA /YES, SET UP COUNTER
6789          10753 R 610770 R          JMP GFNDA /NO, TAKE 'LONG' CUT
6790          10754 R 051325 R          DAC GNOWDS
6791          10755 R 777777 A          LAW -1 /BACKUP
6792          10756 R 340133 R          TAD BUFF2 /REPLACEMENT PTR
6793          10757 R 040016 A          DAC R16
6794          10760 R 777777 A          LAW -1 /BACKUP
6795          10761 R 340017 A          TAD R17 /OVERLAY PTR
6796          10762 R 040017 A          DAC R17
6797          10763 R 220016 A          LAC* R16 /INSERT NEW STRING
6798          10764 R 060017 A          DAC* R17
6799          10765 R 451325 R          ISZ GNOWDS

```

6800	10766	R	610763	R		JMP	.-3	
6801	10767	R	611013	R		JMP	GFND0B	
6802	10770	R	211325	R	GFNDA	LAC	GNOWDS	/RETRIEVE OFFSET
6803	10771	R	300017	A		ADD	R17	
6804	10772	R	040017	A		DAC	R17	
6805	10773	R	200017	A	GFND0	LAC	R17	/COPY EVERYTHING UP OVER OLD
6806	10774	R	551322	R		SAD	GEND	
6807	10775	R	611016	R		JMP	GFND1	/DONE
6808	10776	R	220017	A		LAC*	R17	
6809	10777	R	060016	A		DAC*	R16	
6810	11000	R	610773	R		JMP	GFND0	
6811					/			
6812	11001	R	440017	A	GFND0A	ISZ	R17	/*12JAH11*
6813	11002	R	211327	R		LAC	GLFNCT	/*12JAH11*
6814	11003	R	751101	A		SPA!CLA!CMA		/KILL?
6815	11004	R	610773	R		JMP	GFND0	/YES, NO SHORTCUT
6816	11005	R	540207	R		SAD	VAL	/STRING OR NUMERIC?
6817	11006	R	610773	R		JMP	GFND0	/STRING
6818	11007	R	340017	A		TAD	R17	/NUMERIC, BACKUP PT
6819	11010	R	040017	A		DAC	R17	
6820	11011	R	200207	R		LAC	VAL	/INSERT NEW NUMERIC VALUE
6821	11012	R	060017	A		DAC*	R17	
6822	11013	R	777777	A	GFND0B	LAW	-1	
6823	11014	R	040172	R		DAC	WRTSW	/MAKE SURE BLOCK WRITTEN
6824	11015	R	611054	R		JMP	GFND4A	
6825					/			
6826	11016	R	200016	A	GFND1	LAC	R16	
6827	11017	R	111412	R		JMS	GOFF	/NEW OFFSET
6828	11020	R	211327	R		LAC	GLFNCT	
6829	11021	R	755100	A		SPA!CLL!CLA		/FUNCTION?
6830	11022	R	611225	R		JMP	GTHRED	/KILL
6831	11023	R	213705	R	GFND2	LAC	(2	/PUT
6832	11024	R	051325	R		DAC	GNOWDS	/NEED AT LEAST 2 WORDS
6833	11025	R	750001	A		CLA!CMA		
6834	11026	R	540207	R		SAD	VAL	/DATA TYPE?
6835	11027	R	611057	R		JMP	GFND5	/STRING
6836	11030	R	213663	R		LAC	(200000	/NUMERIC
6837	11031	R	271326	R	GFND3	XOR*	GINDEX	/ADD DATA TYPE BIT (N OR S) TO SUB...
6838	11032	R	071326	R		DAC*	GINDEX	/...AND STORE IN STACK
6839	11033	R	741100	A		SPA		
6840	11034	R	451325	R		ISZ	GNOWDS	/1 MORE FOR POINTER
6841	11035	R	111436	R		JMS	GSPACE	/FIND SPACE
6842	11036	R	200017	A		LAC	R17	
6843	11037	R	040015	A		DAC	R15	
6844	11040	R	231326	R		LAC*	GINDEX	
6845	11041	R	060017	A		DAC*	R17	/PUT IN SUBSCRIPT
6846	11042	R	740100	A		SMA		
6847	11043	R	611046	R		JMP	GFND3A	/*12JAH12*
6848	11044	R	234105	E		LAC*	GTMBUF	/ADD POINTER IF IT EXISTS
6849	11045	R	060017	A		DAC*	R17	
6850	11046	R	200207	R	GFND3A	LAC	VAL	/*12JAH12*
6851	11047	R	553741	R		SAD	(777777	

6852	11050 R 611065 R		JMP	GFND6	/STRING
6853	11051 R 060017 A		DAC*	R17	/NUMERIC
6854	11052 R 200017 A	GFND4	LAC	R17	
6855	11053 R 111412 R		JMS	G0FF	/NEW OFFSET
6856	11054 R 231326 R	GFND4A	LAC*	GINDEX	
6857	11055 R 652000 A		LMQ		/RETURN WITH SUB IN MQ
6858	11056 R 611311 R		JMP	GEXIT	
6859		/			
6860	11057 R 220133 R	GFND5	LAC*	BUFF2	/STRING IN BUFF2
6861	11060 R 653606 A		LMQ!CLAC!LLS+6		
6862	11061 R 311325 R		ADD	GNOWDS	
6863	11062 R 051325 R		DAC	GNOWDS	/UPDATE NO. OF WORDS NEEDED
6864	11063 R 213664 R		LAC	(100000	/S BIT
6865	11064 R 611031 R		JMP	GFND3	
6866		/			
6867	11065 R 340133 R	GFND6	TAD	BUFF2	/COPY STRING INTO BLOCK
6868	11066 R 040016 A		DAC	R16	
6869	11067 R 220016 A		LAC*	R16	
6870	11070 R 060017 A		DAC*	R17	
6871	11071 R 513702 R		AND	(77	/LAST CHAR OF STRING ?
6872	11072 R 740200 A		SZA		
6873	11073 R 611067 R		JMP	.-4	/NO, SO GET ANOTHER WORD
6874	11074 R 611052 R		JMP	GFND4	/YES
6875		/			
6876	11075 R 743110 A	GFND7	SPA!RTL		/CHECK FOR POINTER, SET UP LINK
6877	11076 R 440017 A		ISZ	R17	
6878	11077 R 740502 A		SNL!SMA!CML		
6879	11100 R 670004 A		CLQ!CMQ!GAS		/-0 IN MQ, CLEAR LINK
6880	11101 R 611311 R		JMP	GEXIT	/LEAVE WITH LINK, MQ, R15, AND R17 SET
6881			.EJECT		

```

6882          /BUILD CHAIN. GO TO GFND2 TO ADD DATA.
6883          /
6884          11102 R 040253 R   GUND2  DAC    GHEAD
6885          11103 R 140252 R           DZM    GHI
6886          11104 R 111331 R           JMS    GDISK    /READ FREE LIST BLOCK
6887          11105 R 160704 R           DZM*   DSKBFF   /NO CONT BLOCK
6888          11106 R 200705 R           LAC    GBWD2
6889          11107 R 111412 R           JMS    GOFF     /OFFSET
6890          11110 R 451326 R           ISZ    GINDEX
6891          11111 R 211326 R           LAC    GINDEX
6892          11112 R 551321 R           SAD    GDONE    /LAST SUB?
6893          11113 R 611023 R           JMP    GFND2   /YES - GO STORE DATA
6894          11114 R 231326 R   GUND3  LAC*   GINDEX   /NO - ADD ANOTHER BLOCK
6895          11115 R 253662 R           XOR    (400000
6896          11116 R 060017 A           DAC*   R17      /ADD SUB WITH P BIT
6897          11117 R 134103 E           JMS*   GFREE   /*12JAH12* NO - ADD ANOTHER BLK
6898          11120 R 051323 R           DAC    GPOINT  /SAVE BLOCK ADDRESS OF NEWLY ALLOCATED BLOCK
6899          11121 R 060017 A           DAC*   R17      /POINTER TO NEW BLOCK 4 LATER
6900          11122 R 200017 A   GUND4  LAC    R17
6901          11123 R 111412 R           JMS    GOFF
6902          11124 R 200253 R           LAC    GHEAD
6903          11125 R 040255 R           DAC    GHDDLD  /SAVE HEAD ADDRESS
6904          11126 R 211323 R           LAC    GPOINT  /ADDRESS OF NEW BLOCK
6905          11127 R 611102 R           JMP    GUND2
6906          /
6907          /
6908          /UNDEFINED SUBSCRIPT AT LATER LEVEL. FIND SPACE FOR SUBSCRIPT AND
6909          /AND POINTER, GO ADD THEM.
6910          /
6911          11130 R 211327 R   GUND5  LAC    GLFNCT
6912          11131 R 745300 A           SPA!SNA!CLL  /PUT?
6913          11132 R 611304 R           JMP    GXTBAD  /NO
6914          11133 R 211326 R           LAC    GINDEX
6915          11134 R 551321 R           SAD    GDONE
6916          11135 R 611023 R           JMP    GFND2   /DEFINE FINAL VALUE
6917          11136 R 213705 R           LAC    (2
6918          11137 R 051325 R           DAC    GNOWDS  /2 WDS FOR PTR
6919          11140 R 111436 R           JMS    GSPACE
6920          11141 R 611114 R           JMP    GUND3   /GO ADD ANOTHER BLOCK
6921          /
6922          /
6923          /NO POINTER TO NEXT LEVEL (SUB IS THERE). COPY SUB AND DATA INTO
6924          /GTMBUF. ADD POINTER (TO NEW BLOCK). FIND SPACE AND COPY BACK.
6925          /
6926          11142 R 253662 R   GUND6  XOR    (400000
6927          11143 R 074105 E           DAC*   GTMBUF   /STORE SUB WITH P BIT
6928          11144 R 211327 R           LAC    GLFNCT
6929          11145 R 755301 A           SPA!SNA!CLC!CLL /PUT?
6930          11146 R 611304 R           JMP    GXTBAD  /NO
6931          11147 R 340017 A           TAD    R17    /COPY SUB AND VALUE INTO GTMBUF
6932          11150 R 051324 R           DAC    GTEMP
6933          11151 R 214106 E           LAC    GTMB.1

```

6934	11152	R	040016	A		DAC	R16	
6935	11153	R	234105	E		LAC*	GTMBUF	
6936	11154	R	742010	A		RTL		
6937	11155	R	744400	A		SNL!CLL		
6938	11156	R	611217	R		JMP	GUND9	/STRING
6939	11157	R	220017	A		LAC*	R17	/NUMERIC
6940	11160	R	060016	A		DAC*	R16	
6941	11161	R	214105	E	GUND7	LAC	GTMBUF	
6942	11162	R	740031	A		TCA		
6943	11163	R	300016	A		ADD	R16	
6944	11164	R	051325	R		DAC	GNOWDS	/SPACE NEEDED
6945	11165	R	211324	R		LAC	GTEMP	/COPY REST OF BLOCK OVER DELETED DATA
6946	11166	R	040016	A		DAC	R16	
6947	11167	R	200017	A		LAC	R17	
6948	11170	R	551322	R		SAD	GEND	
6949	11171	R	611175	R		JMP	GUND8	
6950	11172	R	220017	A		LAC*	R17	
6951	11173	R	060016	A		DAC*	R16	
6952	11174	R	611167	R		JMP	.-5	
6953					/			
6954	11175	R	200016	A	GUND8	LAC	R16	/OFFSET
6955	11176	R	111412	R		JMS	GOFF	
6956	11177	R	111436	R		JMS	GSPACE	/FIND BLOCK WITH SPACE IN IT
6957	11200	R	134103	E		JMS*	GFREE	/*12JAH12* & A NEW BLK FOR LOWER LEVEL
6958	11201	R	051323	R		DAC	GPOINT	/SAVE BLOCK ADDRESS
6959	11202	R	074106	E		DAC*	GTMB.1	/ POINTER IN GTMBUF IMAGE
6960	11203	R	211325	R		LAC	GNOWDS	
6961	11204	R	740001	A		CMA		
6962	11205	R	051324	R		DAC	GTEMP	
6963	11206	R	451324	R		ISZ	GTEMP	/WORD COUNTER
6964	11207	R	777777	A		LAW	-1	
6965	11210	R	354105	E		TAD	GTMBUF	
6966	11211	R	040016	A		DAC	R16	
6967	11212	R	220016	A		LAC*	R16	/COPY INTO BLOCK (NOW HAS POINTER)
6968	11213	R	060017	A		DAC*	R17	
6969	11214	R	451324	R		ISZ	GTEMP	
6970	11215	R	611212	R		JMP	.-3	
6971	11216	R	611122	R		JMP	GUND4	/GO ADD LEVELS AS NECESSARY
6972					/			
6973	11217	R	220017	A	GUND9	LAC*	R17	/COPY STRING INTO GTMBUFF
6974	11220	R	060016	A		DAC*	R16	
6975	11221	R	513702	R		AND	(77	
6976	11222	R	740200	A		SZA		
6977	11223	R	611217	R		JMP	.-4	
6978	11224	R	611161	R		JMP	GUND7	
6979						.EJECT		

```

6980          /KILL STUFF
6981          / 1) NO LOWER LEVEL - QUIT.
6982          / 2) LOWER LEVEL JOINED - DECREMENT JOIN COUNTER, BUT DON'T ADD
6983          /      TO GARBAGE TREE.
6984          / 3) LOWER LEVEL NOT JOINED - PUT KILLED TREE AT TOP OF GARBAGE
6985          /      TREE.
6986          /
6987          11225 R 574105 E   GTHRED  SAD*   GTMBUF   /MUST ARRIVE WITH 0 IN AC
6988          11226 R 611305 R           JMP     GEXITO   /0 MEANS NO POINTER - KILL IS DONE
6989          11227 R 234105 E           LAC*   GTMBUF
6990          11230 R 043744 R           DAC     STOR1    /SAVE POINTER
6991          11231 R 214100 E           LAC     GBUFF1
6992          11232 R 740030 A           IAC
6993          11233 R 044340 R           DAC     PLACE   /SECOND WORD OF BUFFER
6994          11234 R 214100 E           LAC     GBUFF1
6995          11235 R 134060 E           JMS*   DSKRDE   /*12JAH12*
6996          11236 R 200247 R           LAC     GUNIT
6997          11237 R 203744 R           LAC     STOR1   /READ POINTERED BLOCK
6998          11240 R 224340 R           LAC*   PLACE   /SECOND WORD
6999          11241 R 513646 R           AND     (777700 /CHECK POINTER
7000          11242 R 745300 A           SPA!SNA!CLL
7001          11243 R 611262 R           JMP     GTHR3   /CAN KILL IT
7002          11244 R 224340 R           LAC*   PLACE   /JOIN - DON'T PUT ON GARBAGE TREE
7003          11245 R 353646 R           TAD     (777700
7004          11246 R 064340 R           DAC*   PLACE   /DECREMENT JOIN COUNTER
7005          11247 R 214100 E           LAC     GBUFF1
7006          11250 R 134061 E           JMS*   DSKWRT   /*12JAH12* WRITE BACK BLK
7007          11251 R 200247 R           LAC     GUNIT
7008          11252 R 203744 R           LAC     STOR1
7009          11253 R 611305 R           JMP     GEXITO   /AND LEAVE
7010          /
7011          11254 R 234100 E   GTHR2  LAC*   GBUFF1
7012          11255 R 043744 R           DAC     STOR1   /GET NEXT BLOCK
7013          11256 R 214100 E           LAC     GBUFF1
7014          11257 R 134060 E           JMS*   DSKRDE   /*12JAH12*
7015          11260 R 200247 R           LAC     GUNIT
7016          11261 R 203744 R           LAC     STOR1
7017          11262 R 214101 E   GTHR3  LAC     GCHAIN   /ANY GARBAGE?
7018          11263 R 340247 R           TAD     GUNIT
7019          11264 R 051330 R           DAC     GCH     /GARB CHAIN FOR THIS SUPER UNIT
7020          11265 R 231330 R           LAC*   GCH
7021          11266 R 741200 A           SNA
7022          11267 R 611301 R           JMP     GTHR4   /NO - JUST ADD CHAIN TO GCHAIN
7023          11270 R 234100 E           LAC*   GBUFF1   /CONT BLOCK?
7024          11271 R 744200 A           SZA!CLL
7025          11272 R 611254 R           JMP     GTHR2   /YES
7026          11273 R 231330 R           LAC*   GCH
7027          11274 R 074100 E           DAC*   GBUFF1   /ADD GARBAGE TREE TO END
7028          11275 R 214100 E           LAC     GBUFF1
7029          11276 R 134061 E           JMS*   DSKWRT   /*12JAH12*
7030          11277 R 200247 R           LAC     GUNIT
7031          11300 R 203744 R           LAC     STOR1   /WRITE OUT WITH NEW CHAIN POINTER

```

7032	11301	R	234105	E	GTHR4	LAC*	GTMBUF	/PUT ORIGINAL POINTER
7033	11302	R	071330	R		DAC*	GCH	/...INTO GARBAGE POINTER
7034	11303	R	134057	E		JMS*	DMPOLD	/*12JAH12*
7035	11304	R	650000	A	GXIBAD	CLQ		/UNSUCCESSFUL 'GETS' COME HERE
7036	11305	R	200272	R	GEXIT0	LAC	GLOBAL	/UNSUBSCRIBED KILL?
7037	11306	R	513657	R		AND	(77777	/*12JAH12*
7038	11307	R	554010	R		SAD	(GKILL4	
7039	11310	R	610514	R		JMP	BACK5B	/YES, GO KILL NEXT NODE.
7040	11311	R	211327	R	GEXIT	LAC	GLFNCT	/DON'T ALLOW SWAP IF IN JOIN
7041	11312	R	553705	R		SAD	(2	/JOIN HAS MORE TO DO
7042	11313	R	620272	R		JMP*	GLOBAL	
7043	11314	R	140577	R		DZM	DSKACT	/FREE DISK
7044	11315	R	440652	R		ISZ	SWAPSW	/TIME SLICE UP?
7045	11316	R	620272	R		JMP*	GLOBAL	/NO
7046	11317	R	101533	R		JMS	HANG	/YES, LET OTHERS RUN
7047	11320	R	620272	R		JMP*	GLOBAL	
7048					/			
7049	11321	R	000000	A	GDONE	0		
7050	11322	R	000000	A	GEND	0		
7051	11323	R	000000	A	GPOINT	0		
7052	11324	R	000000	A	GTEMP	0		
7053	11325	R	000000	A	GNOWDS	0		
7054	11326	R	000000	A	GINDEX	0		
7055	11327	R	000000	A	GLFNCT	0		/FUNCTION
7056	11330	R	000000	A	GCH	0		
7057					.EJECT			

```

7058          /SUBROUTINES FOR GLOBALS
7059          /
7060          /DISK READ SUBROUTINE FOR GLOBALS.  CALL WITH DISK ADDRESS IN AC.
7061          /WRITES OLD BUFFER IF NEC.  BEFORE READING
7062          /
7063          11331 R 000000 A      GDISK  0
7064          11332 R 040653 R          DAC    NBPNT
7065          11333 R 641002 A          LACQ
7066          11334 R 051412 R          DAC    GOFF          /SAVE MQ
7067          11335 R 777775 A          LAW    -3          /TRY 3 TIMES ON CHPNT ERRORS BEFORE GIVING UP
7068          11336 R 051411 R          DAC    CHKCNT
7069          11337 R 134102 E      GDISK4  JMS*   GETNEW          /*12JAH12* GET NEW BLK
7070          11340 R 220705 R          LAC*   GBWD2          /2ND WORD IN BLOCK
7071          11341 R 513646 R          AND    (777700          /CHECK POINTER
7072          11342 R 741200 A          SNA
7073          11343 R 611370 R          JMP    GDISK2          /NO JOIN COUNTER OR CHECK POINTER
7074          11344 R 740100 A          SMA
7075          11345 R 611405 R          JMP    GDISK3          /POS MEANS JOIN COUNTER
7076          11346 R 640503 A          LRS    3          /NEG MEANS CHECK POINTER
7077          11347 R 074103 E          DAC*   GFREE          /*12JAH12*
7078          11350 R 200253 R          LAC    GHEAD
7079          11351 R 540171 R          SAD    OBPNT
7080          11352 R 200255 R          LAC    GHDOLD
7081          11353 R 274103 E          XOR*   GFREE          /*12JAH12* COMPARE IT TO PROPER HEAD BLK ADDR
7082          11354 R 514012 R          AND    (37770          /LEFT 11 BITS OF HEAD BLOCK ADDRESS
7083          11355 R 741200 A          SNA          /0 IF PERFECT MATCH
7084          11356 R 611364 R          JMP    GDISK1          /OK
7085          11357 R 440645 R          ISZ   DKCHK          /INCREMENT CHPNT COUNTER
7086          11360 R 740000 A          NOP
7087          11361 R 451411 R          ISZ   CHKCNT
7088          11362 R 611337 R          JMP    GDISK4          /TRY AGAIN
7089          11363 R 000024 A          ERR24
7090          11364 R 211412 R      GDISK1  LAC    GOFF          /GIVE UP, "CHPNT" ERROR
7091          11365 R 652000 A          LMQ
7092          11366 R 111426 R          JMS   GLSET
7093          11367 R 631331 R          JMP*   GDISK
7094          /
7095          11370 R 211327 R      GDISK2  LAC    GLFNCT          /BEFORE WE UPDATE CHECKPOINT, MAKE SURE
7096          11371 R 741200 A          SNA          /WE'RE ALLOWED TO--IS THIS A "WRITE" FUNCTION?
7097          11372 R 611364 R          JMP    GDISK1          /NOT A "WRITE"--MAY NOT HAVE GLOBAL OPEN SO JUST GO AWAY.
7098          11373 R 200253 R          LAC    GHEAD          /YES, A "WRITE", SO WE CAN SAFELY UPDATE IT.
7099          11374 R 540171 R          SAD    OBPNT          /WILL BE EQUAL IF FIRST BLOCK OF LEVEL
7100          11375 R 200255 R          LAC    GHDOLD          /GET CORRECT CHECK POINTER VALUE
7101          11376 R 640603 A          LLS    3
7102          11377 R 514006 R          AND    (377700
7103          11400 R 253662 R          XOR    (400000          /LINE UP, MASK, OR IN C.P. BIT
7104          11401 R 260705 R          XOR*   GBWD2          /OR IN OFFSET
7105          11402 R 060705 R          DAC*   GBWD2          /STORE
7106          11403 R 440172 R          ISZ   WRTSW          /ALWAYS WRITE IF NEW BLK ADDED TO LEVEL
7107          11404 R 611364 R          JMP    GDISK1
7108          /
7109          11405 R 211327 R      GDISK3  LAC    GLFNCT          /JOINED BLOCK ENCOUNTERED

```

7110	11406 R 553705 R		SAD	(2	
7111	11407 R 000013 A		ERR13		/"GLOBE" - ATTEMPTED TO DO ILLEGAL JOIN
7112		/			
7113	11410 R 611364 R		JMP	GDISK1	/OK
7114		/			
7115	11411 R 000000 A	CHKCNT	0		/TRY 3 TIMES ON CHKPNT ERRORS
7116			.EJECT		

```

7117                /GOFF CALCULATES BLOCK OFFSET, RETURNS WITH R17 POINTING TO
7118                /LAST USED WORD IN BLOCK
7119                /
7120                11412 R 000000 A    GOFF      0
7121                11413 R 051322 R    DAC       GEND
7122                11414 R 040017 A    DAC       R17
7123                11415 R 200704 R    LAC       DSKBFF
7124                11416 R 744001 A    CMA!CLL
7125                11417 R 311322 R    ADD       GEND
7126                11420 R 040172 R    DAC       WRTSW      /WRTSW SET MEANS THERE'S SOMETHING NEW
7127                /
7128                11421 R 220705 R    LAC*      GBWD2
7129                11422 R 513646 R    AND       (777700      /OFFSET WILL EQUAL 1 IF NEW GLOBAL
7130                11423 R 300172 R    ADD       WRTSW      /WRTSW IS OFFSET
7131                11424 R 060705 R    DAC*      GBWD2
7132                11425 R 631412 R    JMP*      GOFF
7133                /
7134                /
7135                /GLSET INITIALIZES R17 AND GEND
7136                /
7137                11426 R 000000 A    GLSET     0
7138                11427 R 200705 R    LAC       GBWD2      /PLACE GBWD2 POINTER IN R17
7139                11430 R 040017 A    DAC       R17
7140                11431 R 220705 R    LAC*      GBWD2      /GET SECOND BUFFER WORD AND EXTRACT THE OFFSET
7141                11432 R 513702 R    AND       (77
7142                11433 R 300704 R    ADD       DSKBFF      /ADD BEGINNING OF BUFFER TO OFFSET
7143                11434 R 051322 R    DAC       GEND      /PLACE CURRENT END OF VALID BUFFER IN GEND
7144                11435 R 631426 R    JMP*      GLSET
7145                .EJECT

```

```

7146          /GSPACE FINDS A BLOCK WITH ENOUGH ROOM FOR GNOWDS WORDS AS FOLLOWS:
7147          / 1) CHECKS PRESENT BLOCK
7148          / 2) CHECKS PREVIOUS BLOCKS (GHI)
7149          / 3) SEARCHES FORWARD ALONG CHAIN
7150          / 4) GETS BLOCK FROM FREE LIST
7151          / READS IN SELECTED BLOCK
7152          /
7153          11436 R 000000 A      GSPACE 0
7154          11437 R 220705 R      GSPAC1 LAC*   GBWD2      /*12JAH12* IS THERE ROOM?
7155          11440 R 513702 R              AND      (77          /MASK OFFSET
7156          11441 R 311325 R              ADD      GNOWDS
7157          11442 R 353646 R              TAD      (777700      /77 FITS
7158          11443 R 745100 A              SPA!CLL      /FITS IF NEG
7159          11444 R 631436 R              JMP*     GSPACE      /ROOM IN CURRENT BLK
7160          11445 R 200252 R              LAC      GHI          /NO OF AVAIL LOC IN LARGEST PREVIOUS BLOCK
7161          11446 R 741200 A              SNA
7162          11447 R 611461 R              JMP      GSPAC4      /NO
7163          11450 R 311325 R              ADD      GNOWDS
7164          11451 R 353646 R              TAD      (777700      /77 FITS
7165          11452 R 740100 A              SMA
7166          11453 R 611461 R              JMP      GSPAC4      /NO ROOM
7167          11454 R 200251 R              LAC      GBLKHI
7168          11455 R 111331 R              JMS      GDISK      /READ
7169          11456 R 040017 A              DAC      R17
7170          11457 R 140252 R              DZM      GHI
7171          11460 R 631436 R              JMP*     GSPACE
7172          /
7173          11461 R 220704 R      GSPAC4 LAC*   DSKBFF      /IS THERE CONT BLK
7174          11462 R 741200 A              SNA
7175          11463 R 611467 R              JMP      GSPAC6      /NO
7176          11464 R 111331 R              JMS      GDISK      /READ
7177          11465 R 040017 A              DAC      R17
7178          11466 R 611437 R              JMP      GSPAC1      /*12JAH12*
7179          /
7180          11467 R 134103 E      GSPAC6 JMS*   GFREE      /*12JAH12* GET BLK FROM FREE LIST
7181          11470 R 060704 R              DAC*    DSKBFF      /CONT BLK
7182          11471 R 040172 R              DAC      WRTSW
7183          11472 R 111331 R              JMS      GDISK      /READ
7184          11473 R 160704 R              DZM*    DSKBFF
7185          11474 R 200705 R              LAC      GBWD2      /SET REG ADD OF BUFF
7186          11475 R 111412 R              JMS      GOFF
7187          11476 R 631436 R              JMP*     GSPACE

```

```

7188          .TITLE COMMAND PROCESSORS
7189          /PROCESS THE ASK COMMAND
7190          11477 R 107107 R   ASK   CALL   SETIO
7191          11500 R 744400 A       SNL!CLL
7192          11501 R 000003 A       ERR3           /ASK ALL ILLEGAL, SYNTAX ERROR
7193          /ASK A NUMERIC VARIABLE
7194          11502 R 106673 R   CALL   FORMAT           /HANDLE FORMAT CONT CHARS
7195          11503 R 741400 A       SZL
7196          11504 R 000003 A       ERR3           /' PREFIX NOT ALLOWED
7197          11505 R 105304 R   CALL   ASSEM           /ASSEM VAR NAME
7198          11506 R 220144 R   LAC*   WVAR           /SETUP WATCHDOG COUNTDOWN
7199          11507 R 740031 A       TCA
7200          11510 R 060154 R   DAC*   PRTEAD
7201          11511 R 213730 R   LAC    (XCI1           /*12JAH14* CHAR POS =
7202          11512 R 040221 R   DAC    IPLACE           /*12JAH14* MIDDLE
7203          11513 R 200133 R   LAC    BUFF2           /ADDR OF STR AC
7204          11514 R 040220 R   DAC    IPOINT
7205          11515 R 103325 R   CALL   PACKEM
7206          11516 R 611513 R   JMP    .-3
7207          /##### SAVE ON STACK #####
7208          11517 R 777776 A       LAW    -2           /*12JAH14*
7209          11520 R 105040 R   CALL   SAVE
7210          11521 R 200222 R   LAC    OPOINT
7211          11522 R 200223 R   LAC    OPLACE           /*12JAH14* CHAR POS
7212          /#####          #####
7213          11523 R 213732 R   LAC    (XCO1           /*12JAH14* CHAR POS =
7214          11524 R 040223 R   DAC    OPLACE           /*12JAH14* MIDDLE
7215          11525 R 200133 R   LAC    BUFF2
7216          11526 R 040222 R   DAC    OPOINT
7217          11527 R 744000 A       CLL
7218          11530 R 107163 R   CALL   EVAL           /NUMERIC ONLY
7219          /##### POPSTACK #####
7220          11531 R 777776 A       LAW    -2           /*12JAH14*
7221          11532 R 105103 R   CALL   UNSAVE
7222          11533 R 040223 R   DAC    OPLACE           /*12JAH14* CHAR POS
7223          11534 R 040222 R   DAC    OPOINT
7224          /#####          #####
7225          11535 R 606770 R   JMP    FORM2
7226          .EJECT

```

```
7227          /PROCESS THE BREAK COMMAND
7228          /##### SAVE ON STACK #####
7229      11536 R 777772 A      BREAK    LAW      -6          /*12JAH14*
7230      11537 R 105040 R          CALL    SAVE
7231      11540 R 203535 R          LAC     DISPAT
7232      11541 R 200203 R          LAC     FORTST
7233      11542 R 200206 R          LAC     LOC
7234      11543 R 220145 R          LAC*   LVAR
7235      11544 R 200222 R          LAC     OPOINT
7236      11545 R 200223 R          LAC     OPLACE      /*12JAH14* CHAR POS
7237          /#####          #####
7238      11546 R 211552 R          LAC     BREAK1
7239      11547 R 040020 A          DAC     R20
7240      11550 R 634071 E          JMP*   ERROR2      /*12JAH12*
7241      11551 R 000002 A          ERR2
7242      11552 R 011552 R      BREAK1  .          /*"BREAK"
7243          .EJECT
```

```

7244          /DO - PROCESS THE DO COMMAND
7245          /
7246          11553 R 744400 A DO      SNI!CLL      /ANY OPERAND(S)?
7247          11554 R 000004 A          ERR4          /NO, "CMMND"
7248          11555 R 744002 A DOO     STL           /*12JAH12* YES, SET FOR NUM OR STR RESULT AND
7249          11556 R 107163 R          CALL    EVAL      /GO EVALUATE OPERAND.
7250          /##### SAVE ON STACK #####
7251          11557 R 777772 A          LAW      -6          /*12JAH14*
7252          11560 R 105040 R          CALL    SAVE        /SAVE 5 WORDS
7253          11561 R 203535 R          LAC     DISPATCH
7254          11562 R 200203 R          LAC     FORTST
7255          11563 R 200206 R          LAC     LOC
7256          11564 R 220145 R          LAC*    LVAR
7257          11565 R 200222 R          LAC     OPOINT
7258          11566 R 200223 R          LAC     OPLACE      /*12JAH14* CHAR POS
7259          /#####          #####
7260          11567 R 200207 R          LAC     VAL
7261          11570 R 553741 R          SAD     (777777      /DID OPERAND EVAL AS STR?
7262          11571 R 611611 R          JMP     DOSTR        /YES, GO HANDLE "DO STR"
7263          11572 R 653323 A          IDIV;   144         /NO.
7264          11574 R 741200 A          SNA
7265          11575 R 611624 R          JMP     DO1          /FRACT PORTION (STEP NO) IN AC.
7266          11576 R 105123 R          CALL    STEP        /DO A PART
7267          11577 R 060145 R          DAC*    LVAR        /DO A STEP - FIND STEP NO HELD IN VAL.
7268          11600 R 200016 A          LAC     R16         /MAKE REF STEP "CURRENT" STEP
7269          11601 R 040206 R          DAC     LOC         /UPDATE PTR INTO STEP BUFFER.
7270          11602 R 740030 A          IAC
7271          11603 R 040222 R          DAC     OPOINT      /SETUP PTR TO GET REST OF STEP.
7272          11604 R 213732 R          LAC     (XC01       /*12JAH14* CHAR POS =
7273          11605 R 040223 R          DAC     OPLACE      /*12JAH14* MIDDLE
7274          11606 R 103535 R          CALL    DISPATCH    /GO TO CMND DISPATCH.
7275          11607 R 740000 A          NOP
7276          11610 R 611662 R          JMP     DO3         /QUIT RETURN
7277          /
7278          /DO STRING
7279          /
7280          11611 R 200133 R          DOSTR   LAC     BUFF2
7281          11612 R 040222 R          DAC     OPOINT
7282          11613 R 213732 R          LAC     (XC01       /*12JAH14* CHAR POS =
7283          11614 R 040223 R          DAC     OPLACE      /*12JAH14* MIDDLE
7284          11615 R 104660 R          CALL    OCHARI
7285          11616 R 103724 R          CALL    SYNTAX
7286          11617 R 000003 A          ERR3
7287          11620 R 611622 R          JMP     DOSTR2      /A- "SYNTAX"
7288          11621 R 000003 A          ERR3              /N
7289          .EJECT      /P- "SYNTAX"

```

```

7290          /DO STEP
7291          /
7292          11622 R 104340 R  DOSTR2  CALL  PLACE  /*12JAH14*
7293          11623 R 611662 R      JMP    DO3
7294          /
7295          /DO OR GOTO A PART
7296          /
7297          11624 R 105151 R  DO1     CALL  PART
7298          11625 R 200016 A  DO1.1  LAC   R16  /*12JAH12*
7299          11626 R 040206 R      DAC   LOC
7300          11627 R 740030 A      IAC
7301          11630 R 040222 R      DAC   OPOINT
7302          11631 R 213732 R      LAC   (XC01  /*12JAH14* CHAR POS =
7303          11632 R 040223 R      DAC   OPLACE  /*12JAH14* MIDDLE
7304          11633 R 220206 R      LAC*  LOC
7305          11634 R 060145 R      DAC*  LVAR
7306          11635 R 103535 R  DO2     CALL  DISPAT  /GOTO A STEP
7307          11636 R 611662 R      JMP    DO3  /QUIT RETURN
7308          11637 R 220145 R      LAC*  LVAR
7309          11640 R 653323 A      IDIV;  144
          11641 R 000144 A
7310          11642 R 641002 A      LACQ
7311          11643 R 040207 R      DAC   VAL
7312          11644 R 200206 R      LAC   LOC
7313          11645 R 040016 A      DAC   R16
7314          11646 R 220016 A      LAC*  R16
7315          11647 R 653606 A      LMQ!CLAC!LLS+6
7316          11650 R 300016 A      ADD   R16
7317          11651 R 540167 R      SAD   END
7318          11652 R 611662 R      JMP    DO3
7319          11653 R 040016 A      DAC   R16
7320          11654 R 220016 A      LAC*  R16
7321          11655 R 653323 A      IDIV;  144
          11656 R 000144 A
7322          11657 R 641002 A      LACQ
7323          11660 R 540207 R      SAD   VAL
7324          11661 R 611625 R      JMP    DO1.1  /*12JAH12*
7325          /##### POPSTACK #####
7326          11662 R 777772 A  DO3     LAW   -6  /*12JAH14*
7327          11663 R 105103 R      CALL  UNSAVE
7328          11664 R 040223 R      DAC   OPLACE  /*12JAH14* CHAR POS
7329          11665 R 040222 R      DAC   OPOINT
7330          11666 R 060145 R      DAC*  LVAR
7331          11667 R 040206 R      DAC   LOC
7332          11670 R 040203 R      DAC   FORTST
7333          11671 R 043535 R      DAC   DISPAT
7334          /##### #####
7335          11672 R 105201 R      CALL  CHECK
7336          11673 R 104645 R      CALL  OCHAR  /GET A CHAR.
7337          11674 R 553767 R      SAD   (54  /,          /IS IT A ,
7338          11675 R 611555 R      JMP    DO0  /*12JAH12* YES, DO NEXT OPERAND
7339          11676 R 603574 R      JMP    DTABLE+1  /NO MORE OPERANDS, GO TO COMMON RET PROC.

```

7340

.EJECT

```

7341          /PROCESS THE ERASE COMMAND
7342          11677 R 745400 A      ERASE  SZL!CLL
7343          11700 R 611710 R      JMP      ERASE1
7344          11701 R 220145 R      LAC*    LVAR      /ERASE ALL
7345          11702 R 744202 A      SZA!STL  /DIRECT MODE?
7346          11703 R 612362 R      JMP      ELSE     /E IS ELSE
7347          11704 R 200134 R      LAC     BUFF     /YES
7348          11705 R 740030 A      IAC
7349          11706 R 103744 R      CALL    STOR1
7350          11707 R 603573 R      JMP     DTABLE
7351          /ERASE A STEP OR A PART
7352          11710 R 107163 R      ERASE1  CALL    EVAL      /FIND OUT WHAT STEP OR PART USER HAS IN MIND
7353          11711 R 745100 A      SPA!CLL
7354          11712 R 000005 A      ERR5
7355          11713 R 653323 A      IDIV;   144      /MUST BE POSITIVE- "MINUS"
7356          11714 R 000144 A
7356          11715 R 741200 A      SNA
7357          11716 R 611750 R      JMP     ERASE3    /STEP OR PART?
7358          11717 R 105123 R      CALL    STEP     /PART (REMAINDER IS ZERO)
7359          11720 R 777776 A      LAW     -2       /STEP
7360          11721 R 300016 A      ADD     R16
7361          11722 R 040014 A      DAC     R14      /ONE BEFORE FIRST WORD
7362          11723 R 220016 A      LAC*    R16
7363          11724 R 653606 A      LMQ!CLAC!LLS+6
7364          11725 R 140652 R      ERASE5  DZM     SWAPSW    /INHIBIT SWAP DURING PRG UPDATE
7365          11726 R 300016 A      ADD     R16
7366          11727 R 040015 A      DAC     R15      /LAST WORD
7367          11730 R 540167 R      SAD     END      /IS LAST WORD THE END?
7368          11731 R 611736 R      JMP     ERASE6
7369          11732 R 220015 A      LAC*    R15      /NO
7370          11733 R 060014 A      DAC*    R14      /MOVEUP
7371          11734 R 200015 A      LAC     R15
7372          11735 R 611730 R      JMP     .-5
7373          11736 R 200014 A      ERASE6  LAC     R14
7374          11737 R 103744 R      CALL    STOR1
7375          11740 R 440652 R      JSZ     SWAPSW    /TIME SLICE UP?
7376          11741 R 741000 A      SKP
7377          11742 R 101533 R      JMS     HANG     /NO
7378          11743 R 105201 R      CALL    CHECK    /YES, LET OTHERS RUN
7379          11744 R 104645 R      CALL    OCHAR
7380          11745 R 553767 R      SAD     (54      /.
7381          11746 R 611710 R      JMP     ERASE1
7382          11747 R 603573 R      JMP     DTABLE
7383          /ERASE A PART
7384          11750 R 105151 R      ERASE3  CALL    PART     /EXIT WITH VAL=INTEGER PART
7385          11751 R 777776 A      LAW     -2
7386          11752 R 300016 A      ADD     R16
7387          11753 R 040014 A      DAC     R14      /ONE BEFORE FIRST WORD
7388          11754 R 220016 A      ERASE4  LAC*    R16
7389          11755 R 653606 A      LMQ!CLAC!LLS+6
7390          11756 R 300016 A      ADD     R16
7391          11757 R 540167 R      SAD     END      /REACHED THE END YET?

```

7392	11760 R 611736 R	JMP	ERASE6	/YES
7393	11761 R 040016 A	DAC	R16	
7394	11762 R 220016 A	LAC*	R16	
7395	11763 R 653323 A	IDIV;	144	
	11764 R 000144 A			
7396	11765 R 641002 A	LACQ		
7397	11766 R 540207 R	SAD	VAL	/NEW PART YET?
7398	11767 R 611754 R	JMP	ERASE4	/NO
7399	11770 R 777776 A	LAW	-2	
7400	11771 R 611725 R	JMP	ERASE5	
7401		.EJECT		

```

7402          /PROCESS THE FOR COMMAND
7403          / FOR VAR=EXP,EXP:EXP:EXP,EXP:EXP UNTIL OR WHILE BOOL
7404          /
7405          11772 R 744400 A   FOR      SNL!CLL
7406          11773 R 634073 E           JMP*    FILE6           /*12JAH12* ARG MISSING THEREFORE FILE COMMAND
7407          11774 R 104660 R           CALL    OCHARI
7408          11775 R 040214 R           DAC     M
7409          11776 R 104710 R           CALL    SUBOPT
7410          11777 R 200214 R           LAC     M
7411          12000 R 553724 R           SAD     (33           /LEFT SQUARE BRACKET
7412          12001 R 634072 E           JMP*    FILE           /*12JAH12* FILE WITH KEY CITED
7413          /##### SAVE ON STACK #####
7414          12002 R 777774 A           LAW     -4
7415          12003 R 105040 R           CALL    SAVE
7416          12004 R 200235 R           LAC     FORVAR           /NAME OF FOR VARIABLE
7417          12005 R 200243 R           LAC     T1              /POINTER TO FORVAR VALUE
7418          12006 R 200244 R           LAC     T2              /VALUE OF INCREMENT
7419          12007 R 200245 R           LAC     T3              /TERMINAL VALUE
7420          /#####
7421          12010 R 105304 R           CALL    ASSEM
7422          12011 R 553762 R           SAD     (75           / =
7423          12012 R 741000 A           SKP
7424          12013 R 000003 A           ERR3           /"SYNTAX"
7425          12014 R 200205 R           LAC     SCRIPS
7426          12015 R 744200 A           SZA!CLL
7427          12016 R 000003 A           ERR3           / INDEX VARIABLE CAN'T HAVE ANY SUBSCRIPTS
7428          /"SYNTAX" ERROR
7429          12017 R 107163 R           CALL    EVAL           / EVALUATE INITIAL VALUE OF INDEX
7430          12020 R 200204 R           LAC     SYMS
7431          12021 R 040177 R           DAC     SYM
7432          12022 R 040235 R           DAC     FORVAR
7433          12023 R 140200 R           DZM     SCRIP
7434          12024 R 754002 A           STL!CLA           /SETUP FOR SYMBOL "PUT"
7435          12025 R 105434 R           CALL    SYMBOL           /PUT "FOR VAR" INTO USER SYM TBL.
7436          12026 R 200017 A           LAC     R17
7437          12027 R 040243 R           DAC     T1           / POINTER TO FOR VARIABLE'S VALUE
7438          12030 R 104645 R   FOR1    CALL    OCHAR
7439          12031 R 553764 R           SAD     (72           / :
7440          12032 R 612214 R           JMP     FORIT1
7441          12033 R 750001 A           CLA!CMA
7442          12034 R 040244 R           DAC     T2           / SET SWITCH
7443          /##### POPSTACK #####
7444          12035 R 777775 A   FOR2    LAW     -3           /*12JAH14*
7445          12036 R 105040 R           CALL    SAVE
7446          12037 R 200222 R           LAC     OPOINT
7447          12040 R 200223 R           LAC     OPLACE           /*12JAH14* CHAR POS
7448          12041 R 200206 R           LAC     LOC
7449          /#####
7450          12042 R 440203 R   FOR3    ISZ     FORTST
7451          12043 R 104645 R           CALL    OCHAR
7452          12044 R 741000 A           SKP
7453          /

```

```

7454      12045 R 104660 R      FOR4      CALL      OCHARI
7455      12046 R 553653 R              SAD      (40          /SPACE
7456      12047 R 603537 R              JMP      FORXIT
7457      12050 R 553764 R              SAD      (72          /:
7458      12051 R 741000 A              SKP
7459      12052 R 612045 R              JMP      FOR4
7460      12053 R 104660 R      CALL      OCHARI
7461      12054 R 553764 R              SAD      (72          /:
7462      12055 R 612045 R              JMP      FOR4
7463      12056 R 553653 R              SAD      (40          /SPACE
7464      12057 R 741000 A              SKP
7465      12060 R 612053 R              JMP      .-5
7466      12061 R 104660 R      CALL      OCHARI
7467      12062 R 553653 R              SAD      (40          /SPACE
7468      12063 R 741000 A              SKP
7469      12064 R 612061 R              JMP      .-3
7470      12065 R 104660 R      FOR5      CALL      OCHARI
7471      12066 R 553653 R              SAD      (40          /SPACE
7472      12067 R 603537 R              JMP      FORXIT
7473      12070 R 553755 R              SAD      (42          /"
7474      12071 R 741000 A              SKP
7475      12072 R 612065 R              JMP      FOR5
7476      12073 R 104660 R      CALL      OCHARI
7477      12074 R 553755 R              SAD      (42          /"
7478      12075 R 612065 R              JMP      FOR5
7479      12076 R 612073 R              JMP      .-3
7480      /
7481      /
7482      / RETURN HERE FROM DISPATCH TO ITERATE FOR LOOPS
7483      /
7484      /##### SAVE ON STACK #####
7485      12077 R 777775 A      FORSIP    LAW      -3          /*12JAH14*
7486      12100 R 105103 R              CALL     UNSAVE
7487      12101 R 040206 R              DAC      LOC
7488      12102 P 040223 R              DAC      OPLACE     /*12JAH14* CHAR POS
7489      12103 R 040222 R              DAC      OPOINT
7490      /#####
7491      12104 R 105201 R              CALL     CHECK
7492      12105 R 777777 A              LAW      -1
7493      12106 P 340203 R              TAD     FORTST
7494      12107 R 040203 R              DAC     FORTST     / DECREMENT FORTST
7495      12110 R 777776 A              LAW     -2
7496      12111 R 300243 R              ADD     T1
7497      12112 R 040207 R              DAC     VAL        / PTR TO INDEX-VARIABLE'S NAME IN SYMBOL TABLE
7498      12113 R 740001 A              CMA
7499      12114 R 300170 R              ADD     SYMBEG
7500      12115 R 744001 A              CMA!CLL
7501      12116 R 741300 A              SPA!SNA     /OUTSIDE THE TABLE TEST
7502      12117 R 612124 R              JMP     .+5
7503      12120 R 220207 R              LAC*    VAL
7504      12121 R 240235 R              XOR     FORVAR
7505      12122 R 553662 R              SAD     (400000

```

```

7506      12123 R 612141 R          JMP      FORIT2
7507      12124 R 200235 R          LAC      FORVAR
7508      12125 R 040177 R          DAC      SYM
7509      12126 R 140200 R          DZM     SCRIP
7510      12127 R 754000 A          CLL!CLA  /SETUP FOR SYMBOL GET.
7511      12130 R 105434 R          CALL     SYMBOL / LOOK FOR "FOR VARIABLE"
7512      12131 R 641002 A          LACQ
7513      12132 R 741200 A          SNA
7514      12133 R 000010 A          ERR10   / FOR VARIABLE HAS BEEN KILLED- "UNDEF"
7515
7516      12134 R 745400 A          /
7517      12135 R 000011 A          SZL!CLL
7518
7519      12136 R 200017 A          /
7520      12137 R 740030 A          SZL!CLL
7521      12140 R 040243 R          ERR11   / FOR VARIABLE IS NOW A STRING- "MIXED"
7522      12141 R 200244 R          /
7523      12142 R 553741 R          LAC      R17 / FOR VARIABLE HAD MOVED
7524      12143 R 612167 R          IAC
7525      12144 R 320243 R          DAC      T1 / NEW ADDRESS OF FOR VARIABLE'S VALUE
7526      12145 R 741401 A          LAC      T2
7527      12146 R 000006 A          SAD     (777777
7528      12147 R 744200 A          JMP     FORNEXT
7529      12150 R 740001 A          ADD*    T1 / INCREMENT FOR VARIABLE
7530      12151 R 060243 R          SZL!CMA
7531      12152 R 200245 R          ERR6    /"MAXIM"
7532      12153 R 553741 R          SZA!CLL
7533      12154 R 612230 R          CMA
7534
7535      12155 R 560243 R          DAC*    T1
7536      12156 R 612035 R          LAC      T3
7537      12157 R 744001 A          SAD     (777777
7538      12160 R 320243 R          JMP     FORIF
7539      12161 R 745400 A          /
7540      12162 R 220243 R          FORIT3  SAD*    T1 / COMPARE INDEX AND TERMINATOR
7541      12163 R 740001 A          JMP     FOR2 / THEY'RE =; LOOP ONCE MORE
7542      12164 R 240244 R          CLL!CMA
7543      12165 R 740100 A          ADD*    T1
7544      12166 R 612035 R          SZL!CLL
7545      12167 R 104645 R          LAC*    T1
7546      12170 R 553767 R          CMA
7547      12171 R 612210 R          XOR     T2
7548      12172 R 553653 R          SMA
7549      12173 R 751001 A          JMP     FOR2 / INDEX<TERMINATOR; LOOP SOME MORE
7550      12174 R 000003 A          FORNEXT CALL  OCHAR / INDEX>TERMINATOR; CHECK FOR END
7551      12175 R 260243 R          SAD     (54 /
7552      12176 R 300244 R          JMP     FORON2
7553      12177 R 740001 A          SAD     (40 /SPACE
7554      12200 R 060243 R          SKP!CLA!CMA
7555      12201 R 777774 A          ERR3    /"SYNTAX"
7556      12202 R 105103 R          XOR*    T1
7557
/##### POPSTACK #####
FORDONE LAW -4
CALL UNSAVE

```

```

7558      12203 R 040245 R          DAC      T3
7559      12204 R 040244 R          DAC      T2
7560      12205 R 040243 R          DAC      T1
7561      12206 R 040235 R          DAC      FORVAR
7562      /#####                #####
7563      12207 R 603602 R          JMP      DTATST
7564      /
7565      12210 R 744000 A          FORON2  CLL          /NUMERIC RESULT ONLY
7566      12211 R 107163 R          CALL     EVAL
7567      12212 R 060243 R          DAC*    T1
7568      12213 R 612030 R          JMP      FOR1
7569      /
7570      12214 R 744000 A          FORIT1  CLL          /NUMERIC RESULT ONLY
7571      12215 R 107163 R          CALL     EVAL          /EVALUATE INCREMENT
7572      12216 R 040244 R          DAC      T2
7573      12217 R 104645 R          CALL     OCHAR
7574      12220 R 553653 R          SAD      (40          /SPACE
7575      12221 R 612230 R          JMP      FORIF
7576      12222 R 553764 R          SAD      (72          /:
7577      12223 R 745000 A          SKP!CLL
7578      12224 R 000003 A          ERR3          /"SYNTAX"
7579      12225 R 107163 R          CALL     EVAL          /EVALUATE TERMINATOR
7580      12226 R 040245 R          DAC      T3
7581      12227 R 612155 R          JMP      FORIT3
7582      /
7583      12230 R 140245 R          FORIF   DZM      T3
7584      12231 R 777776 A          LAW      -2          /*12JAH14*
7585      12232 R 105040 R          CALL     SAVE          /*12JAH14*
7586      12233 R 200222 R          LAC      OPOINT          /*12JAH14* PTR WORD
7587      12234 R 200223 R          LAC      OPLACE          /*12JAH14* CHAR WITHIN WORD
7588      12235 R 104660 R          CALL     OCHARI
7589      12236 R 554013 R          SAD      (25          / UNTIL
7590      12237 R 612244 R          JMP      .+5
7591      12240 R 554014 R          SAD      (27          / WHILE
7592      12241 R 741000 A          SKP
7593      12242 R 000004 A          ERR4          /"CMMND"
7594      12243 R 440245 R          ISZ      T3
7595      12244 R 104660 R          CALL     OCHARI
7596      12245 R 553653 R          SAD      (40
7597      12246 R 741000 A          SKP
7598      12247 R 612244 R          JMP      .-3
7599      12250 R 106203 R          CALL     BOOL
7600      12251 R 240245 R          XOR      T3
7601      12252 R 751201 A          SNA!CLA!CMA          /LOOP FINISHED?          /*12JAH18*
7602      12253 R 612261 R          JMP      FORIF1          /NO          /*12JAH18*
7603      12254 R 777776 A          LAW      -2          /YES          /*12JAH18*
7604      12255 R 105103 R          CALL     UNSAVE          /POP START          /*12JAH18*
7605      12256 R 740000 A          NOP          /OF LOOP PTRS          /*12JAH18*
7606      12257 R 740000 A          NOP          /DON'T NEED THEM ANYMORE          /*12JAH18*
7607      12260 R 612201 R          JMP      FORDONE /GO FINISH UP          /*12JAH18*
7608      12261 R 040245 R          FORIF1  DAC      T3          /INSERT -1          /*12JAH18*
7609      /#####                SAVE ON STACK #####

```

7610	12262 R 777777 A	LAW	-1	
7611	12263 R 105040 R	CALL	SAVE	
7612	12264 R 200206 R	LAC	LOC	
7613		/#####		#####
7614	12265 R 612042 R	JMP	FOR3	
7615		.EJECT		

/*12JAH18*

```

7616          /PROCESS THE GOTO COMMAND
7617          12266 R 740002 A      GO      CML          /O LINK=ARGUMENT
7618          12267 R 220145 R      LAC*    LVAR         /O LVAR=DIRECT
7619          12270 R 741600 A      SZL!SNA /SKIP ON ARGUMENT AND INDIRECT
7620          12271 R 612326 R      JMP     GD1        /GO OR DIRECT MODE
7621          12272 R 200203 R      LAC     FORTST
7622          12273 R 740200 A      SZA
7623          12274 R 000004 A      ERR4      /NO GO'S IN FOR COMMANDS
7624          12275 R 107163 R      CALL    EVAL
7625          12276 R 653323 A      IDIV;   144      /"CMMND"
              12277 R 000144 A
7626          12300 R 741200 A      SNA
7627          12301 R 612317 R      JMP     GD2        /GOTO A PART
7628          12302 R 105123 R      CALL    STEP      /GOTO A STEP
7629          12303 R 060145 R      DAC*    LVAR
7630          12304 R 200016 A      LAC     R16
7631          12305 R 040206 R      DAC     LOC
7632          12306 R 740030 A      IAC
7633          12307 R 040222 R      DAC     OPOINT
7634          12310 R 213732 R      LAC     (XCO1     /*12JAH14*
7635          12311 R 040223 R      DAC     OPLACE    /*12JAH14*
7636          12312 R 203535 R      LAC     DISPAT
7637          12313 R 513657 R      AND     (77777
7638          12314 R 554045 E      SAD     CALL71    /*12JAH12*
7639          12315 R 634044 E      JMP*    CALL7     /*12JAH12*
7640          12316 R 611635 R      JMP     D02
7641          /
7642          12317 R 203535 R      GO2    LAC     DISPAT
7643          12320 R 513657 R      AND     (77777
7644          12321 R 554045 E      SAD     CALL71    /*12JAH12*
7645          12322 R 741000 A      SKP
7646          12323 R 611624 R      JMP     D01        /*12JAH17*
7647          12324 R 105151 R      CALL    PART      /FIND PART        /*12JAH17*
7648          12325 R 634055 E      JMP*    DL2        /*12JAH17*
7649          /
7650          /PROCESS THE GO COMMAND
7651          12326 R 744400 A      GO1    SNL!CLL    /SKIP ON NO ARGUMENT AND DIRECT
7652          12327 R 000004 A      ERR4      /"CMMND"
7653          12330 R 611662 R      JMP     D03
7654          .EJECT

```

```
7655 /PROCESS THE HANG COMMAND
7656 12331 R 107163 R HANG1 CALL EVAL
7657 12332 R 140652 R DZM SWAPSW
7658 12333 R 741231 A SNA!CMA!IAC /*12JAH12*
7659 12334 R 612345 R JMP HANG2
7660 12335 R 047163 R DAC EVAL
7661 12336 R 777775 A LAW 17775 /777775
7662 12337 R 705522 A INTOFF
7663 12340 R 520654 R AND* PRINMB
7664 12341 R 253705 R XOR (2
7665 12342 R 060654 R DAC* PRINMB /SET HANG BIT
7666 12343 R 207163 R LAC EVAL
7667 12344 R 060153 R DAC* SYMEND /STORE
7668 12345 R 101533 R HANG2 JMS HANG
7669 12346 R 603573 R JMP DTABLE
7670 .EJECT
```

```
7671          /PROCESS THE IF COMMAND
7672 12347 R 744400 A  IF      SNL!CLL
7673 12350 R 612353 R          JMP      IF3      /*12JAH12* IF NO ARGUMENTS
7674 12351 R 106203 P  IF2    CALL    BOOL    /*12JAH12* EVAL LOGICAL ARG
7675 12352 R 040246 R          DAC      IFSWT   /STORE BOOLEAN VALUE
7676 12353 R 200246 R  IF3    LAC      IFSWT   /*12JAH12* RETRIEVE LAST BOOLEAN VALUE
7677 12354 R 741200 A          SNA
7678 12355 R 603602 R          JMP      DTATST  /FALSE, SKIP REST OF LINE
7679 12356 R 104645 R          CALL    OCHAR   /SO FAR ALL ARE TRUE
7680 12357 R 553767 R          SAD      (54    /MORE ARGUMENTS ?
7681 12360 R 612351 R          JMP      IF2    /*12JAH12* YES
7682 12361 R 603574 R          JMP      DTAB1   /*12JAH12* NO, CONTINUE IN LINE
7683          /PROCESS THE ELSE COMMAND
7684 12362 R 200246 R  ELSE    LAC      IFSWT   /RETRIEVE THE LAST IF RESULT
7685 12363 R 740200 A          SZA
7686 12364 R 603602 R          JMP DTATST  /TRUE IGNORE REST OF LINE
7687 12365 R 603573 R          JMP      DTABLE  /FALSE, CONTINUE SCAN
7688          .EJECT
```

```

7689          /PROCESS THE JOIN COMMAND
7690          /
7691          12366 R 744400 A      JOIN      SNL!CLL          /ARGUMENTS?
7692          12367 R 000004 A          ERR4          /'CMMND' - MUST BE ARGUMENTS
7693          12370 R 104660 P      JOIN0     CALL      OCHARI        /*12JAH12*
7694          12371 R 554001 R          SAD      (36          /
7695          12372 R 741000 A          SKP          /
7696          12373 R 000013 A          ERR13         /'GLOBE'
7697          12374 R 140254 R          DZM      GSYMBOL      /NO NAKED VARIABLES
7698          12375 R 100270 R          CALL      GSYM          /PACK SUBSCRIPTS
7699          12376 R 553655 R          SAD      (37          /
7700          12377 R 751001 A          SKP!CLC        /
7701          12400 R 000003 A          ERR3          /'SYNTAX'
7702          12401 R 340166 R          TAD      SPOINT
7703          12402 R 043712 R          DAC      OTEXT          /SAVE WHERE SUBSCRIPTS END
7704          12403 R 207107 R          LAC      SETIO
7705          12404 R 340165 R          TAD      STACK
7706          12405 R 050005 R          DAC      REMOVE        /SAVE WHERE THEY BEGIN
7707          12406 R 104660 R          CALL      OCHARI
7708          12407 R 554001 R          SAD      (36
7709          12410 R 741000 A          SKP
7710          12411 R 000013 A          ERR13         /'GLOBE'
7711          12412 R 100270 R          CALL      GSYM          /PACK SUBSCRIPTS OF 2ND
7712          12413 R 207107 R          LAC      SETIO
7713          12414 R 340165 R          TAD      STACK
7714          12415 R 047107 R          DAC      SETIO
7715          12416 R 227107 R          LAC*     SETIO
7716          12417 R 570005 R          SAD*    REMOVE        /CHECK NAME AGREEMENT
7717          12420 R 751001 A          SKP!CLC        /
7718          12421 R 000003 A          ERR3          /'SYNTAX'
7719          12422 R 340166 R          TAD      SPOINT
7720          12423 R 043174 R          DAC      OUT           /SAVE WHERE 2ND SUBS END
7721          12424 R 447107 R          JOIN1    ISZ      SETIO        /CHECK FOR LOOP CASE
7722          12425 R 450005 R          ISZ      REMOVE
7723          12426 R 210005 R          LAC      REMOVE
7724          12427 R 543712 R          SAD      OTEXT          /RUN OUT OF FIRST SUBS?
7725          12430 R 000013 A          ERR13         /YES, JOINING FROM NODE WHICH HAS POINTER
7726          /"GLOBE" ERROR
7727          12431 R 207107 R          LAC      SETIO
7728          12432 R 543174 R          SAD      OUT           /RUN OUT OF 2ND SUBS?
7729          12433 R 000003 A          ERR3          /'SYNTAX' - ALL SUBS EQUAL, CAUSES LOOP
7730          12434 R 230005 R          LAC*    REMOVE
7731          12435 R 567107 R          SAD*    SETIO          /SUBSCRIPTS EQUAL?
7732          12436 R 612424 R          JMP      JOIN1         /YES - COULD BE LOOP, KEEP LOOKING
7733          12437 R 754000 A          CLL!CLA        /NO - PROCEED
7734          12440 R 100272 R          CALL      GLOBAL      /'GET' 2ND
7735          12441 R 200253 R          LAC      GHEAD
7736          12442 R 047107 R          DAC      SETIO
7737          12443 R 440015 A          ISZ      R15
7738          12444 R 220015 A          LAC*    R15           /SHOULD BE POINTER
7739          12445 R 040207 R          DAC      VAL
7740          12446 R 641002 A          LACQ

```

7741	12447	R	754102	A	SMA!CLA!STL	
7742	12450	R	000010	A	ERR10	/'UNDEF' - NOT POINTER OR NOT DEFINED
7743	12451	R	742010	A	RTL	/AC=2 STATES "JOIN PUT", EXIT FROM GLOBAL ROUTINE WILL NOT ISZ SWAPS
7744	12452	R	100272	R	CALL GLOBAL	/'JOIN PUT'
7745	12453	R	140172	R	DZM WRTSW	/DON'T WRITE UNLESS ALL IS OK
7746	12454	R	641002	A	LACQ	
7747	12455	R	741100	A	SPA	
7748	12456	R	000013	A	ERR13	/'GLOBE' - ALREADY HAS POINTER
7749	12457	R	253650	R	XOR (600000	/FUDGE N BIT TO P BIT
7750	12460	R	060015	A	DAC* R15	/STORE IN BUFFER
7751	12461	R	207107	R	LAC SETIO	/SAVED HEAD ADDRESS
7752	12462	R	040255	R	DAC GHDDLD	
7753	12463	R	220015	A	LAC* R15	/GET POINTER
7754	12464	R	040253	R	DAC GHEAD	/GHEAD AND GHDDLD SET UP TO AVOID 'CHPNT'
7755	12465	R	440172	R	ISZ WRTSW	/WRITE OUT OLD BUFFER ON READ
7756	12466	R	151327	R	DZM GLFNCT	/AVOID TRAPPING 'JOIN PUT'
7757	12467	R	111331	R	JMS GDISK	/READ JOINED BLOCK
7758	12470	R	220705	R	LAC* GBWD2	
7759	12471	R	741100	A	SPA	/CHECK POINTER?
7760	12472	R	513702	R	AND (77	/YES, ZORCH IT
7761	12473	R	353723	R	TAD (100	/INCREMENT JOIN COUNTER
7762	12474	R	060705	R	DAC* GBWD2	
7763	12475	R	134057	E	JMS* DMPOLD	/*12JAH12* WRITE UPDATED BLK
7764	12476	R	140577	R	DZM DSKACT	/FREE DISK
7765	12477	R	440652	R	ISZ SWAPSW	/TIME SLICE UP?
7766	12500	R	741000	A	SKP	/NO
7767	12501	R	101533	R	JMS HANG	/YES, LET ANOTHER RUN
7768	12502	R	104645	R	CALL OCHAR	/NEXT CHAR
7769	12503	R	553767	R	SAD (54	/,
7770	12504	R	612370	R	JMP JOINO	/*12JAH12* MORE TO JOIN
7771	12505	R	603574	R	JMP DIAB1	/*12JAH12* DONE
7772					.EJECT	

```

7773                /PROCESS THE KILL COMMAND
7774      12506 R 745400 A      KILL  SZL!CLL
7775      12507 R 612513 R                JMP    KILL1
7776      12510 R 209135 R                LAC    SYMBOT
7777      12511 R 104034 R                CALL   STOR2
7778      12512 R 603573 R                JMP    DTABLE
7779                /KILL A VARIABLE OR AN ARRAY
7780      12513 R 104660 R      KILL1  CALL   OCHARI
7781      12514 R 554001 R                SAD    (36
7782      12515 R 610104 R                JMP    GKILL
7783      12516 R 104710 R                CALL   SUBOPT
7784      12517 R 105304 R                CALL   ASSEM
7785      12520 R 200204 R                LAC    SYMS
7786      12521 R 040177 R                DAC    SYM
7787      12522 R 200205 R                LAC    SCRIPS
7788      12523 R 040200 R                DAC    SCRIP
7789      12524 R 744002 A                STL
7790      12525 R 213756 R                LAC    (500000
7791      12526 R 105434 R                CALL   SYMBOL
7792      12527 R 104645 R      KILL15 CALL   OCHAR
7793      12530 R 553767 R      KILL2  SAD    (54
7794      12531 R 612513 R                JMP    KILL1
7795      12532 R 603574 R                JMP    DTAB1
7796                .EJECT

```

/

/SETUP FOR SYMBOL KILL.

/ KILL THE SYMBOL.

/*12JAH12*

/*12JAH12*

```

7797             .IFPNZ  MODIFY
7798             /
7799             /
7800             /             MODIFY COMMAND
7801             /
7802             / M(MODIFY) <S> : <D> <ID> <D> <R> (<D> (, <D> <ID> <D> <R> <D> ...)) <CR>
7803             /
7804             / M(MODIFY) <S> # <N> (, <S> # <N> ...) <CR>
7805             /
7806             / <S> = NUMERIC EXP EQUATING TO STEP NO. BEING MODIFIED
7807             / <D> = SINGLE CHAR DELIMITER NOT IN <ID> OR <R>
7808             / <ID> = ID STRING
7809             / <R> = REPLACEMENT STRING
7810             / <N> = NUMERIC EXP EQUATING TO NEW STEP NO.
7811             / <CR> = CARRIAGE RETURN
7812             /
7813             / FORM 1 ABOVE REPLACES THE FIRST OCCURRENCE OF ID STRING IN STEP NO.
7814             /WITH REPLACEMENT STRING. FORM 2 CHANGES THE OLD STEP NO. TO THE NEW
7815             /STEP NO.
7816             /
7817             /USE OF MODIFY IN INDIRECT MODE REQUIRES EXTREME CARE. AN INDIRECT
7818             /MODIFY MAY NOT MODIFY ITSELF NOR MAY IT RENUMBER ANOTHER STEP
7819             /WITH ITS STEP NUMBER.
7820             /
7821             12533 R 740400 A  MODIFY  SNL             /ANY ARGUMENT?
7822             12534 R 000003 A  ERR3             /NO, SYNTAX ERROR
7823             12535 R 744000 A  MODMET  CLL             /NUMERIC RESULT ONLY
7824             12536 R 107163 R  CALL      EVAL          /GO GET STEP NO TO BE MOD
7825             12537 R 745100 A  SPA!CLL          /STEP NUMBER NEGATIVE?
7826             12540 R 000005 A  ERR5             /YES, MINUS ERROR
7827             12541 R 560145 R  SAD*      LVAR          /MODIFYING STEP BEING EXECUTED?
7828             12542 R 000015 A  ERR15          /YES, STEPS ERROR
7829             12543 R 105123 R  CALL      STEP          /GO FIND LOC OF STEP IN PARTITION
7830             12544 R 104645 R  CALL      OCHAR          /GET COMMAND CHAR
7831             12545 R 553764 R  SAD      (72          / :
7832             12546 R 612552 R  JMP      CTEXT          / #
7833             12547 R 553770 R  SAD      (43          / #
7834             12550 R 612745 R  JMP      SWNUM          / #
7835             12551 R 000003 A  ERR3             /SYNTAX ERROR
7836             /
7837             /START OF : MODIFY STEP CONTENTS
7838             /
7839             12552 R 200207 R  CTEXT  LAC      VAL          /SET UP R16 AGAIN
7840             12553 R 105123 R  CALL      STEP          /IN CASE OF ADDITIONAL MOD TO THIS STEP
7841             12554 R 200016 A  LAC      R16           /MOVE STEP TO
7842             12555 R 104573 R  CALL      MBUF2          /TO STRING BUFFER
7843             12556 R 104660 R  CALL      OCHAR1         /FETCH NEXT COMMAND CHAR
7844             12557 R 741200 A  SNA             /EOL?
7845             12560 R 000003 A  ERR3             /YES, SYNTAX ERROR
7846             12561 R 040224 R  DAC      TEMP1          /*12JAH14* SAVE DELIMITER
7847             12562 R 213732 R  LAC      (XC01         /*12JAH14* CHAR POS=
7848             12563 R 040231 R  DAC      PLACE2         /*12JAH14* MIDDLE

```

7849	12564	R	045040	R	DAC	SAVE	/*12JAH14*
7850	12565	R	200016	A	LAC	R16	/LOAD PTR TO FIRST WORD OF STEP TO BE MOD
7851	12566	R	740030	A	IAC		/INCR TO FIRST DATA WORD
7852	12567	R	040230	R	DAC	POINT2	
7853	12570	R	044034	R	RSET	DAC	STOR2
7854	12571	R	200222	R	LAC	OPOINT	/*12JAH14*
7855	12572	R	040225	R	DAC	POINT1	/SET UP CHAR1 TO READ FIRST CHAR OF ID STRING
7856	12573	R	200223	R	LAC	OPLACE	/*12JAH14* SAVE
7857	12574	R	040226	R	DAC	PLACE1	/*12JAH14* CHAR POS
7858	12575	R	104770	R	TRYNSC	CALL	CHAR1
7859	12576	R	612620	R	JMP	FOUND	/YES, STRING MATCH
7860	12577	R	741200	A	SNA		/NO, EOL BEFORE 2ND OCCUR OF DELIMITER?
7861	12600	R	000003	A	ERR3		/YES, SYNTAX ERROR
7862	12601	R	040227	R	DAC	TEMP2	/SAVE CHAR
7863	12602	R	105014	R	CALL	CHAR2	/FETCH STEP CHAR
7864	12603	R	612575	R	JMP	TRYNSC	/*12JAH14* IGNORED RETURN
7865	12604	R	741200	A	SNA		/EOL?
7866	12605	R	000003	A	ERR3		/YES, ID STRING NOT FOUND IN STEP, SYNTAX ERROR
7867	12606	R	204034	R	LAC	STOR2	/NO, RESTORE POINT2 TO CHAR OF STEP
7868	12607	R	040230	R	DAC	POINT2	/MATCHED TO FIRST CHAR OF ID STRING
7869	12610	R	205040	R	LAC	SAVE	/*12JAH14* RESTORE
7870	12611	R	040231	R	DAC	PLACE2	/*12JAH14* CHAR POS
7871	12612	R	105014	R	CALL	CHAR2	/ADVANCE 1 CHAR IN STEP
7872	12613	R	740000	A	NOP		/IGNORED RETURN
7873	12614	R	200231	R	LAC	PLACE2	/*12JAH14* SAVE
7874	12615	R	045040	R	DAC	SAVE	/*12JAH14* CHAR POS
7875	12616	R	200230	R	LAC	POINT2	/SEEK MATCH WITH NEW SUBSTRING OF STEP
7876	12617	R	612570	R	JMP	RSET	
7877							/MATCH HAS BEEN FOUND
7878	12620	R	200225	R	FOUND	LAC	POINT1
7879	12621	R	040222	R	DAC	OPOINT	/SET UP OPOINT TO READ FIRST CHAR
7880	12622	R	200226	R	LAC	PLACE1	/OF REPLACEMENT STRING?
7881	12623	R	040223	R	DAC	OPLACE	/*12JAH14* SET UP
7882	12624	R	200133	R	LAC	BUFF2	/*12JAH14* CHAR POS
7883	12625	R	740031	A	TCA		/GET ADDR OF STRING BUFFER
7884	12626	R	040202	R	DAC	TEMP	/*12JAH14* SAVE ITS 2'S COMPL
7885	12627	R	340016	A	TAD	R16	/ADD STEP ADDR
7886	12630	R	740031	A	TCA		
7887	12631	R	344034	R	TAD	STOR2	/ADD START OF REPLACEMENT IN STEP
7888	12632	R	040220	R	DAC	IPOINT	/START OF REPLACEMENT IN STRING BUFFER
7889	12633	R	205040	R	LAC	SAVE	/*12JAH14* RESTORE
7890	12634	R	723061	A	AAC	XCII-XCOI	/*12JAH14* UPDATED
7891	12635	R	040221	R	DAC	IPLACE	/*12JAH14* CHAR POS
7892	12636	R	200133	R	LAC	BUFF2	
7893	12637	R	354015	R	TAD	(STRACC-1	
7894	12640	R	040014	A	DAC	R14	/LAST POSSIBLE ADDR FOR STEP IN STRING BUFFER
7895	12641	R	104660	R	REPCH	CALL	OCHARI
7896	12642	R	741200	A	SNA		/FETCH CHAR OF REPLACEMENT STRING
7897	12643	R	612657	R	JMP	ESTR	/EOL?
7898	12644	R	540224	R	SAD	TEMP1	/YES
7899	12645	R	612657	R	JMP	ESTR	/*12JAH14* DELIMITER?
7900	12646	R	104725	R	CALL	ICHARI	/YES
							/PUT CHAR IN STRING BUFFER

7901	12647	R	200220	R	LAC	IPOINT		
7902	12650	R	540014	A	SAD	R14	/2ND CHAR OF 25TH WORD OF STEP BUFFER?	
7903	12651	R	741000	A	SKP		/*12JAH14* LAST WORD OF BUFFER	
7904	12652	R	612641	R	JMP	REPCH	/*12JAH14* NOT LAST WORD	
7905	12653	R	200221	R	LAC	IPLACE	/*12JAH14* CHAR POS =	
7906	12654	R	553727	R	SAD	(XCI3	/*12JAH14* LAST?	
7907	12655	R	000006	A	ERR6		/YES, MAXIM ERROR --- TOO MANY CHARS	
7908	12656	R	612641	R	JMP	REPCH		
7909	12657	R	200222	R	ESTR	LAC	OPOINT	/SAVE COMMAND PTR
7910	12660	R	044034	R	DAC	STOR2		
7911	12661	R	200223	R	LAC	OPLACE	/*12JAH14* SAVE	
7912	12662	R	044340	R	DAC	PLACE	/*12JAH14* CHAR POS	
7913	12663	R	105014	R	REP2CH	CALL	CHAR2	/GET REMAINING CHARS IN STEP
7914	12664	R	740000	A	NOP			
7915	12665	R	741200	A	SNA		/DONE?	
7916	12666	R	612700	R	JMP	FINRPL	/YES	
7917	12667	R	104725	R	CALL	ICHARI	/PUT CHAR IN STRING BUFFER	
7918	12670	R	200220	R	LAC	IPOINT		
7919	12671	R	540014	A	SAD	R14	/2ND CHAR OF 25TH WD OF STEP BUFFER?	
7920	12672	R	741000	A	SKP		/*12JAH14* LAST WORD OF BUFFER	
7921	12673	R	612663	R	JMP	REP2CH	/*12JAH14* NOT LAST WORD	
7922	12674	R	200221	R	LAC	IPLACE	/*12JAH14* CHAR POS =	
7923	12675	R	553727	R	SAD	(XCI3	/*12JAH14* LAST?	
7924	12676	R	000006	A	ERR6		/YES, MAXIM ERROR --- TOO MANY CHARS	
7925	12677	R	612663	R	JMP	REP2CH		
7926	12700	R	104725	R	FINRPL	CALL	ICHARI	/ADD EOL
7927	12701	R	200133	R	LAC	BUFF2		
7928	12702	R	723001	A	AAC	1		
7929	12703	R	040227	R	DAC	TEMP2	/2ND WRD OF STEP (IN STRING BUFFER)	
7930	12704	R	220227	R	LAC*	TEMP2		
7931	12705	R	653606	A	LMQ!CLAC!LLS+6			
7932	12706	R	200220	R	LAC	IPOINT		
7933	12707	R	340202	R	TAD	TEMP	/*12JAH14* SUBTRACT START OF STRING BUFFER	
7934	12710	R	723777	A	AAC	-1	/DECREMENT TO GET PROPER STEP COUNT	
7935	12711	R	640614	A	LLS	14	/INSERT NEW STEP COUNT	
7936	12712	R	060227	R	DAC*	TEMP2	/RETURN TO 2ND WORD	
7937	12713	R	200207	R	LAC	VAL		
7938	12714	R	104616	R	CALL	ERMOD	/ERASE OLD STEP	
7939	12715	R	213732	R	LAC	(XCO1	/*12JAH14* CHAR POS =	
7940	12716	R	040223	R	DAC	OPLACE	/*12JAH14* MIDDLE	
7941	12717	R	200227	R	LAC	TEMP2		
7942	12720	R	040222	R	DAC	OPOINT	/SET UP PTR TO NEW STEP	
7943	12721	R	104357	R	CALL	PLAC1	/PUT NEW STEP IN PARTITION	
7944	12722	R	204034	R	LAC	STOR2		
7945	12723	R	040222	R	DAC	OPOINT	/RESTORE COMMAND PTR	
7946	12724	R	204340	R	LAC	PLACE	/*12JAH14* RESTORE	
7947	12725	R	040223	R	DAC	OPLACE	/*12JAH14* CHAR POS	
7948	12726	R	105201	R	CALL	CHECK	/RESTORE PTRS TO EXECUTING STEP IF NEC	
7949	12727	R	104645	R	CALL	OCHAR	/GET CHAR AFTER REPLACEMENT STRING	
7950	12730	R	741200	A	SNA		/EOL?	
7951	12731	R	612735	R	JMP	NRLINE	/YES, WRITE STEP IF IN DIRECT MODE	
7952	12732	R	104660	R	CALL	OCHAR	/NO,CLOSING DELIMITER INSERTED; FETCH NXT CHAR	

7953	12733	R	553767	R	SAD	(54	/COMMA MEANS ADDIT. MODS. TO THIS STEP
7954	12734	R	612552	R	JMP	CTEXT	
7955	12735	R	220145	R	WRLINE LAC*	LVAR	
7956	12736	R	740200	A	SZA		/IN DIRECT MODE?
7957	12737	R	603573	R	JMP	DTABLE	/NO
7958	12740	R	200207	R	LAC	VAL	
7959	12741	R	105123	R	CALL	STEP	
7960	12742	R	200207	R	LAC	VAL	
7961	12743	R	107066	R	CALL	WRSTEP	
7962	12744	R	603573	R	JMP	DTABLE	
7963					/		
7964					/START OF #MODIFY S EP NUMBER		
7965					/		
7966	12745	R	200016	A	SWNUM LAC	R16	
7967	12746	R	045123	R	DAC	STEP	/SAVE PTR TO 1ST WORD OF STEP TO BE RENUM.
7968	12747	R	744000	A	CLL		
7969	12750	R	107163	R	CALL	EVAL	/GET NEW STEP NO
7970	12751	R	560145	R	SAD*	LVAR	/REPLACING EXECUTING STEP?
7971	12752	R	000015	A	ERR15		/YES, STEPS ERROR
7972	12753	R	741300	A	SPA!SNA		/NEG OR ZERO?
7973	12754	R	000005	A	ERR5		/YES, MINUS ERROR
7974	12755	R	653323	A	IDIV ; 144		
	12756	R	000144	A			
7975	12757	R	741200	A	SNA		/ZERO FRAC STEP?
7976	12760	R	000014	A	ERR14		/YES,FRACT ERR
7977	12761	R	205123	R	LAC	STEP	
7978	12762	R	104573	R	CALL	MBUF2	/MOVE STEP TO STRING BUFFER
7979	12763	R	200207	R	LAC	VAL	
7980	12764	R	060133	R	DAC*	BUFF2	/REPLACE STEP #
7981	12765	R	225123	R	LAC*	STEP	
7982	12766	R	104616	R	CALL	ERMOD	/ERASE OLD STEP
7983	12767	R	220133	R	LAC*	BUFF2	
7984	12770	R	040207	R	DAC	VAL	/NEW STEP #
7985	12771	R	200222	R	LAC	OPOINT	
7986	12772	R	045123	R	DAC	STEP	/SAVE COMMAND PTR
7987	12773	R	200223	R	LAC	OPLACE	/*12JAH14*
7988	12774	R	044616	R	DAC	ERMOD	/*12JAH14*
7989	12775	R	213730	R	LAC	(XCI1	/*12JAH14* SET UP
7990	12776	R	040221	R	DAC	IPLACE	/*12JAH14* OUTPUT CHAR POS
7991	12777	R	213732	R	LAC	(XC01	/*12JAH14* AND
7992	13000	R	040223	R	DAC	OPLACE	/*12JAH14* INPUT CHAR POS
7993	13001	R	200133	R	LAC	BUFF2	
7994	13002	R	723001	A	AAC	1	/POINT TO
7995	13003	R	040222	R	DAC	OPOINT	/2ND
7996	13004	R	340214	R	TAD	M	/ (STEP COUNT)
7997	13005	R	040220	R	DAC	IPOINT	/AND LAST WORD OF STEP (IN STRING BUFFER)
7998	13006	R	104357	R	CALL	PLAC1	/INSERT NEW STEP
7999	13007	R	205123	R	LAC	STEP	/RESTORE COMMAND PTR
8000	13010	R	040222	R	DAC	OPOINT	
8001	13011	R	204616	R	LAC	ERMOD	/*12JAH14* RESTORE
8002	13012	R	040223	R	DAC	OPLACE	/*12JAH14* CHAR POS
8003	13013	R	105201	R	CALL	CHECK	/RESTORE PTRS TO EXECUTING STEP IF NEC

8004 13014 R 104645 R
8005 13015 R 553767 R
8006 13016 R 612535 R
8007 13017 R 603573 R
8008
8009

CALL DCHAR
SAD (54
JMP MODMET
JMP DTABLE
.ENDC
.EJECT

/COMMA MEANS ADDITIONAL STEP NO CHANGES

```

8010 /PROCESS THE PRINT COMMAND - OUTPUT ANY ASCII CODE
8011 13020 R 107107 R PRINT CALL SETIO /SET UP FOR CORRECT DEVICE
8012 13021 R 744400 A SNL!CLL /MUST HAVE A NUMERIC ARGUMENT
8013 13022 R 000004 A ERR4 /"CMMND"
8014 13023 R 107163 R PRINT0 CALL EVAL /*12JAH12*
8015 13024 R 144710 R DZM SUBOPT /SUBOPT USED AS SPACING COUNTER IN OUT
8016 13025 R 513657 R AND (77777
8017 13026 R 652000 A LMQ
8018 13027 R 200020 A LAC R20 /IS PRINCIPAL DEVICE
8019 13030 R 540162 R SAD PRNC DV /USER'S TERMINAL?
8020 13031 R 741000 A SKP /...YES
8021 13032 R 613037 R JMP PRINT3 /...NO
8022 13033 R 641002 A LACQ /FETCH P VALUE
8023 13034 R 514016 R AND (277 /GET 8 BITS
8024 13035 R 554017 R SAD (200 /REMOTE BREAK TO SELF?
8025 13036 R 613050 R JMP PRINT1 /REMOTE BREAK TO SELF NOT PERMITTED
8026 13037 R 554020 R PRINT3 SAD (12 /LF
8027 13040 R 741000 A SKP
8028 13041 R 613045 R JMP PRINT2
8029 13042 R 220151 R LAC* YVAR
8030 13043 R 353715 R TAD (144
8031 13044 R 060151 R DAC* YVAR
8032 13045 R 214021 R PRINT2 LAC (PRINT1 /FUDGE OUTPUT
8033 13046 R 043174 R DAC OUT
8034 13047 R 603253 R JMP XOUT4A
8035 13050 R 104645 R PRINI1 CALL OCHAR
8036 13051 R 553767 R SAD (54
8037 13052 R 613023 R JMP PRINT0 /*12JAH12* COMMA, PROCESS NEXT ARG
8038 13053 R 607010 R JMP NEXT2 /DONE
8039 .EJECT

```

```
8040          /PROCESS THE QUIT COMMAND
8041      13054 R 200203 R      QUIT      LAC      FORTST
8042          SNA!CLA!CMA
8043      13056 R 623535 R      JMP*      DISPAT      /FIRST EXIT
8044      13057 R 340203 R      TAD      FORTST      /QUIT OUT OF A FOR COMMAND
8045      13060 R 040203 R      DAC      FORTST
8046          /##### POPSTACK      #####
8047      13061 R 777771 A      LAW      -7      /*12JAH14*
8048      13062 R 105103 R      CALL     UNSAVE
8049      13063 R 740000 A      NOP
8050      13064 R 740000 A      NOP      /DON'T RESTORE *12-3*
8051      13065 R 740000 A      NOP      /*12JAH14*
8052      13066 R 040245 R      DAC      T3      /LAST TWO WORDS OF STACK *12-3*
8053      13067 R 040244 R      DAC      T2
8054      13070 R 040243 R      DAC      T1
8055      13071 R 040235 R      DAC      FORVAR
8056          /#####      #####
8057      13072 R 603573 R      JMP      DTABLE
8058          .EJECT
```

```

8059          /PROCESS THE READ COMMAND
8060          /
8061          13073 R 107107 R   READ   CALL   SETIO
8062          13074 R 745400 A          SZL!CLL          /READ ALL?
8063          13075 R 613152 R          JMP     READ1
8064          13076 R 220145 R          LAC*   LVAR   /YES, MUST BE IN DIRECT MODE
8065          13077 R 750201 A          SZA!CLC          /SET ACC TO -1 FOR SAVE *12JAH9*
8066          13100 R 000004 A          ERR4          / CMMND ERROR
8067          13101 R 777776 A          LAW    -2     /*12JAH14*
8068          13102 R 105040 R          CALL   SAVE   /SAVE DIRECT *12JAH9*
8069          13103 R 200222 R          LAC    OPOINT /COMMAND POINTER *12JAH9*
8070          13104 R 200223 R          LAC    OPLACE /*12JAH14* CHAR POS
8071          13105 R 213732 R   READ0  LAC    (XCO1  /*12JAH14* INIT
8072          13106 R 040223 R          DAC    OPLACE /*12JAH14* CHAR POS PTR
8073          13107 R 213730 R          LAC    (XCI1  /*12JAH14* INIT
8074          13110 R 040221 R          DAC    IPLACE /*12JAH14* CHAR POS PTR
8075          13111 R 200133 R          LAC    BUFF2  /*12JAH14* SETUP TO STR AC
8076          13112 R 040220 R          DAC    IPOINT /*12JAH14* AS IF IT WERE THE
8077          13113 R 040222 R          DAC    OPOINT /*12JAH14* DIRECT COMMAND BUFFER
8078          13114 R 103325 R          CALL   PACKEM /FILL BUFFER
8079          13115 R 000015 A          ERR15        /NO ^U ALLOWED
8080          13116 R 104660 R          CALL   OCHARI /GET 1ST CHAR OF LINE AND TEST IT
8081          13117 R 103724 R          CALL   SYNTAX
8082          13120 R 000003 A          ERR3          /A-"SYNTAX"
8083          13121 R 613142 R          JMP     READ0A /N--NUMERIC GO PLACE STEP
8084          13122 R 740200 A          SZA          /P--IS IT EOM
8085          13123 R 000004 A          ERR4          /NO CMMND ERROR
8086          13124 R 213732 R          LAC    (XCO1  /*12JAH14* INIT
8087          13125 R 040223 R          DAC    OPLACE /*12JAH14* INPUT
8088          13126 R 213730 R          LAC    (XCI1  /*12JAH14* AND OUTPUT
8089          13127 R 040221 R          DAC    IPLACE /*12JAH14* CHAR POS PTR
8090          13130 R 200133 R          LAC    BUFF2  /*12JAH14* READ IN NEXT LINE
8091          13131 R 040220 R          DAC    IPOINT /*12JAH14*
8092          13132 R 040222 R          DAC    OPOINT /*12JAH14*
8093          13133 R 103325 R          CALL   PACKEM
8094          13134 R 000015 A          ERR15
8095          13135 R 104660 R          CALL   OCHARI /GET 1ST CHAR
8096          13136 R 103724 R          CALL   SYNTAX /AND CHECK IT
8097          13137 R 000003 A          ERR3          /A--SYNTAX ERROR
8098          13140 R 741000 A          SKP          /N--GO PLACE STEP
8099          13141 R 613144 R          JMP     READ0B /P--TERMINATE PROGRAM LOADING *12JAH9*
8100          13142 R 104340 R   READ0A CALL   PLACE
8101          13143 R 613105 R          JMP     READ0  /GO BACK FOR NEXT COMMAND LINE
8102          13144 R 777776 A   READ0B LAW    -2     /*12JAH14*
8103          13145 R 105103 R          CALL   UNSAVE /RESTORE DIRECT *12JAH9*
8104          13146 R 040223 R          DAC    OPLACE /*12JAH14* CHAR POS PTR
8105          13147 R 040222 R          DAC    OPOINT /COMMAND POINTER *12JAH9*
8106          13150 R 104645 R          CALL   OCHAR  /FETCH NEXT COMMAND CHAR *12JAH9*
8107          13151 R 607010 R          JMP     NEXT2  /EXECUTE NEXT COMMAND *12JAH9*
8108          /
8109          /READ A STRING VARIABLE
8110          /

```

```

8111      13152 R 106673 R      READ1  CALL  FORMAT
8112      13153 R 741400 A              SZL
8113      13154 R 000003 A              ERR3          /' PREFIX ILLEGAL
8114      13155 R 105304 R      CALL    ASSEM
8115      13156 R 220144 R      LAC*    WVAR          /SETUP WATCHDOG COUNTDOWN
8116      13157 R 740031 A              TCA
8117      13160 R 060154 R      DAC*    PRIEND
8118      13161 R 213730 R      LAC     (XCI1      /*12JAH14* INIT OUTPUT
8119      13162 R 040221 R      DAC     IPLACE    /*12JAH14* CHAR POS PTR
8120      13163 R 200133 R      LAC     BUFF2     /*12JAH14*
8121      13164 R 040220 R      DAC     IPOINT    /*12JAH14*
8122      13165 R 103325 R      CALL    PACKEM
8123              013166 R      READ1A=.          /THIS LABEL USED BY MUDDC TO DETERMINE IF READ
8124              013166 R              /OF STRING VARIABLE UPON DENTRY FROM PACKEM
8125      13166 R 613161 R              JMP     .-5        /*12JAH14*
8126      13167 R 200221 R      LAC     IPLACE    /*12JAH14* POS PTR
8127      13170 R 553727 R      SAD     (XCI3    /*12JAH14* PT TO LAST CHAR IN WORD
8128      13171 R 613175 R      JMP     .+4        /*12JAH14* YES
8129      13172 R 750000 A      CLA     /*12JAH14* NO OUTPUT NULL CHAR
8130      13173 R 104725 R      CALL    ICHARI   /*12JAH14*
8131      13174 R 613167 R      JMP     .-5        /*12JAH14* TEST FOR LAST CHAR AGAIN
8132      13175 R 200133 R      LAC     BUFF2
8133      13176 R 740001 A      CMA
8134      13177 R 300220 R      ADD     IPOINT
8135      13200 R 513702 R      AND     (77
8136      13201 R 553702 R      SAD     (77          /IS IT ZERO LENGTH?          /*12JAH17*
8137      13202 R 750000 A      CLA     /*YES, MAKE SURE OFFSET IS ZERO /*12JAH17*
8138      13203 R 650506 A      CLQ!LRS+5
8139      13204 R 220133 R      LAC*    BUFF2
8140      13205 R 513645 R      AND     (7777
8141      13206 R 640002 A      DMQ
8142      13207 R 060133 R      DAC*    BUFF2
8143      13210 R 754001 A      CLA!CMA!CLL
8144      13211 R 040207 R      DAC     VAL
8145      13212 R 606770 R      JMP     FORM2
8146              .EJECT

```

```

8147          /PROCESS THE SET COMMAND
8148          13213 R 744400 A      SET      SNL!CLL
8149          13214 R 000004 A      ERR4          /"CMMND"
8150          13215 R 104660 R      CALL      OCHARI
8151          13216 R 554001 R      SAD      (36      /"
8152          13217 R 610073 R      JMP      GPUT
8153          13220 R 104710 R      CALL      SUBOPT
8154          13221 R 105304 R      CALL      ASSEM
8155          13222 R 553762 R      SAD      (75      /=
8156          13223 R 741000 A      SKP
8157          13224 R 000003 A      ERR3          /"SYNTAX"
8158          /##### SAVE ON STACK #####
8159          13225 R 777776 A      LAW      -2
8160          13226 R 105040 R      CALL      SAVE
8161          13227 R 200204 R      LAC      SYMS
8162          13230 R 200205 R      LAC      SCRIPS
8163          /#####          #####
8164          .IFPNZ  FMONEY
8165          13231 R 750030 A      CLA!IAC /FLOAT=1 MEANS THIS IS THE ONLY PLACE FROM
8166          13232 R 040212 R      DAC      FLOAT  /WHICH SA CAN BE CALLED!!!!
8167          .ENDC
8168          13233 R 744002 A      STL
8169          13234 R 107163 R      CALL      EVAL
8170          .IFPNZ  FMONEY
8171          13235 R 140212 R      DZM      FLOAT  /SA CANNOT BE USED WHEN FLOAT=0
8172          .ENDC
8173          /##### POPSTACK          #####
8174          13236 R 777776 A      LAW      -2
8175          13237 R 105103 R      CALL      UNSAVE
8176          13240 R 040200 R      DAC      SCRIP
8177          13241 R 040177 R      DAC      SYM
8178          /#####          #####
8179          13242 R 754002 A      STL!CLA
8180          13243 R 105434 R      CALL      SYMBOL      / PUT
8181          13244 R 104645 R      SET2    CALL      OCHAR
8182          13245 R 553767 R      SAD      (54      /COMMA
8183          13246 R 613215 R      JMP      SET+2
8184          13247 R 603574 R      JMP      DTABLE+1
8185          .EJECT

```

```

8186 /PROCESS THE TYPE COMMAND
8187 13250 R 107107 R TYPE CALL SETIO
8188 13251 R 744400 A SNL!CLL
8189 13252 R 613301 R JMP TALL /TYPE ALL
8190 13253 R 106673 R CALL FORMAT
8191 13254 R 745402 A SZL!CLL!CML
8192 13255 R 613274 R JMP TYPE3
8193 13256 R 107163 R CALL EVAL
8194 13257 R 553741 R SAD (777777
8195 13260 R 613263 R JMP TYPE2
8196 13261 R 104255 R CALL DECOU
8197 13262 R 606777 R JMP NEXT
8198 13263 R 200133 R TYPE2 LAC BUFF2 /TYPE A STRING
8199 13264 R 040225 R DAC POINT1
8200 13265 R 213732 R LAC (XC01 /*12JAH14* INIT
8201 13266 R 040226 R DAC PLACE1 /*12JAH14* CHAR POS
8202 13267 R 140224 R DZM TEMP1 /TERMINATOR FOR CHAR1
8203 13270 R 104770 R CALL CHAR1 /GET CHAR
8204 13271 R 606777 R JMP NEXT /DONE
8205 13272 R 103174 R CALL OUT
8206 13273 R 613270 R JMP .-3
8207 13274 R 107163 R TYPE3 CALL EVAL
8208 13275 R 553741 R SAD (777777
8209 13276 R 613263 R JMP TYPE2 /STRING
8210 13277 R 104320 R CALL OCTOUT
8211 13300 R 606777 R JMP NEXT
8212 /
8213 13301 R 200170 R TALL LAC SYMBEG / TYPE ALL VARIABLES IN CORE
8214 13302 R 040017 A DAC R17
8215 13303 R 750000 A TALL1 CLA
8216 13304 R 103174 R CALL OUT / CRLF
8217 13305 R 200017 A LAC R17
8218 13306 R 740030 A IAC
8219 13307 R 540153 R SAD SYMEND
8220 13310 R 606777 R JMP NEXT / NO MORE VARIABLES IN CORE
8221 13311 R 220017 A LAC* R17 / GET NEXT VAR'S NAME
8222 13312 R 045040 R DAC SAVE
8223 13313 R 777775 A LAW -3
8224 13314 R 045103 R DAC UNSAVE
8225 13315 R 205103 R TALL2 LAC UNSAVE
8226 13316 R 043712 R DAC OTEXT
8227 13317 R 205040 R LAC SAVE
8228 13320 R 741000 A SKP
8229 13321 R 640505 A LRS 5
8230 13322 R 443712 R ISZ OTEXT
8231 13323 R 613321 R JMP .-2
8232 13324 R 513655 R AND (37
8233 13325 R 553751 P SAD (35
8234 13326 R 760045 A LAW 45
8235 13327 R 553713 R SAD (34
8236 13330 R 760044 A LAW 44
8237 13331 R 740200 A SZA

```

```

8238      13332 R 103174 R      CALL      OUT
8239      13333 R 445103 R      ISZ       UNSAVE
8240      13334 R 613315 R      JMP       TALL2      /NEXT CHAR.
8241      13335 R 205040 R      LAC       SAVE
8242      13336 R 742110 A      SMA!RTL
8243      13337 R 613357 R      JMP       TARRAY     / ITS AN ARRAY; GO TYPE IT
8244      13340 R 760075 A      LAW       75
8245      13341 R 103174 R      CALL      OUT      / =
8246      13342 R 205040 R      LAC       SAVE
8247      13343 R 742010 A      RTL
8248      13344 R 741400 A      SZL
8249      13345 R 613351 R      JMP       .+4
8250      13346 R 220017 A      LAC*      R17      / ITS A NUMERIC VAR; GET ITS VALUE
8251      13347 R 104255 R      CALL      DECOU     / TYPE VAR'S VALUE
8252      13350 R 613303 R      JMP       TALL1
8253      13351 R 107021 R      CALL      TSTR      / IT'S A STRING VAR; GO TYPE IT
8254      13352 R 613303 R      JMP       TALL1
8255
8256      /
8257      13353 R 200017 A      TALL4     .IFPNZ  FMONEY
8258      13354 R 723004 A      LAC       R17      /SKIP OVER 4 WORDS
8259      13355 R 040017 A      AAC       4        /((LENGTH OF F.P. VAR)
8260      13356 R 613303 R      DAC       R17
8261      .ENDC
8262
8263      /
8264      013357 R      TARRAY=.
8265      13357 R 740400 A      .IFPNZ  FMONEY
8266      13360 R 613353 R      SNL      /FLOATING POINT?
8267      .ENDC
8268      13361 R 220017 A      JMP       TALL4     /YES, PRINT NULL STRING
8269      13362 R 300017 A      LAC*      R17      /NO, ARRAY
8270      13363 R 045103 R      ADD       R17
8271      13364 R 760050 A      TARR1     DAC       UNSAVE     / POINTER TO NEXT VAR
8272      13365 R 103174 R      LAW       50
8273      13366 R 220017 A      CALL      OUT      / (
8274      13367 R 741100 A      LAC*      R17      / GET SUBSCRIPT
8275      13370 R 613407 R      SPA
8276      13371 R 104255 R      JMP       TASTR     / GO TYPE STRING VAR
8277      13372 R 760051 A      CALL      DECOU     / TYPE SUBSCRIPT
8278      13373 R 103174 R      LAW       51
8279      13374 R 760075 A      CALL      OUT      / )
8280      13375 R 103174 R      LAW       75
8281      13376 R 220017 A      CALL      OUT      / =
8282      13377 R 104255 R      LAC*      R17      / GET VAR'S VALUE
8283      CALL      DECOU     / AND TYPE IT
8284
8284      /
8284      13400 R 200017 A      TARR2     LAC       R17
8285      13401 R 545103 R      SAD       UNSAVE
8286      13402 R 613303 R      JMP       TALL1     / END OF ARRAY
8287      13403 R 750000 A      CLA
8288      13404 R 103174 R      CALL      OUT      / CRLF
8289      13405 R 760040 A      LAW       40      / SPACE; INDENT REST OF ARRAY

```

8290	13406 R 613365 R	JMP	TARR1+1	
8291		/		
8292	13407 R 740001 A	TASTR	CMA	
8293	13410 R 104255 R	CALL	DECOU	/ TYPE SUBSCRIPT
8294	13411 R 760051 A	LAW	51	
8295	13412 R 103174 R	CALL	OUT	/)
8296	13413 R 760075 A	LAW	75	
8297	13414 R 103174 R	CALL	OUT	/ =
8298	13415 R 107021 R	CALL	TSTR	
8299	13416 R 613401 R	JMP	TARR2+1	
8300		.EJECT		

```

8301          /PROCESS THE WRITE COMMAND
8302      13417 R 107107 R      WRITE  CALL  SETIO
8303      13420 R 745400 A          SZL!CLL
8304      13421 R 613441 R          JMP    WRITE1
8305      13422 R 200134 R          LAC    BUFF          /WRITE ALL
8306      13423 R 040016 A          DAC    R16
8307      13424 R 220016 A          LAC*   R16
8308      13425 R 653606 A          LMQ!CLAC!LLS+6
8309      13426 R 300016 A          ADD    R16
8310      13427 R 540167 R          SAD    END
8311      13430 R 613462 R          JMP    WRITE3  /PGM BUFFER EMPTY
8312      13431 R 040016 A          DAC    R16
8313      13432 R 220016 A          LAC*   R16
8314      13433 R 653323 A          IDIV;  144
            13434 R 000144 A
8315      13435 R 641002 A          LACQ
8316      13436 R 040207 R          DAC    VAL
8317      13437 R 107041 R          CALL  WRPART
8318      13440 R 613437 R          JMP    .-1
8319          /WRITE A PART OR A STEP
8320      13441 R 744000 A      WRITE1  CLL          /NUMERIC RESULT ONLY
8321      13442 R 107163 R          CALL  EVAL
8322      13443 R 745100 A          SPA!CLL
8323      13444 R 000005 A          ERR5          /"MINUS"
8324      13445 R 653323 A          IDIV;  144
            13446 R 000144 A
8325      13447 R 740200 A          SZA
8326      13450 R 613460 R          JMP    WRITE2
8327      13451 R 105151 R          CALL  PART          /WRITE A PART
8328      13452 R 107041 R          CALL  WRPART
8329      13453 P 760000 A      WRITE4  LAW    0          /END OUTPUT OF A PART
8330      13454 R 103174 R          CALL  OUT          /OR A WRITE ALL WITH 2 NULL STRINGS
8331      13455 R 760000 A          LAW    0
8332      13456 R 103174 R          CALL  OUT
8333      13457 P 613462 R          JMP    WRITE3  /
8334      13460 R 105123 R      WRITE2  CALL  STEP          /WRITE A STEP
8335      13461 R 107066 R          CALL  WRSTEP
8336      13462 R 104645 R      WRITE3  CALL  OCHAR
8337      13463 R 553767 R          SAD    (54
8338      13464 P 613441 R          JMP    WRITE1
8339      13465 R 652000 A          LMQ
8340      13466 R 607011 R          JMP    NEXT1
8341          .EJECT

```



```

8392 /
8393 /
8394 /
8395 /DATA BLOCK READ ROUTINE--THERE ARE TWO ENTIRES INTO THIS ROUTINE.
8396 /XCDKR IS THE ENTRY POINT IF THE USER HAS EXPLICITLY STATED THE UNIT
8397 /IN HIS COMMAND STRING. XCDKR1 IS THE ENTRY POINT IF HE IS USING THE
8398 /DEFAULT UNIT SETTING OF 0.
8399 /
8400 /
8401 13527 R 744000 A XCDKR CLL /SET SO EVAL WILL ALLOW ONLY NUMERIC RESULT
8402 13530 R 107163 R CALL EVAL /EVALUATE 2ND ARG (I.E. UNIT#)
8403 13531 R 741100 A SPA /RETURNS WITH RESULT IN AC
8404 13532 R 000005 A ERR5 /"MINUS"
8405 13533 R 040247 R DAC GUNIT /SAVE UNIT
8406 .IFZER RP /*12JAH8*
8407 SZA
8408 .ENDC
8409 .IFPNZ RP /*12JAH8*
8410 13534 R 354022 R TAD (-RP /CHECK FOR OUT OF RANGE *12JAH8*
8411 13535 R 740100 A SMA
8412 .ENDC
8413 13536 R 000006 A ERR6 /"MAXIM"
8414 13537 R 103641 R XCDKR1 CALL PRODA /GET DISK
8415 13540 R 200252 R LAC GHI
8416 13541 R 040253 R DAC GHEAD
8417 13542 R 113600 R JMS XCDIS /GO CHECK OUT BLOCK NUMBER
8418 13543 R 134102 E JMS* GETNEW /*12JAH12* GET NEW BLOCK
8419 13544 R 220705 R LAC* GBWD2 /SETUP A FAKEOUT CHECKPOINT CALCULATION
8420 13545 R 640503 A LRS 3 /IN GHDDLD SO THAT IF USER WISHES TO DO
8421 13546 R 040255 R DAC GHDDLD /NAKED REFERENCES AFTER THIS HE CAN WITHOUT ERROR.
8422 13547 R 140577 R XCOMEX DZM DSKACT /FREE DISK *12JAH4*
8423 13550 R 140252 R XCOREX DZM GHI /CLEAR OUT OUR TEMP CELL *12JAH4*
8424 13551 R 440652 R ISZ SWAPSW /TIME SLICE UP?
8425 13552 R 603573 R JMP DTABLE /NO
8426 13553 R 101533 R JMS HANG /YES, LET OTHERS RUN
8427 13554 R 603573 R JMP DTABLE
8428 /
8429 /
8430 /
8431 /CORE WORD WRITE ROUTINE
8432 /
8433 /
8434 13555 R 200607 R XCCORW LAC ACSWR /GET CURRENT DATA SWITCH SETTING
8435 13556 R 513637 R AND (10000 /SEE IF XCOM WRITE SWITCH IS SET
8436 13557 R 745202 A SNA!STL /ALSO STL SO EVAL SVE CAN YIELD 777777
8437 13560 R 000032 A ERR32 /DATA SW NOT SET--TREAT AS KEY ERROR
8438 13561 R 107163 R CALL EVAL /GET SECOND ARG (CONTENTS TO BE PLACED)
8439 13562 R 722000 A PAL /SAVE NEW CONTENTS FOR NOW
8440 13563 R 200252 R LAC GHI /GET CORE ADDRESS
8441 13564 R 721000 A PAX /SAVE IT
8442 13565 R 741100 A SPA
8443 13566 R 000005 A ERR5 /"MINUS"

```

8444	13567	R	354023	R	TAD	(-..C	/SEE IF NONEXM	
8445	13570	R	740100	A	SMA			
8446	13571	R	000006	A	ERR6		/"MAXIM"	
8447					.DBREL ; DBA			/*12JAH18*
	13572	R	707762	A				
8448	13573	R	730000	A	PLA		/PUT NEW CONTENTS IN AC	
8449	13574	R	073577	R	DAC*	ZERO,X	/AND NOW INTO LOCATION DESIRED	/*12JAH18*
8450					.EBREL ; EBA		/RESET BANK MODE	/*12JAH18*
	13575	R	707764	A				
8451	13576	R	613550	R	JMP	XCOREX	/GO AWAY. *12JAH4*	
8452					/			
8453	13577	R	000000	A	ZERO	0	/MUST CONTAIN ZERO, IN PAGE 1	/*12JAH18*
8454					/			
8455					.EJECT			

```

8456 /
8457 /
8458 /THIS IS A GENERAL ROUTINE USED BY ALL THE OTHER XCOM FUNCTION ROUTINES
8459 /TO CHECK OUT BLOCK NUMBERS TO SEE IF THEY ARE OK.
8460 /
8461 /
8462 13600 R 000000 A XCDIS 0
8463 13601 R 040653 R DAC NBPNT /SAVE BLOCK #
8464 .IFPNZ RP-1 /*12JAH8*
8465 13602 R 200247 R LAC GUNIT
8466 13603 R 741200 A SNA /UNIT 0?
8467 13604 R 613612 R JMP XCDIS1 /YES
8468 13605 R 200653 R LAC NBPNT /NO, NON-ZERO UNIT--CHECK OUT BLOCK #
8469 13606 R 134054 E JMS* DISTR /*12JAH12* WHAT DEVICE IS IT ON?
8470 13607 R 613623 R JMP XCERR /CORE--NOT ON THIS UNIT--ERROR
8471 13610 R 613623 R JMP XCERR /RF15--NOT ON THIS UNIT--ERROR
8472 13611 R 613616 R JMP XCDIS2 /RP--SO FAR OK--GO SEE IF IN BOUNDS.
8473 .ENDC
8474 13612 R 200653 R XCDIS1 LAC NBPNT /THIS IS FOR UNIT 0; CHECK OUT BLOCK #
8475 13613 R 134054 E JMS* DISTR /*12JAH12* WHAT DEVICE?
8476 13614 R 613625 R JMP XCDIS4 /CORE--GO CHECK RANGE
8477 .IFPNZ RF15
8478 JMP XCDIS3 /RF--CHECK RANGE
8479 .ENDC
8480 .IFZER RF15
8481 13615 R 613623 R JMP XCERR /NO RF IN SYS. -- ERROR
8482 .ENDC
8483 .IFZER RP /*12JAH8*
8484 JMP XCERR /NO RP IN SYS. -- ERROR
8485 .ENDC
8486 .IFPNZ RP /*12JAH8*
8487 13616 R 740100 A XCDIS2 SMA
8488 13617 R 633600 R JMP* XCDIS /POS #--DEFINITELY ON RP--GO AWAY *12JAH8*
8489 13620 R 354024 R TAD (307400 /BLOCK # TOO BIG?
8490 13621 R 741100 A SPA
8491 13622 R 633600 R JMP* XCDIS /NO, ALL OK.
8492 .ENDC
8493 13623 R 140252 R XCERR DZM GHI /ALL XCOM ERRORS COME HERE--CLEAR TEMP CELL
8494 13624 R 000013 A ERR13 /"GLOBE" (THIS PARTICULAR MESSAGE IS SCREWY)
8495 .IFPNZ RF15
8496 XCDIS3 TAD (-..A /IS THIS BLOCK ON A NON-EXISTENT RF UNIT?
8497 SPA
8498 JMP* XCDIS /NO, GO AWAY
8499 JMP XCERR /YES! ERROR
8500 .ENDC
8501 13625 R 354025 R XCDIS4 TAD (-..D /CHECK AGAINST MAX CORE BLOCK #
8502 13626 R 740100 A SMA
8503 13627 R 613623 R JMP XCERR /TOO BIG--ERROR
8504 13630 R 354002 R TAD (.D
8505 13631 R 353635 R TAD ..BM /CHECK TO SEE IF TOO SMALL (I.E. IN SYS AREA)
8506 13632 R 741100 A SPA
8507 13633 R 613623 R JMP XCERR /TOO SMALL

```

8508	13634 R 633600 R		JMP#	XCDIS	/ALL OK -- GO AWAY.
8509		/			
8510		/			
8511	13635 R 000000 A	..BM			
8512		/			
8513			.EJECT		

8514

.LTCRG

13636	R	007000	A	*L
13637	R	010000	A	*L
13640	R	020000	A	*L
13641	R	030000	A	*L
13642	R	040000	A	*L
13643	R	050000	A	*L
13644	R	250600	A	*L
13645	R	007777	A	*L
13646	R	777700	A	*L
13647	R	000160	A	*L
13650	R	600000	A	*L
13651	R	710000	A	*L
13652	R	570000	A	*L
13653	R	000040	A	*L
13654	R	001075	R	*L
13655	R	000037	A	*L
13656	R	777727	A	*L
13657	R	077777	A	*L
13660	R	001534	R	*L
13661	R	700000	A	*L
13662	R	400000	A	*L
13663	R	200000	A	*L
13664	R	100000	A	*L
13665	R	000301	R	*L
13666	R	004000	A	*L
13667	R	000001	A	*L
13670	R	004377	A	*L
13671	R	000506	A	*L
13672	R	177777	A	*L
13673	R	003657	R	*L
13674	R	741000	A	*L
13675	R	740000	A	*L
13676	R	777477	R	*L
13677	R	000135	R	*L
13700	R	001007	R	*L
13701	R	000175	A	*L
13702	R	000077	A	*L
13703	R	000031	A	*L
13704	R	000004	A	*L
13705	R	000002	A	*L
13706	R	777773	A	*L
13707	R	623174	R	*L
13710	R	001000	A	*L
13711	R	177313	R	*L
13712	R	777776	A	*L
13713	R	000034	A	*L
13714	R	000177	A	*L
13715	R	000144	A	*L
13716	R	000015	A	*L
13717	R	000134	A	*L
13720	R	000014	A	*L

13721 R 000023 A *L
13722 R 777745 A *L
13723 R 000100 A *L
13724 R 000033 A *L
13725 R 777640 A *L
13726 R 777726 A *L
13727 R 004767 R *L
13730 R 004765 R *L
13731 R 004764 R *L
13732 R 004704 R *L
13733 R 000073 A *L
13734 R 603606 R *L
13735 R 377777 A *L
13736 R 004706 R *L
13737 R 777752 A *L
13740 R 777765 A *L
13741 R 777777 A *L
13742 R 000056 A *L
13743 R 000017 A *L
13744 R 000003 A *L
13745 R 000006 A *L
13746 R 000060 A *L
13747 R 004705 R *L
13750 R 004703 R *L
13751 R 000035 A *L
13752 R 000045 A *L
13753 R 000050 A *L
13754 R 000051 A *L
13755 R 000042 A *L
13756 R 500000 A *L
13757 R 000005 A *L
13760 R 000047 A *L
13761 R 000074 A *L
13762 R 000075 A *L
13763 R 000076 A *L
13764 P 000072 A *L
13765 R 000046 A *L
13766 R 000041 A *L
13767 R 000054 A *L
13770 P 000043 A *L
13771 R 600413 R *L
13772 R 000062 A *L
13773 R 777715 A *L
13774 R 177302 R *L
13775 R 000332 R *L
13776 R 000055 A *L
13777 R 000053 A *L
14000 P 000044 A *L
14001 R 000036 A *L
14002 R 001600 A *L
14003 R 000052 A *L
14004 R 000057 A *L

14005 R 770000 A *L
14006 R 377700 A *L
14007 R 010105 R *L
14010 R 010112 R *L
14011 R 477777 A *L
14012 R 037770 A *L
14013 R 000025 A *L
14014 R 000027 A *L
14015 R 000030 A *L
14016 R 000277 A *L
14017 R 000200 A *L
14020 R 000012 A *L
14021 R 013050 R *L
14022 R 777775 A *L
14023 R 620000 A *L
14024 R 307400 A *L
14025 R 776200 A *L

8515

000750 R
14041 R 014041 E *E
14042 R 014042 E *E
14043 R 014043 E *E
14044 R 014044 E *E
14045 R 014045 E *E
14046 R 014046 E *E
14047 R 014047 E *E
14050 R 014050 E *E
14051 R 014051 E *E
14052 R 014052 E *E
14053 R 014053 E *E
14054 R 014054 E *E
14055 R 014055 E *E
14056 R 014056 E *E
14057 R 014057 E *E
14060 R 014060 E *E
14061 R 014061 E *E
14062 R 014062 E *E
14063 R 014063 E *E
14064 R 014064 E *E
14065 R 014065 E *E
14066 R 014066 E *E
14067 R 014067 E *E
14070 R 014070 E *E
14071 R 014071 E *E
14072 R 014072 E *E
14073 R 014073 E *E
14074 R 014074 E *E
14075 R 014075 E *E
14076 R 014076 E *E
14077 R 014077 E *E
14100 R 014100 E *E
14101 R 014101 E *E
14102 R 014102 E *E

.END VBLOCK

14103 R 014103 E *E
14104 R 014104 E *E
14105 R 014105 E *E
14106 R 014106 E *E
14107 R 014107 E *E
14110 R 014110 E *E
14111 R 014111 E *E
14112 R 014112 E *E
14113 R 014113 E *E
14114 R 014114 E *E
14115 R 014115 E *E
14116 R 014116 E *E
14117 R 014117 E *E
14120 R 014120 E *E
14121 R 014121 E *E
14122 R 014122 E *E
14123 R 014123 E *E
14124 R 014124 E *E
14125 R 014125 E *E
14126 R 014126 E *E
14127 R 014127 E *E
14130 R 014130 E *E
14131 R 014131 E *E
14132 R 014132 E *E
14133 R 014133 E *E
14134 R 014134 E *E
14135 R 014135 E *E
14136 R 014136 E *E
14137 R 014137 E *E
14140 R 014140 E *E
14141 R 014141 E *E
14142 R 014142 E *E
14143 R 014143 E *E
14144 R 014144 E *E
14145 R 014145 E *E
14146 R 014146 E *E
14147 R 014147 E *E
14150 R 014150 E *E
14151 R 014151 E *E
14152 R 014152 E *E
14153 R 014153 E *E
14154 R 014154 E *E
14155 R 014155 E *E
14156 R 014156 E *E
14157 R 014157 E *E
14160 R 014160 E *E
14161 R 014161 E *E

PRG>4K

SIZE=14162

2 ERROR LINES

AC	00156	985*	2215	2223	2242	2408	2691	2692	2734
		3057							
ACSWR	00607	1283*	2151	2159	2164	2179	8378	8434	
ACSWRD	00662	284	1354*	2177	2270				
ACSWRX	00610	1284*	2175						
ACTEMP	00217	284	1037*	2496	2822	3717	3738	3748	3764
		3767	3770	3778	3811	3824	3833	3837	3848
		3852	3857	3872	3875	3881	3883	5759	6693
		6696	6697	6701					
ALTMOD	03434	442	3033	3264	3328*				
APT	000001	111	112*	114	239	328	444	452	766
		767	1248	1574	1585	1738	1797	1804	1992
		2022	2079	2201	2312	2335			
APPARR	005621	4746	4793*						
APPEND	04516	3951	4014*						
APPEP	05641	4749	4817*	4940					
APPNUM	05563	4757*	4945						
APPSTR	05575	4753	4770*	4944					
ARRAY1	05523	4714*	4723	4736					
ARRAY2	05535	4716	4727*						
ARRSTR	05661	4800	4835*						
ASEM15	05323	4524	4528*						
ASK	11477	3459	7190*						
ASSEM	05304	284	2616	2942	4512*	4539	4563	4581	7197
		7421	7784	8114	8154				
ASSEM1	05316	4523*	4534						
ASSEM2	05327	4527	4532*						
ASSEM4	05361	4520	4565*						
ASSEM5	05326	4517	4531*	4566					
AVAR	00136	441	957*	975	3017				
BACK	05451	4661*	4681	4685	4689				
BACK1	05452	4662*	4672						
BACK5A	10535	6593	6611*						
BACK5B	10514	6594*	7039						
BACK5C	10525	6597	6603*						
BAKCHR	000031								
BAND	06363	5223	5236*						
BEGIN	03460	284	3352*	3391	3398	3407			
BEGIN1	03500	3369*	3379						
BEGID	14041	310	933	1567					
BEG1	03461	3353*	3363						
BEG2	03474	3361	3365*						
BEG3	03476	3359	3367*						
BEDM	06416	5227	5229	5231	5276*				
BKCHAR	00242	284	1056*	2534	2860	4461	4476	5156	5715
		5810	5817	5820	5840	5876	5922	5939	5964
		6384	6404						
BLOCK	000100	211*	1103	2703	8345	8346			
BOOL	06203	284	2562	2888	5113*	5151	5258	5260	5265
		5280	7599	7674					
BOOLE	00260	284	1076*	2566	2892	5131	5137	5197	5214
BOOLG	00262	1078*	2568	2894	5132	5139	5199	5212	
BOOLL	00256	1074*	2564	2890	5130	5135	5195	5210	

BOOLNM	06240	5142*	5147						
BOOLX	14042	310							
BOR	06367	5225	5243*						
BPWRIT	00723	408	425	1401*					
BREAK	11536	3460	7229*						
BREAK1	11552	7238	7242*						
BSUM	00232	1048*	2518	2844	5116	5153	5246	5256	5267
		5279							
BTERM	00233	1049*	2520	2846	5118	5152	5237	5244	5257
		5266	5277						
BUFF	00134	285	952*	2708	3380	3649	3652	3943	4371
		4393	7347	8305					
BUFF1	00132	285	950*	1377	2704	3376	3382		
BUFF2	00133	356	951*	2706	3251	4076	4771	4773	4824
		4836	4838	4987	4989	5167	5707	5980	5995
		6056	6061	6064	6785	6792	6860	6867	7203
		7215	7280	7882	7892	7927	7980	7983	7993
		8075	8090	8120	8132	8139	8142	8198	
BUNIT	00250	285	1062*	2546	2872	8382			
BVAL	00234	1050*	2522	2848	5108	5115	5144	5154	5218
		5220	5236	5238	5243	5247	5255	5261	5268
		5270	5276						
CALL	100000	527*	3284	3330	3339	3341	3343	3368	3370
		3378	3386	3387	3397	3403	3404	3405	3413
		3414	3424	3434	3437	3444	3558	3561	3720
		3721	3823	3836	3841	3843	3845	3851	3856
		3861	3863	3880	3897	3898	3904	3908	3978
		3981	3985	3990	3995	4009	4018	4027	4039
		4092	4108	4429	4459	4468	4471	4479	4482
		4515	4516	4532	4533	4543	4547	4555	4558
		4562	4571	4574	4577	4581	4587	4595	4611
		4615	4760	4782	4804	4820	4847	4924	4973
		4999	5119	5124	5126	5133	5135	5137	5139
		5150	5160	5172	5176	5193	5195	5197	5199
		5200	5201	5221	5254	5260	5264	5271	5286
		5289	5294	5302	5305	5320	5333	5335	5340
		5348	5353	5360	5371	5373	5381	5383	5398
		5400	5414	5416	5451	5454	5466	5469	5494
		5506	5511	5533	5538	5543	5546	5555	5556
		5567	5568	5594	5596	5606	5610	5626	5628
		5635	5637	5640	5719	5724	5746	5750	5751
		5758	5761	5762	5778	5779	5802	5826	5870
		5901	5906	5920	5929	5937	5947	5952	5958
		5967	6025	6027	6039	6041	6045	6076	6081
		6090	6107	6110	6119	6120	6130	6151	6155
		6157	6169	6173	6206	6207	6216	6217	6221
		6225	6236	6238	6246	6250	6253	6259	6262
		6265	6279	6280	6293	6294	6319	6323	6338
		6343	6345	6348	6351	6360	6367	6370	6375
		6382	6393	6396	6411	6422	6424	6432	6442
		6446	6453	6455	6458	6475	6485	6524	7190
		7194	7197	7205	7209	7218	7221	7230	7249
		7252	7266	7274	7284	7285	7292	7297	7306

		7327	7335	7336	7349	7352	7358	7374	7378
		7379	7384	7407	7409	7415	7421	7429	7435
		7438	7445	7451	7454	7460	7466	7470	7476
		7486	7491	7511	7545	7557	7566	7571	7573
		7579	7585	7588	7595	7599	7604	7611	7624
		7628	7647	7656	7674	7679	7693	7698	7707
		7711	7734	7744	7768	7777	7780	7783	7784
		7791	7792	7824	7829	7830	7840	7842	7843
		7858	7863	7871	7895	7900	7913	7917	7926
		7938	7943	7948	7949	7952	7959	7961	7969
		7978	7982	7998	8003	8004	8011	8014	8035
		8048	8061	8068	8078	8080	8081	8093	8095
		8096	8100	8103	8106	8111	8114	8122	8130
		8150	8153	8154	8160	8169	8175	8180	8181
		8187	8190	8193	8196	8203	8205	8207	8210
		8216	8238	8245	8251	8253	8272	8276	8278
		8280	8282	8288	8293	8295	8297	8298	8302
		8317	8321	8327	8328	8330	8332	8334	8335
		8336	8351	8360	8362	8384	8402	8414	8438
CALLP	14043	310	3461						
CALL7	14044	310	7639						
CALL71	14045	310	7638	7644					
CDSTAT	00641	511	1332*						
CFUN1	14046	310	3245						
CHAR1	04770	285	344	2570	2896	3168	3185	3219	3233
		4230*	4246	4247	4248	5286	5288	5291	5302
		5304	5308	5340	5353	5381	5414	5454	5469
		5594	7858	8203					
CHAR2	05014	285	2572	2898	4259*	4275	4276	4277	5289
		5294	5305	5333	5348	5352	5355	5360	5371
		5377	5391	5398	5413	5424	5451	5453	5457
		5466	5468	5472	5635	6025	7863	7871	7913
CHECK	05201	285	2574	2900	4412*	4438	4444	5465	5474
		7335	7378	7491	7948	8003			
CHECK1	05217	4422	4426*	4443					
CHECK2	05234	4415	4425	4439*					
CHKARR	006011	4937	4947*						
CHKCNT	11411	7068	7087	7115*					
CLAC	641000	563*	4153	4157	4244	4273	5990		
CLKCNT	00174	286	1014*	2106	2107	2109	2450	2776	
CLKHNG	01476	2188	2194*						
CLKHZ	000074	163	164*	259	2115	2230			
CLKTMP	01532	2231*							
CLOCK	14047	319	2078	2118	2122	3015			
CLOCK1	01456	2114	2172	2175*					
CLOCK2	01416	2134	2142*						
CLOCK3	01523	2213	2218*						
CLOCK4	01454	2163	2173*						
CLOCK5	01441	2154	2162*						
CLOCK6	01425	2124	2149*						
CLOF	700004	586*	2095						
CLOK	01342	1640	2095*						
CLOK1	01361	2104	2110*						

CLON	700044	587*	2101						
CLOSE	10036	286	6206*						
CLOSE0	10040	6172	6208*	6224					
CLOSE1	10050	286	6216*						
CLOSE2	10051	6217*	6227						
CLOSE3	10055	6196	6221*						
CLRCHR	000037								
CLSF	700001	585*	1638	2092					
COMBUF	000031	224*	1103	2705	3252	3253			
COMMAN	000111	214	215*	217	218*	220	221*	223*	223
		224	3393	7188					
CONSW	00576	367	1244*	2071	2076	2173			
CONT	06621	286	2560	2886	3628	3637	3643	3644	3681
		3689	3691	3694	4311	4315	4317	4320	5320
		5445*	5452	5467	5486				
CONTAI	06452	5184	5319*						
CONT1	06627	5451*	5484						
CONT2	06632	5454*	5459						
CONT3	06646	5466*	5473						
CONT4	06671	5455	5470	5485*					
CORE	000070	261	672	677					
CRFIL	00715	286	1388*						
CRSI	706721	650*	1597						
CRSKP	706701	653*	1760						
CRTRUB	000001								
CRXX	000000	132*	136*	145*	254	507	796	799	1162
		1188	1214	1268	1406	1469	1472		
CR3B	000000	130	131*	138	144	1593			
CR15	000000	134	135*	139	144	1759			
CTEXT	12552	7832	7839*	7954					
CTT	01057	1731*							
DABSPT	00706	286	1381*						
DABSXF	14050	310	926	930					
DATE	14051	319	3020						
DATE.1	14052	319	2129	2135	2138	2148	3022		
DATE.2	14053	319	2127	2140	2142	2146	3024		
DBK	703304	2033	2193*	2202	2313	2336			
DCOUNT	000012	121	122*	1246	1389				
DC01	000005	253	363	377	380	383	386	389	392
		395	832	835	838	841	844	847	850
		853	856	859	862	865	868	871	874
		877	1407	1419	1422	1643	1654	1663	1672
		1681	1690	1699	1708	1805	1826	1848	1870
		1892	1914	1936	1958	3335			
DDTSW	00574	1242*	2044	2058	2153	2161			
DECER3	04164	3722	3752	3761*					
DECER4	04167	3741	3764*	3788					
DECER5	04173	3745	3767*	3792	3795				
DECER6	04203	3737	3774*						
DECIN	04103	286	2608	2934	3713*	3797	3799	3898	4778
		4787	4827	4830	4843	4900	4915	4923	4953
		4994	4997	5003	5028	5802			
DECNUM	04121	3723	3727*						

DECOU	04255	286	2610	2936	3831*	3854	3864	3874	3884
		5626	8196	8251	8276	8282	8293		
DECPT	04151	3725	3750*						
DECPT0	04156	3731	3755*						
DECR53	04176	3766	3769*						
DECR55	04201	3747	3772*						
DEC2	04235	2624	2950	3810*	3825	3841	3843	3845	3851
		3861	3863						
DEFIN	06460	5174	5328*						
DEFNUM	05571	4763*	4811						
DEFSTR	05614	4785*	4856						
DEF0	06465	5333*	5351	5380	5437				
DEF1	06504	5339	5348*	5356					
DEF10	06570	5336	5413*						
DEF11	06606	5415	5427*						
DEF11A	06613	5430	5432*						
DEF11B	06617	5431	5436*						
DEF12	06571	5414*	5426						
DEF2	06522	5337	5368*						
DEF21	06533	5374	5377*						
DEF3	06515	5354	5360*	5364	5404				
DEF4	06534	5378*	5393						
DEF5	06554	5363	5382	5398*	5401	5402			
DEF6	06563	5342	5405*						
DEGRAD	00673	287	1365*						
DEL	03405	3268	3300*						
DELELE	006104	5017*							
DELELO	06110	5020	5024*						
DELEL1	06120	5033*	5044						
DELEL2	06113	5028*	5060						
DELEND	05771	4913	4923*						
DELETE	05746	4898*	5048						
DELET1	05747	4871	4899*						
DELET2	05750	4886	4900*	5077					
DELET3	05751	4902*	5036						
DELMOV	05757	4912*	4921						
DELNEE	06141	5058*	5068						
DELNL1	006144	5057	5061*						
DELNVA	06154	5075*	5079	5083					
DEL2	03414	3303	3307*						
DEL3	03421	3310	3312*						
DEL4	03425	3311	3316*						
DEV	000062	254*	672	673	5582	5656	5665		
DTGIT	07306	5726	5738	5802*					
DIRECT	03527	3388	3402*						
DIRPTR	00676	287	1373*						
DISLOP	00264	287	1080*	2578	2904				
DISPAT	03535	287	2576	2902	3405	3411*	3454	3455	7231
		7253	7274	7306	7333	7636	7642	8043	
DISPT1	03552	3424*	3431						
DISPT2	03563	3433*	3436						
DISPT3	03550	3415	3422*						
DISTR	14054	310	8469	8475					

DIVO	07354	5843	5851*						
DIV2	07364	5853	5858*						
DIV3	07370	5859	5861*						
DKCHK	00645	1339*	7085						
DLAH	707064	620*							
DLAL	707024	619*							
DLOK	707202	624*							
DL2	14055	311	7648						
DMP	14056	311	1569						
DMPOLD	14057	311	6133	6210	7034	7763	8387		
DO	11553	140	3462	7246*					
DOLLAR	07236	5730	5750*						
DOSTR	11611	7262	7280*						
DOSTR2	11622	7287	7292*						
DOO	11555	7248*	7338						
DO1	11624	7265	7297*	7646					
DO1.1	11625	7298*	7324						
DO2	11635	7306*	7640						
DO3	11662	7276	7293	7307	7318	7326*	7653		
DPCA	706344	597*							
DPCS	706324	601*							
DPLA	706304	598*							
DPLF	706464	599*							
DPRSA	706312	591*	898						
DPRSB	706332	592*	895						
DPSA	706321	594*							
DPSE	706361	595*	893						
DPSF	706301	600*	1622						
DPSJ	706341	593*	890						
DPWC	706364	596*							
DRAH	707062	618*							
DRAL	707022	617*							
DSCC	707021	616*							
DSCD	707242	625*							
DSCF	707041	621*							
DSCN	707044	623*							
DSFX	707042	622*							
DSKACT	00577	287	1245*	2303	2342	2367	3034	3506	3520
		6134	6211	7043	7764	8422			
DSKBFF	00704	287	1379*	2700	6587	6603	6608	6651	6709
		6714	6725	6887	7123	7142	7173	7181	7184
DSKCNT	00600	287	1246*	2199	3514	3517			
DSKRDE	14060	311	6995	7014					
DSKREP	00716	1389*	2198	3516					
DSKWRT	14061	311	7006	7029					
DSRS	707262	626*	908						
DSSF	707001	615*	906	1617					
DTA	000002	254	437	772	775	1094	1151	1152	1177
		1178	1203	1204	1250	1404	1462	1465	1601
		2637	2962	3152	3159	3160	3318	3527	
DTAB	03607	3422	3459*						
DTABLE	03573	288	3444*	5585	6213	6215	7339	7350	7382
		7669	7687	7778	7957	7962	8007	8057	8184

		8425	8427						
DTAB1	03574	288	3445*	7682	7771	7795			
DTACT	14062	450	3541	3545					
DTATST	03602	288	3418	3448	3451*	7563	7678	7686	
DTBSZ	000400	184	185*						
DTBUFF	00666	442	1360*						
DTCA	707541	611*							
DTDF	707601	610*	1608						
DTEF	707561	608*	1605						
DTERRA	00624	442	1309*						
DTERRB	00625	443	1310*						
DTERRS	00626	443	1311*						
DTIN	14063	450	1205						
DTIN1	14064	450	3319						
DTJMPI	03427	443	3319*						
DTJMPO	03203	443	3153	3154*					
DTLA	707545	606*							
DTOUT	14065	450	1179						
DTOUT1	14066	450	3154						
DIRA	707552	607*							
DIRB	707572	609*							
DTTRAN	00603	443	773	1251*					
DTWHO1	00403	441	1153*						
DTXA	707544	605*							
DUMMY	000005	172	173*	254	1169	1170	1195	1196	1221
		1222	1408						
DVHANG	01713	288	2387*						
DVHNG1	01721	288	2393*						
DXKEY	14067	319	8352						
EBCDIC	000001								
ELSE	12362	7346	7684*						
EMERG	00570	1234*	2306	3052					
END	00167	288	1008*	2440	2766	3615	3645	3651	3653
		3668	3686	3950	3965	4002	4036	4101	4309
		4376	4400	4441	5614	7317	7367	7391	8310
EQUAL	06423	5186	5286*	5292					
EQUAL1	05501	4665	4692*						
EQUAL2	06433	5287	5294*						
ERASE	11677	3463	7342*						
ERASE1	11710	7343	7352*	7381					
ERASE3	11750	7357	7384*						
ERASE4	11754	7388*	7398						
ERASE5	11725	7364*	7400						
ERASE6	11736	7368	7373*	7392					
ERMDD	04616	2657	2983	4090*	4110	4112	7938	7982	7988
		8001							
ERROR	000000	530*							
ERROR0	14070	311	1565						
ERROR2	14071	311	7240						
ERR1	000001	534*	3654	3688					
ERR10	000010	541*	4590	5829	6304	7514	7742		
ERR11	000011	542*	4592	5900	5973	7517			
ERR12	000012	543*							

ERR13	000013	544*	6176	6228	6555	7111	7696	7710	7725
		7748	8494						
ERR14	000014	545*	3903	7976					
ERR15	000015	546*	3941	4377	4401	7828	7971	8079	8094
ERR16	000016	547*	4310	4346					
ERR17	000017	548*	4702	4705	4796	4880	4897	4905	4950
		5022	5064	6640					
FRR2	000002	535*	7241						
ERR20	000020	549*	6565						
ERR21	000021	550*	6650						
ERR22	000022	551*							
ERR23	000023	552*	3043						
ERR24	000024	553*	7089						
ERR25	000025	554*							
ERR26	000026	555*							
ERR27	000027	556*	1189	1197	1209	1212	1223		
ERR3	000003	536*	3392	3416	3421	3450	3535	3907	4518
		4521	4535	4561	5189	5232	5334	5349	5361
		5372	5375	5376	5399	5540	5739	5753	5754
		5764	5781	5903	5955	6080	6121	6156	6218
		6249	6282	6296	6346	6451	6468	6473	7192
		7196	7286	7288	7424	7427	7550	7578	7701
		7718	7729	7822	7835	7845	7861	7866	8082
		8097	8113	8157					
ERR30	000030	557*							
ERR31	000031	558*	5676						
ERR32	000032	559*	6161	6364	8356	8381	8437		
ERR33	000033	560*							
ERR4	000004	537*	3406	3475	3477	3485	3488	3489	6106
		7247	7593	7623	7652	7692	8013	8066	8085
		8149							
ERR5	000005	538*	4551	5513	5653	5951	7354	7826	7973
		8323	8404	8443					
ERR6	000006	539*	5658	5863	5886	5913	5993	6034	6414
		6421	7527	7907	7924	8413	8446		
ERR7	000007	540*							
ER55	04631	4101*	4106						
ER6	04637	4102	4107*						
ESTR	12657	7897	7899	7909*					
ESYMBD	06173	5090	5095*						
EVAL	07163	288	2598	2924	3423	3426	3428	3440	4460
		4468	4477	4547	5126	5201	5511	5694*	5894
		5921	5929	5938	5947	5965	6067	6082	6250
		6383	6393	6405	7218	7249	7352	7429	7566
		7571	7579	7624	7656	7660	7666	7824	7969
		8014	8169	8193	8207	8321	8360	8402	8438
EVALD	07174	5704	5711*						
EVAR	00152	288	972*	975					
EXEC	001644	2301	2334*						
EXEC1	01672	2344	2359*						
EXEC2	01677	2341	2364*						
EXEC3	01704	2363	2369*						
EXEC4	001705	2346	2356	2361	2366	2370*			

EXIT	01075	289	442	1790*	2049	2077	2157	2185	2191
EXTSI	05060	4297	4308*						
FILE	14072	311	7412						
FILE6	14073	312	7406						
FINRPL	12700	7916	7926*						
FLOAT	00212	356	1031*	2485	2811	8166	8171		
FLOAT1	05723	4874*	4892						
FLOAT2	05735	4877	4884*						
FMONEY	000001	150	151*	153	352	1029	2480	2805	4540
		4552	4675	4686	4747	4794	4813	4873	4890
		4929	4934	4938	4948	5018	5051	5054	5062
		5080	5563	5700	6084	6089	6525	6636	8164
		8170	8256	8264					
FNDARR	005740	4862	4889*						
FNDELE	06076	4718	4729	5011*					
FNDSYM	05707	4699	4860*						
FOLLOW	06436	5188	5302*	5313					
FOR	11772	3464	7405*						
FORDON	12201	7556*	7607						
FORIF	12230	7533	7575	7583*					
FORIF1	12261	7602	7608*						
FORIT1	12214	7440	7570*						
FORIT2	12141	7506	7522*						
FORIT3	12155	7535*	7581						
FORMAT	06673	2614	2940	5493*	5505	5508	5549	7194	8111
		8190							
FORM2	06770	5559*	7225	8145					
FORNEX	12167	7524	7545*						
FURON2	12210	7547	7565*						
FORSTP	12077	3453	7485*						
FORTST	00203	289	1021*	2464	2790	3412	3451	7232	7254
		7332	7450	7493	7494	7621	8041	8044	8045
FORVAR	00235	1051*	2524	2850	7416	7432	7504	7507	7561
		8055							
FORXIT	03537	3413*	3420	3446	7456	7472			
FOR1	12030	7438*	7568						
FOR2	12035	7444*	7536	7544					
FOR3	12042	7450*	7614						
FOR4	12045	7454*	7459	7462					
FOR5	12065	7470*	7475	7478					
FOUND	12620	7859	7878*						
FPERRC	10564	6527	6529	6637*					
FTABLE	14074	312							
FUNCT	14075	312	5766						
FUNOUT	14076	312	5931						
FVAR	00142	961*	3202	3212					
GARB	14077	312	2325						
GAS	660000	564*							
GASS	00266	1082*	2618	2944	3618	3629	3631	3638	3640
		3641	3677	3682	3684	3690	3692	4314	4316
		4318	6120	6217	6283	6299	6306	6319	
GASS1	10122	6284*	6295						
GASS15	10127	6285	6289*						

GASS2	10133	6288	6293*						
GASS5	10132	6281	6292*						
GASS6	10141	6301*	6309						
GBACK1	10415	6531*	6642						
GBACK2	10473	6577*	6746						
GBACK3	10475	6579*	6733						
GBACK4	10504	6586*	6726						
GBACK5	10505	6587*	6659						
GBACK6	10540	6602	6615*	6629					
GBACK7	10541	6616*	6634						
GBAD	00671	289	1363*						
GBLKHI	00251	1063*	2548	2874	6724	7167			
GBLTPT	00710	289	1383*						
GBUFF1	14100	319	6991	6994	7005	7011	7013	7023	7027
		7028							
GRWD2	00705	289	1380*	2702	6888	7070	7104	7105	7128
		7131	7138	7140	7154	7185	7758	7762	8419
GCH	11330	7019	7020	7026	7033	7056*			
GCHAIN	14101	319	7017						
GCHANP	00672	289	1364*						
GCONT	10651	6617	6652	6670	6709*				
GCONT1	10670	6720	6725*						
GDISK	11331	290	6586	6886	7063*	7093	7168	7176	7183
		7757							
GDISK1	11364	7084	7090*	7097	7107	7113			
GDISK2	11370	7073	7095*						
GDISK3	11405	7075	7109*						
GDISK4	11337	7069*	7088						
GDONE	11321	6534	6543	6738	6892	6915	7049*		
GEND	11322	290	6596	6616	6669	6713	6806	6948	7050*
		7121	7125	7143					
GETKEY	05242	290	2644	2970	4456*	4481	4483	6155	6345
GETNEW	14102	312	7069	8418					
GEXIT	11311	6605	6858	6880	7040*				
GEXIT0	11305	6988	7009	7036*					
GFNDA	10770	6781	6784	6789	6802*				
GFNDB	10747	6783	6785*						
GFND0	10773	6775	6805*	6810	6815	6817			
GFND0A	11001	6773	6812*						
GFND0B	11013	6801	6822*						
GFND1	11016	6807	6826*						
GFND2	11023	6831*	6893	6916					
GFND3	11031	6837*	6865						
GFND3A	11046	6847	6850*						
GFND4	11052	6854*	6874						
GFND4A	11054	6824	6856*						
GFND5	11057	6835	6860*						
GFND6	11065	6852	6867*						
GFND7	11075	6764	6876*						
GFJUND	10712	6739	6756*						
GFREE	14103	312	6897	6957	7077	7081	7180		
GGET	10065	5736	6236*						
GGROW	000001	102	103*	2263	2321	2326			

GHDOLD	00255	290	1067*	1069	2556	2882	6570	6744	6903
		7080	7100	7752	8421				
GHEAD	00253	290	1065*	2552	2878	6572	6576	6577	6732
		6743	6884	6902	7078	7098	7735	7754	8416
GHI	00252	290	1064*	2550	2876	6579	6716	6718	6722
		6885	7160	7170	8348	8350	8357	8361	8415
		8423	8440	8493					
GHIGH	10607	6668*	6683						
GHIGH0	10603	6613	6664*						
GHIGH1	10610	6669*	6688						
GHIGH2	10620	6677*	6695	6698					
GHIGH3	10634	6676	6690*						
GHIGH4	10645	6691	6699*						
GINDEX	11326	6540	6550	6566	6578	6600	6611	6621	6657
		6664	6737	6766	6837	6838	6844	6856	6890
		6891	6894	6914	7054*				
GKILL	10104	6259*	7782						
GKILL2	010105	6260*	6466						
GKILL3	10110	6264*	6489						
GKILL4	010112	6266*	6591	7038					
GLFNCT	11327	290	6531	6558	6648	6756	6779	6813	6828
		6911	6928	7040	7055*	7095	7109	7756	
GLOBAL	00272	278	290	1086*	2650	2976	5449	5478	6093
		6238	6253	6262	6265	6581	6584	6589	7036
		7042	7045	7047	7734	7744			
GLSET	11426	6594	6601	6658	7092	7137*	7144		
GNEXT	10700	6622	6737*						
GNDSYM	10600	6568	6657*						
GNOWDS	11325	6778	6787	6790	6799	6802	6832	6840	6862
		6863	6918	6944	6960	7053*	7156	7163	
GO	12266	3465	7617*						
GUREG	01040	715	1566*						
GOCLK	01324	787	2080*	2084					
GODMP	01042	716	1568*						
GOERR	01036	747	1564*	2037					
GOFF	11412	6827	6855	6889	6901	6955	7066	7090	7120*
		7132	7186						
GOSNAF	01044	717	1570*						
G01	12326	7620	7651*						
G02	12317	7627	7642*						
GPOINT	11323	6898	6904	6958	7051*				
GPUT	10073	6246*	8152						
GSCRIP	10242	6381*	6448						
GSCRI1	10332	6410	6441*						
GSPACE	11436	6841	6919	6956	7153*	7159	7171	7187	
GSPAC1	11437	7154*	7178						
GSPAC4	11461	7162	7166	7173*					
GSPAC6	11467	7175	7180*						
GSYM	00270	291	1084*	2648	2974	5463	5476	6236	6246
		6259	6390	6398	6415	6456	6464	6469	7698
		7711							
GSYMBO	00254	291	1066*	2554	2880	6118	6552	6553	7697
GSYMO	10231	6355	6366*	6461					

GSYM1	10235	6371*	6483						
GSYM10	10330	6418	6436*						
GSYM11	10375	6320	6479*						
GSYM2	10204	6342*	6460						
GSYM21	10214	6341	6351*						
GSYM3	10350	6336	6458*						
GSYM4	10354	6373	6462*						
GSYM5	10363	6463	6469*						
GSYM6	10402	6467	6484*						
GTABLE	14104	321	6181	6300					
GTEMP	11324	6932	6945	6962	6963	6969	7052*		
GTHRED	11225	6610	6830	6987*					
GTHR2	11254	7011*	7025						
GTHR3	11262	7001	7017*						
GTHR4	11301	7022	7032*						
GTMBUF	14105	321	6606	6769	6848	6927	6935	6941	6965
		6987	6989	7032					
GTMR.1	14106	321	6933	6959					
GUND2	11102	6884*	6905						
GUND3	11114	6894*	6920						
GUND4	11122	6900*	6971						
GUND5	11130	6730	6911*						
GUND6	11142	6742	6926*						
GUND7	11161	6941*	6978						
GUND8	11175	6949	6954*						
GUND9	11217	6938	6973*						
GUNIT	00247	291	408	1061*	2544	2870	6575	6996	7007
		7015	7018	7030	8367	8383	8405	8465	
GXTBAD	11304	6702	6913	6930	7035*				
GOOFF	10572	6588	6647*						
HALT	14107	312	3466						
HALTER	14110	312	2026						
HALTON	00664	291	1356*						
HALT1	14111	312	3072						
HANG	01533	291	441	2212	2226	2240*	2243	2393	3362
		3511	3533	4012	4111	5094	5678	6135	6214
		7046	7377	7668	7767	8426			
HANG.2	01535	2220	2242*						
HANG.4	01545	2217	2227	2253*					
HANG1	12331	291	7656*						
HANG2	12345	7659	7668*						
HFGLO1	14112	314	6417						
HKBENT	14113	314	2046						
HLOCK	000000	265	266*	1356					
HLTSWT	00663	291	1355*	2273					
HOMCHR	000035								
HOUT	14114	314	2158						
HTPENT	14115	314	2060						
ICHARI	04725	292	344	2622	2948	3284	3330	3938	3975
		3988	4189*	4200	4208	4215	5906	6027	6045
		7900	7917	7926	8130				
ID	00240	292	1054*	2530	2856				
IENTRY	14116	471	1633						

IENTO	04727	4191*	4219						
IENT1	04744	4201*	4217						
IENT2	04756	4209*	4218						
IF	12347	3467	7672*						
IFSWT	00246	1060*	2542	2868	7675	7676	7684		
IF2	12351	7674*	7681						
IF3	12353	7673	7676*						
ILLINT	01253	768	769	770	771	776	782	784	785
		792	794	795	800	806	808	809	810
		811	816	822	824	825	830	836	842
		848	854	860	866	872	878	1981*	1987
IN	03365	442	3284*						
INDIR	03525	3389	3397*						
INDTAB	01007	292	1375	1563*	2348	2690	3131		
INPUT	00163	442	465	1004*	1140	2432	2758	3142	3257
		5583	5666						
INSERT	04535	3959	4029*						
INSNUM	06030	4958	4969*						
INSSTR	06046	4957	4965	4986*					
INTAB	00476	1202*	5667						
INTAC	00560	1226*	1578	1796	1809	1830	1852	1874	1896
		1918	1940	1962	1996	2018	2019	2083	2214
		2218							
INTERU	001046	714	1573*						
INTL	00563	1229*	1584	1790	1817	1838	1860	1882	1904
		1926	1948	1970	2004	2091	2206		
INTMQ	00561	1227*	1580	1794	1813	1834	1856	1878	1900
		1922	1944	1966	2000	2087	2208		
INTOFF	705522	116*	241*	1575	1807	1828	1850	1872	1894
		1916	1938	1960	1982	1994	2081	2241	2294
		2387	3104	3119	3499	3540	5679	5841	7662
INTON	705521	115*	240*	1800	1985	2025	2035	2256	2316
		2338	3107	3122	3503	3523	3546	5685	5861
INTSCO	01110	376	833	1806*	1810				
INTSC1	01133	378	839	1827*	1831				
INTSC2	01157	381	845	1849*	1853				
INTSC3	01203	384	851	1871*	1875				
INTSC4	01227	387	857	1893*	1897				
INTX	00562	1228*	1582	1792	1815	1836	1858	1880	1902
		1924	1946	1968	2002	2089	2204		
IN2	03375	3288	3292*						
IQIMP	00564	1230*	2098	2105	2110	2388	2391		
IOUT	03101	3055	3073*						
IPIH	701764	667*							
IPLACE	00221	292	1039*	2500	2826	3296	3304	3308	3316
		3375	3922	3926	4190	4194	4201	4209	5706
		5911	5985	6011	6019	6032	6046	7202	7891
		7905	7922	7990	8074	8089	8119	8126	
IPDINT	00220	292	343	1038*	2498	2824	3030	3031	3292
		3300	3313	3314	3377	3932	4195	4196	4199
		4205	4207	4212	4214	4775	4840	4851	4992
		5001	5708	5907	5982	6022	6028	6051	6053
		6055	6058	7204	7888	7901	7918	7932	7997

		8076	8091	8121	8134				
ITEXT1	06264	5128	5167*						
IVAR	00143	292	962*	5651					
JBSPTR	00646	1340*							
JOIN	12366	3468	7691*						
JOIN0	12370	7693*	7770						
JOIN1	12424	7721*	7732						
KBOUT	14117	374	1202						
KEEP20	00573	1240*	3047	3049					
KEY	03336	292	3261*						
KEYCHK	05272	292	2642	2968	4496*	4497	4498	4502	4504
		4505	6157	6351	6360	8351			
KEYCH2	05302	4500	4504*						
KEYCR	03430	3266	3323*						
KEY.2	03340	292	3263*						
KILL	12506	3469	7774*						
KILL1	12513	7775	7780*	7794					
KILL15	12527	6263	6267	7792*					
KILL2	12530	7793*							
KRB	700312	572*	2043	2052					
KSF	700301	571*	1731						
KVAR	00137	292	958*	6149	6326	8354			
LDMM	700024	662*							
LETTER	07261	5725	5775*						
LETT0	07263	5777*	5793						
LETT1	07274	5780	5789*						
LETT2	07272	5767	5784*						
LETT3	07264	5778*	5798						
LINE	06763	5497	5553*						
LNTBPT	00707	293	1382*						
LOC	00206	293	1024*	2470	2796	3381	4362	4391	4417
		4423	4431	4437	4439	7233	7255	7269	7299
		7304	7312	7331	7448	7487	7612	7631	
LOCK	14120	314	3470						
LOKDI	03701	441	2639	2965	3539*	3547			
LOKDI1	03675	3533*	3543						
LOKUP1	07325	5828*	6241						
LOOKUP	07322	5785	5825*	5968					
LOSE	06342	5216*	5290	5293	5296	5303	5312	5324	5409
LPAREN	06375	5123	5253*						
LPSF	706501	641*	1743						
LPSF.2	703501	646*	1752						
LPSTA	00642	482	1333*						
LPSTA2	00643	1334*							
LP15	000000	124	125*	254	479	802	805	1156	1182
		1208	1262	1405	1447	1450	1740		
LP15.2	000000	127	128*	254	489	826	829	1159	1185
		1211	1265	1405	1454	1457	1751		
LR	00161	989*	2262	2414	2740	3007			
LVAR	00145	293	964*	3352	3939	4364	4387	4413	4424
		4426	4428	4430	7234	7256	7267	7305	7308
		7330	7344	7618	7629	7827	7955	7970	8064
M	00214	293	1034*	2490	2816	3716	3729	3735	3750

		3753	3773	3780	4070	4465	4472	4514	4523
		4526	5777	5791	6278	6284	6287	6330	6331
		6332	6333	6352	6357	6359	6361	7408	7410
		7996							
MBUF2	04573	2659	2985	4066*	4083	7842	7978		
MDUMP	14121	334	1475						
MEXEC	03055	295	3049*						
MFUNCT	14122	343							
MINUS	07177	5714*	5891						
MODFY	12533	3472	7821*						
MODIFY	000001	199	200*	1097	2655	2980	3471	3474	4060
		7797							
MODMET	12535	7823*	8006						
MONEY	000001	147	148*	154*	339	1026	1412	1415	2475
		2800	4645	5088	6072				
MONEYM	14123	349	1413						
MONIT	00661	293	1353*	2400	2672	2694			
MONTH	14124	321	2132	2141					
MONTHY	00575	367	1243*	2047	2061	2171	2174		
MONYSW	00210	356	1027*	2477	2803	4646	5089	6074	
MOVED1	04562	4047	4050*						
MOVEDN	04550	4040*	4049						
MPCNE	701744	666*	2032						
MPEU	701742	661*	3001	3064					
MPSNE	701741	665*	1777						
MQ	00157	987*	2258	2410	2736	3003			
MSRUF	04610	4079*	4084						
MSTACK	03766	3604	3617*						
MSTKR	04012	3637*	3647						
MSTR	07657	6065*	6091						
MTA	000001	162*	254	461	778	781	1165	1166	1191
		1192	1217	1218	1253	1403	1440	1443	1629
MTA7DN	000000	159	160*	162					
MTA9	000001								
MTA9DN	000001	156	157*	162					
MTBUFF	00667	465	1361*						
MTCOL	000073	243	244*	246	247*	1409			
MTEQM	00674	466	1371*						
MTERRS	00630	465	1316*						
MTGET	14125	466	1219						
MTQVFL	00732	467	1409*						
MTPUT	14126	466	1193						
MTSTAT	00627	465	1315*						
MTTRAN	00604	465	779	1254*					
MTUN1	00631	465	1320*						
MTUN2	00632	1321*							
MTUN3	00633	1322*							
MTUN4	00634	1323*							
MTUN5	00635	1324*							
MTUN6	00636	1325*							
MTUN7	00637	1326*							
MTUN8	00640	1327*							
MTNHD1	00406	465	1167*						

MUBUFF	14127	320	1476						
MUDC	14130	375	1420						
MUDT	14131	451	1463						
MULO	07345	5844*							
MUL2	07352	5846	5849*						
MUL3	07366	5850	5860*						
MUMPS2	14132	313	1410						
MUMT	14133	472	1441						
MUPT	14134	326	1460						
MURP	14135	414	1427						
MVAR	00141	293	960*	3176	5516	5520			
NBPNT	00653	293	1347*	7064	8463	8468	8474		
NEXMR	01262	1779	2032*						
NEXT	06777	5530	5536	5550	5557	5568*	8197	8204	8211
		8220							
NEXT1	07011	5578*	8340						
NEXT2	07010	5572	5574	5577*	8038	8107			
NFLAG	01073	408	425	1785*	1819	1840	1862	1884	1906
		1928	1950	1972	2006	2093			
NKDCHK	10672	6711	6728*						
NMCR	00727	1406*							
NMDC	00730	1407*							
NMDT	00725	1404*							
NMDUM	00731	1408*							
NMGBL	00711	1384*							
NMLP	00726	1405*							
NMMT	00724	1403*							
NMRF	00712	1385*							
NMRP	00713	1386*							
NOARR	06022	4722	4734	4960*					
NOFLAG	00644	329	442	1338*	1785	1983			
NOG	000074	1384							
NOSTAC	04023	3636	3646*						
NOSYM	05547	4696	4740*						
NSTACK	04055	3665	3677*						
NSTKR	04071	3689*	3697						
NUMAT	000051	256*	1149	1175	1201	2167	3156	3286	
NUMDEF	05637	4810*	4981						
NUMELE	006137	5016	5050*						
NUMTI	000050	253*	254	256	2050	2064			
NUMVAR	06152	4868	5073*						
NUMV1	06157	5074	5078*						
NUSEDK	001504	2197	2200*						
OBPNT	00171	293	1010*	2444	2770	6723	7079	7099	8385
OCHAR	04645	293	344	2584	2910	3432	3433	3438	3444
		3897	3904	3936	3970	3974	3980	3982	3984
		3986	3994	3996	4015	4024	4031	4123*	4127
		4131	4479	4558	5133	5172	5221	5370	5378
		5568	5870	5875	5887	5952	6039	6076	6151
		6169	6221	6338	6370	6411	6446	6458	7336
		7379	7438	7451	7545	7573	7679	7768	7792
		7830	7949	8004	8035	8106	8181	8336	8362
OCHARI	04660	293	2586	2912	3386	3413	3424	3434	3558

		3720	3978	3981	3985	3990	3995	4035	4038
		4048	4050	4137*	4142	4148	4155	4482	4515
		4532	4562	4571	4611	4615	5119	5160	5193
		5271	5494	5538	5546	5556	5719	5746	5750
		5758	5761	5778	5901	5967	6041	6090	6107
		6173	6225	6279	6293	6424	6455	7284	7407
		7454	7460	7466	7470	7476	7588	7595	7693
		7707	7780	7843	7895	7952	8080	8095	8150
OCTOUT	04320	297	2612	2938	3871*	3886	8210		
OCT2	04325	3876*	3885						
OFLAG	00213	1033*	2488	2814	6318	6344	6349	6391	6397
		6425	6445	6481					
OF2	14136	314	6427	6478					
OF3	14137	314	6583						
OF5	14140	314	6430	6471					
OLAY	14141	314	6112						
ONLYDM	06124	5032	5038*						
OPEN	07677	3478	6105*						
OPEN0	07707	6109	6114*						
OPEN1	07714	6119*	6140						
OPEN2	07747	6127	6147*						
OPEN3	07775	6169*							
OPEN4	07742	6129	6142*						
OPEN5	07715	6120*	6175						
OPLACE	00223	297	1041*	2504	2830	3385	3557	3917	4126
		4138	4141	4151	4168	4171	4177	4609	5330
		5346	5408	6116	6139	7211	7214	7222	7236
		7258	7273	7283	7303	7328	7447	7488	7587
		7635	7856	7881	7911	7940	7947	7987	7992
		8002	8070	8072	8087	8104			
OPNCNT	00670	297	1362*	6145	6146	6168	6201	6203	
OPDINT	00222	297	1040*	2502	2828	3383	3555	3930	4124
		4139	4149	4152	4174	4175	4434	4435	4607
		5328	5344	5406	6114	6137	7210	7216	7223
		7235	7257	7271	7281	7301	7329	7446	7489
		7586	7633	7854	7879	7909	7942	7945	7985
		7995	8000	8069	8077	8092	8105		
OP1	07332	297	5163	5804	5833*				
OP2	07377	5839	5868*						
OTEXT	03712	297	2630	2956	3553*	3560	7703	7724	8226
		8230							
OUT	03174	297	2620	2946	3056	3061	3062	3146*	3204
		3209	3339	3341	3343	3356	3357	3366	3368
		3370	3403	3561	3823	3836	3856	3880	5129
		5175	5182	5203	5208	5533	5543	5555	5596
		5606	5628	5637	5640	5669	5670	5680	5684
		7720	7728	8033	8205	8216	8238	8245	8272
		8278	8280	8288	8295	8297	8330	8332	
OUTPUT	00164	297	344	443	1005*	1140	2434	2760	3142
		3149	3162	3199	3244	5581	5664		
OUTTAB	00414	297	1176*	1343	3156	5580	5663		
OVERLA	07704	6110*							
OVER1	005465	4668	4674*						

OVER2	05472	4670	4682*						
OVER3	05476	4677	4687*						
P	00300	1104*							
PACKEM	03325	297	2626	2952	3249*	3331	3332	3348	3378
		7205	8078	8093	8122				
PAGE	06764	5501	5554*						
PART	05151	299	2596	2922	4386*	4407	5608	5609	7297
		7384	7647	8327					
PASTAR	14142	321	1376	2714	3134				
PBDLX	00275	298	1089*	2669	2995				
PC	00155	984*	1372	2253	2406	2689	2713	2715	2716
		2732	3059						
PERCEN	07260	5732	5771*						
PERIOD	000003	259*	1398						
PERMSY	000041	1103	1561*						
PFAIL	000000	95	96*	788	791	1587	1991		
PFSF	703201	658*	1588	2005					
PLACE	04340	2582	2908	3397	3896*	3909	6993	6998	7002
		7004	7292	7912	7946	8100			
PLACE1	00226	299	1044*	2510	2836	4231	4233	4242	5170
		5428	5436	5462	5477	5592	7857	7880	8201
PLACE2	00231	299	1047*	2516	2842	4260	4262	4271	5181
		5331	5345	5407	5448	5479	5633	5978	7848
		7870	7873						
PLACE3	04452	3978*	3989						
PLACE4	04476	3992	3998*						
PLACE5	04473	3977	3995*						
PLACE6	04542	3973	4034*						
PLACE7	04446	3974*	4021	4028	4058				
PLACE8	04527	3966	4023*						
PLAC1	04357	2652	2978	3908	3916*	4011	4013	7943	7998
PLAC3A	04453	3976	3979*						
PLAC4A	04510	4003	4008*						
PLAC4B	04512	4001	4010*						
PLENG	00701	299	1376*						
PLUS	07176	5713*	5889						
POINT1	00225	299	343	1043*	2508	2834	4232	4240	4243
		5168	5433	5434	5460	5481	5590	7855	7878
		8199							
POINT2	00230	299	1046*	2514	2840	4261	4269	4272	5179
		5329	5343	5405	5446	5483	5631	5976	5989
		7852	7868	7875					
PODLRP	000062								
PRDCNT	00276	298	1090*	2667	2993				
PRGDRT	00722	299	1399*						
PRINT	13020	3479	8011*						
PRINT0	13023	8014*	8037						
PRINT1	13050	299	8025	8032	8035*				
PRINT2	13045	8028	8032*						
PRINT3	13037	8021	8026*						
PRNCDV	00162	299	1003*	1069	2430	2756	3354	5578	5661
		5674	8019						
PRDDA	03641	299	2580	2906	3498*	3504	3525	6119	6206

		6216	6524	8384	8414				
PRDDRT	03657	2354	3512*						
PRDD1	03650	3501	3505*						
PRDD15	03656	3511*	3518						
PRDD2	03661	3508	3514*						
PRDD3	03666	3510	3515	3519*					
PRDG	00236	300	1052*	2526	2852				
PRTEND	00154	300	977*	2727	3026	3067	3329	7200	8117
PRTIDX	00657	300	1351*	2274	2318	2679	2687		
PRTIN	00660	300	1352*	2371	2673	2680			
PRTNMB	00654	300	409	425	441	1348*	2195	2275	2288
		2292	2300	2322	2327	2345	2347	2357	2372
		2390	2392	2678	3035	3037	3041	3042	3068
		3358	3365	3509	3519	3534	3544	5671	5683
		7663	7665						
PRTNUM	000027	1137	1341	1562					
PRTPIR	00714	300	1387*						
PRTSIZ	002114	1374							
PRVDKA	00601	1247*	2194	3521					
PSF	700201	581*	1771						
PSIZE	00677	1374*							
PSTART	00700	1375*							
PTEMPX	00274	298	1088*	2665	2991				
PTPIN	14143	325	1181						
PTPOUT	14144	325	1773						
PTROUT	14145	325	1207						
PTTRAN	00602	300	786	1249*					
PTWHD	00405	300	1155*						
PUSER	00237	301	1053*	2528	2854				
PUNUM	06164	5069	5085*						
PVAR	00140	959*	3225						
PWRITE	000000	205	206*	1401					
Q	000170	2474*	2476*	2476	2478	2481*	2481	2483	2484*
		2484	2486	2489	2491	2493	2495	2497	2499
		2501	2503	2505	2507	2509	2511	2513	2515
		2517	2519	2521	2523	2525	2527	2529	2531
		2533	2535	2537	2539	2541	2543	2545	2547
		2549	2551	2553	2555	2557	2559	2561	2563
		2565	2567	2569	2571	2573	2575	2577	2579
		2581	2583	2585	2587	2589	2591	2593	2595
		2597	2599	2601	2603	2605	2607	2609	2611
		2613	2615	2617	2619	2621	2623	2625	2627
		2629	2631	2633	2635	2636*	2636	2638*	2638
		2640	2643	2645	2647	2649	2651	2653	2654*
		2654	2656*	2656	2658	2660	2661*	2661	2664
		2666	2668	2670	2799*	2801*	2801	2802	2806*
		2806	2807	2809*	2809	2810	2813	2815	2817
		2819	2821	2823	2825	2827	2829	2831	2833
		2835	2837	2839	2841	2843	2845	2847	2849
		2851	2853	2855	2857	2859	2861	2863	2865
		2867	2869	2871	2873	2875	2877	2879	2881
		2883	2885	2887	2889	2891	2893	2895	2897
		2899	2901	2903	2905	2907	2909	2911	2913

		2915	2917	2919	2921	2923	2925	2927	2929
		2931	2933	2935	2937	2939	2941	2943	2945
		2947	2949	2951	2953	2955	2957	2959	2961*
		2961	2963*	2963	2964	2967	2969	2971	2973
		2975	2977	2979*	2979	2981*	2981	2982	2984
		2986*	2986	2988	2990	2992	2994		
QF7	14146	315	6419						
QF8	14147	315	6435						
QUEST	06713	5503	5510*						
QUIT	13054	3480	8041*						
QUOTE	06747	5499	5538*	5544					
QUOTE1	06756	5542	5546*						
R	000000	687*							
RDCLK	701762	664*							
RDCNT	00176	301	1016*	2454	2780				
RDCNTX	14150	315							
RDMM	700032	663*							
READ	13073	3481	8061*						
READ0	13105	8071*	8101						
READ0A	13142	8083	8100*						
READ0B	13144	8099	8102*						
READ1	13152	8063	8111*						
READ1A	013166	367	8123*						
REGS	000014	999*	1101						
REL2	06340	5143	5205	5214*					
REMOVE	10005	301	2663	2989	3812	3819	3838	3847	6130
		6180*	6185	6207	7706	7716	7722	7723	7730
REMOV1	10010	6183*	6197						
REMOV2	10021	6192*	6204						
REMOV3	10027	6191	6198*						
REPCH	12641	7895*	7904	7908					
REPTRA	05067	4315*	4323						
REP2CH	12663	7913*	7921	7925					
RESTOR	03742	3580	3588*						
RFADR0	00614	425	1291*						
RFADR1	00615	425	1292*						
RFERRS	00616	425	1293*						
RFOPTD	000002	193	194*						
RFOPTG	000004	196	197*						
RFSTAT	00613	425	1290*						
RF15	000000	421	812	815	1256	1385	1433	1436	1613
		8345	8477	8480	8495				
RMOV	10013	6186*	6220						
RP	000003	212*	404	818	821	889	905	925	929
		1259	1386	1426	1429	1621	8406	8409	8410
		8464	8483	8486					
RPADDR	00622	409	1303*						
RPDRV	00620	409	1301*						
RPERRS	00623	408	1305*						
RPERST	00621	408	1302*						
RPFLAG	14151	413	1627						
RPOPTD	000002	187	188*						
RPOPTG	000006	190	191*						

RPSECT	000400	210*							
RPSTAT	00617	408	1297*						
RPTRAN	00605	409	819	1260*					
RP02	000003	212							
RP03	000000	108	109*	212					
RSET	12570	7853*	7876						
RSF	700101	580*	1765						
RUB	03441	3262	3334*						
RUNDDW	00656	301	409	1350*	2102	2182	2184	2255	3010
RUNNMB	00655	1349*	2682	6128	6166	6190	6335	6563	
R07	000007	724*	2096	2100					
R10	000010	728*	2416	2742					
R11	000011	729*	3019	3021	3023	3025			
R12	000012	730*							
R13	000013	731*	2418	2744					
R14	000014	735*	2420	2746	3944	3946	3948	3949	3952
		3960	3969	3987	3998	4000	4005	4008	4014
		4020	4022	4023	4029	4072	4082	4095	4104
		4107	4970	4977	4979	4996	5004	5006	7361
		7370	7373	7387	7894	7902	7919		
R15	000015	736*	2422	2748	3961	3962	3964	3967	3999
		4004	4006	4017	4026	4030	4034	4046	4053
		4078	4080	4099	4103	4105	4772	4786	4825
		4828	4837	4988	6117	6138	6761	6843	7366
		7369	7371	7737	7738	7750	7753		
R16	000016	737*	2424	2750	4094	4096	4098	4363	4372
		4373	4375	4378	4379	4394	4397	4399	4402
		4403	4433	4436	4600	4610	4693	4902	4928
		5180	5607	5611	5613	5616	5617	5629	6115
		6136	6667	6674	6692	6699	6760	6793	6797
		6809	6826	6868	6869	6934	6940	6943	6946
		6951	6954	6966	6967	6974	7268	7298	7313
		7314	7316	7319	7320	7360	7362	7365	7386
		7388	7390	7393	7394	7630	7841	7850	7885
		7966	8306	8307	8309	8312	8313		
R17	000017	738*	2426	2684	2721	2752	4067	4068	4071
		4073	4075	4079	4081	4602	4608	4661	4662
		4671	4679	4680	4682	4684	4687	4694	4706
		4707	4709	4714	4719	4720	4730	4732	4735
		4759	4763	4765	4781	4785	4803	4807	4809
		4819	4822	4829	4846	4850	4853	4869	4884
		4898	4907	4972	4978	4998	5005	5024	5046
		5047	5058	5075	5086	5178	5588	5598	5600
		5601	5832	5974	5998	6003	6545	6547	6557
		6561	6569	6571	6573	6595	6598	6615	6618
		6625	6627	6630	6632	6633	6668	6671	6679
		6681	6684	6686	6687	6745	6759	6768	6776
		6795	6796	6798	6803	6804	6805	6808	6812
		6818	6819	6821	6842	6845	6849	6853	6854
		6870	6877	6896	6899	6900	6931	6939	6947
		6950	6968	6973	7122	7139	7169	7177	7436
		7519	8214	8217	8221	8250	8257	8259	8268
		8269	8273	8281	8284				

R20	000020	744*	2036	2428	2686	2722	2754	3046	3050
		3198	3256	3285	3334	5579	5662	5673	7239
		8018							
R22	000022	754*							
R23	000023	755*							
R30	000030	756*							
R31	000031	757*							
R32	000032	758*							
R33	000033	759*							
R35	000035	760*							
R36	000036	761*							
R37	000037	762*							
SAVE	05040	301	344	2590	2916	3254	3293	4040	4042
		4051	4055	4292*	4298	4299	4307	4342	4355
		4459	4543	4577	4659	4664	4710	4916	4918
		4954	4974	4976	4980	5000	5002	5007	5030
		5033	5034	5254	5447	5482	5920	5937	6323
		6343	6375	6382	6422	6442	6453	6475	6485
		7209	7230	7252	7415	7445	7585	7611	7849
		7869	7874	7889	8068	8160	8222	8227	8241
		8246							
SAVPRD	00201	1019*	2460	2786	3505	3512	3524		
SBA	707761	2247*	2248						
SBACK	05042	4294*	4306						
SCAN	07203	5718*	5812	5821					
SCANCT	14152	374	2067						
SCAND	14153	374	1653	1662	1671	1680	1689	1698	1707
		1716	1824	1846	1868	1890	1912	1934	1956
		1973	2053						
SCAND0	07204	5719*	6043						
SCAN1	07211	5724*	5747						
SCL	704404	632*							
SCOUNT	00565	1231*	2279	2283	2291	2295	2296	2298	2299
		2307	2317	2349	2350	2351	2352		
SCR	704424	636*							
SCRIP	00200	301	1018*	2458	2784	4585	4697	4717	4728
		4744	4810	4854	5562	5784	5948	6394	6407
		6437	6443	7433	7509	7788	8176		
SCRIPS	00205	301	1023*	2468	2794	4538	4548	4579	4584
		4596	5561	7425	7787	8162			
SECOND	01531	301	2111	2112	2116	2230*			
SET	13213	3482	8148*	8183					
SETID	07107	302	2634	2960	5643*	5646	5649	5686	5688
		6316	6389	6399	6423	6433	6454	6477	6488
		7190	7704	7712	7714	7715	7721	7727	7731
		7736	7751	8011	8061	8187	8302		
SETPRT	03155	3096	3109	3127*	3141				
SET2	13244	6254	8181*						
SLASH	07315	5817*	5874						
SLASH1	07317	5813	5819*						
SLICE	000062	166	167*	259					
SLPCHK	03102	2268	3083*						
SLPCK1	03103	3084*	3095						

SLPCK2	03112	3091*	3102	3108	3124				
SLPCNT	00571	1235*	2149	2266	2269	3097	3113		
SLPHNG	03117	3088	3096*						
SLPTMP	00572	1237*	3084	3085	3091	3105	3106	3120	3121
		3128							
SLT	704422	635*							
SNAFU	14154	315	1571						
SPEC02	000035								
SPEC03	000037								
SPEC04	000014								
SPOINT	00166	302	1007*	2438	2764	3607	3619	3627	3633
		3663	3678	3679	4295	4300	4301	4324	4325
		4344	4348	4352	6315	6533	6535	6536	6537
		6539	6541	6728	6731	7702	7719		
SRR	704412	631*	1649	1652	1657	1661	1666	1670	1675
		1679	1684	1688	1693	1697	1702	1706	1711
		1715	1820	1823	1841	1845	1863	1867	1885
		1889	1907	1911	1929	1933	1951	1955	1973
		1977							
SSF	704401	630*	1647	1655	1664	1673	1682	1691	1700
		1709	1818	1839	1861	1883	1905	1927	1949
		1971							
SSL	704421	633*							
STACH0	01562	2269*	3094						
STACH1	01570	2275*	2358						
STACH2	01620	2299*	2309						
STACH3	001632	2287	2297	2311*					
STACH4	01571	302	2276*	2320					
STACH7	01623	2282	2303*						
STACH8	001557	302	2264*	2327*					
STACH9	01600	2283*	2305						
STACK	00165	302	1006*	2436	2762	3601	3625	3635	3666
		3680	3683	4308	4313	4345	6313	6538	7705
		7713							
STAR	07311	5810*	5872						
STATCH	01557	302	2266*	2329					
STATX	00330	302	1139*	2277	3093				
STAT1	00301	302	1138*	1340	2278	2348	2681	2688	3083
		3129							
STEMP	00566	1232*	3132	3137					
STEMP2	00567	1233*	3130	3133	3135	3136			
STEP	05123	302	2594	2920	4092	4361*	4366	4381	4429
		7266	7358	7628	7829	7840	7959	7967	7977
		7981	7986	7999	8334				
STEP1	05137	4370	4373*	4382					
STMDMP	00746	1475*							
STMLP2	000742	1453*							
STMONY	000734	1411*							
STMUBF	00747	1476*							
STMUCR	000745	1468*							
STMUDC	000735	1418*							
STMUDT	000744	1461*							
STMULP	000741	1446*							

STMUMT	000740	1439*							
STMUM2	00733	1410*							
STMUPT	00743	1460*							
STMURF	000737	1432*							
STMURP	000736	1425*							
STOR1	03744	303	2604	2930	3598*	3616	4009	4018	4027
		4039	4108	6990	6997	7008	7012	7016	7031
		7349	7374						
STOR2	04034	2606	2932	3660*	3676	4760	4782	4804	4820
		4847	4924	4973	4999	7777	7853	7867	7887
		7910	7944						
STRACC	000031	236*	1103	2707	3252	5710	5991	5994	7893
STRDEF	05704	4854*	5008						
STRING	000111	226	227*	229	230*	232	233*	235*	235
		236	246	247					
STR0	07432	5734	5898*						
STR0.1	07435	5901*	5910	5914					
STR1	07525	5831	5971*						
STR1.1	007612	5984	5988	6024*					
STR1.2	07601	6009	6015*						
STR1.4	07610	6014	6022*						
STR2	07612	6025*	6031	6035					
STR3	07625	6000	6023	6026	6039*				
STR3.1	07627	5905	6041*	6088					
STR3.2	07645	6048	6055*						
STR3.3	07646	6050	6056*						
STR4	007662	303	343	6071*					
SUBDSK	00717	1390*	2359	2364					
SUBEXP	07453	5728	5919*						
SUBMER	000001	105	106*	1390	1392	1395			
SUBNUM	000061	1093*	1095*	1095	1098*	1098	1101		
SUBOPT	04710	303	2588	2914	3167	3181	3205	3207	3404
		3437	4166*	4172	4178	4574	5124	5200	5506
		6110	7409	7783	8015	8153			
SUBQUE	000720	303	408	1391*	2339	2369	3500		
SUBSCR	07467	5783	5936*						
SUM	00215	303	1035*	2492	2818	4463	4474	5157	5712
		5881	5884	5892	5924	5941	5962	6386	6402
SVAR	00146	303	965*	3610	3612	3613	3670	3672	3673
		4303	4304	4349	4351	6548	6549		
SVE	05363	303	343	2632	2958	4570*	4603	4617	5176
SVE1	05421	4573	4607*						
SWAPSW	00652	303	1344*	2186	2189	3012	3353	3502	3522
		3945	4010	4100	4109	4649	5092	6212	7044
		7364	7375	7657	7765	8424			
SWAPSZ	000174	1101*	1103	2699					
SWIM	00675	1372*							
SWNUM	12745	7834	7966*						
SWPCNT	00173	303	1013*	2448	2774	3013			
SWPIN	02327	2375	2677*	2997					
SWPIN0	03040	3028	3034*						
SWPIN1	03064	3057*	3071						
SWPIN2	03067	3054	3061*						

SWPIN3	003000	2373	2376	2999*					
SWPIN4	02375	2720*	2723						
SWPOK	003052	3040	3044*						
SWPOK2	03062	3036	3055*						
SWPOUT	01725	2374	2399*	2402	2674				
SWPTR	00702	1377*							
SYM	00177	304	1017*	2456	2782	4583	4656	4761	4783
		4805	4821	4848	5560	5757	5775	5794	5797
		5944	5959	6122	6123	6124	6142	6143	6147
		6148	6158	6162	6164	6165	6167	6182	6183
		6186	6187	6188	6192	6193	6194	6198	6200
		6301	6302	6307	6317	6328	6376	6434	6476
		6486	7431	7508	7786	8177			
SYMBEG	00170	304	1009*	2442	2768	3605	3621	3675	3695
		4296	4321	4660	4758	4780	4802	4818	4845
		4912	4969	4995	7499	8213			
SYMBOL	05434	304	2600	2926	4587	4644*	4650	4655	4740
		4863	4874	4894	4925	4960	5011	5039	5096
		5098	5100	5567	5826	7435	7511	7791	8180
SYMBOLT	00135	304	953*	2683	2710	2717	2718	7776	
SYMEND	00153	976*	2725	4657	4695	7667	8219		
SYMEX	006166	4743	4766	4789	4832	4866	4883	4927	4963
		5014	5087*						
SYMS	00204	304	1022*	2466	2792	4529	4531	4578	4582
		4597	5559	6290	6292	6305	7430	7785	8161
SYNTAX	03724	304	344	2602	2928	3387	3414	3574*	3581
		3587	3589	3714	3718	3721	3728	3732	3743
		3755	3759	3762	3775	3777	3796	3899	4516
		4533	5335	5373	5383	5400	5416	5724	5751
		5762	5779	5789	5796	5803	6280	6294	7285
		8081	8096						
SYSID	00611	1285*							
SYSTAB	000606	719	1280*						
SYSVAR	000015	975*	1378	2685					
SYSVNM	00703	1378*							
S2BACK	05046	4298*	4326						
TALL	13301	8189	8213*						
TALL1	13303	8215*	8252	8254	8260	8286			
TALL2	13315	8225*	8240						
TALL4	13353	8257*	8266						
TARRAY	013357	8243	8263*						
TARR1	13364	8271*	8290						
TARR2	13400	8284*	8299						
TASTR	13407	8275	8292*						
TCF	700402	575*	2057						
TEMP	00202	304	343	1020*	2462	2788	4464	4473	5711
		5745	5836	5898	5908	5926	5943	5960	5971
		6029	6388	6400	7884	7933			
TEMPX	14155	315							
TEMP1	00224	304	343	1042*	2506	2832	4245	5171	5593
		7846	7898	8202					
TEMP2	00227	305	1045*	2512	2838	4274	5177	5332	5634
		5979	7862	7929	7930	7936	7941		

TERM	00216	305	1036*	2494	2820	4462	4475	5159	5717
		5838	5847	5854	5869	5878	5923	5940	5963
		6385	6403						
TIMESL	00721	1398*	3009						
PLS	700406	576*	2074	2156					
TPIN	14156	374	1176						
TRYNSC	12575	7858*	7864						
TSF	700401	574*	1734						
TSTR	07021	2646	2972	5587*	5602	8253	8298		
TT	00332	1150*							
TTALEM	000031	178	179*						
TTALFL	000012	181	182*						
TTCHAR	000031	175	176*	257					
TTCHR	000062								
TTSET	14157	374	1651	1660	1669	1678	1687	1696	1705
		1714	1822	1844	1866	1888	1910	1932	1954
		1976	2051	2066					
TTWORD	000015	257*							
TVAR	00147	966*	3016						
TYPE	13250	3483	8187*						
TYPE2	13263	8195	8198*	8209					
TYPE3	13274	8192	8207*						
T1	00243	1057*	2536	2862	7417	7437	7496	7521	7525
		7530	7535	7538	7540	7551	7554	7560	7567
		8054							
T2	00244	1058*	2538	2864	7418	7442	7522	7542	7552
		7559	7572	8053					
T3	00245	1059*	2540	2866	7419	7531	7558	7580	7583
		7594	7600	7608	8052				
UBACK	05105	4343*	4356						
UMINUS	07231	5721	5743*						
UNLOCK	14160	315	3484						
UNDT	06201	5108*	5121						
UNSAVE	05103	305	344	2592	2918	3029	3250	3301	4037
		4041	4044	4045	4054	4057	4293	4305	4341*
		4353	4354	4357	4471	4555	4595	4708	4721
		4733	4914	4917	4920	5150	5264	5461	5480
		5958	6081	6348	6367	6396	6432	7221	7327
		7486	7557	7604	8048	8103	8175	8224	8225
		8239	8270	8285					
UPLUS	07233	5723	5745*						
USABLE	00277	305	1103*	2709					
USER	00241	305	1055*	2532	2858				
USERS	00665	305	1357*						
VAL	00207	305	1025*	2472	2798	3900	3940	3954	4019
		4056	4091	4365	4368	4380	4389	4396	4406
		4418	4419	4427	4752	4764	4799	4943	4956
		4964	5067	5078	5085	5142	5161	5162	5319
		5321	5450	5456	5464	5471	5475	5485	5524
		5531	5534	5620	5622	5718	5744	5834	5835
		5844	5851	5868	5893	5925	5942	5961	6066
		6150	6159	6324	6327	6353	6362	6368	6387
		6401	6782	6816	6820	6834	6850	7260	7311

		7323	7397	7497	7503	7739	7839	7937	7958
		7960	7979	7984	8144	8316			
VALF	00211	356	1030*	2482	2808	4544	4556	4748	4795
		4878	4882	4939	4949	5021	5063	5082	5564
		5701	6085	6639					
VARLEN	000001								
VARNUM	000074	1069*	1101						
VBLOCK	00750	305	1491*	1561	8515				
VEND	001006	306	1541*	1561					
V11	000001	202	203*						
W	000001	688*							
WATHNG	03134	3090	3109*						
WATH1	03152	3112	3116	3123*					
WHOPTR	00650	1342*							
WHOTAB	00331	306	1148*	1342	1343	3355	5668		
WIN	06343	5217*	5272	5295	5306	5310	5323	5347	
WMASK	03154	3118	3125*						
WRITE	13417	3486	8302*						
WRITE1	13441	8304	8320*	8338					
WRITE2	13460	8326	8334*						
WRITE3	13462	8311	8333	8336*					
WRITE4	13453	5615	8329*						
WRLINE	12735	7951	7955*						
WRPART	07041	306	2558	2884	5095	5099	5604*	5623	8317
		8328							
WRSTEP	07066	2628	2954	4952	5035	5038	5042	5610	5625*
		5641	7961	8335					
WRTCNT	00175	306	1015*	2452	2778				
WRTSW	00172	306	1011*	2446	2772	6131	6208	6607	6647
		6823	7106	7126	7130	7182	7745	7755	
WTIMER	000074	169	170*	1517					
WVAR	00144	306	963*	3032	7198	8115			
XBOOL1	06207	5117*	5248						
XBOOL2	06210	5118*	5239						
XBOOL3	06211	5109	5119*						
XCCDRW	13555	8366	8434*						
XCDIS	13600	8386	8417	8462*	8488	8491	8498	8508	
XCDIS1	13612	8467	8474*						
XCDIS2	13616	8472	8487*						
XCDIS4	13625	8476	8501*						
XCDKR	13527	8364	8401*						
XCDKR1	13537	8368	8414*						
XCDKW	13514	8359	8378*						
XCERR	13623	8470	8471	8481	8484	8493*	8499	8503	8507
XCH1.1	05003	4236	4240*						
XCH1.2	05010	4234	4239	4245*					
XCH2	04675	4144	4149*						
XCH2.1	05027	4265	4269*						
XCH2.2	05034	4263	4268	4274*					
XC11	04765	307	3305	3309	3374	3927	4193	4217*	5705
		5986	6017	6018	6047	7201	7890	7989	8073
		8088	8118						
XC12	04766	4218*	6017						

XCT3	04767	307	3297	3315	3923	4219*	5912	6010	6033
		6049	7906	7923	8127				
XCOM	13467	3487	8348*						
XCOMEX	13547	8388	8422*						
XCOREX	13550	8423*	8451						
XCO1	04704	307	3384	3918	4150	4157*	4169	4241	4270
		4599	5169	5429	5591	5632	5977	7213	7272
		7282	7302	7634	7847	7890	7939	7991	8071
		8086	8200						
XCO2	04705	3920	4158*						
XCO3	04706	3556	4159*	4176	5435				
XCO4	04707	4160*							
XDECM5	04105	3715*	3733						
XDEC00	04112	3720*	3734	3749	3754	3757			
XDEC2A	04246	3816	3819*						
XDEC2B	04253	3818	3824*						
XDISLO	14161	315	1081						
XDOUT	04206	3726	3777*						
XDOUTE	04207	3760	3763	3776	3778*				
XDOUT1	04222	3784	3789*						
XDOUT2	04227	3782	3794*						
XFORMA	06674	5494*	5570						
XFR	MACRO	672							
XFRDN	00100	306	890*	906*					
XFR.1	00024	749*	896						
XFR.2	00025	750*	900						
XFO	06675	5495*	5576						
XF1	06710	5506*							
XGASS	10113	1083	6277*						
XGLOBA	10410	1087	6524*						
XGSYM	10152	1085	6313*						
XLOKDT	03702	3537	3540*						
XOUT	03175	3147*							
XOUTDT	03260	441	3204*						
XOUT1	03214	3158	3166*						
XOUT10	03322	3151	3243*						
XOUT2	03225	3176*	3214	3218					
XOUT3	03255	367	3201*						
XOUT4	03233	3182*	3210						
XOUT4A	03253	306	2395	3198*	8034				
XOUT5	03236	3185*	3227	3231	3238				
XOUT5A	03241	305	3188*	3241					
XOUT5B	03247	3194*							
XOUT6	03267	3170	3212*						
XOUT7	03277	3172	3221*						
XOUT8	03314	3174	3235*						
XOUT9	03320	3165	3240*						
XPACK2	03334	2396	3256*	3272	3279	3283	3295	3299	3306
		3317	3337						
XPART1	05164	4397*	4408						
XPLAC2	04371	3921	3926*						
XPLAC3	04374	3919	3924	3929*					
XPLAC4	04373	3925	3928*						

.USER	MACRO				
.WAIT	MACRO				
.WAITR	MACRO				
.WRITE	MACRO				
.XVMDF	MACRO				
.XVMON	MACRO				
..A	000000	8345*	8496		
..B^	13635	306	8505	8511*	
..C	160000	261*	8346	8444	
..D	001600	8346*	8501	8504	
.32K	000040	262	263*		