```
LL                      RRRRRRRR    UU        UU  NN        NN              888888       0000      5555555
LL                      RRRRRRRRR   UU        UU  NN        NN            88888888     000000      5555555
LL                      RR      RRR UU        UU  NNN       NN          888     888   00     00     55
LL                      RR        RR UU       UU  NNN       NN          88       88   00     00     55
LL                      RR        RR UU       UU  NNNN      NN          88       88   00       00   55
LL                      RR      RRR UU        UU  NNNNN     NN           88     88   00       00   5555555
LL                      RRRRRRRRR   UU        UU  NN NNN    NN            888388    00       00   5555555
LL                      RRRRRRRR    UU        UU  NN  NNN NN             888888    00       00
LL                      RR RR       UU        UU  NN    NNNNN           88     88   00       00
LL                      RR  RR      UU        UU  NN     NNNN          88       88   00       00
LL                 **   RR   RR     UU        UU  NN      NNN   **     88       88   00     00
LL                ****  RR     RR   UUU      UUU  NN      NNN  ****    888     888   00     00     55
LLLLLLL LLLL      ****  RR      RR   UUUUUUUUU   NN       NN  ****     88888888     000000      5555555
LLLLLLLLLLL        **   RR       RR    UUUUUU    NN       NN   **       888888       0000        555555
```

```
  33333333
3333333333
33      333
          33
          33
        3333
     333333
     333333
        333
          33
          33
        333
33      333
3333333333
  33333333
```

File: 18/L.RUN.8053 Printed: MAR 8, 1988 0:51:27
File size: 559 blocks Copy 1 of 1

Printed for RELEASE CONTROLL

```
.EOT    ;IRIS UCL FILE FOR "RUN"
.EOT
.EOT    ;DECIMAL ARITHMETIC DEFINITIONS FOR R9.0
.EOT    ;  "BASIC/RUN" R9.0 DEFINITIONS
.EOT    ;  "RUN" R9.0 SOURCE #1
.EOT    ;  "RUN" R9.0 SOURCE #2
.EOT    ;  "RUN"  SOURCE #3
.EOT    ;  "RUN" R9.0 SOURCE #4
.EOT    ;  "RUN" R9.0 SOURCE #5
.EOT    ;  "RUN" R9.0 SOURCE #6
        .END    ;"RUN" R9.0 SOURCE #7
```

```
ASM ,@18/L.RUN.8053!,B330,-B331,B332,B333,B334
FEB 26, 1988  18:32:49


                    ;       Batchfile:   R92JCL.RUN

                    ;       ; D = 8053
                    ;       ; NAME = RUN
                    ;       ; TYPE = 33602
                    ;
                    ;
                    ;       -R92DEFSPZ
                    ;       -R90DECDEFSA
                    ;       -R92BRDEFSB
                    ;       R92RUNS1D
                    ;       R92RUNS2A
                    ;       R92RUNS3E
                    ;       R92RUNS4B
                    ;       R90RUNS5A
                    ;       R92RUNS6A
                    ;       R90RUNS7A
                    ;
                    .EOT   ;IRIS JCL FILE FOR "RUN"
```

```
                ;                              << SI = R92RUNS1D; BO = 18/A.RUN.8053! >>
                ;  "RUN" == "BUSINESS BASIC" INTERPRETER FOR "IRIS" R9.0
                ;                WRITTEN BY DAN PAYMAR
                ;
                ;  SOURCE #1 OF 7
                ;
                ;  July 6, 1982   (2187)   WAM   Modified for list protect
                ;  May 24, 1983   (3144)   TWM   Convert Enhanced BASIC to R8.2
                ;  June 7, 1983   (3158)   TWM   Correct bugs in R8.2 conversion
                ;  Oct  1, 1984   (4275)   MLR   Edited for R9.0
                ;  Mar  3, 1986   (6062)   PLR   Changed Polyfile Access Logic for Node relief
                ;  Nov  5, 1987   (7309)   MEP   Added control mnemonic capability from INPUT
                ;                                statements.  Removed 'RD' mnemonic.  Added
                ;                                translation of error code 143 to 90.
                ;  DEC 10, 1987   (7309)   RDC   INCORPORATE PATCHES




              1         .TXTM  1
            200         .LOC   IN+0-400

           1000 RUNRV =       2*K+0      ;RUN revision number, should be changed whenever
                                         ;  RUN is re-assembled or patched so that the structure
                                         ;  either RUN or the partition is changed.  The upper
                                         ;  byte of the revision number is incremented for a
                                         ;  re-assembly and the lower byte incremented for a
                                         ;  substantial patch (as described above).  The revision
                                         ;  number is included in the RUN parameter table.

    200        0         0       ;Address of "CALL" table in pre-R8.2 RUN

    201        0 D.BSC:  0       ;DISC ADDRESS OF "BASIC"
    202        0 D.MAT:  0       ;DISC ADDRESS OF "RUNMAT"

    203      477 .PRMT:PARMT     ;Pointer to the RUN parameter table

    204     6267 ERRSY:RUNERROR ;SYNTAX ERROR
    205   100510       1*K+SE

    206     6267 ERRST:RUNERROR ;ILLEGAL STRING OPERATION
    207   101110       2*K+SE
```

```
                ;                              << SI = R92RUNS1D; BO = 18/A.RUN.8053! >>

210       30 C30:   30      ;CONSTANTS
211      125 C125:  125
212      136 C136:  136
213      336 C336:  336
214      340 C340:  340
215      342 C342:  342
216      344 C344:  344
217      347 C347:  347
220      351 C351:  351
221      355 C355:  355
222      360 C360:  360
223      361 C361:  361
224      362 C362:  362
225      365 C365:  365
226      366 C366:  366
227      375 C375:  375
230      376 C376:  376

                ;LIST & STACK POINTERS FOR EVEX

231    43100 .EL:   EL
232    43450 .EN:   EN
233    41042 .ADIM: ADIM    ;POINTERS
234    37010 .CESF: CESF
235    32352 .CSEN: CSEN
236    33063 .EVCH: EVCHX   ;Routine to evalute channel and address specification
237    42232 .EVG:  EVG
240    41617 .EVP4: EVP4
241    42162 .EVQ:  EVQ
242    34605 .EVSE: EVSE
243    41264 .EVSU: EVSU
244    41277 .EV1:  EV1
245    40701 .FIXS: FIXS
246    33546 .EDST: ENDST   ;Check for end of statement and then exec next stmt
247    33536 .NXLN: NXLIN   ;Execute next line
250    32720 .SKST: SKSTM   ;Routine to over a BASIC statement to the next
251    40710 .LUVD: LUVD

                ; POINTERS TO TEMPORARY STORAGE AREAS

252      311 .TSX:  TSX
253      322 .TSF:  TSF
254      330 .TSE:  TSE
255      353 .TSN:  TSN
256    41230 .VLOC: VLOC
```

```
                ;                                  << SI = R92RUNS1D;  BO = 1B/A.RUN.8053! >>

      6257 ALGEX  =JSR  @. ;ALGEXPRESSION
257  41255        EVAX

      6260 EXPRE  =JSR  @. ;EXPRESSION
260  41263        EVEX

      6261 MTOST  =JSR  @. ;EMPTY OUTPUT STREAM
261  36162        FLUS.

      6262 GOGO   =JSR  @. ;GET OUTPUT GOING
262  36541        GOGO.

      6263 INTEG  =JSR  @. ;INTEGER EXPRESSION
263  40260        EVIN

      6264 NEXTB  =JSR  @. ;GET NEXT PROGRAM BYTE
264  32554        ACNB

      6265 QBYTE  =JSR  @.
265  36322        QBYT.

      6266 QTXT.  =JSR  @.
266  36142        QT..

      6267 RUNER  =JSR  @. ;RUN ERROR
267  32603        ERR

      6270 XFLUS  =JSR  @.
270  36467        XFLU.

      6271 SETBD  =JSR  @.
271  32524        BDSET
```

```
                ;                              << SI = R92RUNS1D; BO = 18/A.RUN.8053! >>

272       0 RFBA:  0        ;XOB FIRST BYTE ADDRESS
273       0 RLBA:  0        ;XOB LAST BYTE ADDRESS
274       0 EVSW:  0        ;EVALUATOR SWITCH
275       0 I:     0        ;EVALUATOR LIST COUNTER
276       0 J:     0        ;EVALUATOR NUMBER COUNTER
277       0 EVPF:  0        ;EVALUATOR PROCESS FLAG
300       0 EVLF:  0        ;EVALUATOR LEN FUNCTION FLAG

                ;TEMPORARY STORAGE AREAS

                ;TS:   .BLK 10    ;FOR INTERPRETER

301       0 TS:    0
302       0 TS1:   0
303       0 TS2:   0
304       0 TS3:   0
305       0 TS4:   0
306       0 TS5:   0
307       0 TS6:   0
310       0 TS7:   0

                ;TSX:  .BLK 11    ;FOR MATRIX ROUTINES

311       0 TSX:   0
312       0 TSX1:  0
313       0 TSX2:  0
314       0 TSX3:  0
315       0 TSX4:  0
316       0 TSX5:  0
317       0 TSX6:  0
320       0 TSX7:  0
321       0 TSX10: 0

                ;TSF:  .BLK  6    ;FOR FUNCTION ROUTINES

322       0 TSF:   0
323       0 TSF1:  0
324       0 TSF2:  0
325       0 TSF3:  0
326       0 TSF4:  0
327       0 TSF5:  0
```

```
                ;                              << SI = R92RUNS1D;  BD = 18/A.RUN.8053! >>

                ;TSE:  .BLK 10    ;FOR EXPONENTIATION "^"
330         0 TSE:   0
331         0 TSE1:  0
332         0 TSE2:  0
333         0 TSE3:  0
334         0 TSE4:  0
335         0 TSE5:  0
336         0 TSE6:  0
337         0 TSE7:  0
                ;TSP:  .BLK 13   ;FOR PRINT, IF, AND SIGNAL
340         0 TSP:   0
341         0 TSP1:  0
342         0 TSP2:  0
343         0 TSP3:  0
344         0 TSP4:  0
345         0 TSP5:  0
346         0 TSP6:  0
347         0 TSP7:  0
350         0 TSP10: 0
351         0 TSP11: 0
352         0 TSP12: 0
                ;TSN:  .BLK 11    ;FOR FILE TRANSFERS
353         0 TSN:   0
354         0 TSN1:  0
355         0 TSN2:  0
356         0 TSN3:  0
357         0 TSN4:  0
360         0 TSN5:  0
361         0 TSN6:  0
362         0 TSN7:  0
363         0 TSN10: 0
364         0 TSN11: 0        ;Used for delay lmt in EVCHX & RWIC; Also length in INPUT


365         0 ERPBC: 0        ;Value of starting PBC of stmt following erred stmt

366   1777/7 .BUS:  -1        ;BEGINNING OF USER STORAGE (SET BY SWAPI)
367   1777/7 .EUS:  -1        ;END OF USER STORAGE (SET BY SWAPI, SWAPO)
370         0 RRSZ: 0         ;REQUEST FOR MORE SPACE (SET BY ASKM,ALOCO - USED BY SWAPO)
371         0 CNODE:0         ;NODE ADDR POINTER (SET BY CALLC)


            0         .LOC    372-.     ;PAGE ZERO OVERLAP CHECK  * * *


          372         .LOC    372       ;SPECIAL ENTRY POINTERS

372    367/0 .FMRD: FINMRD    ;END of MAT READ from RUNMAT
373    32735 .RUN2: RUN2      ;CALCULATOR (EXECUTE) MODE
374    33562 .NXST: NXSTM     ;RETURN FROM "RUNMAT"
375    33605 .EXST: EXSTM
376    32735        RUN2      ;(LINE #) "RUN"


377   134431 D.EN:  101-EN
```

```
                        ;                               << SI = R92RUNSID; BO = 18/A.RUN.8053! >>
          400                   .LOC    400             ;MUST BE LOCATION 400
           65           .PTBL =         C400
400         1           PCTBL:  .BLK    1               ;BA(XOB) in partition
401         1                   .BLK    1               ;XOB size in bytes - 1

402         0           SWPOK:  0
403     54301           SWPI:   STA     3,TS            ;START UP AFTER SWAP-IN
404     5476            SWPI:   STA     3,SWPOK
405      6101                   CALL
406    100006                   LOADUSER
407      2476                   JMP     @SWOOP          ;Types don't match
410    102400                   SUB     0,0             ;CLEAR FLAGS
411      4071                   STA     0,SWPOK
412      4006                   STA     0,ERRF
413      4037                   STA     0,RRSZ          ;SIZE INCREASE VALUE
414      4037                   STA     0,CNODE         ;AND NODE POINTER
415      4103                   JSR     SATUP           ;Setup for running
416      2401                   JMP     @.+1            ;Finish up swap-in
417     32306                   SWPI2
```

```
                ;                                << SI = R92RUNS1D;  BO = 1B/A.RUN.8053! >>
                ; +
                ; **  Routine to setup .BUS, .EUS, and IOB/XOB related pointers

  420   54302 SATUP: STA     3,TS1       ;Save return
  421   32004        LDA     2,@PIB      ;A(PARTITION)
  422   50366        STA     2,.BUS
  423   24133        LDA     1,REVNO     ;CHECK REVISION NUMBER
  424   21000        LDA     0,REV.,2
  425 106654         SUBOR#  0,1,SZR     ;REV R8.0 OR LATER ?
  426    2427        JMP     @SWOOP      ;   NO
  427   34004        LDA     3,PIB       ;   YES, load address of Partition Info Block
  430   21401        LDA     0,SZP.,3    ;SIZE OF PARTITION
  431   34023        LDA     3,CM400     ;ROUND DOWN MOD 400
  432 117400         AND     0,3
  433 100400         NEG     0,0
  434 100000         COM     0,0
  435   41010        STA     0,EUS.,2
  436 143000         ADD     2,0         ;END OF PARTITION
  437   40367        STA     0,.EUS
  440   21051        LDA     0,XOB.,2    ;XOB word offset
  441 101125         MOVZL   0,0,SNR     ;XOB byte offset
  442    6142        TRAPFAULT
  443   40272        STA     0,RFBA      ;BA(XOB) in partition
  444   40734        STA     0,PCTBL     ;Setup $DEC output control block
  445   25062        LDA     1,XOBE,2    ;XOB end + 1 word offset
  446 125120         MOVZL   1,1         ;BA(XOB end +1) in partition
  447 106000         ADC     0,1         ;Size of XOB - 1
  450   44731        STA     1,PCTBL+1
  451 123000         ADD     1,0
  452   40273        STA     0,RLBA      ;Set partition BA(last byte of XOB)
  453    6131        LOADDA
  454    2302        JMP     @TS1        ;return

  455   32323 SWOOP: NOTBS               ;Not a BASIC file
  456 134261 REVNO: BSCRV                ;Revision number "81"
```

```
                        ;                                  << SI = R92RUNS1D; BD = 18/A.RUN.8053! >>
457   54330 SWPO:  STA     3,TSE      ;WRAP UP FOR SWAP-OUT
460   24722        LDA     1,SWPOK
461  125014        SKZ     1,1        ;WAS SWAP-IN SUCCESSFUL ?
462    1401        JMP     1,3        ;   NO, RELEASE PARTITION
463    6137        STORDA             ;   YES

                    ;   The following code was used with the R8.0 and R8.1 dynamic partition
                    ;   system.
                    ;
                    ;       LDA     3,.BUS
                    ;       LDA     0,NVS.,3 ;TRY TO GET 400 MORE WORDS
                    ;       LDA     1,RRSZ   ;REQUEST FOR MORE SPACE (IF ANY)
                    ;       ADD     1,0
                    ;       LDA     2,CM400  ;ROUND UP MOD 400
                    ;       ADC     2,0      ;(COM(-400)=add 377)
                    ;       SUB     2,0      ;(-(-400)=add 400)
                    ;       AND     2,0      ;MOD 400 (truncate to block boundry)
                    ;       LDA     2,C6     ;ALLOW FOR TABLE
                    ;       ADD     2,0      ;PART'N SPACE REQ'D BY USER
                    ;       LDA     3,.INFO  ;CALC PART'N SPACE AVAILABLE
                    ;       LDA     3,.UPT.,3
                    ;       LDA     2,EPA.,3
                    ;       SUB     3,2
                    ;       INC     2,2      ;AVAIL PART'N SPACE
                    ;       SGE     2,0      ;WILL THE USER SIZE + 400 FIT ?
                    ;       SUBZ    0,0,SKP  ;   Yes
                    ;       LDA     0,C400   ;   NO, ADD 400 TO REQUEST FOR MORE SPACE
                    ;       ADD     0,1

464   34356        LDA     3,.BUS
465   21407        LDA     0,NVS.,3
466  100400        NEG     0,0
467  100000        COM     0,0
470   41410        STA     0,EUS.,3 ;EUS=NVS-1
471  163000        ADD     3,0      ;CALC LAST ADDRESS
472   30104        LDA     2,RSVTB
473    6101        CALL
474      67        AFSETUP
475    2330        JMP     @TSE

476   32240 RSVTB: SVTB     ;RUN'S SAVE TABLE

                    ;   RUN parameter table  -  this table specifies installation
                    ;     configurable values for RUN.

      477 PARMT:
477   1000 PREVN: RUNRV     ;RUN revision number


       100          .LOC    INFO-.    ;PAGE ZERO OVERFLOW CHECK * * *
```

```
                        ;                               << SI = R92RUNS1D;  BO = 18/A.RUN.8053! >>
          32200                 .LOC    BPS             ;ENTRY POINTERS

 32200      403                 SWPI    ;SWAP-IN
 32201      457                 SWPO    ;SWAP-OUT
 32202    37023                 ESCP    ;ESCAPE
 32203    32505        CTCEN:   CTLC    ;CONTROL C

                        ;         The following .TXTF is the equivalent of:
                        ;                   MOVS    2,2,SNR
                        ;                   ADDOR   2,1
                        ;                   COMCR#  1,3,SNR
                        ;                   NEGCR     1,2
                        ;         Note:  This code is effectively a NOP
 32204  151325                 .TXTF   "RU
 32205  147240        N
 32206  134556        9.
 32207  130610        1
 32210  120000          "

 32211    34103                 IOCALL                  ;RESTORE IOB (FROM XMEM)
 32212     7741                 IOBRESTORE              ;TO PRESERVE REST OF COMMAND LINE FOR CALL 82
 32213     6271                 SETBD                   ;SET B & D BITS
 32214     6266                 QTXT.                   ;Output Initial CR
 32215  106100                 215*K
 32216     6262                 GOGO
 32217  102400        RUN1:    SUB     0,0
 32220    40076                 STA     0,ERRF
 32221     4574                 JSR     CVSP            ;CLEAR ALL VARIABLES & STACKS
 32222    30366                 LDA     2,.BUS
 32223     4404                 JSR     RESTR           ;"RESTOR" DATA statements (reset ptr&cnt)
 32224     2413                 JMP     @.EXLN          ;And execute first line


                        ;  RESTORE statement

 32225     4102        REST:    JSR     RESTR           ;Reset DATA statement ptr and data cnt
 32226     2374                 JMP     @.NXST          ;And then execute next statement

                        ;  RESTR  -  Reset pointer to current DATA statement and clear data
                        ;            count.  Uses a non-skip return.
                        ;
                        ;       JSR     RESTR

 32227    30366        RESTR:   LDA     2,.BUS
 32230    21005                 LDA     0,SLT.,2        ;"RESTORE" routine
 32231    41022                 STA     0,DSC.,2        ;Reset DATA stmt pointer to program start
 32232  102000                 ADC     0,0             ;=-1
 32233    41020                 STA     0,DWC.,2        ;Flag no reads yet
 32234     1400                 JMP     0,3             ;Return

                        ;  Entry Point for BASIC "calculator" mode

 32235     6271        RUN2:    SETBD                   ;SETUP
 32236     2401                 JMP     @.EXLN          ;  & GO
 32237    33542        .EXLN:   EXLIN

 32240      340        SVTB:    TSP                     ;Start of PZ save area
 32241       25                 .BUS-TSP-1              ;Area size - 1
```

```
                ;                                   << SI = R92RUNS1D; BO = 18/A.RUN.8053! >>
 32242 177763           177763
 32243  54307 FNDS:     STA     3,TS6       ;FIND NEXT DATA STATEMENT
 32244    403           JMP     FNDS2

 32245 126400 FNDS1:    SUB     1,1
 32246  45020           STA     1,DWC.,2    ;ASSUME NO DATA STATEMENTS
 32247  25001 FNDS2:    LDA     1,VDT.,2
 32250  35022           LDA     3,DSC.,2
 32251 166033           SLS     3,1         ;END OF SLT ?
 32252   2307           JMP     @TS6        ;   YES, NO DATA STATEMENT
 32253  11022           ISZ     DSC.,2      ;   NO, STEP DSC
 32254  11022           ISZ     DSC.,2
 32255 157000           ADD     2,3         ;GET DSC ABSOLUTE POINTER
 32256  35401           LDA     3,1,3       ;STATEMENT RELATIVE POSITION
 32257 165400           INC     3,1         ;Assume DATA statement
 32260 157000           ADD     2,3         ;ABSOLUTE STATEMENT LOCATION
 32261  45020           STA     1,DWC.,2
 32262  25400           LDA     1,0,3       ;FIRST WORD OF STATEMENT
 32263  20064           LDA     0,C377
 32264 123400           AND     1,0
 32265 106700           SUBS    0,1         ;STATEMENT TYPE BYTE
 32266  34211           LDA     3,C125
 32267 136414           SEQ     1,3         ;"DATA" STATEMENT ?
 32270    755           JMP     FNDS1       ;   NO
 32271  24062           LDA     1,C300      ;   YES
 32272 107400           AND     0,1         ;NUMBER TYPE * 2^6
 32273 122400           SUB     1,0         ;NUMBER OF DATA ELEMENTS
 32274  41061           STA     0,DEC.,2    ;SET DATA ELEMENT COUNTER
 32275 127120           ADDZL   1,1
 32276 127120           ADDZL   1,1
 32277 125300           MOVS    1,1         ;NUMBER TYPE * 2^2
 32300  21024           LDA     0,FLAG.,2
 32301  34741           LDA     3,FNDS-1
 32302 163400           AND     3,0
 32303 125000           ADD     1,0         ;NEW NUMBER TYPE IN FLAG WORD
 32304  41024           STA     0,FLAG.,2
 32305   2307           JMP     @TS6
```

TWH 07/15/88

(E) HSB 8/18/88

```
                        ;                              << SI = R92RUNS1D; 80 = 18/A.RUN.8053! >>

032306  030100 SWPI2:LDA     2,.INFO   ;   Yes
032307  021042         LDA     0,CPLU.,2;Current processor LU
032310  101113         SSN     0,0       ;DSP breakpoint set in RUN ?
032311  002301         JMP     @TS       ;   No
032312  030005         LDA     2,RUP
032313  006102         FLACCHANGE        ;Disable DSP
032314  100012         RESET+FLW.
032315  130000         130000
032316  034004         LDA     3,PIB
032317  031403         LDA     2,AFP.,3  ;A(AF Header)
032320  020016         LDA     0,BPI     ;Original BP contents
032321  043052         STA     0,@DSPS+2,2
032322  002301         JMP     @TS

                        ;***** I M P R O P E R   A C T I V E   F I L E ******


032323  006103 NOTBS:OUTTEXT
032324  106607 .TXTF "<215><207>
032325  137316 >N
032326  147724 OT
032327  120301  A
032330  120302  B
032331  140723 AS
032332  144703 IC
032333  120306  F
032334  144714 IL
032335  142610 E
032336  147722 OR
032337  120327  W
032340  151317 RO
032341  147307 NG
032342  120326  V
032343  142722 ER
032344  151711 SI
032345  147716 ON
032346       0 "

032347  006141         STOUTPUT
032350  006101         CALL
032351  100000         SCOPE
```

```
                        ;                               << SI = R92RUNS1D; BO = 18/A.RUN.8053! >>

032352   24111 CSEN:  LDA     1,CSENT   ;CONVERT STATUS TO ERROR NUMBER
032353  167000        ADD     3,1
032354    6124        GETBYTE
032355   20402        LDA     0,..+2
032356  143301        ADDS    2,0,SKP
032357   44200        44200
032360   40402        STA     0,..+2
032361    6267        RUNERROR
032362  100110        SE

              12        .RDX 10

032363   64750 CSENT:..+1*2  ;ERROR NUMBER CONVERSION TABLE
                     ;    BASIC ERROR   ;SYSTEM ERROR NUMBER (ONE PER BYTE)
032364   16050        28*K+40 ;   0  1
032365   30501        49*K+65 ;   2  3
032366   41103        66*K+67 ;   4  5
032367   42101        68*K+65 ;   6  7
032370   27460        47*K+48 ;  10 11
032371   25451        43*K+41 ;  12 13
032372   15052        26*K+42 ;  14 15
032373   27516        47*K+78 ;  16 17
032374   26055        44*K+45 ;  20 21
032375   27052        46*K+50 ;  22 23
032376   42514        69*K+76 ;  24 25
032377   31573        51*K+123 ;  26 27
032400   32466        53*K+54 ;  30 31
032401   32106        52*K+70 ;  32 33
032402   43510        71*K+72 ;  34 35
032403   44512        73*K+74 ;  36 37
032404   46517        77*K+79 ;  40 41
032405   50121        80*K+81 ;  42 43
032406   51123        82*K+83 ;  44 45
032407   52101        84*K+89 ;  46 47
032410   53000        86*K+0  ;  50 51
032411   57502        95*K+90 ;  52 53
032412   52055        84*K+45 ;  54 55

              10        .RDX 8
```

```
                ;                                  << SI = R92RUNSID; BO = 18/A.RUN.8053! >>
 02413 115774/4         115774
 02414  77477           77477
 02415  54301  CVSP:    STA     3,TS       ;CLEAR VARIABLES & STACK POINTERS
 02416  30005           LDA     2,RUP
 02417 102400           SUB     0,0
 02420  41033           STA     0,OCC.,2   ;INITIALIZE OCC
 02421  34366           LDA     3,.BUS
 02422  25405           LDA     1,SLT.,3
 02423  45402           STA     1,PLC.,3   ;PROGRAM COUNTER
 02424  21412           LDA     0,UFS.,3
 02425  41413           STA     0,UFC.,3   ;USER FUNCTION STACK
 02426  25416           LDA     1,GSS.,3
 02427  45417           STA     1,GSC.,3   ;GOSUB STACK
 02430  25414           LDA     1,FNS.,3
 02431  45415           STA     1,FSC.,3   ;FOR-NEXT STACK
 02432  31411           LDA     2,UFT.,3   ;CLEAR TABLES AND STACKS
 02433 173000           ADD     3,2
 02434 167000           ADD     3,1
 02435 102400           SUB     0,0
 02436  41000           STA     0,0,2
 02437 151400           INC     2,2
 02440 146002           SGE     2,1
 02441      775         JMP     .-3
 02442  21424           LDA     0,FLAG.,3; INITIALIZE FLAG WORD
 02443  24750           LDA     1,CVSP-2
 02444 123100           AND     1,0
 02445 101400           INC     0,0        ;Default precision is 2%
 02446  41424           STA     0,FLAG.,3
 02447   4155           JSR     BDSET      ;Set B & D flag bits
```

```
                        ;                                << SI = R92RUNS1D;  BO = 18/A.RUN.8053! >>
032450  021454          LDA     0,FL1.,3
032451  101222          MOVZR   0,0,SZC   ;De-allocate variable storage?
032452  000427          JMP     CVSP2     ;   No
032453  102400          SUB     0,0
032454  041435          STA     0,RNOI.,3;Clear random number seed
032455  021406          LDA     0,UVS.,3  ;   Yes, reset Next Variable Storage
032456  041407          STA     0,NVS.,3
032457  031401          LDA     2,VDT.,3
032460  173000          ADD     3,2
032461  025404          LDA     1,UPS.,3  ;Compute end of VDT
032462  167000          ADD     3,1       ;A(UPS)
032463  176120          ADCZL   3,3       ;=-2
032464  167000          ADD     3,1       ;End of VDT = UPS - 2
032465  034727          LDA     3,CVSP-1
032466  000411          JMP     CVSP3

032467  021000  CVSP1: LDA     0,0,2     ;DEALLOCATE VARIABLE SPACE
032470  163400          AND     3,0
032471  041000          STA     0,0,2
032472  102400          SUB     0,0
032473  041001          STA     0,1,2
032474  041003          STA     0,3,2
032475  151400          INC     2,2
032476  151400          INC     2,2
032477  146032  CVSP3: SGE     2,1
032500  000767          JMP     CVSP1
032501  002301  CVSP2: JMP     @TS
```

```
                                                  << SI = R92RUNS1D; BO = 18/A.RUN.8053! >>
032502   37047          ESCP2
032503      20          EMOD1
032504   37046          ESCP1
032505   34366 CTLC:    LDA     3,.BUS    ;CONTROL C ENTRY
032506   25452          LDA     1,ERBP.,3
032507  125014          SKZ     1,1       ;ERROR BRANCH SET ?
032510     404          JMP     CTLC2     ;   Yes, check IF ERR mode
032511    4704          JSR     CVSP      ;   NO
032512    6101          CALL
032513  100000          SCOPE

032514   25460 CTLC2:   LDA     1,FL4.,3;Load flag word 4
032515   34366          LDA     3,CTLC-2
032516  137415          AND#    1,3,SNR   ;IF ERR 1 mode set?
032517    2765          JMP     @CTLC-1   ;   No, Report as error 99 (same as ESC)
032520   20402          LDA     0,C307    ;   Yes, Report as error 199
032521    2761          JMP     @CTLC-3

032522     307 C307:    307
```

```
                ;                                  << SI = R92RUNS1D; BO = 18/A.RUN.8053! >>
 02523  10200 BDMSK: 10200   ;Mask for security bits B & D

                ;Readout security bits B & D and copy them into TSU.4

 02524  54310 BDSET: STA     3,TS7      ;Save return
 02525  34366         LDA     3,.BUS
 02526  21424         LDA     0,FLAG.,3
 02527  25455         LDA     1,PSTS.,3
 02530  30773         LDA     2,BDMSK
 02531 143400         AND     2,0        ;Select B & D
 02532 147400         AND     2,1        ;  bits
 02533 131024         MOVZ    1,2,SZR    ;FLAG. XOR PSTS. [Ok if skip]
 02534 113500         ANDL    0,2
 02535 107020         ADDZ    0,1
 02536 146400         SUB     2,1
 02537  30022         LDA     2,C774C
 02540 147020         ADDZ    2,1        ;Force B bit to sign (15)
 02541 130400         NEG     1,2
 02542  51477         STA     2,TSU.4,3;Set B & D flag
 02543  20272         LDA     0,RFBA
 02544  41423         STA     0,ROBP.,3;INITIALIZE POINTER TO XOB
 02545 102400         SUB     0,0        ;CLEAR '@' SEEN FLAG
 02546  40350         STA     0,ATF
 02547  41457         STA     0,FL3.,3 ;Clear FL3 (Assume normal input)
 02550  41454         STA     0,INLEN,3
 02551 102000         ADC     0,0
 02552  41475         STA     0,TSU.2,3;INITIALIZE I/O TO TERMINAL
 02553   2310         JMP     @TS7


 02554  30366 ACNB:  LDA     2,.BUS     ;ACCESS NEXT PROGRAM BYTE
 02555  25003         LDA     1,PBC.,2
 02556  11003         ISZ     PBC.,2
 02557   2144         JMP     @XGETB&377;RETURNS DIRECTLY


 02560  54301 SNO:   STA     3,TS       ;STATEMENT NUMBER OUTPUT
 02561   6133         OUTTEXT
 02562 120341         .TXTF   ; a
 02563 172240 t
 02564      0 ;

 02565  30366         LDA     2,.BUS
 02566  35002         LDA     3,PLC.,2
 02567 157023         ADDZ    2,3,SNC    ;[Never skips]
 02570  25400         LDA     1,0,3
 02571 127024         ADDO    1,1,SZR    ;Force sign [will only skip on 0 or @0]
 02572 125423         MOVZR   1,1,SNC    ;  bit to 0 [Never skips]
 02573  20032         LDA     0,C12
 02574 152400         SUB     2,2        ;=0, No leading blanks
 02575   6101         CALL
 02576      7         CIA
 02577   6133         OUTTEXT            ;Terminate output
 02600 106400         215*K
 02601   2301         JMP     @TS
```

```
                ;                                  << SI = R92RUNS1D; BO = 18/A.RUN.8053! >>
032602   33732 ENDAO:ENDA1
032603   21400 ERR:   LDA     0,0,3      ;N*K!NOP
032604   30366        LDA     2,.BUS
032605   41073        STA     0,TSU.,2   ;REMEMBER THE ERROR #
032606   30371        LDA     2,CNODE
032607  151014        SKZ     2,2        ;IS A NODE IN USE ?
032610    6107        FREENODE           ;   YES, RETURN IT
032611  152400        SUB     2,2        ;CLEAR NODE POINTER
032612   50371        STA     2,CNODE
032613   30366        LDA     2,.BUS
032614   21073        LDA     0,TSU.,2   ;Recover error #
032615   30366        LDA     2,.BUS
032616  126400        SUB     1,1
032617   45057        STA     1,FL3.,2   ;Reset INPUT length
032620   24022        LDA     1,C774C
032621  107700        ANDS    0,1        ;EXTRACT ERROR NUMBER
032622   45051        STA     1,ERRN.,2
032623   35002        LDA     3,PLC.,2
032624  157000        ADD     2,3
032625   25400        LDA     1,0,3      ;SAVE LINE NUMBER
032626   45053        STA     1,ERLN.,2
032627   25003        LDA     1,STPBC,2  ;Save starting PBC of error statement
032630   44345        STA     1,ERPBC
032631    4152        JSR     CKFM       ;Execute (calculator) mode?
032632     413        JMP     ERR2       ;   Yes, Ignore error branch
032633   25052        LDA     1,ERBP.,2
032634   34053        LDA     3,C200
032635  117405        AND     0,3,SNR    ;NON-ABORTIVE ERROR . . .
032636  125015        SNZ     1,1        ;AND ERROR BRANCH SET ?
032637      406       JMP     ERR2
032640   54076        STA     3,ERRF     ;   YES
032641   45003        STA     1,PBC.,2
032642    2401        JMP     @.+1       ;EXECUTE ERROR BRANCH
032643   32627        EXST1
```

```
                ;                              << SI = R92RUNSID; BO = 18/A.RUN.8053! >>

032644  033717          END
032645  126400 ERR2:    SUB     1,1
032646  044370          STA     1,RRSZ      ;  NO, CLEAR REQ FOR MORE SPACE
032647  102000          ADC     0,0
032650  041075          STA     0,TSU.2,2;Direct I/O to terminal
032651  006241          MTOST               ;Empty buffers
032652  030366          LDA     2,.BUS
032653  021073          LDA     0,TSU.,2    ;OUTPUT ERROR MESSAGE
032654  006101          CALL
032655  000012          ERROR
032656  034724          LDA     3,ENDAO
032657  054301          STA     3,TS
032660  004423          JSR     CKEM        ;Immediate execute (calculator) mode?
032661  000417          JMP     ERR4        ;  YES
032662  102400          SUB     0,0         ;  NO
032663  040076          STA     0,ERRF
032664  004674          JSR     SNO         ;OUTPUT ERROR LINE NUMBER
032665  006141          STOUTPUT
032666  034366          LDA     3,.BUS
032667  021473          LDA     0,TSU.,3
032670  024053          LDA     1,C200
032671  123414          AND#    1,0,SZR     ;ABORTIVE ERROR ?
032672  002752          JMP     @ERR2-1     ;  Yes
032673  025463          LDA     1,STPBC,3;  No, continue execution at next statement
032674  045403          STA     1,PBC.,3    ;Reset PBC to start of statement with error
032675  006270          JSR     @.SKST      ;Skip over statement - Last statement in pgm?
032676  002746          JMP     @ERR2-1     ;  Yes, End of program
032677  002375          JMP     @.EXST      ;  No, Continue execution

032700  006141 ERR4:    STOUTPUT            ;Output error message
032701  002743          JMP     @ERR2-1     ;And END

                ;  CKEM  -  Check for Execute (calculator) Mode
                ;               If in execute mode, CKEM uses a non-skip return
                ;               If NOT in execute mode, CKEM uses a skip return

032702  000000          0
032703  054777 CKEM:    STA     3,CKEM-1    ;Save return address
032704  034366          LDA     3,.BUS
032705  035424          LDA     3,FLAG.,3
032706  177100          ADDL    3,3         ;Shift execute mode flag into sign bit
032707  175112          SSP     3,3         ;In Execute mode?
032710  002772          JMP     @CKEM-1     ;  Yes, Non-skip return
032711  034771          LDA     3,CKEM-1;   No, Skip return
032712  001401          JMP     1,3

                .EOT    ; "RUN" R9.0 SOURCE #1
```

```
                ;                                  << SI = R92RUNS2A, BD = 18/A.RUN.8053! >>
                ; 25 AUG 86, RB.

                ; "RUN" SOURCE #2 OF 7 FOR "IRIS"


                ;
                ;  SKSTM  -  Skip over current BASIC statement to start of next
                ;            statement setting PBC. and PLC.. If the end of
                ;            the program is reached, the routine will use a
                ;            non-skip return.  If not, a skip return will be
                ;            performed with A2 set to 342 octal (end of line)
                ;            if a line was ended or 343 octal (stmt follows) if
                ;            only a statement was ended.  PBC. must point to the start
                ;            of a statement at entry to SKSTM.
                ;
                ;  SKRST  -  Skip over rest of BASIC statement.  Alternate entry point
                ;            for SKSTM which assumes that PBC. points inside of a
                ;            statement.
                ;
                ;  Calling Sequence:
                ;
                ;                           ;PBC. and PLC. are assumed to be valid and
                ;                           ;  pointing at a token (not inside a literal)
                ;      JSR     SKSTM
                ;      JMP     ENDPGM       ;Non-skip return if end of pgm hit
                ;      STA     2,LSTBYT;Termination type of stmt returned in A2,
                ;                           ;  end of stmt (342) or stmt follows (343)
                ;
                ;      Scratch:             TSE7
                ;

 32713  54404 SKRST: STA     3,SKSTM-1;Save return address
 32714 102400        SUB     0,0
 32715  40337        STA     0,TSE7       ;Clear statement code
 32716    424        JMP     SKSNO        ;And jump into SKSTM to skip rest of stmt

 32717      0        0                    ;Return address save cell
 32720  54777 SKSTM: STA     3,SKSTM-1;Save return address

 32721   6264 SKSNS: NEXTBYTE             ;Load statement code
 32722  50307        STA     2,TSE7       ;Save statement code
 32723  20051        LDA     0,C100
 32724 112655        SUBOR#  0,2,SNR      ;GOTO or GOSUB?
 32725    452        JMP     SKSGO        ;  Yes, Skip over line nbr list
 32726  20211        LDA     0,C125
 32727 112655        SUBOR#  0,2,SNR      ;DATA or REM?
 32730    462        JMP     SKSNL        ;  Yes, Skip over entire line
 32731  20530        LDA     0,C130
 32732 112414        SEQ     0,2          ;MAT STMT?
 32733    407        JMP     SKSNO        ;  NO, SHIP SPECIAL CHECK
 32734   6264        NEXTBYTE
 32735  20215        LDA     0,C342
 32736 112015        ADC#    0,2,SNR      ;NEXT-STMT CODE?
 32737    403        JMP     SKSNO        ;  YES, SKIP IT
 32740  34356        LDA     3,.BUS       ;  NO, BACKUP PBC
 32741  15403        DSZ     PBC.,3
```

```
                ;                                   << SI = R92RUNS2A; BO = 18/A.RUN.8053! >>

 32742   6264 SKSNO: NEXTBYTE                    ;Skip over individual tokens, Get next token
 32743  20215        LDA     0,C342
 32744 112415        SNE     0,2                ;End of statement (actually line)?
 32745    445        JMP     SKSNL              ;  Yes, Skip to next line
 32746 112015        ADC#    0,2,SNR            ;Statement Follows?
 32747    451        JMP     SKSSF              ;  Yes, Check for compound stmt (IF or ON)
 32750  20031        LDA     0,C11
 32751 142432        SGR     2,0                ;"CALL" subroutine name?
 32752    404        JMP     SKST1              ;  No, Check if floating point constant
 32753  24034        LDA     1,C14
 32754 146132        SGR     2,1                ;Above subroutine name token range?
 32755    447        JMP     SKSCST             ;  No, Skip over multi-word constant
 32756  20502 SKST1: LDA     0,C173             ;  Yes,  Check if floating point constant
 32757 142432        SGR     2,0                ;Floating point constant?
 32760    404        JMP     SKST2              ;  No, Continue testing
 32761  24052        LDA     1,C177
 32762 146132        SGR     2,1                ;Above Floating Point constants?
 32763    461        JMP     SKSCST             ;  No, Process constant
 32764  20270 SKST2: LDA     0,C351
 32765 112415        SNE     0,2                ;Quote?
 32766    465        JMP     SKSST              ;  Yes, Skip over string literal
 32767  20267        LDA     0,C375             ;Single Quote?
 32770 112414        SEQ     0,2
 32771    751        JMP     SKSNO              ;  No, Was single byte token, try next
                                                ;  Yes, Skip over $TERM mnemonic
 32772   6264 SKSTS: NEXTBYTE                   ;Skip until closing single quote
 32773  20267        LDA     0,C375
 32774 112414        SEQ     0,2                ;Single quote?
 32775    775        JMP     SKSTS              ;  No, continue search
 32776    744        JMP     SKSNO              ;  Yes, Try next token

 32777   6264 SKSGO: NEXTBYTE                   ;Skip over GOTO/GOSUB line nbr list
                                                ;  A2 now contains # of line nbrs in list
 33000  34346        LDA     3,.BUS
 33001  21403        LDA     0,PBC.,3           ;Skip PBC. over line nbr list
 33002 143000        ADD     2,0                ;  by adding size of list
 33003 143000        ADD     2,0                ;  to PBC.
 33004  41403        STA     0,PBC.,3
 33005   6264        NEXTBYTE                   ;Load byte following list
 33006  20215        LDA     0,C342
 33007 112015        ADC#    0,2,SNR            ;Statement follows code?
 33010    420        JMP     SKSSF              ;  Yes, Handle it
 33011    401        JMP     SKSNL              ;  No, Advance to next line
```

```
                ;                                << SI = R92RUNS2A; BO = 18/A.RUN.8053! >>

 00012  34366 SKSNL: LDA     3,.BUS    ;Advance to next line and return
 00013  11402        ISZ     PLC.,3    ;Increment PLC. to next line
 00014  11402        ISZ     PLC.,3    ;  (each entry is two words long)
 00015  31402        LDA     2,PLC.,3  ;Load PBC of first stmt using from SLT using PLC
 00016  25401        LDA     1,VDT.,3
 00017 146033        SLS     2,1       ;End of Program reached?
 00020   2677        JMP     @SKSTM-1  ;  Yes, perform non-skip return
 00021 173000        ADD     3,2
 00022  21001        LDA     0,1,2     ;Load relative word addr of first stmt of line
 00023 101120        MOVZL   0,0       ;Convert to byte address
 00024  41403        STA     0,PBC.,3  ;And set PBC.
 00025  30215        LDA     2,C342    ;Return End of statement code in A2
 00026  34671 SKSSR: LDA     3,SKSTM-1;Skip return with terminater code in A2
 00027   1401        JMP     1,3

 00030  34366 SKSSF: LDA     3,.BUS    ;Handle Statement Follows code
 00031  21403        LDA     0,PBC.,3  ;Advance PBC. to next word boundary
 00032 101620        INCZR   0,0       ;  where next stmt starts
 00033 101120        MOVZL   0,0
 00034  41403        STA     0,PBC.,3
 00035  20307        LDA     0,TSE/    ;Load statement code of skipped stmt
 00036  24121        LDA     1,C114
 00037 122655        SUBOR#  1,0,SNR   ;IF or ON statement?
 00040    661        JMP     SKSNS     ;  Yes, Skip over following stmt also
 00041  30402        LDA     2,C343    ;Return Stmt Follows code in A2
 00042    744        JMP     SKSSR     ;  No, All done, perform skip return
 00043    343 C343:  343

 00044 112400 SKSCST:        SUB       0,2;Skip Over Multi-word Constant
                                       ;  A2 now has word length of cst
 00045  34366        LDA     3,.BUS
 00046  21403        LDA     0,PBC.,3  ;Advance PBC past literal
 00047 101620        INCZR   0,0       ;Roundup and convert PBC to word addr
 00050 143120        ADDZL   2,0       ;Add constant length and convert to byte addr
 00051  41403        STA     0,PBC.,3  ;Set new PBC.
 00052    670        JMP     SKSNO     ;And then process next token

 00053   6264 SKSST: NEXTBYTE          ;Skip over string literal
 00054 151014        SKZ     2,2       ;Zero byte terminator?
 00055    776        JMP     SKSST     ;  No, Try next byte
 00056    664        JMP     SKSNO     ;  Yes, Process next token

 00057    114 C114:  114
 00060    173 C173:  173
 00061    130 C130:  130
```

```
                ;                              << SI = R92RUNS2A; BO = 18/A.RUN.8053! >>

                ;   EVCHX   -  Evaluate Channel-Address-DelayLimit Specification
                ;   EVCHS   -  Evaluate Channel-Mode-Dir Specification for SEARCH
                ;
                ;              Prepare item control block for I/O statements and
                ;              SEARCH.  Uses non-skip return if the channel, prefixed
                ;              by "#", is not specified.  Otherwise a skip return is
                ;              is used.  May abort via RUNERROR on various parameter
                ;              errors.  Sets the following cells:
                ;                   TSN6   -  Contents of A0 on entry
                ;                   TSN5   -  Channel number (if specified)
                ;                   TSN11  -  Delay limit (EVCHX only, defaults to -1 (unlimited))
                ;                   TSN    -  Record number, defaults to -1
                ;                   TSN1   -  Item Nbr or Mode, defaults to 0
                ;                   TSN2   -  Upper 6 bits used for POLYFILE record nbr
                ;              Various scratch cells.

000062 152521 EVCHS: SUBZL   2,2,SKP   ;Evaluate Channel-Mode-Dir for SEARCH
000063 152400 EVCHX: SUB     2,2       ;Evaluate Channel-Address-DelayLimit for I/O statements
000064  50364         STA     2,TSN11   ;Save type of call (0 if EVCHX, 1 if EVCHS)
000065  30366         LDA     2,.BUS
000066  54357         STA     3,TSN4    ;EVALUATE CHANNEL ADDRESS
000067  40361         STA     0,TSN6
000070  25003         LDA     1,PBC.,2
000071   6144         XGETBYTE
000072  34064         LDA     3,C377
000073 156114         SEQ     2,3       ;FIRST BYTE A "#" ?
000074   2357         JMP     @TSN4     ;  NO
000075  10357         ISZ     TSN4      ;  YES
000076  34366         LDA     3,.BUS
000077  11403         ISZ     PBC.,3
000100   6263         INTEGER
000101  44360         STA     1,TSN5    ;CHANNEL NUMBER
000102 101014         SKZ     0,0       ;Negative channel number ?
000103   6267 EVCPE: RUNERROR           ;  YES, Parameter error
000104 116110         34*K!SE
000105 121000         MOV     1,0       ;Channel #
000106  54363         STA     3,TSN10   ;Save terminating character
000107   6101         CALL              ;Find DFT
000110 100002         CHKCHANNEL        ; and check if channel open
000111 102401         SUB     0,0,SKP   ;Not found or not open - Fake not poly
000112  21007         LDA     0,CLP,2   ;Get "open in polyfile mode" flag from DFT
000113 101213         SKO     0,0       ;Is this file open in polyfile mode?
000114 102400         SUB     0,0       ;  No - clear poly flag
000115  34363         LDA     3,TSN10   ;Terminating character
000116  40363         STA     0,TSN10   ;Save poly flag
```

```
                ;                                  << SI = R92RUNS2A; BO = 18/A.RUN.8053! >>

                ;  Evaluate Record/Mode Number

00117 102520            SUBZL   0,0       ;Setup default value of -1 for record number
00120 126520            SUBZL   1,1
00121  30222            LDA     2,C360
00122 172405            SUB     3,2,SNR   ;Semi-colon? (Record not specified)
00123    411            JMP     EVCHD     ;  Yes, Use default of -1 in A0, A1, A2
00124 150014            COM#    2,2,SZR   ;  No, Comma?
00125   4204            JSR     ERRSY     ;    No, Syntax Error
00126   6257            ALGEXPRESSION     ;    Yes, Evaluate record number
00127  54362            STA     3,TSN7    ;Save terminator token
00130  34121            LDA     3,FIX&377
00131   5727            JSR     -1,3      ;Perform double precision FIX
00132    407            JMP     EVCHE     ;  Overflow, Parameter Error
00133  34362            LDA     3,TSN7    ;Restore terminator token

00134 101015 EVCHD: SNZ     0,0       ;NEGATIVE RECORD NUMBER ?
00135    412            JMP     EVCHB     ;  NO
00136  20004            LDA     0,C4      ;  YES
00137 151015            SNZ     2,2       ;Single word value?
00140 122403            SLE     1,0       ;Between -1 and -4 inclusive?
00141   6267 EVCHE: RUNERROR              ;  NO
00142 131510            63*K!SE
00143  20363            LDA     0,TSN10   ;  Yes
00144 101014            SKZ     0,0       ;Poly ?
00145  30050            LDA     2,C77     ;  Yes - H.O. 6 bits
00146 124400            NEG     1,1       ;2's comp
00147  44353 EVCHB: STA     1,TSN     ;RECORD NUMBER
00150  20050            LDA     0,C77
00151 112002            SGE     0,2       ;Record # > 22 bits?
00152    767            JMP     EVCHE     ;  Yes
00153 151300            MOVS    2,2       ;  No; Bits to upper byte of item type word
00154  50355            STA     2,TSN2

                ;  Evaluate Item/Directory Number

00155 102400            SUB     0,0       ;Setup default value of 0 for item number
00156  24202            LDA     1,C360
00157 166405            SUB     3,1,SNR   ;Semi-colon? (item/dir not specified)
00160    407            JMP     EVCHC     ;  Yes, Default item/dir to zero
00161 124014            COM#    1,1,SZR   ;  No, Is it a Comma?
00162   4204            JSR     ERRSY     ;    No, Syntax error
00163   6263            INTEGER;     Yes, Evaluate item/dir number
00164 101014            SKZ     0,0       ;Negative item or dir number?
00165   6267            RUNERROR          ;  Yes, Error, Bad item number
00166 132510            65*K!SE
00167  44354 EVCHC: STA     1,TSN1    ;  No, Set item/dir number
```

```
                ;                              << SI = R92RUNS2A; BO = 18/A.RUN.8053! >>

                ;  Evaluate Record Locked Delay Limit

 30170  20364          LDA     0,TSN11   ;Is delay limit allowed?
 30171  101014         SKZ     0,0
 30172     417         JMP     EVCHG     ;  No, Check for Semi-colon
 30173  126000         ADC     1,1       ;Default to -1 (unlimited)
 30174   30222         LDA     2,C360    ;  Yes, Check for it
 30175  172405         SUB     3,2,SNR   ;Was it specified?
 30176     412         JMP     EVCHF     ;  No, Use default value of -1 in A1
 30177  124014         COM#    1,1,SZR   ;Required Comma present?
 30200    4204         JSR     ERRSY     ;  No, Syntax error
 30201    6263         INTEGER           ;  Yes, Evaluate delay limit
 30202  101222         MOVZR   0,0,SZC   ;Negative? (also move sign to CARRY)
 30203  124400         NEG     1,1       ;  Yes, Negate it
 30204  127012         ADD#    1,1,SZC   ;Overflow on negate (does sign agree)?
 30205     676         JMP     EVCPE     ;  Yes, Parameter error
 30206  125506         INCZL#  1,1,SEZ   ;Is value between -1 and 32767 inclusive?
 30207     674         JMP     EVCPE     ;  No, Parameter error (was less than -1)
 30210   44364 EVCHF:STA     1,TSN11   ;  No, Set retry count

                ;  All legal fields evaluated, Check for Semi-colon terminator

 30211   20222 EVCHG:LDA     0,C360
 30212  162414         SEQ     3,0       ;Required Semi-colon present?
 30213    4204         JSR     ERRSY     ;  No, Syntax error
 30214  176400         SUB     3,3       ;  Yes, Continue
 30215   30363         LDA     2,TSN10   ;POLY FILE FLAG
 30216   54363         STA     3,TSN10   ;CLEAR
 30217   50321         STA     2,TSX10   ;PRESERVE POLYFILE FLAG
 30220   20355         LDA     0,TSN2    ;H.O.  6 bits of record # * 256.
 30221  101015         SNZ     0,0       ;Record # > 16 bits?
 30222    2357         JMP     @TSN4     ;  No, return
 30223  151014         SKZ     2,2       ;Polyfile ?
 30224    2357         JMP     @TSN4     ;  Yes, RETURN
 30225     714         JMP     EVCHE     ;  No - error
```

```
                ;                               << SI = R92RUNS2A;  BO = 18/A.RUN.8053! >>
 00226   33472          READD
 00227      33          READITEM
 00230  102400 REEAD:   SUB     0,0             ;"READ" STATEMENT
 00231    4602          JSR     EVCHX
 00232    2774          JMP     @REEAD-2
 00233     405          JMP     RWIB

 00234   20004          WRITITEM
 00235   20777 WRITE:   LDA     0,-1            ;"WRITE" STATEMENT
 00236    4625          JSR     EVCHX
 00237    4204          JSR     ERRSY
 00240   34361 RWIB:    LDA     3,TSN6          ;READ OR WRITE AN ITEM
 00241  175014          SKZ     3,3             ;WHICH ?
 00242     410          JMP     RWIBO           ;   WRITE
 00243    6256          JSR     @.VLOC          ;   READ, LOCATE VARIABLE
 00244  121001          MOV     1,0,SKP         ;     STRING
 00245     427          JMP     RWIB3           ;     NUMERIC
 00246  143000          ADD     2,0
 00247  175414          INC#    3,3,SZR         ;TWO SUBSCRIPTS GIVEN ?
 00250   40363          STA     0,TSN10         ;   NO, SAVE ZERO BYTE LOCATION
 00251     463          JMP     RWIS1

 00252    6260 RWIBO:   EXPRESSION              ;EVALUATE EXPRESSION FOR WRITE
 00253     456          JMP     RWIS            ;STRING FOUND
 00254  151015          SNZ     2,2             ;ANY EXPRESSION FOUND ?
 00255     440          JMP     RWIQ            ;   NO
 00256  151414          INC#    2,2,SZR         ;RESULT IN DA ?
 00257     415          JMP     RWIB3           ;   NO
 00260   24024          LDA     1,C4            ;   YES, ASSUME 4 WORDS
 00261  101015          SNZ     0,0             ;ONE-WORD CONSTANT ?
 00262  126520          SUBZL   1,1             ;   YES
 00263   44316          STA     1,TSX5          ;SAVE LENGTH
 00264    6137          STORDA
 00265  102400          SUB     0,0             ;STORE # IN TSX
 00266   24316          LDA     1,TSX5          ;GET LENGTH
 00267   30252          LDA     2,.TSX          ;PLACE TO PUT IT
 00270    6120          DECIMAL
 00271    6131          LOADDA
 00272   24316          LDA     1,TSX5          ;SIZE
 00273   30252          LDA     2,.TSX          ;PLACE ITS LOCATED
 00274   20005 RWIB3:   LDA     0,C5            ;ITEM TYPE IS DECIMAL <<FROM RWIB, RWIBO
 00275   34356          LDA     3,.BUS
 00276   15403          DSZ     PBC.,3
 00277    4142 RWIB1:   JSR     RWIC            ;<< ENTRY FOR STRING ITEM
 00300    6264          NEXTBYTE
 00301   20223          LDA     0,C361
 00302  142415          SNE     2,0             ;COMMA ?
 00303     735          JMP     RWIB            ;   YES
 00304  142014 RWIB2:   ADC#    2,0,SZR         ;UNLOCK RECORD ?
 00305    2246          JMP     @.EDST          ;   NO
 00306   20360 RWIQ2:   LDA     0,TSN5          ;   YES << ENTRY FROM RWIQ
 00307    6101          CALL
 00310  100010          UNLOCK
 00311    6267          RUNERROR                ;CHANNEL NOT OPEN
 00312  130510          61*K!SE
 00313    6264          NEXTBYTE
 00314    2246          JMP     @.EDST          ;SHOULD BE END OF STATEMENT
```

```
                        ;                                << SI = R92RUNS2A;  BO = 18/A.RUN.8053! >>
 33315   24200 RWIQ:  LDA     1,C351     ;NULL EXPRESSION OR LITERAL STRING
 33316  166414         SEQ     3,1        ;WRITING A LITERAL STRING ?
 33317     767         JMP     RWIQ2      ;   NO, RECORD UNLOCK
 33320   30366         LDA     2,.BUS     ;   YES
 33321   25003         LDA     1,PBC.,2
 33322   44357         STA     1,TSN4
 33323    6264         NEXTBYTE           ;FIND END OF STRING
 33324  151014         SKZ     2,2
 33325     776         JMP     .-2
 33326   30357         LDA     2,TSN4
 33327  146400         SUB     2,1        ;STRING LENGTH - 1
 33330     406         JMP     RWIS2

 33331  176500 RWIS:  SUBL    3,3        ;WRITE FROM A STRING VARIABLE << FROM RWIBO
 33332  113000         ADD     0,2
 33333  106401         SUB     0,1,SKP
 33334  136414 RWIS1: SEQ     1,3        ;ENTRY TO READ INTO STRING VARIABLE <<FROM RWIB
 33335  175015         SNZ     3,3        ;SKIP IF TWO SUBSCRIPTS
 33336  125400 RWIS2: INC     1,1        ;ENTRY TO WRITE A LITERAL STRING
 33337   20001         LDA     0,C11
 33340     737         JMP     RWIB1

 33341   54362 RWIC:  STA     3,TSN7     ;SAVE RETURN CALL READI/WRITI<<FROM RWIB1
 33342   50357         STA     2,TSN4     ;SOURCE/DEST PTR
 33343   44356         STA     1,TSN3     ;SIZE
 33344   24023         LDA     1,CM400
 33345   30355         LDA     2,TSN2     ;H.O. record # (polyfiles only)
 33346  133400         AND     1,2        ;Isolate it
 33347  143000         ADD     2,0        ;Merge into item type word
 33350   40355         STA     0,TSN2     ;TYPE

 33351   30255 RWIC1: LDA     2,.TSN     ;PICK UP ICB AND
 33352   20361         LDA     0,TSN6     ;READ/WRITE KEYWORD
 33353  101015         SNZ     0,0        ;WRITE ?
 33354   20653         LDA     0,REEAD-1  ;   NO
 33355   40403         STA     0,RWIC2    ;   YES, SET KEYWORD INTO CHANNEL CALL
 33356   20360         LDA     0,TSN5     ;PICK UP CHANNEL #
 33357    6106         CHANNEL
 33360      33 RWIC2: READITEM           ;*** CODE MODIFIED HERE ***
 33361     432         JMP     RWICE      ;AN ERROR:  FIND OUT WHAT
 33362   20353         LDA     0,TSN      ;GET RECORD #
 33363  175014         SKZ     3,3        ;TEXT FILE ?
 33364     405         JMP     RWIC3      ;   NO
 33365  126000         ADC     1,1        ;   YES
 33366  101113         SSN     0,0        ;RECORD # -1 OR -2 ?
 33367   44353         STA     1,TSN      ;   NO, SET # -1
 33370     407         JMP     RWIC4
```

```
                ;                                   << SI = R92RUNS2A; BO = 18/A.RUN.8053! >>

 00371  101414 RWIC3: INC#    0,0,SZR   ;REC #=-1?
 00372     405        JMP     RWIC4     ;  NO, CHECK FOR STRING
 00373   20321        LDA     0,TSX10   ;  YES, PICK UP POLYFILE FLAG
 00374  101014        SKZ     0,0       ;IS FILE A PLYFILE?
 00375     407        JMP     RWIC5     ;  YES, CHECK H.O. 6 BITS
 00376   14353        DSZ     TSN       ;  NO, SET # -2
 00377  102400 RWIC4: SUB     0,0
 00400   24353        LDA     1,TSN10
 00401  125014        SKZ     1,1       ;STRING ?
 00402    6145        XPUTBYTE          ;  YES, STORE ZERO BYTE
 00403    2362        JMP     @TSN7

 00404   20355 RWIC5: LDA     0,TSN2    ;PICK UP H.O. PART OF REC #
 00405   24023        LDA     1,CM400
 00406  107700        ANDS    0,1       ;ISOLATE H.O. REC #
 00407   20050        LDA     0,C77
 00410  106415        SNE     0,1       ;IS REC # = -1?
 00411   14353        DSZ     TSN       ;  YES, SET = -2
 00412     765        JMP     RWIC4     ;AND EXIT

 00413   20210 RWICE: LDA     0,C30     ;ERROR IN READ/WRITE ITEM
 00414  162014        ADC#    3,0,SZR   ;RECORD LOCKED ?
 00415    2735        JMP     @.CSEN    ;  NO
                                        ;  Yes

 00416   30364        LDA     2,TSN11   ;Load delay limit
 00417  151015        SNZ     2,2       ;Limit of 0?
 00420    2235        JMP     @.CSEN    ;     Yes, Report error to program
 00421  151415        INC#    2,2,SNR   ;Limit of -1 (infinite)?
 00422     405        JMP     RWIRT     ;  Yes, Retry
 00423  146032        SGE     2,1       ;Is the limit greater than the delay?
 00424  145000        MOV     2,1       ;  No, Replace the delay count with the limit
 00425  132400        SUB     1,2       ;Subtract the delay from the limit
 00426   50364        STA     2,TSN11   ;And update the limit

 00427   44301 RWIRT: STA     1,TS      ;Save delay count returned by READITEM/WRITITEM
 00430    6234        JSR     @.CESF    ;Check for ESC
 00431    6262        GOGO              ;START OUTPUT BEFORE BUMPUSER
 00432   30366        LDA     2,.BUS
 00433  150100        NEG     2,2
 00434    4412        JSR     ABREL     ;CONVERT SOURCE/DEST PTR TO REL
 00435   20003        LDA     0,C3
 00436   24301        LDA     1,TS
 00437    6101        CALL
 00440      57        SIGPAUSE          ;WAIT FOR RECORD TO UNLOCK
 00441    6117        BUMPUSER
 00442   30366        LDA     2,.BUS
 00443    4403        JSR     ABREL     ;CONVERT SOURCE/DEST PTR BACK TO ABS
 00444     705        JMP     RWIC1
```

```
                ;                              << SI = R92RUNS2A; BO = 18/A.RUN.8053! >>


 33445      20 VEMD1:EMOD1                     ;IF ERR 1 mode flag

                ;LITTLE ROUTINE TO CONVERT SOURCE/DEST POINTER FOR READITEM/WRITITEM
                ;TO RELATIVE BEFORE SWAPOUT & BACK TO ABSOLUTE AFTER SWAPIN
                 ;CALL IS:
                 ;      (A2) = -(.BUS) FOR RELATIVE
                 ;                (.BUS) FOR ABSOLUTE
                 ;      JSR    ABREL

 33446   20355 ABREL:LDA     0,TSN2
 33447   24064       LDA     1,C377
 33450  123100       AND     1,0         ;Isolate item type
 33451   24031       LDA     1,C11
 33452  106415       SNE     0,1         ;IS ITEM OF TYPE STRING ?
 33453    1400       JMP     0,3         ;   YES, LEAVE IT ALONE
 33454   20357       LDA     0,TSN4
 33455  143000       ADD     2,0         ;   NO, CONVERT POINTER TO ABS/REL
 33456   40357       STA     0,TSN4
 33457    1400       JMP     0,3         ;RETURN
```

```
                ;                                  << SI = R92RUNS2A;  BD = 18/A.RUN.8053! >>


30460  34366 BADST:LDA     3,.BUS     ;IMPOSSIBLE STATEMENT CODE !?
30461  25403        LDA     1,PBC ,3
30462 124400        NEG     1,1
30463 124220        COMZR   1,1
30464  20040        LDA     0,SBA
30465 107000        ADD     0,1
30466   6142        TRAPFAULT
30467 116410        35*K!NOP


30470  32243 FNDS.:FNDS
30471   6777 READA:JSR     @FNDS.
30472  30366 READD:LDA     2,.BUS     ;READ FROM DATA STATEMENT
30473  25020        LDA     1,DWC.,2
30474 124015        COM#    1,1,SNR
30475    774        JMP     READA
30476 125015        SNZ     1,1
30477   6247        RUNERROR           ;OUT OF DATA
30500 107110        16*K+SE
30501   6256        JSR     @.VLOC     ;LOCATE THE VARIABLE
30502   4206        JSR     ERRST      ;IT WAS A STRING !!
30503  54302        STA     3,TS1
30504  34366        LDA     3,.BUS
30505  21424        LDA     0,FLAG.,3
30506  24034        LDA     1,C14
30507 107620        ANDZR   0,1
30510 125220        MOVZR   1,1        ;Data size
30511 125400        INC     1,1
30512  31420        LDA     2,DWC.,3
30513 141000        MOV     2,0
30514 123000        ADD     1,0
30515  41420        STA     0,DWC.,3
30516 173000        ADD     3,2        ;A(Data)
30517 102520        SUBZL   0,0        ;LOAD THE DATA
30520   6120        DECIMAL
30521 102400        SUB     0,0
30522  24310        LDA     1,TS7
30523  30307        LDA     2,TS6      ;STORE THE DATA
30524   6120        DECIMAL
30525  30366        LDA     2,.BUS
30526  15021        DSZ     DEC.,2     ;END OF THIS DATA STATEMENT ?
30527    402        SKIP
30530   6740        JSR     @FNDS.     ;   YES
30531  30302        LDA     2,TS1      ;   NO
30532  34223        LDA     3,C361
30533 156415        SNE     2,3        ;COMMA ?
30534    706        JMP     READD      ;   YES
30535    431        JMP     ENDST      ;   No, Check for end of stmt and then exec nxt stmt
```

```
                        ;                                << SI = R92RUNS2A; BO = 18/A.RUN.8053! >>
                        ;  NXLIN  -  Entry to increment PLC and execute next line

 000536   4116 NXLIN: JSR     CERRF     ;Check for arithmetic errors in current statement
 000537  30366         LDA     2,.BUS
 000540  11002         ISZ     PLC.,2    ;Increment PLC to next line
 000541  11002         ISZ     PLC.,2    ;  (each SLT entry is two words long)

                        ;  EXLIN  -  Entry to execute line currently identified by PLC

 000542  30366 EXLIN: LDA     2,.BUS
 000543  35002         LDA     3,PLC.,2
 000544  25001         LDA     1,VDT.,2
 000545 166003         SLS     3,1       ;At end of program?
 000546    551         JMP     END       ;  Yes, Return to BASIC
 000547 157000         ADD     2,3       ;  No, Set PBC and execute 1st stmt of line
 000550  21401         LDA     0,1,3     ;Load relative word addr of statement
 000551 101120         MOVZL   0,0       ;Convert to relative byte addr
 000552  41003         STA     0,PBC.,2  ;Set PBC
 000553    432         JMP     EXSTM     ;And execute that statement

                        ;  CERRF  -  Use check ERRF flag to detect arithmetic errors in
                        ;            current statement before advancing to the next stmt.

 000554  20076 CERRF: LDA     0,ERRF    ;Load DECIMAL error flag
 000555 101015         SNZ     0,0       ;Any errors?
 000556   1400         JMP     0,3       ;  No Return
 000557   6267         RUNERROR          ;  Yes, Generate error 15
 000560 107510         17*K+SE


                        ;  NXSTM  -  Entry to skip over current statement and execute next.
                        ;            normally called by statements which do NOT check for extra
                        ;            tokens at end of statement (compatibility means not killing
                        ;            bugs without official approval) This entry must NOT be used by
                        ;            compound statements such as IF or ON.

 000561  32713         SKRST
 000562   4772 NXSTM: JSR     CERRF     ;Check for errors in current statement
 000563   4776         JSR     @NXSTM-1; Skip to start of next statement - Is there one?
 000564    533         JMP     END       ;  No, End of program
 000565    420         JMP     EXSTM     ;  Yes, Execute statement
```

```
                ;                              << SI = R92RUNS2A;  BO = 18/A.RUN.8053! >>

                ;   ENDST   -  Check for end of statement or statement follows code,
                ;                error if not found.  If found, execute next statement
                ;   EDSTX   -  Alternate entry point to ENDST.  Entered with the
                ;                terminator byte in A2.

 33566   30366 ENDST: LDA     2,.BUS
 33567   25003        LDA     1,PBC.,2 ;Load current PBC
 33570  124400        NEG     1,1      ;Backup to previous code byte
 33571  124000        COM     1,1
 33572    6144        XGETBYTE         ;Load previous code byte
 33573    4761 EDSTX: JSR     CERRF    ;Alternate entry point, Also check for DECIMAL errors
 33574   20215        LDA     0,C342
 33575  112415        SNE     0,2      ;End of statement (line) code?
 33576     740        JMP     NXLIN    ;  Yes, Execute next line
 33577  112014        ADC#    0,2,SZR  ;Statement follows code?
 33600    4204        JSR     ERRSY    ;  No, Syntax error
 33601   30366        LDA     2,.BUS   ;  Yes, Advance PBC to word boundary for next stmt
 33602   21003        LDA     0,PBC.,2
 33603  101212        SKE     0,0      ;PBC on word boundary?
 33604   11003        ISZ     PBC.,2   ;  No, Advance to it

                ;   EXSTM   -  Execute current statement pointed at by PBC

 33605   30366 EXSTM: LDA     2,.BUS   ;PICK UP CURRENT PBC
 33606   21003        LDA     0,PBC.,2
 33607   41003        STA     0,STPBC,2;SET UP 'START OF STATEMTENT
 33610    6204        JSR     @.CESF   ;CHECK FOR ESC OR CTRL-C

 33611   30366        LDA     2,.BUS
 33612   35024        LDA     3,FLAG.,2;Currently in BASIC calculator mode?
 33613  177100        ADDL    3,3
 33614  175113        SSN     3,3
 33615     405        JMP     EXST2    ;  No, Execute this statement
 33616   35002        LDA     3,PLC.,2;  Yes, Check if we're in line 0
 33617   25005        LDA     1,SLT.,2
 33620  136114        SEQ     1,3      ;Are we on the first line in the SLT?
 33621     476        JMP     END      ;  No, All done, return to BASIC
                                       ;  Yes, Continue execution
 33622   34074 EXST2: LDA     3,ETSF
 33623  175015        SNZ     3,3      ;TIME SLICE ENDED ?
 33624     403        JMP     EXST1    ;  NO
 33625    6262        GOGO             ;  YES
 33626    6117        BUMPUSER
```

```
                ;                          << SI = R92RUNS2A;  BO = 18/A.RUN.8053! >>

                ;  EXST1  -  Alternate entry for EXSTM which skips ESC/CTLC and Swap check

 00627  30366 EXST1: LDA     2,.BUS
 00630  25012         LDA     1,UFS.,2 ;Clear User Function Stack for next stmt
 00631  45013         STA     1,UFC.,2
 00632  25003         LDA     1,PBC.,2 ;Record starting PBC of current statement
 00633  45063         STA     1,STPBC,2
 00634  11003         ISZ     PBC.,2    ;Increment PBC and
 00635   6144         XGETBYTE          ;Load statement code of new statement
 00636  20051         LDA     0,C100
 00637  24212         LDA     1,C136
 00640 142033         SLS     2,0
 00641 146133         SLE     2,1
 00642    616         JMP     BADST     ;IMPOSSIBLE STATEMENT CODE !?
 00643 112036         ADCZ#   0,2,SEZ   ;GOTO or GOSUB ?
 00644    407         JMP     EXST5     ;  No
 00645  34366         LDA     3,.BUS
 00646  35477         LDA     3,TSU.4,3;B & D flags
 00647 174400         NEG     3,3
 00650 151220         MOVZR   2,2       ;L.O. bit of Pseudo Op to carry
 00651 177000         ADD     3,3       ;XOR of B flag with carry
 00652 151100         MOVL    2,2       ;"Jiggled op"
 00653   4410 EXST5: JSR     EXST6
 00654  34230         GOTO
 00655  34214         GOSUB
 00656  40024         SIGNL
 00657  37772         KILL
 00660  37602         BILDF
 00661  37601         OPENF
 00662  37734         CLOSF
 00663  40271         SRCH
 00664  34633         RNDM
 00665  35326         CHAIN
 00666  40440         CALLS
 00667    204         ERRSY   ;SPARE
 00670  34327         IF
 00671  34147         ON
 00672    204         ERRSY   ;FIND
 00673  33746         STOP
 00674  34642         DEF
 00675  33717         END
 00676  34024         RTRN
 00677  37203         FOR
 00700  37427         NEXT
 00701  33536         NXLIN   ;DATA
 00702  33536         NXLIN   ;REM
 00703  34661         LET
 00704  36655         MAT
 00705  35253         DIM
 00706  32725         REST
 00707  36000         INPUT
 00710  35156         PRINT
 00711  33230         REEAD
 00712  33235         WRITE
```

```
                ;                               << SI = R92RUNS2A; BO = 18/A.RUN.8053! >>

 00713 157000 EXST6: ADD     2,3
 00714  30346        LDA     2,.BUS
 00715   3700        JMP     @-100,3   ;BRANCH TO STATEMENT INTERPRETER


                ;   END   -  Terminate program execution and return to BASIC.  Used
                ;            to implement END statement, called at end of pgm if
                ;            no END statement is found, and called at end of calculator
                ;            mode execution.

 00716  32703        CKEM
 00717   6261 END:   MTOST               ;"END" STATEMENT
 00720   6776        JSR     @END-1     ;End of calculator mode execution?
 00721    406        JMP     ENDAA      ;   Yes, Just output carriage return
 00722   6133        OUTTEXT            ;   No, Output "READY"
 00723 106722        .TXTF   ;<215>R
 00724 142701 EA
 00725 142331 DY
 00726      0 ;

 00727  30055 ENDAA: LDA     2,C215     ;Output a carriage return
 00730   6132        OUTBYTE
 00731   6261 ENDA:  MTOST
 00732   6101 ENDA1: CALL
 00733 100003        ALLCLEAR
 00734  34366        LDA     3,.BUS
 00735 102400        SUB     0,0
 00736  41457        STA     0,FL3.,3 ;Reset INPUT length
 00737  34005        LDA     3,RUP
 00740  20201        LDA     0,D.BSC    ;CHAIN TO BASIC
 00741  31425        LDA     2,DFT.,3
 00742  41371        STA     0,FDA+CHM1,2
 00743  34402        LDA     3,.+2
 00744   2140        JMP     @STINP&377;START INPUT
 00745  32004        BPS+4


                ;   STOP  -  STOP statement

 00746   6261 STOP:  MTOST               ;   NO
 00747   6133        OUTTEXT
 00750 106723        .TXTF   ;<215>S
 00751 152317 TO
 00752 150000 P;

 00753   6402        JSR     @SNOW      ;Statment number output
 00754    755        JMP     ENDA

 00755  32540 SNOW:  SNO
```

```
                ;                       << SI = R92RUNS2A; BO = 18/A.RUN.8053! >>
                ;
                ;  FDPLC  -  Search SLT for line which contains current PBC. value.
                ;            When found, set PLC. so that both PBC. and PLC. are
                ;            valid.  Uses non-skip return.
                ;
                ;     Calling Sequence:
                ;
                ;     JSR    FDPLC
                ;                       ;Uses non-skip return
                ;
                ;     Scratch:          TSE6, TSE7
                ;
 33756       0          0                ;Return address save cell
 33757   54777 FDPLC:STA    3,FDPLC-1;Save return address
 33760   34366          LDA    3,.BUS    ;Perform binary search of SLT for line addr
 33761   25405          LDA    1,SLT.,3 ;Load relative word addr of first line
 33762  167000          ADD    3,1       ;Make absolute address
 33763   44306          STA    1,TSE6    ;And set low address for search
 33764   31401          LDA    2,VDT.,3 ;Load addr of last line plus one
 33765  102520          SUBZL  0,0
 33766  112000          ADC    0,2       ;Subtract two to form addr of last line
 33767  173000          ADD    3,2       ;Make absolute
 33770   50307          STA    2,TSE7    ;And set high address for search
 33771   21403          LDA    0,PBC.,3 ;Load value to search for - PBC.
 33772  101220          MOVZR  0,0       ;Convert to relative word addr
 33773  132640 FDPNX:SUBOR  1,2       ;Calc # of lines in group
 33774  151225          MOVZR  2,2,SNR   ;Divide by two to pnt at middle, Is increment zero?
 33775     415          JMP    FDPDN     ;  Yes, Binary search done, almost through
 33776  153000          ADD    2,2       ;Scale index - SLT entries are 2 words each
 33777  133000          ADD    1,2
 34000   25001          LDA    1,1,2     ;Load relative word addr of 1st stmt of line
 34001  106033          SLS    0,1       ;Compare to PBC.
 34002     404          JMP    FDPGE     ;  Greater or Equal than PBC., Check lower lns
                                         ;  Less than PBC., Check higher lines
 34003   50307          STA    2,TSE7    ;Set test address as new high address
 34004   24306          LDA    1,TSE6    ;Reload low address
 34005     766          JMP    FDPNX     ;And check new partition
 34006   50306 FDPGE:STA    2,TSE6    ;Current line below or equal to PBC., Set new low addr
 34007  145000          MOV    2,1       ;Move low address to low register
 34010   30307          LDA    2,TSE7    ;Reload high address
 34011     762          JMP    FDPNX     ;And check new partition

 34012   30307 FDPDN:LDA    2,TSE7    ;Reload high address
 34013   25001          LDA    1,1,2     ;Load line address
 34014  172400          SUB    3,2       ;Calc relative SLT address
 34015  106033          SLS    0,1       ;PBC within this high line?
 34016     403          JMP    FDPRT     ;  Yes, Set PLC and return
 34017  102520          SUBZL  0,0       ;  No, decrement to prev line
 34020  112000          ADC    0,2
 34021   51402 FDPRT:STA    2,PLC.,3 ;Update PLC.
 34022    2734          JMP    @FDPLC-1 ;And return
```

```
                ;                               << SI = R92RUNS2A;  BO = 18/A.RUN.8053! >>

                ;   RETURN [<expression>] statement

34023     343           343
34024   35017 RTRN:   LDA     3,GSC.,2 ;"RETURN" STATEMENT
34025   21016         LDA     0,GSS.,2
34026  162432         SGR     3,0
34027    6267         RUNERROR         ;"RETURN" WITHOUT "GOSUB"
34030  110510         21*K+SE
34031   15017         DSZ     GSC.,2
34032  157000         ADD     2,3
34033   21777         LDA     0,-1,3   ;Load return address - starting PBC of calling stmt
34034   40330         STA     0,TSE
34035   25003         LDA     1,PBC.,2
34036    6144         XGETBYTE
34037   24764         LDA     1,RTRN-1
34040  146615         SUBOR   2,1,SNR  ;End of Statement or Statement Follows code?
34041     411         JMP     RTRN1    ;   YES
34042    6257         ALGEXPRESSION    ;   NO, ASSUME  +-N
34043   30215         LDA     2,C342
34044  156654         SUBOR#  2,3,SZR  ;End of Statement or Statement Follows code?
34045    4204         JSR     ERRSY    ;   NO
34046    6121         FIX              ;GET INTEGER VALUE
34047     475         JMP     ERR6     ;  > 65 K
34050  101014         SKZ     0,0
34051  124400         NEG     1,1
34052  125400 RTRN1: INC     1,1
34053   44331         STA     1,TSE1   ;Save # of statements to skip after returning
34054   30366         LDA     2,.BUS
34055   20330         LDA     0,TSE    ;Set PBC to point at GOSUB statement
34056   41003         STA     0,PBC.,2
34057    4700         JSR     FDPLC    ;Set PLC to match PBC

34060   24331         LDA     1,TSE1   ;Load number of statements to skip from current loc
34061  124536         SZN     1,1      ;Skip backwards?
34062     417         JMP     RTNSK    ;   No, Skip forwards
34063   14331         DSZ     TSE1     ;   Yes, Skip backwards (also adjust cnt by 1)

                ;   Since we can't scan backwards to find preceeding statements, a
                ;   backwards skip is performed skipping entire lines backwards,
                ;   counting the number of statements each line.  When we have skipped
                ;   enough or more than enough statements to satisfy the count, we then
                ;   skip forwards within the line if we have skipped too far.

34064   30366 RTNBK:  LDA     2,.BUS
34065   35002         LDA     3,PLC.,2 ;Set PBC to start of line
34066  157000         ADD     2,3
34067   35401         LDA     3,1,3    ;Load rel word address of line
34070  175120         MOVZL   3,3      ;Convert to byte address
34071   55003         STA     3,PBC.,2
```

```
                        ;                               << SI = R92RUNS24; BO = 18/A.RUN.8053! >>

34072    6250 RTNBC: JSR     @.SKST    ;Skip over current statement
34073   30215        LDA     2,C342    ;  End of pgm hit, Mark as end of line
34074   10331        ISZ     TSE1      ;Increment count of stmts on line
34075  100010        NOP
34076   34215        LDA     3,C342
34077  156415        SNE     2,3       ;Hit end of line?
34100     407        JMP     RTNBD     ;  Yes, count is complete
34101   30366        LDA     2,.BUS
34102   21003        LDA     0,PBC.,2
34103   24330        LDA     1,TSE
34104  106432        SGR     0,1       ;Past original GOSUB statement?
34105     765        JMP     RTNBC     ;  No, Continue counting
34106     404        JMP     RTNBE     ;And check return skip count

34107   30366 RTNBD:LDA     2,.BUS    ;Reached end of line, restore PLC to
34110   15002        DSZ     PLC.,2    ;  to that of the line we just checked
34111   15002        DSZ     PLC.,2

34112   24331 RTNBE:LDA     1,TSE1    ;Add number of stmts on lines to the number
34113  125113        SSN     1,1       ;Skip count still negative (still skip backwards)?
34114     410        JMP     RTNS1     ;  No, Skip forward if we skipped to far back
34115   15002        DSZ     PLC.,2    ;  Yes, Backup PLC to previous line
34116   15002        DSZ     PLC.,2
34117   21002        LDA     0,PLC.,2
34120   35005        LDA     3,SLT.,2
34121  116003        SLS     0,3       ;Below beginning of program?
34122     742        JMP     RTNBK     ;  No, Check previous line
34123     415        JMP     RTNER     ;  Yes, Error - no such line number

34124   35002 RTNS1:LDA     3,PLC.,2;Finished backwards skip, Skip forwards
34125  157000        ADD     2,3       ;  to adjust for over-skipping backwards
34126   21401        LDA     0,1,3     ;Set PBC to beginning of line before skipping
34127  101120        MOVZL   0,0       ;Rel word addr -> byte address
34130   41003        STA     0,PBC.,2

                        ;  Skip statements forward,  skip count in A1 and TSE1

34131  125015 RTNSK:SNZ     1,1       ;Any statements to skip?
34132    2375        JMP     @.EXST    ;  No, Execute current statement
34133    6250 RTNSF:JSR     @.SKST    ;  Yes, Skip a statement
34134     404        JMP     RTNER     ;  Error, hit end of pgm
34135   14331        DSZ     TSE1      ;Need to skip any more?
34136     775        JMP     RTNSF;  Yes, skip another
34137    2375        JMP     @.EXST    ;  No, Execute this statement

34140   30366 RTNER:LDA     2,.BUS    ;Error - No such line number
34141   21063        LDA     0,STPBC,2
34142   41003        STA     0,PBC.,2 ;Restore PBC to that of RETURN statement
34143    4614        JSR     FDPLC     ;Restore PLC too
                                       ;And fall into ERR6


                        ;  ERR6  -  RUNTIME error, NO SUCH LINE NUMBER

34144    6267 ERR6:  RUNERROR         ;  ERROR - NO SUCH LINE NUMBER
34145  103210        6*K+AE
```

```
                ;                              << SI = R92RUNS2A; BO = 18/A.RUN.8053! >>

                ;   ON <expression> GOTO <line#> {,<line#>}
                ;
                ;   ON <expression> GOSUB <line#> {,<line#>}

34146   33542           EXLIN
34147    6257 ON:       ALGEXPRESSION    ;"ON" STATEMENT
34150   30215           LDA     2,C342
34151  156014           ADC#    2,3,SZR  ;STATEMENT FOLLOW ?
34152    4204           JSR     ERRSY    ;   NO
34153    6121           FIX              ;FIX EXPRESSION RESULT
34154     421           JMP     ONSKP    ;   > 65 K
34155  101015           SNZ     0,0
34156  125015           SNZ     1,1
34157     416           JMP     ONSKP    ;   ZERO OR NEGATIVE
34160   30366           LDA     2,.BUS
34161   35003           LDA     3,PBC.,2
34162  175620           INCZR   3,3
34163  161000           MOV     3,0
34164  123120           ADDZL   1,0      ;B(SELECTED LINE #)
34165   40330           STA     0,TSE    ;Save PBC of selected line number
34166   20040           LDA     0,SBA
34167  117000           ADD     0,3
34170   21400           LDA     0,0,3    ;GOTO/GOSUB, # BRANCHES
34171   34064           LDA     3,C377
34172  117400           AND     0,3
34173  136402           SGR     1,3      ;EXPRESSION > # BRANCHES ?
34174     410           JMP     ONDO     ;   No, Perform GOTO or GOSUB
                                          ;   Yes, Skip to next statement

34175   30366 ONSKP:   LDA     2,.BUS   ;Value out of range, Skip over GOTO/GOSUB
34176   21003           LDA     0,PBC.,2
34177  101212           SKE     0,0      ;PBC on word boundary?
34200   11003           ISZ     PBC.,2   ;   No, put it there
34201    6250           JSR     @.SKST   ;Skip over GOTO/GOSUB statement
34202    2744           JMP     @ON-1    ;   End of pgm, Use execute line entry
34203    2375           JMP     @.EXST   ;Execute that statement

34204   24330 ONDO:    LDA     1,TSE
34205   45003           STA     1,PBC.,2;Point PBC at line number
34206  162700           SUBS    3,0      ;
34207   25077           LDA     1,TSU.4,2;B & D flags
34210  124400           NEG     1,1
34211  101220           MOVZR   0,0      ;Pseudo Op L.O. bit to carry
34212  127003           ADD     1,1,SNC  ;Carry XOR B flag, GOSUB?
34213     415           JMP     GOTO     ;   NO
                                          ;   Yes, Fall into GOSUB
```

```
                        ;                              << SI = R92RUNS2A;  BO = 18/A.RUN.8053! >>

                        ;   GOSUB <line#>
                        ;
                        ;       The ON GOSUB statement routine falls into this code

34214   35017 GOSUB:LDA     3,GSC.,2 ;"GOSUB" STATEMENT
34215   21061        LDA     0,XOB.,2
34216  162033        SLS     3,0
34217    6267        RUNERROR         ;STACK OVERFLOW
34220  110210        20*K+AE
34221  157000        ADD     2,3
34222   25063        LDA     1,STPBC,2;Load starting PBC of GOSUB statement
34223   21052        LDA     0,ERBP.,2
34224  106415        SNE     0,1      ;Are we executing an IF ERR GOSUB or IF ERR On GOSUB?
34225   24365        LDA     1,ERPBC  ;  Yes, Set return address to stmt which erred
                                      ;  No, Set return address to GOSUB stmt
34226   45400        STA     1,0,3
34227   11017        ISZ     GSC.,2   ;BUMP GOSUB STACK POINTER
                        ;   Fall into GOTO code

                        ;   GOTO <line#>
                        ;
                        ;       The GOSUB statement code falls into GOTO;  the ON GOTO statement
                        ;       jumps to GOTO.

34230   34076 GOTO:  LDA     3,ERRF   ;"GOTO" STATEMENT
34231  175014        SKZ     3,3
34232    6267        RUNERROR         ;ARITHMETIC ERROR IN "ON" OR "IF"
34233  107510        17*K+SE
34234   35003        LDA     3,PBC.,2
34235  175620        INCZR   3,3
34236  157000        ADD     2,3
34237   21400        LDA     0,0,3    ;LINE NUMBER FOR BRANCH
34240   25055        LDA     1,PSTS.,2
34241  124760        NEGCS   1,1
34242  115000        MOV     0,3      ;(Line number) XOR (SWAP(NEG(PSTS)))
34243  137520        ANDZL   1,3
34244  123000        ADD     1,0
34245  162400        SUB     3,0
34246   35005        LDA     3,SLT.,2
34247   25001        LDA     1,VDT.,2 ;START BINARY SEARCH OF SLT
34250  157000        ADD     2,3      ;.SLT
34251  147000        ADD     2,1      ;.VDT
34252  152120        ADCZL   2,2
34253  147000        ADD     2,1      ;.VDT-2
34254   44301        STA     1,TS     ;END LIMIT
34255   54302        STA     3,TS1    ;START LIMIT
```

```
                ;                                  << SI = R92RUNS2A; BO = 18/A.RUN.8053! >>

34256  166640 GOTO1: SUBOR   3,1        ;NUMBER OF STATEMENTS IN GROUP
34257  125620         INCZR   1,1        ;HALVE THE STEP VALUE
34260  137000         ADD     1,3        ;ADD STEP TO POINTER
34261  137000         ADD     1,3
34262   31400         LDA     2,0,3      ;GET LINE NUMBER FROM SLT
34263  112415         SNE     0,2        ;IS THIS IT ?
34264     415         JMP     GOTO4      ;   YES
34265  126120         ADCZL   1,1        ;   NO
34266  112432         SGR     0,2        ;BEYOND DESIRED LINE NUMBER ?
34267     404         JMP     GOTO2      ;   YES
34270  136400         SUB     1,3        ;   NO, STEP POINTER
34271   24301         LDA     1,TS
34272     404         JMP     GOTO3

34273  167000 GOTO2: ADD     3,1        ;DECREMENT POINTER
34274   44301         STA     1,TS       ;NEW END LIMIT
34275   34302         LDA     3,TS1
34276  136003 GOTO3: SLS     1,3        ;IS AREA EMPTY ?
34277     757         JMP     GOTO1      ;   NO, TRY SMALLER AREA
34300     614         JMP     ERR6       ;   YES, NO SUCH LINE NUMBER

34301   30366 GOTO4: LDA     2,.BUS     ;FOUND DESIRED LINE NUMBER
34302  156400         SUB     2,3
34303   55002 GOTOL: STA     3,PLC.,2   ;NEW PLC DISPLACEMENT
34304    2401         JMP     @.+1
34305   33542         EXLIN



34306   54420 IFFIX: STA     3,IFRET    ;Subroutine handles D-flag flip of relational operators
34307   34321         LDA     3,TSX10
34310   24503         LDA     1,C352
34311   44321         STA     1,TSX10
34312   30503         LDA     2,C357
34313  166003         SLS     3,1        ;Relational operator next ?
34314  172433         SLE     3,2        ;   Maybe
34315    2411         JMP     @IFRET     ;   No
34316   10410         ISZ     IFRET      ;   Yes, skip return
34317   30366         LDA     2,.BUS     ;Diddle operator
34320   21077         LDA     0,TSU.4,2  ;B & D flags
34321  100700         NEGS    0,0
34322  175240         MOVOR   3,3        ;L.O. bit of Pseudo Op to carry
34323  103000         ADD     0,0        ;Carry XORs D flag
34324  175100         MOVL    3,3        ;Recover Pseudo Op
34325    2401         JMP     @IFRET

34326       0 IFRET: 0
```

```
                ;                                   << SI = R92RUNS2A;  BO = 18/A.RUN.8053! >>
34327    6260 IF:    EXPRESSION          ;"IF" STATEMENT
34330     517        JMP     IFS         ;   STRING VARIABLE
34331  151015        SNZ     2,2         ;EXPRESSION FOUND ?
34332     502        JMP     IFQ         ;   NO
34333   54321        STA     3,TSX10     ;Save Pseudo Op
34334    4752        JSR     IFFIX       ;Check for relational operator flip
34335     415        JMP     IF1         ;   NOT relation operator return
34336   54321        STA     3,TSX10     ;Save Pseudo Op
34337  102400        SUB     0,0         ;SAVE VALUE OF LEFT SIDE
34340   24025        LDA     1,C5
34341   30436        LDA     2,IFT
34342    6120        DECIMAL
34343    6257        ALGEXPRESSION       ;EVALUATE RIGHT SIDE
34344   54342        STA     3,TSP2      ;SAVE NEXT BYTE
34345   20002        LDA     0,C2        ;SUBTRACT LEFT SIDE
34346   24025        LDA     1,C5
34347   30430        LDA     2,IFT
34350    6120        DECIMAL
34351     402        SKIP
34352   54342 IF1:   STA     3,TSP2
34353    6137        STORDA
34354   34342        LDA     3,TSP2
34355   30216 IF1A:  LDA     2,C344
34356  172014        ADC#    3,2,SZR     ;STATEMENT FOLLOW ?
34357    4204        JSR     ERRSY       ;   NO
34360   34366        LDA     3,..BUS     ;   YES
34361   31403        LDA     2,PBC.,3    ;SET PBC TO WORD BOUNDARY
34362  151620        INCZR   2,2
34363  151120        MOVZL   2,2
34364   51403        STA     2,PBC.,3
34365   30321        LDA     2,TSX10     ;BRANCH ON DIFFERENCE
34366  101014        SKZ     0,0         ;ARE EXPRESSIONS EQUAL ?
34367     427        JMP     IF2         ;   NO
34370   34221        LDA     3,C355      ;   YES
34371  156414        SEQ     2,3         ;WAS RELATION "=" . . .
34372  156015        ADC#    2,3,SNR     ;OR ">=" ?
34373    2375        JMP     @.EXST      ;   YES
34374  172014        ADC#    3,2,SZR     ;WAS IT "<=" ?
34375    2347        JMP     @.NXLN      ;   NO, IT WAS '<>'
34376    2375        JMP     @.EXST      ;   YES

34377   34400 IFT:   .+1                 ; "IF" & "LET" TEMP STORAGE
34400      13        .BLK 13

34413     352 C352:  352
34414     353 C353:  353
34415     357 C357:  357
```

```
                                                  << SI = R92RUNS2A; BO = 18/A.RUN.8053! >>
34416  34776 IF2:   LDA     3,C353    ;EXPRESSIONS NOT EQUAL
34417 156015       ADC#    2,3,SNR   ;WAS RELATION "<>" ?
34420   2375       JMP     @.EXST    ;   YES
34421 125015       SNZ     1,1       ;WAS LEFT SIDE GREATER ?
34422    405       JMP     .+5       ;   NO
34423 156414       SEQ     2,3       ;WAS RELATION ">" . . .
34424 172015       ADC#    3,2,SNR   ;OR ">=" ?
34425   2375       JMP     @.EXST    ;   YES
34426   2247       JMP     @.NXLN    ;   NO

34427  34766       LDA     3,C357    ;RIGHT SIDE WAS GREATER
34430 156414       SEQ     2,3       ;WAS RELATION "<" . . .
34431 156015       ADC#    2,3,SNR   ;OR "<=" ?
34432   2375       JMP     @.EXST    ;   YES
34433   2247       JMP     @.NXLN    ;   NO

34434  30220 IFQ:   LDA     2,C351    ;MAY BE A LITERAL STRING
34435 172414       SEQ     3,2       ;IS IT A QUOTE ?
34436   4204       JSR     ERRSY     ;   NO
34437  34366       LDA     3,.BUS    ;   YES
34440  31403       LDA     2,PBC.,3  ;SAVE B(LEFT STRING)
34441  50311       STA     2,TSP1
34442   6264       NEXTBYTE          ;FIND END OF LEFT STRING ELEMENT
34443 145004       MOV     2,1,SZR
34444    776       JMP     .-2
34445  20341       LDA     0,TSP1
34446    403       JMP     IFS2

34447 106400 IFS:   SUB     0,1       ;LEFT EXPRESSION IS A STRING
34450 143000       ADD     2,0
34451   4501 IFS2:  JSR     IFX       ;SAVE POINTERS
34452   4514       JSR     FESX      ;FIND END OF LEFT STRING ELEMENT
34453  20345       LDA     0,TSP5
34454  54321       STA     3,TSX10   ;Save operator
34455   4631       JSR     IFFIX     ;Check for relational operator flip
34456    677       JMP     IF1A      ;   NOT a relational operator return
34457  54321       STA     3,TSX10   ;   YES, SAVE IT
34460   4525 IFR:   JSR     EVSE      ;EVALUATE RIGHT STRING ELEMENT
34461   4204       JSR     ERRSY
34462  40316       STA     0,TSP6    ;SAVE BYTE "POINTER"
34463  44312       STA     1,TSP2    ;SAVE LENGTH
34464  34366       LDA     3,.BUS
34465  31403       LDA     2,PBC.,3
34466  50313       STA     2,TSP3    ;SAVE NEXT ELEMENT LOCATION
34467  20345       LDA     0,TSP5
34470 101015       SNZ     0,0       ;END OF LEFT STRING ELEMENT ?
34471    421       JMP     IFL       ;   YES
34472  24314 IFC:   LDA     1,TSP4    ;COMPARE STRINGS
34473   6144       XGETBYTE
34474  50315       STA     2,TSP5    ;SAVE BYTE OF LEFT STRING
34475 151015       SNZ     2,2       ;END OF LEFT STRING ELEMENT ?
34476    414       JMP     IFL       ;   YES
```

```
                ;                                  << SI = R92RUNS2A; BO = 18/A.RUN.8053! >>
34477    4461           JSR     IFGB        ;GET BYTE OF LEFT STRING
34500  151015           SNZ     2,2         ;END OF LEFT STRING ELEMENT ?
34501     437           JMP     IFE         ;   YES
34502   10344           ISZ     TSP4        ;   NO, INCR. PONTER
34503   20345           LDA     0,TSP5
34504  142404           SUB     2,0,SZR     ;ARE THE BYTES EQUAL ?
34505     422           JMP     IFD         ;   NO
34506   14342           DSZ     TSP2        ;   YES, ????
34507  100010           NOP
34510   14340           DSZ     TSP         ;END OF LEFT STRING ELEMENT ?
34511     761           JMP     IFC         ;   NO
34512   24341  IFL:     LDA     1,TSP1      ;   YES, GET NEXT LEFT STRING BYTE
34513   34366           LDA     3,.BUS
34514   45403           STA     1,PBC.,3;Set current PBC to Left string expression
34515    6264           NEXTBYTE            ;Load next token from Left expression
34516   34223           LDA     3,C361
34517  156414           SEQ     2,3         ;COMMA ?
34520     405           JMP     IFD-2       ;   NO
34521    4464           JSR     EVSE        ;   YES, EVALUATE NEXT LEFT ELEMENT
34522    4204           JSR     ERRSY
34523    4427           JSR     IFX         ;SAVE POINTERS
34524     746           JMP     IFC         ;CONTINUE COMPARING

34525    4133           JSR     IFGB
34526  140400           NEG     2,0
34527   40344  IFD:     STA     0,TSP4
34530   34366           LDA     3,.BUS
34531   20343           LDA     0,TSP3
34532   41403           STA     0,PBC.,3
34533    4133           JSR     FESX
34534   20344           LDA     0,TSP4
34535  105100  IFF:     MOVL    0,1
34536  126500           SUBL    1,1
34537     616           JMP     IF1A

34540   34366  IFE:     LDA     3,.BUS
34541   20343           LDA     0,TSP3
34542   41403           STA     0,PBC.,3
34543    6264           NEXTBYTE
34544   34223           LDA     3,C361
34545  156415           SNE     2,3         ;COMMA ?
34546     712           JMP     IFR         ;   YES
34547  155000           MOV     2,3         ;   NO
34550   20345           LDA     0,TSP5
34551     764           JMP     IFF

34552   40344  IFX:     STA     0,TSP4      ;SAVE BYTE "POINTER"
34553   44340           STA     1,TSP       ;SAVE STRING LENGTH
34554   30366           LDA     2,.BUS
34555   31003           LDA     2,PBC.,2
34556   50341           STA     2,TSP1      ;SAVE BYTE "POINTER"
34557    1400           JMP     0,3
```

```
                ;                                  << SI = R92RUNS2A;  BO = 18/A.RUN.8053! >>

34560  30342 IFGB:  LDA     2,TSP2     ;GET BYTE OF LEFT STRING
34561 141035        MOV     2,0,SNR
34562   1400        JMP     0,3
34563  24346        LDA     1,TSP6
34564  10346        ISZ     TSP6
34565   2144        JMP     @XGETBYTE&377

34566  54342 FESX:  STA     3,TSP2     ;FIND END OF STRING EXPRESSION
34567 105000        MOV     0,1
34570   6144        XGETBYTE
34571  50345        STA     2,TSP5
34572   6264        NEXTBYTE
34573  34223        LDA     3,C361
34574 156414        SEQ     2,3        ;COMMA ?
34575 155001        MOV     2,3,SKP    ;   NO
34576   4407        JSR     EVSE       ;   YES, EVALUATE NEXT ELEMENT
34577   2342        JMP     @TSP2
34600  24345        LDA     1,TSP5
34601 125015        SNZ     1,1        ;????
34602    765        JMP     FESX+1
34603    767        JMP     FESX+4


34604      0        0
34605  54777 EVSE:  STA     3,.-1      ;EVALUATE STRING ELEMENT
34606   6260        EXPRESSION
34607    417        JMP     EVSE1      ;   STRING VARIABLE
34610  20220        LDA     0,C351
34611 151015        SNZ     2,2        ;NULL EXPRESSION . . .
34612 162414        SEQ     3,0        ;AND QUOTE MARK ?
34613   2771        JMP     @EVSE-1    ;   NO, NOT A STRING
34614  30356        LDA     2,.BUS     ;   YES, LITERAL STRING
34615  21003        LDA     0,PBC.,2   ;SAVE BYTE "POINTER"
34616  40352        STA     0,TSP12
34617   6264        NEXTBYTE           ;FIND END OF LITERAL
34620 151014        SKZ     2,2
34621    776        JMP     .-2
34622  20352        LDA     0,TSP12    ;BYTE "POINTER"
34623 126000        ADC     1,1        ;STRING LENGTH
34624 176400        SUB     3,3        ;FIRST "SUBSCRIPT" - 1
34625    404        JMP     EVDON

34626 115000 EVSE1: MOV     0,3        ;FIRST SUBSCRIPT - 1
34627 106400        SUB     0,1        ;STRING LENGTH
34630 143000        ADD     2,0        ;BYTE "POINTER"
34631  10753 EVDON: ISZ     EVSE-1
34632   2752        JMP     @EVSE-1
```

```
                ;                                  << SI = R92RUNS2A;  BD = 18/A.RUN.8053! >>

                ;   RANDOM <expression>

 34633    6257 RNDM:  ALGEXPRESSION    ;"RANDOM" STATEMENT
 34634  102400        SUB     0,0       ;Set Initialize Seed function code
 34635    6101        CALL              ;And call PRAND
 34636     150          PRAND
 34637   34366        LDA     3,.BUS
 34640   45445        STA     1,RNDI.,3;Save new Seed
 34641    2946        JMP     @.EDST

                .EOT   ;  "RUN" R9.0 SOURCE #2
```

```
                ;                                  << SI = R92RUNS3E; 80 = 18/A.RUN.8053! >>
                ; 25 AUG 86, RB.

                ; "RUN" SOURCE #3 OF 7 TAPES FOR "IRIS"
                ;  EDIT 17 FEB 86 BY RB., TO REMOVE DGC CRT KLOODGE USING CHM3
                ;  EDIT 25 JUN 87 BY RDC, TO ADD DGC CRT KLOODGE USING PCX (MINUS CELL)
                ; LAST EDITED 23 DEC 87 BY RDC, TO CHANGE DGC CRT KLOODGE (DEFS CHANGE OF PCXM4)

                ;  DEF FN<letter> (<letter>)= <expression>

34642    6264 DEF:   NEXTBYTE           ;"DEF" STATEMENT
34643   50302        STA     2,TS1     ;SAVE FN- CODE
34644    6264        NEXTBYTE           ;Skip over dummy argument
34645    6264        NEXTBYTE
34646   34221        LDA     3,C355
34647  156414        SEQ     2,3        ;"=" ?
34650    4204        JSR     ERRSY      ;  NO
34651   30366        LDA     2,.BUS     ;  YES
34652   35011        LDA     3,UFT.,2
34653   20302        LDA     0,TS1
34654  157000        ADD     2,3        ;.UFT
34655  117000        ADD     0,3        ;TABLE ENTRY LOCATION + 101
34656   25003        LDA     1,PBC.,2
34657   45677        STA     1,-101,3   ;DISPLACEMENT OF DEFINITION
34660    2374        JMP     @.NXST     ;Skip over function code to next statement


                ;  [LET] <variable> = <expression>
                ;  [LET] <variable> = <expression> USING <string>
                ;  <stringselect> ::= <string> [TO <string> [:<variable>]]
                ;  [LET] <stringvariable> = <stringselect> {,<stringselect>}

34661    6256 LET:   JSR     @.VLOC     ;"LET" STATEMENT
34662     437        JMP     LETS       ;ASSIGNMENT TO A STRING
34663  162414        SEQ     3,0        ;"=" SEPARATOR ?
34664    4204        JSR     ERRSY      ;  NO
34665    6260        EXPRESSION         ;  YES
34666    2421        JMP     @.LTVS     ;STRING VARIABLE (CONVERT TO MNEMONIC)
34667   20220        LDA     0,C351
34670  162415        SNE     3,0        ;LITERAL STRING ?
34671    2416        JMP     @.LTVS     ;  YES
34672   20215        LDA     0,C342     ;  NO
34673  151014        SKZ     2,2        ;NON-NULL EXPRESSION . . .
34674  116554        SUBOR#  0,3,SZR    ;And end of statement (or statement follows)
34675    4204        JSR     ERRSY      ;  NO
34676   54414 LETX:  STA     3,TBYTE    ;Save statement terminator byte
34677  102400        SUB     0,0        ;  YES
34700   24310        LDA     1,TS7
34701   30307        LDA     2,TS6
34702    6120        DECIMAL            ;STORE RESULT
34703   30407        LDA     2,TBYTE    ;Load statement terminator byte
34704    2401        JMP     @.+1       ;And execute next statement
34705   33573        EDSTX


34706   37073 .MOST: MOST
34707   35104 .LTVS: LETVS
```

```
                ;                                  << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>
                ;STRING ASSIGNMENT

                ;FLAGS -- 0 = FALSE,   -1 = TRUE

34710       0 SER:   0          ;STRING ERROR
34711       0 APEND: 0          ;APPENDING

                ;TEMP CELLS

34712       0 TBYTE: 0          ;TERMINATOR (FROM EVSE ETC.)
34713       0 SBP:   0          ;STRING BYTE POINTER
34714       0 SS:    0          ;STRING SIZE
34715       0 SFS:   0          ;STRING'S FIRST SUBSCRIPT
34716       0 ATFH:  0          ;ALTERNATE TERMINATOR FOUND HERE
34717       0 DPS1:  0          ;SAVE DESTINATION PTR 1
34720       0 DPS2:  0          ;SAVE DESTINATION PTR 2

34721   54343 LETS:  STA     3,TSP3    ;SET A (SWAPPABLE) FLAG
34722    6264        NEXTBYTE
34723   34221        LDA     3,C355
34724  156414        SEQ     2,3       ;"=" ?
34725    4204        JSR     ERRSY     ;   NO, SYNTAX ERROR
34726   20223        LDA     0,C361
34727   40763        STA     0,TBYTE   ;TBYTE := ","
34730  102400        SUB     0,0
34731   40757        STA     0,SER     ;STRING.ERROR := FALSE
34732   40757        STA     0,APEND   ;APPENDING := FALSE
34733   20757 LETSS: LDA     0,TBYTE
34734   24223        LDA     1,C361
34735  106414        SEQ     0,1       ;TBYTE = "," ?
34736     500        JMP     LDONE     ;   NO, DONE
34737   20752        LDA     0,APEND
34740  101015        SNZ     0,0       ;APPENDING ?
34741     404        JMP     LEISA     ;   NO
34742   20310        LDA     0,TS7     ;   YES
34743  101015        SNZ     0,0       ;DESTINATION FULL ?
34744    2374        JMP     @.NXST    ;   YES, ALL DONE, EXECUTE NEXT STMT
34745  102000 LETSA: ADC     0,0
34746   40743        STA     0,APEND   ;APPENDING := TRUE
34747    4606        JSR     EVSE      ;EVALUATE SOURCE STRING
34750     457        JMP     LETSV     ;NOT A STRING, TRY FOR VARIABLE
34751   40742        STA     0,SBP     ;SAVE BYTE POINTER ...
34752   44742        STA     1,SS      ;AND LENGTH ...
34753   54742        STA     3,SFS     ;AND FIRST SUBSCRIPT
34754    6264        NEXTBYTE
34755   50735        STA     2,TBYTE   ;SAVE TERMINATOR
34756   34216        LDA     3,C344
34757  156414        SEQ     2,3       ;CODE FOR "TO" ?
34760     451        JMP     NOTTO     ;   NO
34761    4624        JSR     EVSE      ;   YES, EVALUATE TERMINATOR STRING
34762    4204        JSR     ERRSY     ;NOT A STRING, SYNTAX ERROR
```

```
                ;                                  << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>
34763 105000            MOV     0,1             ;GET FIRST BYTE OF STRING
34764   6164            XGETBYTE                ;THIS IS OUR ALTERNATE TERMINATOR
34765  20726            LDA     0,SBP           ;FETCH BYTE POINTER TO SOURCE
34766  24726            LDA     1,SS            ;AND ITS SIZE
34767   6717            JSR     @.MOST          ;MOVESTRING USING ALTERNATE TERMINATOR
34770   6264            NEXTBYTE
34771  50721            STA     2,TBYTE         ;SAVE TERMINATOR
34772  34230            LDA     3,C376
34773 156414            SEQ     2,3             ;IS IT CODE FOR ":" ?
34774    737            JMP     LETSS           ;   NO, PROCESS NEXT SOURCE STRING
34775  20716            LDA     0,SBP           ;INITIAL SOURCE BYTE PTR
34776  24317            LDA     1,TSX6          ;CURRENT SOURCE PTR + 1
34777 106400            SUB     0,1             ;OFFSET WHERE ALT TERM FOUND
35000  20715            LDA     0,SFS           ;FIRST SUBSCRIPT - 1
35001 107000            ADD     0,1             ;SUBSCRIPT WHERE ALT TERM FOUND
35002  20320            LDA     0,TSX7
35003 101414            INC#    0,0,SZR         ;ALT TERM FOUND ?
35004 126400            SUB     1,1             ;   NO
35005  44711            STA     1,ATFH          ;SAVE ... ALTERNATE TERM FOUND HERE
35006  20307            LDA     0,TS6
35007  40710            STA     0,DPS1          ;SAVE DESTINATION PTR 1
35010  20310            LDA     0,TS7
35011  40707            STA     0,DPS2          ;SAVE DESTINATION PTR 2
35012   6256            JSR     @.VLOC          ;LOCATE VARIABLE
35013   4204            JSR     ERRSY           ;   STRING NOT ALLOWED HERE
35014  54676            STA     3,TBYTE         ;SAVE TERMINATOR
35015 102400            SUB     0,0
35016  24700            LDA     1,ATFH          ;FETCH PTR TO ALT TERMINATOR
35017   6122            FLOAT                   ;FLOAT IT
35020 102400            SUB     0,0
35021  24310            LDA     1,TS7
35022  30307            LDA     2,TS6
35023   6120            DECIMAL                 ;STORE IT IN LOCATED VARIABLE
35024  20673            LDA     0,DPS1
35025  40307            STA     0,TS6           ;RESTORE DESTINATION PTR 1
35026  20672            LDA     0,DPS2
35027  40310            STA     0,TS7           ;RESTORE DESTINATION PTR 2
35030    703            JMP     LETSS           ;PROCESS NEXT STRING

35031  20652 NOTTO:     LDA     0,SBP           ;FETCH BYTE POINTER TO SOURCE
35032  24652            LDA     1,SS            ;AND ITS SIZE
35033 152400            SUB     2,2             ;OUR TERMINATOR CODE
35034   6652            JSR     @.MOST          ;MOVESTRING USING 0
35035    676            JMP     LETSS           ;PROCESS NEXT SOURCE STRING

35036   2246 LDONE:     JMP     @.EDST          ;All done, execute next statement
```

```
                        ;                                  << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>
35037   30215 LETSV:LDA     2,C342      ;CONVERT NUMERIC TO STRING
35040 156654        SUBDR#  2,3,SZR     ;End of statement ?
35041    515        JMP     LETSU       ;   NO
                                        ;   YES
35042   20307        LDA     0,TS6       ;PROTECT SOME VOLITILE INFO AGAINST SWAPPING
35043   40315        STA     0,TSP5
35044   20310        LDA     0,TS7
35045   40317        STA     0,TSP7
35046    6270        XFLUSH              ;CLEAN OUT XOB BEFORE CLOBBERING IT
35047   24055        LDA     1,.PTBL     ;GET PTR TO TABLE FOR DECIMAL OUTPUT
35050   20027        LDA     0,C7
35051    6120        DECIMAL             ;DECIMAL OUTPUT
35052 102400        SUB     0,0
35053   40305        STA     0,TS4
35054   24055        LDA     1,@.PTBL    ;GET ADR OF OUTPUT STRING (XOB)
35055   44306        STA     1,TS5
35056   24306 LETS1:LDA     1,TS5       ;GET BYTE OF OUTPUT STRING
35057   10306        ISZ     TS5
35060    6144        XGETBYTE
35061 141000        MOV     2,0
35062   24345        LDA     1,TSP5      ;PUT BYTE IN STRING VARIABLE
35063   10315        ISZ     TSP5
35064    6145        XPUTBYTE
35065   34305        LDA     3,TS4
35066 175015        SNZ     3,3         ;FIRST BYTE ?
35067     404        JMP     LETSC       ;   YES
35070   34056        LDA     3,C240      ;   NO
35071 116415        SNE     0,3         ;LAST BYTE ?
35072     404        JMP     LETSD       ;   YES
35073   10305 LETSC:ISZ     TS4         ;   NO, BUMP BYTE COUNT
35074   14347        DSZ     TSP7        ;END OF STRING ?
35075     741        JMP     LETS1       ;   NO
35076   20313 LETSD:LDA     0,TSP3      ;   YES
35077   24315        LDA     1,TSP5
35100 101015        SNZ     0,0         ;ZERO BYTE NEEDED ?
35101    6145        XPUTBYTE            ;   YES
35102    2946        JMP     @.EDST      ;Execute next statement
```

```
                ;                                  << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>
35103   34676 .LETX:LETX

35104   30366 LETVS:LDA     2,.BUS     ;CONVERT STRING TO NUMERIC VARIABLE
35105   15003         DSZ     PBC.,2
35106   15003         DSZ     PBC.,2     ;SEARCH BACK TO "="
35107    6264         NEXTBYTE
35110   34221         LDA     3,C355
35111  156114         SEQ     2,3        ;"=" ?
35112     772         JMP     LETVS      ;   NO
35113    6242         JSR     @.EVSE     ;   YES, GET STRING DOPE
35114    4204         JSR     ERRSY      ;NOT A STRING
35115   40305         STA     0,TS4      ;B(STRING)
35116   44306         STA     1,TS5      ;BYTE COUNT
35117    6264         NEXTBYTE
35120   34215         LDA     3,C342
35121  172654         SUBOR#  3,2,SZR    ;End of statement ?
35122    4204         JSR     ERRSY      ;   NO
35123   50302         STA     2,TS1      ;Save statement terminator byte
35124   20305         LDA     0,TS4      ;   YES
35125   24306         LDA     1,TS5
35126  125415         INC#    1,1,SNR    ;LITERAL STRING ?
35127     407         JMP     LETV1      ;   YES
35130  107000         ADD     0,1        ;   NO
35131   44304         STA     1,TS3      ;B(END OF STRING)
35132    6144         XGETBYTE
35133   50303         STA     2,TS2      ;SAVE LAST BYTE
35134  102400         SUB     0,0
35135    6145         XPUTBYTE           ;TEMPORARY TERMINATOR
35136   20026 LETV1:LDA     0,C6
35137   24305         LDA     1,TS4
35140    6120         DECIMAL            ;DECIMAL INPUT
35141    4410         JSR     LETV2      ;INPUT ERROR
35142   20303         LDA     0,TS2
35143   24304         LDA     1,TS3
35144   34306         LDA     3,TS5
35145  175414         INC#    3,3,SZR    ;LITERAL STRING USED ?
35146    6145         XPUTBYTE           ;   NO, RESTORE SAVED BYTE
35147   34302         LDA     3,TS1      ;Load statement terminator byte
35150    2703         JMP     @.LETX

35151  102400 LETV2:SUB     0,0        ;INPUT ERROR, USE ZERO
35152  126400         SUB     1,1
35153    2122         JMP     @FLOAT&377


35154     307 .LUTB:TS6      ;FOR FORMATTED OUTPUT IN LET USING
                           ;THIS ASSUMES TS6 CONTAINS THE DEST PTR AND
                           ;TS7 CONTAINS THE DEST SIZE WHEN FORMATTED
                           ;OUTPUT IS CALLED
```

```
                ;                               << SI = R92RUNSGE; BO = 18/A.RUN.8053! >>
035155  34400           IFT+1
035156  20217 LETSU:LDA     0,C347      ;MAY BE "LET USING"
035157 162404           SUB     3,0,SZR     ;"USING" SEPARATOR ?
035160   4204           JSR     ERRSY       ;   NO
035161  24025           LDA     1,C5        ;   YES
035162  30773           LDA     2,LETSU-1
035163   6120           DECIMAL             ;SAVE DA IN IFT
035164   6242           JSR     @.EVSE      ;GET STRING DOPE
035165   4204           JSR     ERRSY       ;NOT A STRING
035166  40330           STA     0,TSE       ;B(FORMAT $)
035167  44304           STA     1,TS3       ;LEN(FORMAT $) AS RETURNED BY EVSE
035170 126400           SUB     1,1         ;GEN A ZERO
035171  44331           STA     1,TSE1      ;INITIALIZE COUNTER
035172  24330           LDA     1,TSE       ;FIND LEN(FORMAT $)
035173  44303           STA     1,TS2
035174  24303 LETU6:LDA     1,TS2
035175  10303           ISZ     TS2
035176   6144           XGETBYTE
035177 151015           SNZ     2,2         ;LAST BYTE ?
035200    403           JMP     LETS3       ;   YES
035201  10331           ISZ     TSE1        ;   NO, BUMP BYTE COUNT
035202    7/2           JMP     LETU6

035203  30304 LETS3:LDA     2,TS3       ;LENGTH COMPUTED BY EVSE
035204  24331           LDA     1,TSE1      ;LENGTH COMPUTED HERE
035205 132403           SLE     1,2         ;USE THE SMALLEST
035206 145000           MOV     2,1
035207  30310           LDA     2,TS7       ;LEN(TARGET $)
035210 132403           SLE     1,2
035211 145000           MOV     2,1         ;SMALLEST LENGTH
035212  30330           LDA     2,TSE
035213  34307           LDA     3,TS6
035214 137000           ADD     1,3
035215  54313           STA     3,TSX2      ;B(END OF TARGET $)
035216 147000           ADD     2,1
035217  44304           STA     1,TS3       ;B(END OF FORMAT $)
035220   6144           XGETBYTE
035221  50305           STA     2,TS4       ;SAVED BYTE
035222 102400           SUB     0,0
035223  40331           STA     0,TSE1      ;INITIALIZE "FORMAT STRING OK" FLAG TO NOT OK
035224   6145           XPUTBYTE            ;TEMPORARY TERMINATOR
035225 102520           SUBZL   0,0
035226  24025           LDA     1,C5
035227  30726           LDA     2,LETSU-1
035230   6120           DECIMAL             ;RESTORE DA
035231  20031           LDA     0,C11
035232  24722           LDA     1,.LU1B     ;PTR TO TARGET $
035233  30330           LDA     2,TSE       ;B(FORMAT $)
035234   6120           DECIMAL             ;FORMATTED OUTPUT
035235    407           JMP     LETS4       ;ERROR, REPLACE BYTE
035236  14331           DSZ     TSE1        ;MARK ASSIGNMENT OK
035237 102400           SUB     0,0
035240  24313           LDA     1,TSX2      ;B(END OF TARGET $)
035241  34313           LDA     3,TSP3      ;2ND SS FLAG
035242 175015           SNZ     3,3         ;TERMINATOR NEEDED ?
035243   6145           XPUTBYTE            ;   YES
```

```
                        ;                               << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>
035244   20305 LETS4:LDA     0,TS4
035245   24304         LDA     1,TS3      ;B(END OF FORMAT $)
035246    6145         XPUTBYTE           ;RESTORE SAVED BYTE
035247   10331         ISZ     TSE1       ;FLAG MARKED OK?
035250    6267         RUNERROR           ;   NOPE, STRING ERROR
035251  135110         72*K+SE            ;ERROR CODE
035252    2374         JMP     @.NXST


                        ;   DIM statement

035253    6264 DIM:   NEXTBYTE           ;"DIM" STATEMENT
035254   24024         LDA     1,C4
035255  146403         SLE     2,1        ;SET PRECISION ?
035256     421         JMP     DIM1       ;   NO
035257   50310         STA     2,TS7      ;   MAYBE
035260    6264         NEXTBYTE
035261   24213         LDA     1,C336
035262   20310         LDA     0,TS7
035263  100104         NEG     0,0,SZR    ;PRECISION ZERO . . .
035264  146114         SEQ     2,1        ;OR NO PERCENT SIGN ?
035265    4204         JSR     ERRSY      ;   YES
035266  100000         COM     0,0        ;   NO, OK
035267   34366         LDA     3,.BUS
035270   25424         LDA     1,FLAG.,3
035271   30405         LDA     2,DIM1-1
035272  147400         AND     2,1
035273  107000         ADD     0,1        ;SET PRECISION SWITCH
035274   45424         STA     1,FLAG.,3
035275     424         JMP     DIM3

035276  177774         -4
035277   34366 DIM1:  LDA     3,.BUS     ;DIMENSION A VARIABLE
035300   15403         DSZ     PBC.,3
035301    6243         JSR     @.EVSU
035302  101015         SNZ     0,0        ;SUBSCRIPTS BOTH ZERO ?
035303  125014         SKZ     1,1
035304     412         JMP     DIM2       ;   NO
035305    6251         JSR     @.LUVD     ;   YES, FIND THE VARIABLE
035306  103112         ADDL#   0,0,SZC    ;STRING VARIABLE ?
035307     407         JMP     DIM2       ;   YES
035310  151014         SKZ     2,2        ;LOCATION ASSIGNED ?
035311    6267         RUNERROR           ;   YES, CAN'T CHANGE PRECISION
035312  113510         27*K+SE
035313    6402         JSR     @.+2       ;   NO, ASSIGN LOC & PREC
035314     405         JMP     DIM3
035315   40770         ALOC

035316  151015 DIM2:  SNZ     2,2        ;LOCATION ALREADY ASSIGNED ?
035317    4204         JSR     ERRSY      ;   NO
035320    6203         JSR     @.ADIM     ;   YES
035321    6264 DIM3:  NEXTBYTE
035322   34203         LDA     3,C361
035323  156415         SNE     2,3        ;COMMA ?
035324     727         JMP     DIM        ;   YES
035325    2246         JMP     @.EDST     ;   NO
```

```
                ;                                  << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>
035326  34366 CHAIN: LDA     3,.BUS     ;"CHAIN" STATEMENT
035327  25403        LDA     1,PBC.,3
035330   6144        XGETBYTE           ;GET NEXT BYTE OFFSET
035331  20212        LDA     0,C136     ;"WRITE" TOKEN
035332 142415        SNE     2,0        ;IS IT "CHAINREAD"?
035333   6267        RUNERROR
035334 116610        35*K+AE
035335 142015        ADC#    2,0,SNR    ;IS IT "CHAINREAD" ?
035336   2247        JMP     @.NXLN     ;   YES, IGNORE
035337   6261        MTOST              ;   NO, FLUSH XOB
035340   6242        JSR     @.EVSE     ;EVALUATE STRING ELEMENT
035341   4204        JSR     ERRSY
035342  40301        STA     0,TS       ;RELATIVE BYTE ADDRESS
035343  44302        STA     1,TS1      ;STRING LENGTH
035344  20231        LDA     0,.EL
035345 101120        MOVZL   0,0        ;MOVE STRING TO "EL"
035346  40306        STA     0,TS5
035347  24053        LDA     1,C200
035350   4171        JSR     MOVST
035351  20055        LDA     0,C215     ;APPEND A RETURN CODE
035352   6134        PUTBYTE
035353  20303        LDA     0,TS2
035354  24306        LDA     1,TS5      ;GET THE NEW PROGRAM
035355   6101        CALL
035356    100        LINKPROGRAM
035357    417        JMP     CHAER      ;   ERROR
035360 175015        SNZ     3,3        ;WAS A PROGRAM LOADED ?
035361    422        JMP     CHAI1
035362  34366        LDA     3,.BUS     ;   No
035363  31405        LDA     2,SLT.,3
035364 173000        ADD     3,2
035365  21000        LDA     0,0,2      ;Statement #
035366  25001        LDA     1,1,2      ;Statement offset
035367 137000        ADD     1,3        ;A(Statement)
035370  24150        LDA     1,RMCON    ;"REM" token
035371 101015        SNZ     0,0        ;Statement 0 ?
035372  45400        STA     1,0,3      ;   Yes, make into a REM
035373   6101        CALL               ;NO
035374 100000        SCOPE


035375 104200        104200
035376  20777 CHAER: LDA     0,.-1      ;ERROR IN CHAIN
035377 163300        ADDS    3,0
035400  40402        STA     0,.+2      ;<ERROR #> *K+AE
035401   6267        RUNERROR
035402 100210        AE
```

```
                        ;                               << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>

 35403    6431 CHAI1: JSR     @CHAI3    ;SETUP for run
 35404    6433        JSR     @CHAI6    ;"RESTOR" (reset) DATA ptr and cntr
 35405  102400        SUB     0,0
 35406  40076         STA     0,ERRF    ;Clear $DEC arithmetic error flag
 35407    6427        JSR     @CHAI5    ;Clear all variable, stacks, and reset PLC

                        ;  Search for first executable statement (not DATA or REM) and
                        ;  start execution at that statement.  Skip ESCAPE/CNTL-C check
                        ;  on first statement to allow IF ERR to be set on first statement.
                        ;  This provides ESCAPE control during CHAINing.

 35410   34366            LDA     3,.BUS
 35411   31402            LDA     2,PLC.,3
 35412   25401            LDA     1,VDT.,3
 35413 1460033            SLS     2,1       ;Is the current line within the pgm?
 35414    2421            JMP     @CHAI4    ;  No, end of program
 35415  173000            ADD     3,2
 35416   25001            LDA     1,1,2     ;Load relative word addr of line
 35417  125120            MOVZL   1,1       ;Convert to byte address
 35420   45403            STA     1,PBC.,3  ;Set PBC to current statement
 35421     403            JMP     CHNCK     ;And check if line is executable

 35422    6250 CHNNX: JSR     @.SKST    ;Skip to next statement
 35423    2412        JMP     @CHAI4    ;  No next statement, End of program
 35424   30366 CHNCK: LDA     2,.BUS
 35425   25003        LDA     1,PBC.,2
 35426    6144        XGETBYTE           ;Load statement code of statement
 35427   24211        LDA     1,C125
 35430 1332655        SUBOR#  1,2,SNR    ;DATA or REM statement?
 35431     771        JMP     CHNNX     ;  Yes, try next statement
 35432    2401        JMP     @CHAI2    ;  No, Execute statement

 35433   33627 CHAI2: EXST1
 35434     420 CHAI3: SATUP
 35435   33717 CHAI4: END
 35436   32415 CHAI5: CVSP              ;Routine to init program
 35437   32227 CHAI6: RESTR
 35440   53000 RMCON: 53000             ;REM Token
```

```
                ;                                  << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>

                ; MOVE STRING FROM USER PARTITION TO LOWER CORE
                ;   TS    = FIRST SOURCE BYTE ADDRESS (RELATIVE)
                ;   TS1 = MAX NUMBER OF SOURCE BYTES
                ;   A0    = FIRST DEST BYTE ADDRESS (ABSOLUTE)
                ;   A1    = MAX NUMBER OF DEST BYTES

 055441   54305 MOVST: STA     3,TS4      ;MOVE STRING INTO RUN AREA
 055442   40303         STA     0,TS2
 055443   44304         STA     1,TS3
 055444   24301         LDA     1,TS       ;GET A SOURCE BYTE
 055445   10301         ISZ     TS
 055446    6144         XGETBYTE
 055447   24303         LDA     1,TS2
 055450  141005         MOV     2,0,SNR    ;END OF SOURCE ?
 055451    2305         JMP     @TS4       ;   YES
 055452   10303         ISZ     TS2        ;   NO, STORE IN DEST
 055453    6134         PUTBYTE
 055454   14304         DSZ     TS3        ;DEST. FULL ?
 055455     402         JMP     .+2        ;   NO
 055456     403         JMP     .+3        ;   YES
 055457   14302         DSZ     TS1        ;END OF SOURCE ?
 055460     764         JMP     MOVST+3    ;   NO
 055461   24303         LDA     1,TS2      ;   YES
 055462    2305         JMP     @TS4       ;(A1) = NEXT DEST BYTE ADDRESS
```

```
                    ;                              << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>

                    ;  PRINT statement

 055463   32703          CKEM
 055464   20034          WRITITEM
 055465     275          275
 055466  102400 PRINT:   SUB     0,0          ;CLEAR @ FLAG
 055467   40350          STA     0,ATF
 055470   20774          LDA     0,PRINT-2    ;"PRINT" STATEMENT
 055471    6236          JSR     @.EVCH       ;PRINT TO CHANNEL ?
 055472     407          JMP     PRNTA        ;   NO
 055473   20350          LDA     0,TSN5       ;   YES
 055474    6101          CALL
 055475  100002          CHKCHANNEL
 055476     403          JMP     PRNTA        ;CHANNEL NOT OPEN
 055477    6261          MTOST
 055500  126401          SUB     1,1,SKP      ;CHANNEL OPEN
 055501  126000 PRNTA:   ADC     1,1          ;CHANNEL NOT OPEN
 055502   34346          LDA     3,.BUS
 055503   45175          STA     1,TSU.2,3    ;SET PRINT MODE FLAG (-1=TERMINAL)
 055504   25403          LDA     1,PBC.,3
 055505    6144          XGETBYTE
 055506  102400          SUB     0,0
 055507   34217          LDA     3,C347
 055510  156414          SEQ     2,3          ;PRINT USING ?
 055511     404          JMP     .+4          ;   NO
 055512    6264          NEXTBYTE             ;   YES
 055513    6242          JSR     @.EVSE       ;FIND THE FORMAT STRING
 055514    4204          JSR     ERRSY        ;   NOT A STRING
 055515   34346          LDA     3,.BUS
 055516   41473          STA     0,TSU.,3
 055517   41474          STA     0,TSU.1,3
 055520  101015          SNZ     0,0          ;PRINT USING ?
 055521     405          JMP     PRNTE        ;   NO
 055522    6264          NEXTBYTE             ;   YES
 055523   34222          LDA     3,C360
 055524  156414          SEQ     2,3          ;";" ?
 055525    4204          JSR     ERRSY        ;   NO
 055526   30737 PRNTE:   LDA     2,PRINT-1    ;"="
 055527    6734          JSR     @PRINT-3     ;In Calculator mode?
 055530    6265          QBYTE                ;   YES

                     ;*** NO CODE HERE ***
```

```
                ;                              << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>

                ;**** NO CODE HERE ****

035531    6260 PRTNF: EXPRESSION          ;PROCESS NEXT FIELD
035532    2461        JMP     @.PRTS      ;STRING VARIABLE
035533  151014        SKZ     2,2         ;EXPRESSION ?
035534     450        JMP     PRNTN       ;  YES
035535   30220        LDA     2,C351      ;  NO
035536  172415        SNE     3,2         ;LITERAL STRING ?
035537     551        JMP     PRNTL       ;  YES
035540   54313        STA     3,TSP3      ;SAVE TERMINATOR
035541   30216        LDA     2,C344
035542  156015        ADC#    2,3,SNR     ;TAB FUNCTION ?
035543    2513        JMP     @.PRTT      ;  YES
035544   30225        LDA     2,C365
035545  172415        SNE     3,2         ;"@" ?
035546     560        JMP     PRTPC       ;  YES, POSITION CURSOR
035547   30227        LDA     2,C375
035550  172415        SNE     3,2         ;SINGLE QUOTE ?
035551    2431        JMP     @.PRCF      ;  TERMINAL CONTROL FUNCTION
035552   34313 PRNTD: LDA     3,TSP3      ;DONE WITH A FIELD OR NULL FIELD
035553   20222        LDA     0,C360
035554  116015        ADC#    0,3,SNR     ;COMMA ?
035555     512        JMP     PRNTC       ;  YES
035556  162415        SNE     3,0         ;SEMI-COLON ?
035557     530        JMP     PRTSC       ;  YES
035560   30215        LDA     2,C342      ;  NO
035561  156655        SUBOR#  2,3,SNR     ;End of statement ?
035562     466        JMP     PRNTR       ;  YES
                                           ;  No, Check if legal PRINT item
035563  172402        SGR     3,2         ;Variable, Function, Or Literal?
035564     421        JMP     PRNDS       ;  Yes, Start of expression, Assume semicolon
035565   24220        LDA     1,C351
035566   30227        LDA     2,C375
035567  136114        SEQ     1,3         ;String or
035570  156115        SNE     2,3         ;  $TERMS codes?
035571     414        JMP     PRNDS       ;  Yes, Assume semicolon
035572   24224        LDA     1,C362
035573   30225        LDA     2,C365
035574  136414        SEQ     1,3         ;Left parenthesis (expression) or
035575  156415        SNE     2,3         ;  AT sign (position cursor)?
035576     407        JMP     PRNDS       ;  Yes, Assume semicolon
035577   24112        LDA     1,VMINUS
035600   30216        LDA     2,C344
035601  156014        ADC#    2,3,SZR     ;TAB function or
035602  136655        SUBOR#  1,3,SNR     ;  Plus or Minus sign?
035603     402        JMP     PRNDS       ;  Yes, Start of expression, Assume semicolon
035604    4204        JSR     ERRSY       ;  No, error
035605   40313 PRNDS: STA     0,TSP3      ;Assume semicolon
035606   34366        LDA     3,.BUS
035607   15403        DSZ     PBC,,3      ;BACK UP, AND
035610     721 JPRTF: JMP     PRTNF       ;PROCESS THE NEXT FIELD

035611     370 VMINUS:        370         ;Token code for MINUS
035612   35762 .PRCF: PRTCF
035613   35732 .PRTS: PRNTS
```

```
                ;                                  << SI = R92RUNS3E; B0 = 18/A.RUN.8053! >>
035614   54343 PRNTN: STA     3,TSP3     ;NUMERIC EXPRESSION
035615   34366        LDA     3,.BUS     ;SAVE DEST BYTE ADDR
035616    6270        XFLUSH
035617   34366        LDA     3,.BUS
035620   31473        LDA     2,TSU.,3
035621  151015        SNZ     2,2        ;PRINT USING ?
035622     407        JMP     PRTN1      ;   NO
035623   20031        LDA     0,C11      ;   YES
035624   24065        LDA     1,.PTBL
035625    6120        DECIMAL
035626    6267        RUNERROR           ;FORMAT STRING ERROR
035627  135110        72*K+SE
035630     404        JMP     PRTN2


035631   20027 PRTN1: LDA     0,C7       ;OUTPUT
035632   24065        LDA     1,.PTBL
035633    6120        DECIMAL
035634   34366 PRTN2: LDA     3,.BUS
035635  151015        SNZ     2,2        ;END OF FORMAT STRING ?
035636   31474        LDA     2,TSU.1,3  ;   YES, USE IT AGAIN
035637   51473        STA     2,TSU.,3   ;    NO, REMEMBER WHERE TO START
035640   45473        STA     1,ROBP.,3  ;UPDATE POINTER TO XOB
035641   20272        LDA     0,RFBA
035642  106400        SUB     0,1        ;# BYTES OUTPUT
035643   30005        LDA     2,RUP
035644   21033        LDA     0,OCC.,2
035645  123000        ADD     1,0        ;ADD TO OLD OCC
035646   41033        STA     0,OCC.,2   ;UPDATING IT
035647     703 JPRND: JMP     PRNTD
```

```
                ;                                  << SI = R92RUNS3E;  BO = 18/A.RUN.8053! >>
35650  30055 PRNTR:LDA     2,C215     ;END OF STATEMENT, OUTPUT A CR  << FROM PRNTD
35651   6265        QBYTE
35652 152400 PRNTB:SUB     2,2        ;OUTPUT A LINE
35653  50341        STA     2,TSP1
35654   6234        JSR     @.CESF     ;CHECK FOR ESC
35655  34366        LDA     3,.BUS
35656  31475        LDA     2,TSU.2,3
35657 151014        SKZ     2,2        ;CHANNEL OUTPUT ?
35660    406        JMP     PRDON      ;   NO
35661   6265        QBYTE              ;   YES, TERMINATE "RECORD" (WITH ZERO BYTE)
35662   6270        XFLUSH             ;FLUSH XOB
35663  34366        LDA     3,.BUS
35664 126000        ADC     1,1        ;RESET 'PRINT TO CHANNEL' FLAG
35665  45475        STA     1,TSU.2,3
35666   2246 PRDON:JMP     @.EDST

35667   6264 PRNTC:NEXTBYTE
35670  50343        STA     2,TSP3
35671  24037        LDA     1,C17      ;COLUMN TAB
35672  34005        LDA     3,RUP
35673  21433        LDA     0,OCC.,3
35674 122422        SUBZ    1,0,SZC    ;MODULO 15
35675    777        JMP     .-1
35676 123000        ADD     1,0
35677 106400        SUB     0,1
35700  44345 PRNT3:STA     1,TSP5     ;OUTPUT (A1) SPACES
35701  30056        LDA     2,C240
35702   6265        QBYTE
35703  14345        DSZ     TSP5
35704    775        JMP     .-3
35705    615        JMP     PRNTD

35706  35751 .PRTT:PRNTT

35707   6265        QBYTE
35710   6264 PRNTL:NEXTBYTE            ;LITERAL STRING
35711 151014        SKZ     2,2        ;END OF STRING ?
35712    775        JMP     PRNTL-1    ;   NO
35713   6264        NEXTBYTE           ;   YES
35714  50343        STA     2,TSP3
35715    635        JMP     PRNTD

35716  30343 PTEOL:LDA     2,TSP3     ;FETCH TERMINATOR
35717  34215        LDA     3,C342
35720 172655        SUBOR#  3,2,SNR    ;End of statement?
35721    731        JMP     PRNTB      ;   YES
35722  30366        LDA     2,.BUS     ;   NO
35723  15003        DSZ     PBC.,2     ;RESTORE "NEXTBYTED" BYTE
35724    664        JMP     JPRTF

35725  36275 .GCP:  GCP
35726   6777 PRTPC:JSR     @.GCP
35727   6264 PRTSC:NEXTBYTE            ;SEMI-COLON
35730  50343        STA     2,TSP3
35731    765        JMP     PTEOL
```

```
                ;                                << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>
035732 106400 PRNTS: SUB     0,1        ;STRING VARIABLE
035733  44345         STA     1,TSP5     ;STRING LENGTH
035734 113000         ADD     0,2
035735  50347         STA     2,TSP7     ;BYTE POINTER TO STRING
035736   6264         NEXTBYTE
035737  50313         STA     2,TSP3     ;SAVE CORRECT TERMINATOR
035740  24347 PRNS1: LDA     1,TSP7     ;B(STRING)
035741  10347         ISZ     TSP7
035742   6144         XGETBYTE
035743 151015         SNZ     2,2        ;END OF STRING ?
035744    606         JMP     PRNTD      ;   YES
035745   6265         QBYTE              ;   NO
035746  14345         DSZ     TSP5       ;END OF FIELD ?
035747    771         JMP     PRNS1      ;   NO
035750    677         JMP     JPRND      ;   YES

035751   6257 PRNTT: ALGEXPRESSION       ;TAB FUNCTION
035752  54313         STA     3,TSP3
035753   6245         JSR     @.FIXS
035754  34005         LDA     3,RUP
035755  31433         LDA     2,OCC.,3
035756 146400         SUB     2,1
035757 124513         NEGL#   1,1,SNC     ;AT TAB STOP ?
035760    667         JMP     JPRND      ;   YES
035761    717         JMP     PRNT3      ;   NO

035762   6264 PRTCF: NEXTBYTE            ;CONTROL FUNCTION
035763  24297         LDA     1,C375     ;SINGLE QUOTE
035764 132415         SNE     1,2        ;DONE WITH FIELD ?
035765    403         JMP     PTCF1      ;   YES
035766   6265         QBYTE              ;   NO, OUTPUT CONTROL CHAR . . .
035767    773         JMP     PRTCF      ;AND PROCESS NEXT BYTE

035770   6264 PTCF1: NEXTBYTE            ;FETCH . . .
035771  50313         STA     2,TSP3     ;AND SAVE TERMINATOR
035772    655         JMP     JPRND
```

```
                ;                                  << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>
                ;"INPUT" STATEMENT

                ;TEMP CELL ASSIGNMENTS:

              343 IFLW= TSP3    ;B15 - NEED PROMPT, B0 - NOT END OF STATEMENT
              345 ITAC= TSP5    ;INPUT TERMINATOR A COMMA FLAG
              347 PBCS= TSP7

                ;FOR FLAGS:  0=TRUE   1=FALSE


                ;ROUTINE TO CHECK IF INPUT TERMINATOR IS A COMMA.
                ;RUNERROR("TOO MANY NUMBERS INPUT") IF SO

 35773  20345 IITAC:LDA     0,ITAC    ;LOAD INPUT TERMINATOR A COMMA FLAG
 35774 101015        SNZ     0,0       ;TRUE ?
 35775   6267        RUNERROR          ;   YES
 35776 136110        74*K+SE;  NO
 35777   1400        JMP     0,3


 36000 102400 INPUT: SUB     0,0       ;CLEAR @ FLAG
 36001  40350        STA     0,ATF
 36002   6270        XFLUSH
 36003  30366        LDA     2,.BUS
 36004 126000        ADC     1,1       ;I/O TO TERMINAL, NOT CHANNEL
 36005  45075        STA     1,TSU.2,2
```

```
                        ;                                  << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>
036006  102400                SUB     0,0        ;INITIALIZE FLAGS
036007  40343                 STA     0,IFLW     ;NEED PROMPT AND NOT END OF STATEMENT
036010  40363                 STA     0,TSN10    ;SET TIMOUT TO NONE
036011  40364                 STA     0,TSN11    ;SET INPUT LENGTH TO UNLIMITED
036012  102020                ADCZ    0,0
036013  40365                 STA     0,ITAC     ;INPUT TERMINATOR NOT A COMMA
036014  20343   INP2:         LDA     0,IFLW
036015  101212                MOVR#   0,0,SZC    ;HIT END OF STATEMENT YET ?
036016    2464                JMP     @.IEND     ;  YES
036017    6264  INP2A:        NEXTBYTE           ;Get next code byte
036020  34297                 LDA     3,C375
036021  172414                SEQ     3,2        ;Single quote ?
036022     424                JMP     INP2B      ;  No
036023    4750                JSR     IITAC      ;  Yes, check input terminator
036024    6264  INPMN:        NEXTBYTE           ;Get next code byte
036025  34297                 LDA     3,C375
036026  172415                SNE     3,2        ;Terminating single quote ?
036027     403                JMP     CHKSC      ;  Yes, check semicolon terminator
036030    6265                QBYTE   ;  No, output mnemonic code
036031     773                JMP     INPMN
036032    6264  CHKSC:        NEXTBYTE
036033  34292                 LDA     3,C360
036034  172414                SEQ     3,2        ;Semicolon terminator ?
036035    4204                JSR     ERRSY      ;  No, Syntax Error
036036    6264                NEXTBYTE
036037  34215                 LDA     3,C342
036040  172655                SUBOR#  3,2,SNR    ;End of statement ?
036041    2246                JMP     @.EDST     ;  Yes
036042  34366                 LDA     3,.BUS     ;  No, restore NEXTBYTED byte
036043  15403                 DSZ     PBC.,3
036044    4414                JSR     INPS2      ;Advance retry pointer
036045     752                JMP     INP2A      ;And check for more options
036046  34225   INP2B:        LDA     3,C365     ;"@"
036047  172414                SEQ     3,2        ;IS FIELD A CURSOR POSITIONING ?
036050     414                JMP     INP2L      ;  No, Check for length option
036051    4722                JSR     IITAC      ;CHECK INPUT TERMINATOR
036052    6553                JSR     @.GCP      ;GENERATE CURSOR POSITION
036053     452                JMP     INP3B      ;PROCESS NEXT FIELD

036054  20343   INPSF:        LDA     0,IFLW     ;WE DON'T NEED A PROMPT
036055  101100                MOVL    0,0
036056  101240                MOVOR   0,0
036057  40343                 STA     0,IFLW
036060  30366   INPS2:        LDA     2,.BUS
036061  31003                 LDA     2,PBC.,2;Save pointer to current INPUT item
036062  50317                 STA     2,PBCS     ;  to allow retry
036063    1400                JMP     0,3        ;RETURN

036064  34176   INP2L:        LDA     3,C166
036065  156414                SEQ     2,3        ;LEN option?
036066     417                JMP     INP2T      ;  No, Check for other options
036067    4704                JSR     IITAC      ;  Yes, Prohibit Comma mode
036070    6263                INTEGER            ;Evaluate INPUT length specification
036071  44364                 STA     1,TSN11    ;SET INPUT LENGTH
036072  101014  INP2P:        SKZ     0,0        ;Negative length?
036073    6267                RUNERROR           ;  Yes, Parameter error
036074  116110                34*K+SE
036075  20222                 LDA     0,C360     ;  No
```

```
---

36076 162414        SEQ     3,0         ;Terminated by a Semi-colon?
36077    204        JMP     ERRSY       ;   No, Syntax error
36100   4760        JSR     INPS2       ;Advance retry pointer
36101    716        JMP     INP2A       ;And then check for more options
```

```
                ;                                  << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>
 36102  36221  .IEND: INEND
 36103  36014  .INP2: INP2
 36104  35773  .ITAC: IITAC

 36105  34492  INP2T: LDA     3,C52      ;CHECK FOR TIMEOUT OPT.
 36106 156114          SEQ     2,3
 36107    405          JMP     INP3       ;NOT TIM, CHECK OTHER
 36110   4653          JSR     IITAC      ;IS TIM, PROHIBIT COMING MODE
 36111   6263          INTEGER            ;GET LIMIT
 36112  44363          STA     1,TSN10    ;SET TIMEOUT
 36113    757          JMP     INP2P      ;AND CHK FOR ERRORS

 36114  34220  INP3:   LDA     3,C351     ;DOUBLE QUOTE
 36115 156114          SEQ     2,3        ;IS FIELD A PROMPT ?
 36116    413          JMP     INP4       ;   NO
 36117   4654          JSR     IITAC      ;CHECK INPUT TERMINATOR
 36120   6264          NEXTBYTE           ;OUTPUT PROMPT
 36121 151015          SNZ     2,2
 36122    403          JMP     INP3B
 36123   6265          QBYTE
 36124    774          JMP     .-4

 36125   4727  INP3B:  JSR     INPSF      ;SET FLAGS
 36126    671          JMP     INP2A      ;PROCESS NEXT FIELD

 36127     52  C52:    52      ;TIM TOKEN VALUE

 36130    254  C254:   254

 36131  34366  INP4:   LDA     3,.BUS
 36132  15403          DSZ     PBC.,3     ;RESTORE "NEXTBYTED" BYTE
 36133  20343          LDA     0,IFLW
 36134 101112          MOVL#   0,0,SZC    ;NEED A PROMPT ?
 36135    405          JMP     INPLI      ;   NO
 36136   6266          QTXT.
 36137 137640          .TXTF   "?
 36140      0  "

 36141   4713          JSR     INPSF      ;SET SOME FLAGS
 36142  20345  INPLI:  LDA     0,ITAC
 36143 101015          SNZ     0,0        ;INPUT TERMA A COMMA ?
 36144    412          JMP     INPL1      ;   YES
 36145   6270          XFLUSH             ;   NO, TOTALLY EMPTY XOB
 36146   6262          GOGO               ;START OUTPUT GOING BUT DON'T SWAP
 36147  30366          LDA     2,.BUS
```

```
                ;                                  << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>

 86150  20364          LDA     0,TSN11   ;Load length of fixed length INPUT
 86151 101015          SNZ     0,0       ;Is it set?
 86152  21057          LDA     0,FL3.,2  ;  No, Use Input control flag
 86153  41054          STA     0,INLEN,2 ;  Yes
 86154  24353          LDA     1,TSN10   ;LOAD TIMEOUT
 86155   6146          SPINPUT           ;Call Special Start Input
 86156   6256  INPL1:  JSR     @.VLOC    ;DO INPUT (FIRST FIND VARIABLE)
 86157    445          JMP     INPTS     ;  STRING
 86160  20343          LDA     0,IFLW    ;SET NOT END OF STATEMENT FLAG
 86161 101200          MOVR    0,0
 86162  24215          LDA     1,C342
 86163 136655          SUBOR#  1,3,SNR   ;End of statement?
 86164 101141          MOVOL   0,0,SKP   ;FALSE IF AT END
 86165 101120          MOVZL   0,0       ;ELSE TRUE
 86166  40343          STA     0,IFLW
 86167  20026          LDA     0,C6
 86170 126400          SUB     1,1
 86171   6120          DECIMAL           ;DECIMAL INPUT
 86172    460          JMP     INPNL     ;  ILLEGAL INPUT
 86173  20735          LDA     0,C254
 86174 112414          SEQ     0,2       ;IF TERMINATOR == <COMMA>
 86175 102021          ADCZ    0,0,SKP   ;     ITAC = TRUE
 86176 102400          SUB     0,0       ;ELSE
 86177  40345          STA     0,ITAC    ;     ITAC = FALSE
 86200  24055          LDA     1,C215
 86201 101014          SKZ     0,0
 86202 132415          SNE     1,2       ;LEGAL TERMINATOR ?
 86203    402          SKIP              ;  YES
 86204    446          JMP     INPNL     ;  NO
 86205 102400  INPST:  SUB     0,0       ;STORE THE INPUT VALUE
 86206  24310          LDA     1,TS7
 86207  30307          LDA     2,TS6
 86210   6120          DECIMAL
 86211  20345  INDON:  LDA     0,ITAC
 86212 101015          SNZ     0,0       ;Input terminator a Comma?
 86213    601          JMP     INP2      ;  Yes, Process remaining text in IOB
 86214  20343          LDA     0,IFLW    ;  No, Prompt for more text
 86215 101100          MOVL    0,0       ;Clear PromptNotNeeded flag
 86216 101220          MOVZR   0,0
 86217  40343          STA     0,IFLW
 86220   2643          JMP     @.INP2    ;And input next variable
```

```
                ;                                  << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>
 36221   16663 INEND: JSR     @.ITAC    ;End of INPUT - Error if still in Comma Mode
 36222    2246         JMP     @.EDST    ;All's fine, Execute next statement

 36223   37073         MOST
 36224  145000 INPTS: MOV     2,1
 36225  161005         MOV     3,0,SNR
 36226    6145         XPUTBYTE
 36227  102400         SUB     0,0       ;SOURCE IS IOB
 36230  126000         ADC     1,1       ;NO MAX SOURCE SIZE
 36231   30055         LDA     2,C215    ;ALTERNATE TERMINATOR
 36232   34005         LDA     3,RUP
 36233   35415         LDA     3,ABN.,3
 36234  177100         ADDL    3,3
 36235  175112         SSP     3,3       ;Transparent input mode active ?
 36236  131000         MOV     1,2       ;   Yes - Use "no terminator" code (177777)
 36237    6764         JSR     @INPTS-1  ;MOVESTRING USING ALTERNATE TERMINATOR
 36240    6264         NEXTBYTE
 36241   34215         LDA     3,C342
 36242  172655         SUBOR#  3,2,SNR   ;End of statement?
 36243    2246         JMP     @.EDST    ;   YES
 36244   34293         LDA     3,C361    ;   NO
 36245  172414         SEQ     3,2       ;COMMA ?
 36246    4204         JSR     ERRSY     ;   NO
 36247  102000         ADC     0,0
 36250   40345         STA     0,ITAC    ;ITAC = FALSE
 36251     740         JMP     INDON

 36252   34366 INPNL: LDA     3,.BUS    ;NO LEGAL INPUT YET
 36253   21452         LDA     0,ERBP.,3
 36254  101015         SNZ     0,0       ;Error branching in effect?
 36255     406         JMP     INPBT     ;   No, give error beep
 36256   20055         LDA     0,C215    ;   Yes
 36257  112415         SNE     0,2       ;Null input (carraige return)?
 36260     725         JMP     INPST     ;   Yes, Accept as zero
 36261    6267         RUNERROR          ;   NO, GIVE AN ERROR
 36262  161110         142*K+SE
 36263   34366 INPBT: LDA     3,.BUS
 36264   20317         LDA     0,PBCS
 36265   41403         STA     0,PBC.,3  ;RESTORE POINTER TO PROGRAM
 36266  102000         ADC     0,0
 36267   40315         STA     0,ITAC    ;Reset Comma mode state
 36270    6266         QTXT.
 36271  103734         .TXTF   "<7>\
 36272  137640 ?
 36273       0 "

 36274     646         JMP     INPLI
```

```
                        ;                                << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>
                        ;TEMP CELLS FOR OUTPUT STREAM ROUTINES:

                    340 T1=    TSP
                    342 T2=    TSP2
                    344 T3=    TSP4
                    346 T4=    TSP6
                    350 ATF=   TSP10
                    351 QBRTN=TSP11
                    352 WRTN=  TSP12

 026275   54344 GCP:   STA     3,T3        ;GENERATE CURSOR POSITION
 026276   30052         LDA     2,C17/
 026277    6265         QBYTE
 026300    6263 GCP1:  INTEGER
 026301   30277         LDA     2,C375
 026302  101015         SNZ     0,0         ;POSITION EXPRESSION < 0 . . .
 026303  132033         SLS     1,2         ;OR > 375 ?
 026304    6267         RUNERROR            ;   YES, OUT OF RANGE
 026305  107510         17*K+SE
 026306  131400         INC     1,2         ;INCREMENT (TO PREVENT ZERO)
 026307    6265         QBYTE
 026310   24273         LDA     1,C361
 026311   34362         LDA     3,TSN7      ;RETRIEVE TERMINATOR
 026312  136415         SNE     1,3         ;TERMINATOR A "," ?
 026313     765         JMP     GCP1        ;   YES
 026314   24272         LDA     1,C360
 026315  136414         SEQ     1,3         ;TERMINATOR A ";" ?
 026316    4204         JSR     ERRSY       ;   NO
 026317   30064         LDA     2,C37/      ;   YES, OUTPUT THE TERMINATOR
 026320    6265         QBYTE
 026321    2344         JMP     @T3         ;RETURN


 026322   50340 QBYT.:STA     2,T1        ;QUEUE BYTE
 026323   54351         STA     3,QBRTN     ;SAVE RETURN ADR
 026324   30366         LDA     2,.BUS
 026325   21023         LDA     0,ROBP.,2
 026326   24273         LDA     1,RLBA
 026327  106432         SGR     0,1         ;XOB FULL ?
 026330     415         JMP     QB2         ;   NO
 026331    6410 QB1:   JSR     @.GGOQ      ;   YES, TRY TO MAKE ROOM (don't STOUT unless IOB full)
 026332   30366         LDA     2,.BUS
 026333   21023         LDA     0,ROBP.,2
 026334   24273         LDA     1,RLBA
 026335  106432         SGR     0,1         ;SUCCEEDED ?
 026336     407         JMP     QB2         ;   YES
 026337    6403         JSR     @.WAIT      ;   NO
 026340     771         JMP     QB1         ;TRY AGAIN

 026341   36540 .GGOQ:GOGOQ
 026342   36540 .WAIT:WAIT
 026343      45 CECOD:45                     ;'CE' = Clear to End of Line
 026344      13 RNDGC:13                     ;TERMINAL TYPE CODE FOR DGC 6052 & 6053
```

```
                ;                                << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>
                ;THIS SECTION OF CODE IS TO MAINTAIN THE OUTPUT COLUMN COUNTER (OCC)
                ;TO ASSURE PROPER TABBING

 36345  30005 QB2:   LDA     2,RUP
 36346  20340         LDA     0,T1        ;RETRIEVE CHAR
 36347  24350         LDA     1,ATF
 36350 125014         SKZ     1,1         ;DOING A CURSOR POSITION ?
 36351    446         JMP     QPAT        ;   YES, GO PROCESS
 36352  24056         LDA     1,C240
 36353 106032         SGE     0,1         ;PRINTING CHAR ?
 36354    404         JMP     QSP         ;   NO, TREAT SPECIAL
 36355  11033         ISZ     OCC.,2      ;   YES, BUMP OCC
 36356 100010         NOP                 ;PREVENT ISZ FROM SKIPPING
 36357    455         JMP     QST

 36360  24052 QSP:    LDA     1,C177
 36361 106415         SNE     0,1         ;"@" ?
 36362    446         JMP     QPT1        ;   YES, SET FLAG
 36363 123400         AND     1,0         ;MASK TO 7 BITS
 36364  24757         LDA     1,CECOD     ;'CE' CODE
 36365 106414         SEQ     0,1         ;DOING 'CE' ?
 36366    410         JMP     QSPO        ;   NO
 36367  35035         LDA     3,RDE.,2    ;   YES
 36370  24754         LDA     1,RNDCC     ;DGC TERMINAL TYPE CODE
 36371 136414         SEQ     1,3         ;DCC TERMINAL ?
 36372    404         JMP     QSPO        ;   NO
 36373  35017         LDA     3,PCX.,2    ;   YES
 36374  25033         LDA     1,OCC.,2
 36375  45772         STA     1,PCXMC,3;SAVE CURRENT OCC FOR DGC TERMINAL DRIVER
 36376  24034 QSPO:   LDA     1,C14
 36377  34024         LDA     3,C4
 36400 106432         SUBO    0,1,SZC     ;FF,CR,MH OR CS ?
 36401 137020         ADDZ    1,3
 36402 126462         SUBC    1,1,SZC
 36403  45033         STA     1,OCC.,2    ;   YES, ZERO OCC
 36404  24030         LDA     1,C10
 36405 106414         SEQ     0,1         ;ML OR CTRL H ?
 36406    405         JMP     QSP1        ;   NO
 36407  35033         LDA     3,OCC.,2
 36410 174404         NEG     3,3,SZR
 36411 174000         COM     3,3
 36412  55033         STA     3,OCC.,2    ;   YES, DECREMENT OCC IF > 0
 36413  24014 QSP1:   LDA     1,C40
 36414 106415         SNE     0,1         ;MR ?
 36415  11033         ISZ     OCC.,2      ;   YES, BUMP OCC
 36416    416         JMP     QST

 36417  24064 QPAT:   LDA     1,C377
 36420 106415         SNE     0,1         ;DONE WITH CURSOR POSITIONING ?
 36421    411         JMP     QDAT        ;   YES
 36422  24350         LDA     1,ATF
 36423 125204         MOVZR#  1,1,SZR     ;ON FIRST CHAR PAST "@" ?
 36424    404         JMP     QPT1        ;   NO
 36425 100400         NEG     0,0         ;   YES, UPDATE OCC.
 36426 100000         COM     0,0
 36427  41033         STA     0,OCC.,2
 36430  10350 QPT1:   ISZ     ATF
 36431    403         JMP     QST
```

```
                        ;                                  << SI = R92RUNS3E; BO = 18/A.RUN.8053! >>
 36432 126400 QDAT:  SUB     1,1        ;DONE WITH "@"
 36433  44350       STA     1,ATF      ;CLEAR "@" FLAG
 36434  20340 QST:   LDA     0,T1
 36435  30366       LDA     2,.BUS
 36436  25023       LDA     1,ROBP.,2
 36437  11023       ISZ     ROBP.,2    ;INCREMENT XOB POINTER
 36440   6145       XPUTBYTE           ;STORE THE BYTE
 36441   2351       JMP     @QBRTN

                        ;       Queue Inline String (from processor area)
 36442 175120 QT..:  MOVZL   3,3        ;FORM BYTE POINTER
 36443  54316       STA     3,T4       ;SAVE IT
                        ;P-N CODE
 36444  34201       LDA     3,.EL
 36445  25400       LDA     1,0,3
 36446 124005       COM     1,1,SNR    ;PNT7 still there ?
 36447   1401 PNT7T: JMP     1,3        ;   Yes, go there
                        ;END OF P-N CODE
 36450  24316 QT.1:  LDA     1,T4       ;   NO
 36451  10316       ISZ     T4         ;Bump byte pointer
 36452   6124       GETBYTE            ;GET BYTE FROM SOURCE STRING
 36453 151015       SNZ     2,2        ;ZERO (TERMINATOR) ?
 36454    403       JMP     QT.2       ;   YES
 36455   6265       QBYTE              ;NO, QUEUE IT
 36456    7/2       JMP     QT.1       ;PROCESS NEXT BYTE

 36457  34316 QT.2:  LDA     3,T4       ;DONE
 36460 175620       INCZR   3,3        ;FORM RETURN ADDRESS
 36461   1400       JMP     0,3        ;RETURN

 36462  54340 FLUS.:STA     3,T1       ;FLUSH, EMPTY OUTPUT STREAM
 36463   6270       XFLUSH             ;FIRST EMPTY XOB
 36464   6262       GOGO               ;Start IOB output
 36465   4553       JSR     WAIT       ;AND WAIT FOR OUTPUT TO FINISH
 36466   2340       JMP     @T1        ;RETURN
```

```
                ;                                << SI = R92RUNS3E;  BO = 18/A.RUN.8053! >>

 36467  54342 XFLU.: STA     3,T2        ;XFLUSH, EMPTY XOB
 36470  30366        LDA     2,.BUS
 36471  21023        LDA     0,ROBP.,2
 36472  24272        LDA     1,RFBA
 36473 106132        SGR     0,1         ;XOB ALREADY EMPTY ?
 36474   2342        JMP     @T2         ;  YES, RETURN
 36475  21075        LDA     0,TSU.2,2
 36476 101015        SNZ     0,0         ;Output to Channel?
 36477    515        JMP     CHAPR       ;  Yes, Enter Channel Print
                                         ;  No, Output to user terminal

 36500  34005        LDA     3,RUP
 36501  21412        LDA     0,FLW.,3
 36502 101300        MOVS    0,0
 36503 101112        SSP     0,0         ;OUTPUT IN PROGRESS ?
 36504   4504        JSR     WAIT        ;  YES, WAIT 'TILL IT'S DONE
 36505  20272        LDA     0,RFBA
 36506  40344        STA     0,T3
 36507  30366 XFL1:  LDA     2,.BUS
 36510  20344        LDA     0,T3
 36511  25023        LDA     1,ROBP.,2
 36512 106032        SGE     0,1         ;TRANSFERRED WHOLE XOB ?
 36513    404        JMP     XFL1A       ;  NO
 36514  20272        LDA     0,RFBA      ;  YES, RESET ROBP
 36515  41023        STA     0,ROBP.,2
 36516   2342        JMP     @T2         ;RETURN

 36517  34005 XFL1A: LDA     3,RUP
 36520  21405        LDA     0,OBP.,3
 36521  24004        LDA     1,C14       ;ALLOW FOR 14 CHAR TRANSLATION SEQUENCE
 36522 123000        ADD     1,0
 36523  25403        LDA     1,LBA.,3
 36524 106132        SGR     0,1         ;IOB ALMOST FULL ?
 36525    403        JMP     XFL2        ;  NO
 36526   6141        STOUTPUT            ;  YES, EMPTY IT
 36527   4511        JSR     WAIT
 36530  24344 XFL2:  LDA     1,T3
 36531   6144        XGETBYTE            ;TRANSFER A BYTE
 36532   6132        OUTBYTE             ;FROM XOB TO IOB
 36533  10314        ISZ     T3
 36534    753        JMP     XFL1        ;PROCESS NEXT BYTE
```

```
                ;                                  << SI = R92RUNS3E;  BO = 18/A.RUN.8053! >>
 36535       0 GIBP:  0          ;GO'S IBP
 36536       0 GOT3:  0          ;RETURN ADDR.
 36537       0 GFLAG: 0          ;If -1 don't STOUTPUT unless IOB is full

 36540  102001 GOGOQ: ADC    0,0,SKP   ;Don't STOUTPUT unless IOB if full
 36541  102400 GOGO.: SUB    0,0       ;STOUTPUT even if IOB isn't full
 36542   40775        STA    0,GFLAG   ;Set GFLAG to control STOUTPUT usage
 36543   30366        LDA    2,.BUS    ;GET OUTPUT GOING
 36544   21023        LDA    0,ROBP.,2
 36545   24272        LDA    1,RFBA
 36546  106432        SGR    0,1       ;XOB NONEMPTY . . .
 36547     404        JMP    GO1       ;   (NO)
 36550   21075        LDA    0,TSU.2,2
 36551  101015        SNZ    0,0       ;. . . And a channel print?
 36552     472        JMP    CHAPR     ;   Yes, Print to channel (can BUMP if locked)

 36553   54763 GO1:   STA    3,GOT3    ;SAVE RETURN ADR
 36554   34005        LDA    3,RUP
 36555   21412        LDA    0,FLW.,3
 36556  101300        MOVS   0,0
 36557  101112        SSP    0,0       ;OUTPUT IN PROGRESS ?
 36560    2756        JMP    @GOT3     ;   YES, RETURN
 36561   20272        LDA    0,RFBA    ;   NO, SET GO'S IBP
 36562   40753        STA    0,GIBP
 36563   30366 GO2:   LDA    2,.BUS
 36564   20751        LDA    0,GIBP    ;MOVE SOME XOB TO IOB UNTIL IOB FULL
 36565   25023        LDA    1,ROBP.,2
 36566  106003        SLS    0,1       ;XOB STILL got stuff in it ?
 36567     436        JMP    GO5       ;   (NO)
 36570  101212        SKE    0,0       ;Even XOB BA
 36571     410        JMP    GO3       ;No
 36572   34005        LDA    3,RUP
 36573   21405        LDA    0,OBP.,3
 36574   24034        LDA    1,C14     ;ALLOW FOR 14 CHAR TRANSLATION SEQUENCE
 36575  123000        ADD    1,0
 36576   25403        LDA    1,LBA.,3
 36577  106003        SLS    0,1       ;. . . AND ROOM FOR IT IN IOB ?
 36600     406        JMP    GO4       ;   NO
 36601   24734 GO3:   LDA    1,GIBP
 36602    6144        XGETBYTE         ;MOVE A BYTE
 36603    6132        OUTBYTE          ;FROM XOB TO IOB
 36604   10731        ISZ    GIBP
 36605     756        JMP    GO2       ;PROCESS NEXT BYTE
```

```
                        ;                                  << SI = R92RUNS3E;  BO = 18/A.RUN.8053! >>
 36606   25023 GO4:     LDA     1,ROBP.,2;Partition BA(End of data in XOB)
 36607   20726          LDA     0,GIBP    ;Partition BA(Next byte out of XOB)
 36610  106400          SUB     0,1       ;# bytes left
 36611   34272          LDA     3,RFBA    ;Partition BA(XOB)
 36612  137000          ADD     1,3       ;Partition BA(New end of XOB) (After move)
 36613   55023          STA     3,ROBP.,2
 36614  101220          MOVZR   0,0       ;Partition A(Start of data left in XOB)
 36615  125220          MOVZR   1,1       ;# words to move
 36616   35061          LDA     3,XOB.,2  ;Partition A(XOB)
 36617  143000          ADD     2,0       ;A(Start of source of move)
 36620  107000          ADD     0,1       ;A(End of source of move)
 36621  173000          ADD     3,2       ;A(Destination start)
 36622    6101          CALL              ;Shuffle remaining XOB data forward
 36623  100015          MOVEWORDS         ;  to front of XOB
 36624     412          JMP     GO6       ;Go flush IOB

 36625   20272 GO5:     LDA     0,RFBA    ;Partition BA(XOB)
 36626   41023          STA     0,ROBP.,2;Reset XOB input pointer
 36627   34005          LDA     3,RUP
 36630   21405          LDA     0,OBP.,3
 36631   25402          LDA     1,FBA.,3
 36632   30705          LDA     2,GFLAG
 36633  151015          SNZ     2,2       ;Inhibit STOUTPUT if flag set
 36634  106432          SGR     0,1       ;IOB EMPTY ?
 36635    2701          JMP     @GOT3     ;   YES, DONE
 36636    6141 GO6:     STOUTPUT          ;   NO, GET OUTPUT GOING (!)
 36637    2677          JMP     @GOT3


                        ; Routine to centralize WONA calls.

 36640   54352 WAIT:    STA     3,WRTN
 36641    6101          CALL
 36642  100011          WONA
 36643    2352          JMP     @WRTN     ;RETURN


                        ;   CHAPR   -   Output XOB to channel; Entered via jump from XFLUSH
                        ;               and GOGO.

 36644   54341 CHAPR:   STA     3,TSP1    ;Save return address
 36645   30272          LDA     2,RFBA
 36646   34366          LDA     3,.BUS
 36647   25423          LDA     1,ROBP.,3;Load XOB start and current end addresses
 36650   51423          STA     2,ROBP.,3;Reset ROBP to make XOB empty on errors
 36651   20031          LDA     0,C11     ;Specify string output type
 36652  146400          SUB     2,1       ;Bufferlen:= CurretBfrEnd - BfrStart
 36653    6511          JSR     @.RWIC    ;Output to channel
 36654    2341          JMP     @TSP1     ;And return
```

```
                ;                                  << SI = R92RUNS3E;  BO = 18/A.RUN.8053! >>

 36655   6264 MAT:   NEXTBYTE                ;"MAT" STATEMENT
 36656  34216        LDA     3,C344
 36657 156005        ADC     2,3,SNR
 36660    775        JMP     MAT
 36661  20212        LDA     0,C136
 36662 142415        SNE     2,0             ;MAT WRITE ?
 36663    406        JMP     MAT1            ;   YES
 36664 142005        ADC     2,0,SNR         ;MAT READ ?
 36665   6236        JSR     @.EVCH          ;   YES, EVALUATE CHANNEL ADDRESS
 36666    454        JMP     MAT2            ;   NO, OR NO "#" GIVEN
 36667    405        JMP     MAT13

 36670  20034        WRITITEM
 36671  20777 MAT1:  LDA     0,.-1           ;MAT WRITE
 36672   6236        JSR     @.EVCH          ;EVALUATE CHANNEL ADDRESS
 36673   4204        JSR     ERRSY           ;   NO "#"
 36674   6264 MAT13: NEXTBYTE                ;<< ENTRY FROM ABOVE
 36675  20053        LDA     0,C200          ;VARIABLE NUMBER ?
 36676  24213        LDA     1,C336
 36677 146032        SGE     2,1
 36700 142032        SGE     2,0
 36701   4204        JSR     ERRSY           ;   NO
 36702   6251        JSR     @.LUVD          ;   YES
 36703 151015        SNZ     2,2             ;LOCATION ASSIGNED ?
 36704   6267        RUNERROR                ;   NO
 36705 114510        31*K+SE
 36706 105100        MOVL    0,1             ;   YES
 36707  34366        LDA     3,.BUS
 36710 173000        ADD     3,2
 36711 125112        SSP     1,1             ;STRING VARIABLE ?
 36712 151400        INC     2,2             ;   YES
 36713  50357        STA     2,TSN4
 36714  21376        LDA     0,-2,2
 36715  31377        LDA     2,-1,2
 36716 125102        MOVL    1,1,SZC         ;STRING VARIABLE ?
 36717    436        JMP     MATS            ;   YES
 36720 101400        INC     0,0             ;   NO
 36721 151400        INC     2,2
 36722 125213        MOVR#   1,1,SNC         ;ARRAY VARIABLE ?
 36723 102521        SUBZL   0,0,SKP         ;   NO, ONE ELEMENT
 36724   6116        BINMULTIPLY             ;   YES, COMPUTE NUMBER OF ELEMENTS
 36725  30414        LDA     2,MAT2-1
 36726 133700        ANDS    1,2             ;NUMBER TYPE
 36727 151400        INC     2,2
 36730   6116        BINMULTIPLY
 36731 105000        MOV     0,1             ;NUMBER OF WORDS
 36732  20032 MATRW: LDA     0,C12           ;BINARY MODE
 36733  30357        LDA     2,TSN4          ;A(MATRIX)
 36734   6130        JSR     @.RWIC
 36735   6264        NEXTBYTE
 36736  20223        LDA     0,C361
 36737   2401        JMP     @.+1
 36740  33304        RWIB2
```

```
                ;                               << SI = R92RUNS3E;  BO = 18/A.RUN.8053! >>
 36741    1400          1400
 36742    6270  MAT2:   XFLUSH                  ;CLEAN OUT XOB
 36743    6262          GOGO                    ;GET OUTPUT GOING
 36744   34005          LDA     3,RUP           ;NOT MAT READ# OR MAT WRITE#
 36745   31425          LDA     2,DFT.,3
 36746   20202          LDA     0,D.MAT
 36747  101015          SNZ     0,0
 36750    6267          RUNERROR                ;"RUNMAT" NOT LOADED
 36751  135610          73*K+AE
 36752   41371          STA     0,FDA+CHM1,2
 36753   34113          LDA     3,RNMRT         ;BUMP TO RUNMAT
 36754    2117          JMP     @BUMPUSER&377

 36755   34357  MATS:   LDA     3,TSN1          ;MAT READ or MAT WRITE a string variable
 36756   20041          LDA     0,DBA
 36757  116540          SUBOL   0,3             ;Calculate relative byte address of string
 36760  157000          ADD     2,3             ;Calculate byte address of the end of the string
 36761   54363          STA     3,TSN10         ;Set address for read/write rtn to store terminator
 36762  145620          INCZR   2,1             ;WordsToTransfer:= (SizeInBytes+1)/2 (roundup)
 36763     747          JMP     MATRW           ;And now read or write the string

 36764   33341 .RWIC:  RWIC

 36765     143 C143:   143

 36766  100374 RNMRT:  @374

 36767  177763 VFLDP:  177763 ;MASK TO CLEAR DATA PRECISION IN FLAG.

                ;RE-ENTRY POINT FROM RUNMAT TO FINISH MAT READ FROM DATA STMTS

 36770   30366 FINMRD:          LDA     2,.BUS;MAT READ SUCCESSFUL, UPDATE DATA PTRS
 36771   21073          LDA     0,TSU.,2
 36772   41020          STA     0,DWC.,2
 36773   21074          LDA     0,TSU.1,2
 36774   41021          STA     0,DEC.,2
 36775   21076          LDA     0,TSU.3,2
 36776   41022          STA     0,DSC.,2
 36777   21101          LDA     0,TSU.6,2;UPDATE DATA NUMBER TYPE (n%)
 37000  103120          ADDZL   0,0             ;MOVE CURRENT NUMBER TYPE INTO FLAG. POS
 37001   35024          LDA     3,FLAG.,2;LOAD FLAG VALUE
 37002   24765          LDA     1,VFLDP
 37003  137400          AND     1,3             ;CLEAR DATA NUMBER TYPE
 37004  117000          ADD     0,3             ;UPDATE
 37005   55024          STA     3,FLAG.,2
 37006    2374          JMP     @.NXST          ;EXECUTE NEXT STMT
```

```
                ;                                << SI = R92RUNS3E;  BO = 18/A.RUN.8053! >>

37007  32203 .CTCE:CTCEN

37010  20073 CESF:  LDA     0,ESCF     ;CHECK ESCAPE FLAG
37011 101015        SNZ     0,0        ;Been set ?
37012   1400        JMP     0,3        ;   No
37013 176400        SUB     3,3        ;   Yes, Escape or Control-C
37014  54073        STA     3,ESCF     ;Reset flag
37015  34005        LDA     3,RUP
37016  25415        LDA     1,ABN.,3;Load abort status
37017  34770        LDA     3,.CTCE    ;Load address of cntl-C entry ptr
37020 125213        SKO     1,1        ;Was it a Control-C?
37021   3400        JMP     @0,3       ;   Yes, Treat as Control-C
37022    402        JMP     ESCPB      ;Enter Escape code skipping load of ESCF

37023  20073 ESCP:  LDA     0,ESCF     ;ESCAPE KEY PRESSED
37024 176400 ESCPB:SUB     3,3        ;Clear escape flag
37025  54073        STA     3,ESCF
37026  30366        LDA     2,.BUS
37027  25052        LDA     1,ERBP.,2
37030 125015        SNZ     1,1        ;Error branch set?
37031    434        JMP     ESTOP      ;   No, STOP
37032  30366        LDA     2,.BUS     ;YES
37033  35002        LDA     3,PLC.,2
37034  21400        LDA     0,0,3      ;Current line offset into SLT
37035 157000        ADD     2,3
37036  21400        LDA     0,0,3      ;Current line number
37037 101112        SSP     0,0        ;Protected ?
37040    406        JMP     ESCP1      ;   YES
37041  30005        LDA     2,RUP      ;   NO
37042   6102        FLAGCHECK          ;CTRL Y LAST CHAR IN ?
37043  10012        FLW.+SKIPZ
37044    400        400
37045    420        JMP     ESTOP      ;   YES, STOP !!

37046  20717 ESCP1:LDA     0,C143     ;< < Entry from CTLC in IF ERR 0 mode
37047  30366 ESCP2:LDA     2,.BUS     ;< < Entry from CTLC in IF ERR 1 mode with AC0 = 199
37050 126400        SUB     1,1
37051  45057        STA     1,FL3.,2 ;Reset INPUT length
37052  25052        LDA     1,ERBP.,2;PICK UP B(ERROR LINE #)
37053  45003        STA     1,PBC.,2 ;SET AS NEXT LINE
37054  25063        LDA     1,STPBC,2
37055  44355        STA     1,ERPBC  ;Mark the start of stmt escaped from
37056  41051        STA     0,ERRN.,2;   Set Error# to 99 or 199
37057  35002        LDA     3,PLC.,2;LOAD SLT OFFSET FOR LINE
37060 157000        ADD     2,3        ;MAKE ABS ADDR
37061  21400        LDA     0,0,3      ;LOAD LINE #
37062  41053        STA     0,ERLN.,2;SET ERROR LINE #
37063   2401        JMP     @.+1       ;EXECUTE ERROR LINE
37064  33627        EXST1

37065  30366 ESTOP:LDA     2,.BUS     ;STOP after escape or control-c
37066  20272        LDA     0,RFBA
37067  41023        STA     0,ROBP.,2;Abort XOB
37070   2401        JMP     @.+1       ;And then STOP
37071  33746        STOP

                .EOT   ; "RUN"  SOURCE #3
```

```
                ;                              << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>
                ; "RUN" SOURCE #4 OF 7 FOR "IRIS" R9.0

                 ; "MOST" -- MOVE STRING IN USER AREA
                 ;
                 ;ON ENTRY TO "MOST":
                 ; A0 = FIRST SOURCE BYTE ADDRESS (RELATIVE)
                 ; A1 = MAXIMUM # SOURCE BYTES
                 ; A2 = ALTERNATE TERMINATOR CODE
                 ; A3 = RETURN ADDRESS
                 ; TS6 = FIRST DEST BYTE ADDRESS (RELATIVE)
                 ; TS7 = DESTINATION SIZE (# BYTES)
                 ;
                 ;IF ZERO IS GIVEN AS THE SOURCE BYTE ADDRESS IN A0,
                 ;THEN THE SOURCE IS ASSUMED TO BE THE INPUT BUFFER.
                 ;
                 ;MOVES BYTES FROM SOURCE AREA TO DESTINATION AREA.
                 ;IF A ZERO BYTE IS OVERLAYED IN THE DEST THEN A
                 ;ZERO BYTE IS STORED AFTER THE LAST BYTE MOVED.
                 ;
                 ;IF THE SOURCE IS EXHAUSTED BEFORE THE DEST IS
                 ;FILLED, AND NO ZERO BYTE WAS OVERLAYED, THEN THE
                 ;DEST AREA IS COPIED BACK TO CLOSE THE GAP.

 37072  34712 .TBYT:TBYTE

 37073  54311 MOST: STA     3,TSX      ;MOVE STRING
 37074  54315       STA     3,TSX4     ;ASSUME NO COMMA
 37075  40316       STA     0,TSX5     ;SOURCE BYTE ADDRESS
 37076  40317       STA     0,TSX6     ;CLR TERM FND FLG
 37077  44301       STA     1,TS       ;MAX # SOURCE BYTES
 37100  50313       STA     2,TSX2     ;ALTERNATE TERMINATING CODE
 37101  54304       STA     3,TS3      ;ASSUME NO ZERO OVERLAYED IN DEST
 37102  40306       STA     0,TS5      ;SET "NOT COPYING DEST" FLAG
 37103 102400       SUB     0,0
 37104  40320       STA     0,TSX7
 37105  32765       LDA     2,@.TBYT
 37106  34293       LDA     3,C361
 37107 156404       SUB     2,3,SZR    ;COMMA SEEN ?
 37110    402       JMP     MOST1      ;   NO
 37111  54315       STA     3,TSX4     ;   YES

 37112  24307 MOST1:LDA     1,TS6      ;GET BYTE OF DESTINATION
 37113   6144       XGETBYTE
 37114 151015       SNZ     2,2        ;OVERLAYING ZERO BYTE IN DEST ?
 37115  50304 MOST2:STA     2,TS3      ;   YES, FLAG TO PUT ZERO BYTE AT END
 37116  24316       LDA     1,TSX5     ;FETCH SOURCE BYTE ADDRESS
 37117 125014       SKZ     1,1        ;A REAL ADDRESS GIVEN ?
 37120    406       JMP     MOSTA
 37121 102000       ADC     0,0        ;   NO
 37122   6126       INSTBYTE           ;GET AN INPUT BYTE
 37123 101014       SKZ     0,0        ;End of input ?
 37124    407       JMP     MOSTB      ;   No
 37125    451       JMP     MOST6      ;   Yes
```

```
                        ;                               << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>

037126  10316 MOSTA: ISZ     TSX5        ;   YES, INC SOURCE BYTE PTR
037127  30306        LDA     2,TS5
037130 151014        SKZ     2,2         ;COPYING BACK TO DEST ?
037131  10317        ISZ     TSX6        ;   NO
037132   6144        XGETBYTE            ;AND GET A SOURCE BYTE
037133  34313 MOSTB: LDA     3,TSX2
037134 156415        SNE     2,3         ;ALTERNATE TERMINATOR ?
037135    441        JMP     MOST6       ;   YES
037136 141000        MOV     2,0         ;   NO
037137  24307        LDA     1,TS6       ;STORE BYTE IN DESTINATION
037140  10307        ISZ     TS6         ;INC DESTINATION PTR
037141   6145        XPUTBYTE
037142  14310        DSZ     TS7         ;DESTINATION FILLED ?
037143    402        SKIP
037144    425        JMP     MOST5       ;   YES
037145  24306        LDA     1,TS5       ;   NO
037146 125014        SKZ     1,1         ;COPYING BACK DESTINATION . . .
037147  14301        DSZ     TS          ;OR MORE SOURCE TO BE COPIED ?
037150    742        JMP     MOST1       ;   YES
037151  34304 MOST3: LDA     3,TS3       ;   NO, END OF SOURCE
037152 175015        SNZ     3,3         ;WAS A ZERO BYTE OVERLAYED ?
037153    416        JMP     MOST5       ;   YES
037154  34313        LDA     3,TSX2      ;   NO
037155  20315        LDA     0,TSX4
037156 163015        ADD#    3,0,SNR     ;TERMINATOR = 0 AND COMMA SEEN ?
037157   2311        JMP     @TSX        ;   YES, RETURN
037160  20307        LDA     0,TS6       ;   NO
037161  24310        LDA     1,TS7
037162 107000        ADD     0,1         ;B(END OF DESTINATION)
037163  44316        STA     1,TSX5      ;MAKE IT SOURCE ADR
037164 152400        SUB     2,2
037165  50310        STA     2,TS7       ;DEST SIZE -- TERMINATE ON TERMINATOR
037166  50306        STA     2,TS5       ;COPY BACK DEST FLAG
037167  50313        STA     2,TSX2      ;TERMINATOR = 0
037170    725        JMP     MOST2       ;COPY BACK DESTINATION

037171  20304 MOST5: LDA     0,TS3       ;DESTINATION FILLED
037172  24307        LDA     1,TS6
037173 101015        SNZ     0,0         ;WAS A ZERO BYTE OVERLAYED ?
037174   6145        XPUTBYTE            ;   YES, STORE ZERO BYTE AT END
037175   2311        JMP     @TSX

037176 102000 MOST6: ADC     0,0
037177  34306        LDA     3,TS5
037200 175014        SKZ     3,3         ;Closing destination string?
037201  40320        STA     0,TSX7      ;   No, SET ALT TERM FOUND FLG
037202    747        JMP  MOST3
```

```
                ;                                << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>
37203    6533 FOR:  JSR     @.FNXT    ;"FOR" STATEMENT
37204    4206       JSR     ERRST
37205  162414       SEQ     3,0       ;"=" ?
37206    4204       JSR     ERRSY     ;  NO
37207   34366       LDA     3,.BUS    ;  YES
37210  172400       SUB     3,2       ;INDEX VARIABLE DISPLACEMENT
37211   25414       LDA     1,FNS.,3
37212  137000       ADD     1,3
37213   20577       LDA     0,FNC     ;SCAN FOR-NEXT STACK
37214  162033 FOR1: SLS     3,0       ;END OF STACK ?
37215     417       JMP     FORN      ;  YES, VARIABLE NOT IN STACK
37216   25400       LDA     1,FIVRD,3;  NO
37217  146415       SNE     2,1       ;IS THIS THE INDEX VARIABLE ?
37220     404       JMP     FOR2      ;  Yes
37221   24516       LDA     1,FSESZ   ;  NO
37222  137000       ADD     1,3
37223     771       JMP     FOR1

37224   24513 FOR2: LDA     1,FSESZ   ;  YES
37225  122400       SUB     1,0
37226   40564       STA     0,FNC     ;REMOVE IT FROM THE STACK
37227   25414 FOR3: LDA     1,FESIZ,3
37230   45400       STA     1,FIVRD,3
37231  175400       INC     3,3
37232  162402       SGR     3,0       ;END OF STACK ?
37233     774       JMP     FOR3      ;  NO
37234   34366 FORN: LDA     3,.BUS    ;  YES
37235   25416       LDA     1,CSS.,3
37236  167000       ADD     3,1
37237   34553       LDA     3,FNC
37240  166033       SLS     3,1       ;IS THE STACK FULL ?
37241    6267       RUNERROR          ;  YES, NESTED TOO DEEP
37242  111210       22*K+AE
```

```
                ;                                << SI = R92RUNS4B;  BO = 18/A.RUN.8053! >>
037243  51400          STA     2,FIVRD,3;STACK INDEX VARIABLE DISPLACEMENT
037244  30366          LDA     2,.BUS
037245  20172          LDA     0,FSESZ
037246 163000          ADD     3,0
037247 142400          SUB     2,0
037250  41015          STA     0,FSC.,2 ;UPDATE STACK COUNTER
037251   6257          ALGEXPRESSION    ;EVALUATE INITIAL VALUE
037252  20216          LDA     0,C344
037253 162404          SUB     3,0,SZR  ;"TO" SEPARATOR ?
037254   4204          JSR     ERRSY    ;   NO
037255  24310          LDA     1,TS7    ;   YES
037256  44311          STA     1,TSX
037257  30307          LDA     2,TS6
037260   6120          DECIMAL          ;STORE INITIAL VALUE
037261   6257          ALGEXPRESSION    ;EVALUATE LIMIT VALUE
037262  54301          STA     3,TS
037263  24511          LDA     1,FDLIM
037264   4530          JSR     FORST    ;STORE LIMIT VALUE
037265  34301          LDA     3,TS
037266  30217          LDA     2,C347
037267 172015          ADC#    3,2,SNR  ;STEP VALUE GIVEN  ?
037270    405          JMP     FOR5
037271  20036          LDA     0,C16
037272   6120          DECIMAL          ;   NO, STEP=1
037273  34301          LDA     3,TS
037274    402          SKIP
037275   6257 FOR5:     ALGEXPRESSION    ;   YES, EVALUATE IT
037276  30215          LDA     2,C342
037277 156654          SUBOR# 2,3,SZR   ;End of statement?
037300   4204          JSR     ERRSY    ;   NO
037301   6137          STORDA           ;   YES
037302  34510          LDA     3,FNC    ;STORE SIGN OF STEP
037303  45112          STA     1,FNSTP+MXPRC-1,3
037304  30366          LDA     2,.BUS
037305  15003          DSZ     PBC.,2;  Backup PBC to end of statement
037306   6432          JSR     @.SKRS   ;Skip PBC and PLC to next statement
037307    471          JMP     FORER    ;   No next statement, ERROR
037310  30366          LDA     2,.BUS
037311  34501          LDA     3,FNC
037312  25002          LDA     1,PLC.,2;Save PLC of next statement (strt of loop)
037313  45401          STA     1,FNPLC,3
037314  25003          LDA     1,PBC.,2;Save PBC of next statement (strt of loop)
037315  45406          STA     1,FNPBC,3
037316  24507          LDA     1,FDSTP
037317   4475          JSR     FORST    ;STORE STEP VALUE
037320  24504          LDA     1,FDLIM
037321   4472          JSR     FORLD    ;LOAD LIMIT VALUE
037322  20002          LDA     0,C2
037323  24310          LDA     1,TS7
037324  30307          LDA     2,TS6
037325   6120          DECIMAL          ;SUBTRACT INITIAL VALUE
037326   6137          STORDA
```

```
                ;                                  << SI = R92RUNS4B;  BO = 18/A.RUN.8053! >>

37327  34463            LDA     3,FNC
37330  31412            LDA     2,FNSTP+MXPRC-1,3;SIGN OF STEP VALUE
37331 151212            MOVR#   2,2,SZC   ;NEGATIVE ?
37332    407            JMP     FORS1     ;   YES
37333 125014            SKZ     1,1       ;NEGATIVE DIFFERENCE ?
37334    410            JMP     FORSK;   YES
37335   2375            JMP     @.EXST    ;   NO

37336  37406 .FNXT:     FONXT
37337     14 FSESZ:     FESIZ   ;Size of a stack entry
37340  32713 .SKRS:     SKRST

37341 101014 FORS1:     SKZ     0,0       ;ZERO RESULT . . .
37342 125014            SKZ     1,1       ;OR NEGATIVE ?
37343   2375            JMP     @.EXST    ;   YES, Enter loop and execute next statement
                                          ;   No, Skip over loop

                ;   FOR terminated immediately, Search for matching NEXT to skip loop

37344  30366 FORSK:     LDA     2,.BUS
37345  21063            LDA     0,STPBC,2;Save PBC of FOR statement for possible
37346  40455            STA     0,FORPBC;  use in a missing NEXT error message
37347    403            JMP     FORSC     ;Check first statement after FOR

37350   6250 FORSL:     JSR     @.SKST    ;Search for matching "NEXT"
37351    424            JMP     FORENX    ;   No next statement, FOR without NEXT error
37352  30366 FORSC:     LDA     2,.BUS
37353  25003            LDA     1,PBC.,2
37354  45063            STA     1,STPBC,2;Mark as start of current statement
37355   6144            XGETBYTE          ;Load statement code of statement
37356  24211            LDA     1,C125
37357 146014            ADC#    2,1,SZR   ;"NEXT" STATEMENT ?
37360    770            JMP     FORSL     ;   NO
37361   6264            NEXTBYTE          ;   Yes, Check if control variable matches
37362   6256            JSR     @.VLOC
37363   4206            JSR     ERRST
37364  34366            LDA     3,.BUS
37365  15403            DSZ     PBC.,3    ;Step PBC back to allow skip over rest of stmt
37366 172400            SUB     3,2
37367  26423            LDA     1,@FNC
37370 146415            SNE     2,1       ;FOR THIS VARIABLE ?
37371   2374            JMP     @.NXST    ;   Yes, Execute the NEXT statement
37372   6746            JSR     @.SKRS    ;   No, Skip to next statement
37373    402            JMP     FORENX    ;   Error, There is no next stmt
37374    756            JMP     FORSC     ;Check this statement

37375  20426 FOREN:     LDA     0,FORPBC;FOR without NEXT error while skipping
37376  30366            LDA     2,.BUS
37377  41063            STA     0,STPBC,2;   Restore statement start location
37400  30366 FORER:     LDA     2,.BUS    ;FOR without NEXT error
37401  21063            LDA     0,STPBC,2
37402  41003            STA     0,PBC.,2 ;Restore current PBC to point at FOR stmt
37403   6123            JSR     @.FDPLC   ;Restore PLC also
37404   6257            RUNERROR          ;And then report the error
37405 111610            23*K+AE
```

```
                ;                                  << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>
37406  25015 FONXT: LDA     1,FSC.,2 ;"FOR" & "NEXT" SETUP
37407 147000        ADD     2,1
37410  44402        STA     1,FNC
37411   2956        JMP     @.VLOC   ;LOCATE INDEX VARIABLE

37412      0 FNC:   0                ;FOR-NEXT counter (absolute pointer)

37413 102521 FORLD: SUBZL   0,0,SKP  ;Load step or limit value
37414 102400 FORST: SUB     0,0      ;STORE STEP OR LIMIT VALUE
37415  30775        LDA     2,FNC
37416 133000        ADD     1,2
37417  24310        LDA     1,TS7
37420 125234        MOVZR#  1,1,SZR  ;INTEGER INDEX VARIABLE ?
37421  24024        LDA     1,STPRC  ;  NO
37422   2120        JMP     @DECIMAL&377


37423      0 FORPBC:        0        ;Starting PBC of FOR statement
37424      2 FDLIM: FNLIM   ;Offset to limit value
37425      7 FDSTP: FNSTP   ;Offset to step value
37426  33757 .FDPLC:       FDPLC    ;Routine to set PLC to match PBC
```

```
                ;                                  << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>
037427   4757 NEXT:  JSR     FONXT     ;"NEXT" STATEMENT
037430   4206        JSR     ERRST
037431  34366        LDA     3,.BUS
037432 172400        SUB     3,2
037433  21414        LDA     0,FNS.,3
037434 163000        ADD     3,0
037435  34755 NEXTO: LDA     3,FNC
037436  25764        LDA     1,FIVRD-FESIZ,3
037437 146415        SNE     2,1       ;INDEX VARIABLE ON TOP OF STACK ?
037440    410        JMP     NEXT1     ;   YES
037441  24676        LDA     1,FSESZ   ;   NO, POP THE STACK
037442 136400        SUB     1,3
037443 162432        SGR     3,0       ;STACK NOW EMPTY ?
037444   6267        RUNERROR          ;   YES, "NEXT" WITHOUT A "FOR"
037445 112210        24*K+AE
037446  54744        STA     3,FNC     ;   NO
037447    766        JMP     NEXTO

037450  24667 NEXT1: LDA     1,FSESZ   ;Compute base
037451 136400        SUB     1,3       ; of current entry
037452  54740        STA     3,FNC     ;Save that base away
037453  24310        LDA     1,TS7     ;INDEX VARIABLE AN INTEGER ?
037454 125235        MOVZR#  1,1,SNR
037455    452        JMP     NEXTI     ;   YES
037456  24747        LDA     1,FDSTP
037457   4734        JSR     FORLD     ;Load step value
037460  20003        LDA     0,C3
037461  24310        LDA     1,TS7
037462  30307        LDA     2,TS6
037463   6120        DECIMAL;Add current value
037464   6137        STORDA
037465 102400        SUB     0,0
037466  24310        LDA     1,TS7
037467  30307        LDA     2,TS6
037470   6120        DECIMAL           ;STORE NEW VALUE
037471   6131        LOADDA
037472  20002        LDA     0,C2
037473  24034        LDA     1,C14
037474  30716        LDA     2,FNC
037475  34727        LDA     3,FDLIM
037476 173000        ADD     3,2
037477   6120        DECIMAL           ;SUBTRACT FROM LIMIT VALUE
037500   6137        STORDA
037501  34711 NEXT2: LDA     3,FNC
037502  31412        LDA     2,FNSTP+MXPRC-1,3
037503 151213        MOVR#   2,2,SNC   ;IS STEP NEGATIVE ?
037504    411        JMP     NEXT5     ;   NO
037505 101014        SKZ     0,0       ;   YES
037506 125014        SKZ     1,1
037507    410        JMP     NEXT4
037510  30366 NEXT3: LDA     2,.BUS    ;Exit from loop
037511 156400        SUB     2,3       ;Partition A(FOR-NEXT stack)
037512  55015        STA     3,FSC.,2
037513  15003        DSZ     PBC.,2    ;Backup PBC to last token of statement
037514   2374        JMP     @.NXST    ;Skip over rest of NEXT stmt and execute next stmt
```

```
                ;                                << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>
37515  125014 NEXT5: SKZ     1,1
37516     7/2         JMP     NEXT3
37517   34673 NEXT4: LDA     3,FNC     ;CONTINUE LOOPING
37520   30366         LDA     2,.BUS
37521   21401         LDA     0,FNPLC,3
37522   41002         STA     0,PLC.,2;Reset PLC to start of loop
37523   21406         LDA     0,FNPBC,3
37524   41003         STA     0,PBC.,2;Reset PBC to start of loop
37525    2375         JMP     @.EXST    ;And repeat loop


37526   74631         74631
37527   22307 NEXTI: LDA     0,@TS6    ;INDEX VARIABLE IS AN INTEGER
37530   25407         LDA     1,FNSTP,3;GET STEP VALUE
37531  101100         MOVL    0,0
37532  127000         ADD     1,1       ;XOR signs of
37533  152560         SUBCL   2,2       ; step and index variable
37534  101220         MOVZR   0,0       ;   with result in A2
37535  125220         MOVZR   1,1
37536   34120         LDA     3,DECIMAL &377
37537  151014         SKZ     2,2       ;ARE SIGNS THE SAME ?
37540     406         JMP     NXTIO     ;   NO
37541    7776         JSR     @-2,3     ;   YES, ADD
37542  101113         SSN     0,0       ;OVERFLOW ?
37543  151112         SSP     2,2
37544   20762         LDA     0,NEXTI-1;   YES
37545     407         JMP     NXTI1

37546  122402 NXTIO: SGR     1,0       ;STEP > CURRENT VALUE ?
37547     404         JMP     NXTI1-1   ;   NO
37550  131100         MOVL    1,2       ;   YES
37551  105000         MOV     0,1
37552  141240         MOVOR   2,0
37553    7707         JSR     @-1,3     ;SUBTRACT
37554   26307 NXTI1: LDA     1,@TS6
37555  101124         MOVZL   0,0,SZR
37556  127000         ADD     1,1       ;SIGN OF OLD VALUE
37557  101200         MOVR    0,0
37560   42307         STA     0,@TS6    ;NEW INDEX VALUE
37561   34631         LDA     3,FNC
37562  111000         MOV     0,2
37563  105120         MOVZL   0,1
37564   21402         LDA     0,FNLIM,3;LIMIT VALUE
37565  103013         ADD#    0,0,SNC
37566     404         JMP     NXTI2
37567   35407         LDA     3,FNSTP,3
37570  177012         ADD#    3,3,SZC
37571     726         JMP     NEXT4     ;S(STEP) = S(LIMIT) <> S(CURRENT)
37572  103002 NXTI2: ADD     0,0,SZC
37573  123021         ADDZ    1,0,SKP
37574  122400         SUB     1,0
37575  153000         ADD     2,2
37576  105004         MOV     0,1,SZR
37577  126500         SUBL    1,1       ;SIGN OF DIFFERENCE
37600     701         JMP     NEXT2
```

```
                ;                              << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>
                ;
                ;   OPEN <openchannel> , <filename> {[ , <openchannel>] , <filename>}
                ;   BUILD # <expression> , <filename> {[ , # <expression>] , <filename>}
                ;
                ;   <openchannel> ::= # <expression> [ = <expression>]
                ;

37601 102001 OPENF: ADC     0,0,SKP   ;Process OPEN statement
37602 102400 BILDF: SUB     0,0       ;Process BUILD statement

37603  40314        STA     0,TSX3    ;Set BUILD/OPEN state
37604 102000        ADC     0,0
37605  40312        STA     0,TSX1    ;Set initial channel number to -1 (not set)

37606 102400        SUB     0,0
37607  40513        STA     0,OPNCCB-1;Set initial OPEN mode to standard open

                ;   Process next channel#/filename parameter for BUILD or OPEN

37610   6264 OPNNXT:        NEXTBYTE ;Get next token
37611  20064        LDA     0,C377
37612 142415        SNE     2,0       ;"#" found?
37613    410        JMP     OPNGCH    ;   Yes, Accept channel number
37614  34366        LDA     3,.BUS    ;   No, Backup over token for EXPRESSION
37615  15403        DSZ     PBC.,3
37616  24312        LDA     1,TSX1    ;Was the channel number ever specified?
37617 124015        COM#    1,1,SNR
37620   4204        JSR     ERRSY     ;   No, Syntax error
37621  34223        LDA     3,C361    ;   Yes, Load comma token and
37622    403        JMP     OPNSCH    ;And pretend that the channel number processed

37623   6263 OPNGCH:        INTEGER  ;Accept channel number expression
37624  44312        STA     1,TSX1    ;Set new current channel number
37625  20314 OPNSCH:        LDA      0,TSX3
37626 101015        SNZ     0,0       ;Processing BUILD statement?
37627    411        JMP     OPNCMA    ;   Yes, Don't check for mode on BUILD
37630  20221        LDA     0,C355
37631 116114        SEQ     0,3       ;Channel number followed by EQUALS?
37632    406        JMP     OPNCMA    ;   No, OPEN mode not specified
37633   6263        INTEGER;   Yes, Accept OPEN mode expression
37634 101014        SKZ     0,0       ;Negative OPEN mode?
37635   6267        RUNERROR          ;   Yes, Parameter error
37636 116110        34*K!SE
37637  44463        STA     1,OPNCCB-1;   No, Set OPEN mode

37640  20223 OPNCMA:        LDA      0,C361;End of channel number, Check for comma
37641 162414        SEQ     3,0       ;Comma after channel number clause?
37642   4204        JSR     ERRSY     ;   No, Syntax error

37643  20314        LDA     0,TSX3    ;BUILD or OPEN?
37644 101015        SNZ     0,0
37645    430        JMP     OPNNEW    ;   BUILD!
                                      ;   Do OPEN
```

```
                        ;                               << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>

37646   4510 OPNOLD:            JSR     CPYFNM;Copy filename into low core for OPEN
37647  20312           LDA     0,TSX1  ;Load channel number
37650  30151           LDA     2,.OPNCCB;Load address of OPEN Channel Control Block
37651  34151           LDA     3,OPNCCB-1;Load OPEN mode
37652 175014           SKZ     3,3     ;Standard OPEN?
37653    407           JMP     OPNMN1  ;  No, Use OPENMAINTENANCE
37654  34013           LDA     3,C37   ;  Yes, Use OPEN
37655  54146           STA     3,OPNCCB+0;Require data file type
37656   6106           CHANNEL ;Try to open the file
37657  40022             OPEN
37660   2005             JMP   @.CSEN  ;  Error in OPEN, Report error
37661    406           JMP     OPNDNE  ;Open done, Check for another file

37662 176000 OPNMNT:           ADC     3,3;Use OPENMAINTENANCE for special OPEN modes
37663  54140           STA     3,OPNCCB+0;  Allow any file type
37664   6106           CHANNEL         ;Try to open the file
37665     41             OPENMAINTENANCE
37666   2005             JMP   @.CSEN  ;  Error in open, Report error

                        ;  File opened, Check for additional files to open

37667  10312 OPNDNE:           ISZ     TSX1;Increment current channel number for next file
37670   6264           NEXTBYTE        ;Get next token
37671  20273           LDA     0,C361
37672 112415           SNE     0,2     ;Comma?
37673    715           JMP     OPNNXT  ;  Yes, Open or Build next file
37674   2246           JMP     @.EDST  ;  No, Check for end of statement and then execute next

37675  10074 OPNNEW:           ISZ     ETSF;Build a new file, Force swap after build
37676   6264           NEXTBYTE        ;Get next token
37677  24417           LDA     1,VBLDT ;Load text file type
37700  20420           LDA     0,C371
37701 112415           SNE     0,2     ;Filename prefixed with "+"?
37702    404           JMP     OPNNFN  ;  Yes, Build a text file
37703  34006           LDA     3,.BUS  ;  No, Build a formatted or contiguous file
37704  15403           DSZ     PBC.,3  ;Backup to first token of filename
37705  24412           LDA     1,VBLDF ;Load formatted/contiguous file type

37706  44420 OPNNFN:           STA     1,OPNBCB+0;Set filetype to build
37707   4447           JSR     CPYFNM  ;Copy filename to low core
37710  20312           LDA     0,TSX1  ;Load current channel number
37711  30114           LDA     2,.OPNBCB;Load address of Build Channel Control Block
37712   6106           CHANNEL         ;Try to build file (filename address is in A1)
37713     20             BUILD
37714   2005             JMP   @.CSEN  ;  Error in Build, Report error
37715    752           JMP     OPNDNE  ;Check for additional filenames
```

```
                ;                                  << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>

37716   77000 VBLDT: 77030   ;Text file type (with 77 protection)
37717   77037 VBLDF: 77037   ;Formatted/Contiguous file type (with 77 protection)
37720     371 C371:  371     ;"+", Ascii

37721   37/23 .OPNCCB:       OPNCCB
                             ;OPEN Channel Control Block
37722       0          0     ;  OPEN mode word
37723       0 OPNCCB:        0          ;  OPEN filetype
37724 177/77           -1    ;  OPEN logical unit (default)

37725   37/26 .OPNBCB:       OPNBCB
                             ;BUILD Channel Control Block
37726       0 OPNBCB:        0          ;  BUILD filetype
37727       1          1     ;  BUILD block count
37730 177//7           -1    ;  BUILD start address
37731       0          0     ;  BUILD cost
37732 177//7           -1    ;  BUILD logical unit (default)
37733       0          0     ;  BUILD format map address (nil)
```

```
                ;                                  << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>
037734   6264 CLOSF: NEXTBYTE                      ;"CLOSE" STATEMENT
037735  20064        LDA     0,C377
037736 142414        SEQ     2,0                  ;"#" ?
037737    445        JMP     CLALL                ;  NO, CLOSE ALL
037740   6263        INTEGER                      ;  YES
037741 121000        MOV     1,0
037742   6106        CHANNEL
037743     26        CLOSE
037744   2335        JMP     @.CSEN
037745  30362        LDA     2,TSN/
037746  20233        LDA     0,C361
037747 142415        SNE     2,0                  ;COMMA ?
037750    764        JMP     CLOSF                ;  YES
037751   2246 CLOSX: JMP     @.EDST               ;  NO

037752  35441        MOVST
037753     67        TSN+11-TSE*2-1
037754      0        0
037755    660        TSE*2
037756  54776 CPYFNM:       STA     3,.-2;COPY STRING ELEMENT INTO TSE-TSN
037757   6242        JSR     @.EVSE
037760   4204        JSR     ERRSY
037761  40331        STA     0,TS
037762  44302        STA     1,TS1
037763  20772        LDA     0,CPYFNM-1
037764  24767        LDA     1,CPYFNM-3
037765   6765        JSR     @CPYFNM-4
037766 102400        SUB     0,0
037767   6134        PUTBYTE
037770  24765        LDA     1,CPYFNM-1;Return byte address of filename
037771   2763        JMP     @CPYFNM-2

037772   4764 KILL:  JSR     CPYFNM               ;"KILL" STATEMENT
037773 102000        ADC     0,0
037774   6101        CALL
037775     16        DELETE
037776   2335        JMP     @.CSEN
037777   6264        NEXTBYTE                     ;Load terminator token
040000  24223        LDA     1,C361
040001 146415        SNE     2,1                  ;Comma?
040002    770        JMP     KILL                 ;  Yes, Kill another file
040003   2246        JMP     @.EDST               ;  NO
```

```
                ;                               << SI = R92RUN84B;  BO = 18/A.RUN.8053! >>


40004 102400 CLALL: SUB     0,0
40005  40414        STA     0,CLCHN   ;START WITH CHANNEL #0
40006  26412        LDA     1,@XNDCH; CLOSE ALL CHANNELS
40007  44413        STA     1,CLCNT
40010  20411 CLAL1: LDA     0,CLCHN   ;CURRENT CH #
40011  10410        ISZ     CLCHN     ;BUMP TO NEXT CH #
40012   6106        CHANNEL
40013     26        CLOSE
40014 100010        NOP     ;DON'T CARE ABOUT ERRORS
40015  14405        DSZ     CLCNT     ;CLOSED ALL YET?
40016    772        JMP     CLAL1     ;   NO, GOTO NEXT
40017    732        JMP     CLOSX

40020    603 XNDCH: INFO+NDCH.
40021      0 CLCHN: 0
40022      0 CLCNT: 0
```

```
                ;                               << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>
40023     341           TSP1
40024   20777 SIGNL:    LDA     0,.-1          ;"SIGNAL" STATEMENT
40025   40330           STA     0,TSE
40026   20003           LDA     0,C3
40027   40331           STA     0,TSE1
40030    4566 SGNLA:    JSR     SIGNX          ;EVALUATE A PARAMETER
40031   10330           ISZ     TSE
40032   20332           LDA     0,TSE2
40033   24223           LDA     1,C361
40034  106414           SEQ     0,1            ;COMMA ?
40035    4204           JSR     ERRSY          ;   NO
40036   20341           LDA     0,TSP1         ;   YES
40037  101015           SNZ     0,0            ;"SIGNAL 0" ?
40040    4204           JSR     ERRSY          ;   YES
40041  101204           MOVZR#  0,0,SZR        ;"SIGNAL 1" (SEND) ?
40042     421           JMP     SIGN1          ;   NO
40043   14331 SGNSD:    DSZ     TSE1           ;MORE PARAMETERS ?
40044     764           JMP     SGNLA          ;   YES
40045    4551           JSR     SIGNX          ;LAST PARAMETER
40046    6414           JSR     @XSGNE
40047   20341           LDA     0,TSP1
40050   24005           LDA     1,RUP          ;<< ENTRY FROM AO=6
40051   30752           LDA     2,SIGNL-1
40052  151400           INC     2,2            ;SEND THE SIGNAL
40053    6101           CALL    ;OR CLEAR SIGNALS
40054      57           SIGPAUSE
40055    6267           RUNERROR
40056  137110           76*K+SE
40057    2246           JMP     @.EDST


40060     611 .TIME:     INFO+TSC.              ;POINTER TO SYSTEM CLOCK
40061  106240 TSC.H:     6*12*6*12*12           ;TENTHS OF SECONDS PER HOUR
40062   40252 XSGNE:     SIGNE
```

```
                ;                               << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>
40063  24033 SIGN1:LDA     1,C3        ;"SIGNAL 3" (PAUSE) ?
40064 106414        SEQ     0,1
40065    434        JMP     SIG1A       ;   NO
40066   4530        JSR     SIGNX
40067   4563        JSR     SIGNE
40070  44343        STA     1,TSP3      ;PAUSE DELAY
40071  22767        LDA     0,@.TIME
40072  40345        STA     0,TSP5 ;STARTING TIME
40073   6270        XFLUSH              ;Flush output
40074   6262        GOGO
40075  22763        LDA     0,@.TIME
40076  24345        LDA     1,TSP5
40077  30762        LDA     2,TSC.H     ;TENTH-SECONDS PER HOUR
40100 106032        SGE     0,1         ;IF HOUR CHANGED DURING FLUSH
40101 143000        ADD     2,0         ;...ADD IN AN HOUR'S WORTH OF 10TH-SECONDS
40102 122400        SUB     1,0         ;FLUSH TOOK THIS LONG
40103  24343        LDA     1,TSP3
40104 106033        SLS     0,1         ;FLUSH TIME >= TO PAUSE DELAY
40105   2946        JMP     @.EDST      ;YES, RETURN NOW
40106 106400        SUB     0,1         ;NO, PAUSE FOR TIME REMAINING
40107  20003        LDA     0,C3        ;   NO, PAUSE FOR THE DIFFERENCE
40110   6101        CALL
40111     57        SIGPAUSE            ;Was Pause equal to zero?
40112   6117        BUMPUSER            ;   No, Wait (PDC in PCB has been set)
40113   2946        JMP     @.EDST      ;   Yes


40114      0        0
40115  5477 GBUMP: STA     3,GBUMP-1
40116   6262        GOGO
40117  3475        LDA     3,GBUMP-1
40120   2117        JMP     @BUMPUSER&377
```

```
                ;                                  << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>
40121  106015 SIG1A: ADC#    0,1,SNR   ;"SIGNAL 2" (RECEIVE) ?
40122     406         JMP     SIGNV     ;  YES
40123   24026         LDA     1,C6
40124  106115         SNE     0,1       ;CLEARING
40125     716         JMP     SGNSD     ;  YES
40126  106014         ADC#    0,1,SZR   ;RECEIVING SYSTEM SIGNAL?
40127    4204         JSR     ERRSY     ;  No, Syntax error
40130    6256 SIGNV: JSR     @.VLOC    ;  YES
40131    4206         JSR     ERRST
40132   54332         STA     3,TSE?
40133   20366         LDA     0,.BUS    ;SAVE VARIABLE POINTER & TYPE
40134  112543         SUBOL   0,2,SNC
40135  151122         MOVZL   2,2,SZC
40136    4204         JSR     ERRSY
40137   24310         LDA     1,TS7
40140  124400         NEG     1,1
40141  124000         COM     1,1
40142  147000         ADD     2,1
40143   30356         LDA     2,.BUS
40144   20331         LDA     0,TSE1
40145  112400         SUB     0,2
40146   45076         STA     1,TSU.3,2
40147   20303         LDA     0,C361    ;MORE PARAMETERS ?
40150  162414         SEQ     3,0
40151     414         JMP     SIGN2     ;  NO
40152   14331         DSZ     TSE1      ;  YES
40153     755         JMP     SIGNV
40154    4142         JSR     SIGNX     ;EVALUATE TIME-OUT DELAY
40155    4475         JSR     SIGNE
40156   20024         LDA     0,C4      ;PAUSE FOR RECEIVE WITH TIME-OUT
40157    6101         CALL
40160      57         SIGPAUSE          ;Is signal already waiting in queue?
40161    4724         JSR     GBUMP     ;  No, Wait (PDC in PCB has been set)
40162   20041         LDA     0,SIGNL-1 ;  Yes
40163   40330         STA     0,TSE
40164     405         JMP     SGN2A
```

```
                ;                                  << SI = R92RUNS4B;  BO = 18/A.RUN.8053! >>


40165   20331 SIGN2: LDA     0,TSE1     ;RECEIVE
40166  101204        MOVZR#  0,0,SZR
40167    4204        JSR     ERRSY
40170    4462        JSR     SIGNE
40171   30300 SGN2A: LDA     2,TSE
40172   20331        LDA     0,TSP1     ;RECEIVE A SIGNAL OR SYSTEM SIGNAL
40173    6101        CALL
40174      57        SIGPAUSE           ;Is signal already waiting in queue?
40175     434        JMP     SIGNN      ;   No
40176   20003        LDA     0,C3       ;   Yes
40177   40331        STA     0,TSE1
40200   26330 SIGN3: LDA     1,@TSE     ;STORE THE SIGNAL VALUES
40201  135102        MOVL    1,3,SZC
40202  124400        NEG     1,1
40203  102560        SUBCL   0,0
40204    6122        FLOAT
40205   34356        LDA     3,_BUS
40206   20331        LDA     0,TSE1
40207  116400        SUB     0,3
40210   31476        LDA     2,TSU.3,3
40211    4450        JSR     SIGNS
40212   10330        ISZ     TSE
40213   14331        DSZ     TSE1
40214     764        JMP     SIGN3
40215    2946        JMP     @.EDST
```

```
                ;                                  << SI = R92RUNS4B;  BO = 18/A.RUN.8053! >>
40216   54333 SIGNX: STA     3,TSE3      ;EVALUATE A PARAMETER
40217    6263         INTEGER
40220   54332         STA     3,TSE2
40221  125112         SSP     1,1         ;ABSOLUTE VALUE > 32767 ?
40222    6267         RUNERROR            ;   YES
40223  116110         34*K+SE
40224  101014         SKZ     0,0         ;NEGATIVE ?
40225  124400         NEG     1,1         ;   YES
40226   46320         STA     1,@TSE
40227    2333         JMP     @TSE3

40230   42602         INTF-2
40231  102520 SIGNN: SUBZL   0,0         ;NO SIGNAL RECEIVED
40232  126520         SUBZL   1,1
40233   3075          LDA     2,SIGNN-1
40234    6120         DECIMAL
40235   34366         LDA     3,.BUS
40236   31473         LDA     2,TSU.,3
40237    4102         JSR     SIGNS
40240    2346         JMP     @.EDST

40241   24003 SIGNS: LDA     1,C3        ;STORE A PARAMETER VALUE
40242  147400         AND     2,1
40243  132610         SUBOR   1,2
40244  151220         MOVZR   2,2
40245   20366         LDA     0,.BUS
40246  113000         ADD     0,2
40247  125100         INC     1,1
40250  102400         SUB     0,0
40251    2120         JMP     @DECIMAL&377

40252   20302 SIGNE: LDA     0,TSE2      ;CHECK FOR END OF STATEMENT
40253   30215         LDA     2,C342
40254  142654         SUBOR#  2,0,SZR
40255    4204         JSR     ERRSY
40256    1400         JMP     0,3




40257       0         0
40260   5477 EVIN:   STA     3,.-1       ;EVALUATE SIGNED INTEGER
40261    6257         ALGEXPRESSION
40262   54302         STA     3,TSN7
40263    6121         FIX
40264    6267         RUNERROR
40265  116110         34*K!SE
40266   34362         LDA     3,TSN7
40267    2770         JMP     @EVIN-1
```

```
                        ;                               << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>
 40270   33062  .EVCS:  EVCHS
 40271    6777  SRCH:   JSR     @.EVCS      ;"SEARCH" STATEMENT - Evaluate channel-mode-dir
 40272    4204           JSR     ERRSY
 40273    4470           JSR     CALLC       ;INITIALIZE CALL TABLE
 40274   20360           LDA     0,TSN5      ;CHANNEL #
 40275    6101           CALL
 40276  100002           CHKCHANNEL
 40277    6267           RUNERROR
 40300  130510           61*K!SE
 40301   50301           STA     2,TS        ;A(DFT ENTRY)
 40302   30371           LDA     2,CNODE
 40303   21001           LDA     0,1,2
 40304  101103           MOVL    0,0,SNC     ;IS ITEM 1 A STRING ?
 40305    4204           JSR     ERRSY       ;   NO
 40306  101250           MOVZR   0,0         ;   YES
 40307   41001           STA     0,1,2
 40310   21003           LDA     0,3,2
 40311  101112           SSP     0,0         ;IS ITEM 2 NUMERIC ?
 40312    4204           JSR     ERRSY       ;   NO
 40313   21005           LDA     0,5,2       ;   YES
 40314  101112           SSP     0,0         ;IS ITEM 3 NUMERIC ?
 40315    4204           JSR     ERRSY       ;   NO
 40316  141000           MOV     2,0         ;   YES, COPY A (PARAM NODE)
 40317    4405           JSR     SRCH1       ;Ac3 = Table address
 40320      60           DIRECTORY
 40321   40120           MODE0
 40322   40061           SEARCH
 40323   40122           PFSEARCH

 40324   30301  SRCH1:   LDA     2,TS        ;A(DFT)
 40325   31007           LDA     2,CLP,2
 40326  151212           SKE     2,2         ;Polyfile ?
 40327  175400           INC     3,3         ;   Yes, bias table address for polyfiles
 40330   25400           LDA     1,0,3       ;Assume mode 0
 40331   30353           LDA     2,TSN       ;MODE #
 40332  151014           SKZ     2,2         ;MODE ZERO ?
 40333   25402           LDA     1,2,3       ;   No, get Search routine
 40334   44410           STA     1,SERCL     ;DISCSUB TO CALL
 40335   24004           LDA     1,C377
 40336  133700           ANDS    1,2         ;M*400
 40337   34354           LDA     3,TSN1      ;DIRECTORY #
 40340  167400           AND     3,1
 40341  147000           ADD     2,1         ;M*400+D
 40342   30301           LDA     2,TS        ;A(CHANNEL)
 40343    6101           CALL
 40344       0  SERCL:   0
 40345    6267           RUNERROR            ;FILE IS WRITE PROTECTED
 40346  127110           56*K+SE
 40347  102400           SUB     0,0
 40350    6122           FLOAT               ;FLOAT STATUS
 40351   34371           LDA     3,CNODE
 40352   31404           LDA     2,4,3
 40353   25405           LDA     1,5,3
```

```
                ;                                << SI = R92RUN84B;  BO = 18/A.RUN.8053! >>
40354  102400           SUB     0,0
40355    6120           DECIMAL                  ;STORE IN V2
40356   30371           LDA     2,CNODE          ;RELEASE NODE
40357    6107           FREENODE
40360  102400           SUB     0,0              ;CLEAR NODE POINTER
40361   40371           STA     0,CNODE
40362    2246           JMP     @.EDST

40363   54330 CALLC:    STA     3,TSE            ;INITIALIZE FOR "CALL" OR "SEARCH"<<FROM CALLX,SRCH
40364  152400           SUB     2,2
40365    6107           FREENODE                 ;GRAB A NODE
40366   50371           STA     2,CNODE          ;REMEMBER WHICH ONE
40367   50332           STA     2,TSE2
40370   20310           LDA     0,C30
40371  105220           MOVZR   0,1
40372   44334           STA     1,TSE4
40373  120100           NEG     1,0
40374   24253           LDA     1,.TSF           ;FILL LIST WITH DUMMY POINTERS
40375  176400           SUB     3,3
40376   45000 CALLO:    STA     1,0,2
40377  151400           INC     2,2
40400   55000           STA     3,0,2
40401  151400           INC     2,2
40402  101404           INC     0,0,SZR
40403     773           JMP     CALLO
40404    6256 CALL2:    JSR     @.VLOC           ;GATHER UP ARGUMENT POINTERS
40405     417           JMP     CALL3
40406   52332 CAL22:    STA     2,@TSE2          ;A(VARIABLE)
40407   10332           ISZ     TSE2
40410   46332           STA     1,@TSE2          ;NUMBER TYPE
40411   10332           ISZ     TSE2
40412   20223           LDA     0,C361
40413  162414           SEQ     3,0              ;COMMA ?
40414     404           JMP     .+4              ;   NO
40415   14334           DSZ     TSE4             ;ARGUMENT LIST FULL ?
40416     766           JMP     CALL2            ;   NO
40417    4204           JSR     ERRSY            ;   YES

40420   20215           LDA     0,C342
40421  116654           SUBOR#  0,3,SZR          ;End of statement?
40422    4204           JSR     ERRSY            ;   NO
40423    2330           JMP     @TSE             ;   YES

40424   50331 CALL3:    STA     2,TSE1           ;ARGUMENT IS A STRING
40425  101014           SKZ     0,0
40426    4204           JSR     ERRSY            ;SUBSCRIPTS NOT ALLOWED
40427    6264           NEXTBYTE
40430  155000           MOV     2,3
40431   30331           LDA     2,TSE1
40432  151220           MOVZR   2,2
40433   22004           LDA     0,@PIB
40434  113000           ADD     0,2              ;ABSOLUTE WORD ADDRESS
40435   25377           LDA     1,-1,2
40436  127240           ADDOR   1,1              ;DIMENSION & STRING FLAG
40437     747           JMP     CAL22
```

```
                ;                                << SI = R92RUNS4B; BO = 18/A.RUN.8053! >>
                ;   CALL <subroutinenumber> , <variable> { , <variable> }
                ;   CALL <subroutinename> , <variable> { , <variable> }

 40440  25003 CALLS: LDA     1,PBC.,2;Load first byte of subroutine# or name
 40441   6144         XGETBYTE
 40442  20031         LDA     0,C11
 40443  24034         LDA     1,C14
 40444 142433         SLE     2,0       ;Within CALL name token range?
 40445 146433         SLE     2,1
 40446    427         JMP     CALLN     ;   No, Must be subroutine number rather than name
 40447 112400         SUB     0,2       ;   Yes, Calculate length of name in words
 40450 141000         MOV     2,0       ;Move length to AO
 40451  34346         LDA     3,.BUS
 40452  25403         LDA     1,PBC.,3 ;Load current object location
 40453 125400         INC     1,1
 40454 125620         INCZR   1,1       ;Roundup to word boundary and convert to word offset
 40455  30040         LDA     2,SBA
 40456 133000         ADD     1,2       ;Calculate word address of name
 40457 107120         ADDZL   0,1       ;Calculate byte address of following name
 40460  45403         STA     1,PBC.,3 ;Update current object location
 40461  36153         LDA     3,@.LCALT;Load address of CALL table from INFO
 40462 175415         INC#    3,3,SNR   ;Does the call table exist?
 40463   6257         RUNERROR          ;   No, Error 97, $CALLTBL not enabled
 40464 160510         141*K!AE
 40465   5776         JSR     LCFNM.,3 ;Search CALL table for match
 40466    447         JMP     CALLE     ;Match not found, Error
 40467  40432         STA     0,CALLD   ;Set discsub code
 40470   6264         NEXTBYTE          ;Get next token
 40471  34233         LDA     3,C361
 40472 172414         SEQ     3,2       ;Name followed by comma?
 40473   4204         JSR     ERRSY     ;   No, Syntax error
 40474    416         JMP     CALLX     ;   Yes, Do call

 40475   6257 CALLN: ALGEXPRESSION     ;"CALL" by subroutine number
 40476  30233         LDA     2,C361
 40477 172414         SEQ     3,2
 40500   4204         JSR     ERRSY
 40501   6121         FIX
 40502    433         JMP     CALLF
 40503  36131         LDA     3,@.LCALT;Load address of CALL table
 40504 175415         INC#    3,3,SNR   ;Does the call table exist?
 40505   6257         RUNERROR          ;   No, Error 97, $CALLTBL not enabled
 40506 160610         141*K!AE
 40507   5727         JSR     LCFNBR.,3;Search table for match to subroutine number
 40510    425           JMP   CALLE     ;   Match not found, Error
 40511  40110         STA     0,CALLD   ;SAVE DISCSUB NUMBER
```

```
                        ;                               << SI = R92RUNS4B;  BO = 18/A.RUN.8053! >>

 40512    4591 CALLX: JSR     CALLC       ;SET UP PARAMETER LIST
 40513   34366         LDA     3,.BUS
 40514   24367         LDA     1,.EUS      ;A(END USER'S PARTITION)
 40515   30371         LDA     2,CNODE     ;A(PARM LIST)
 40516   21407         LDA     0,NVS.,3
 40517  163000         ADD     3,0         ;START OF AVAIL PARTITION AREA
 40520    6101         CALL
 40521       0 CALLD: 0
 40522  126001         ADC     1,1,SKP     ;FLAG ERROR
 40523  126400         SUB     1,1         ;NO ERROR
 40524   30371         LDA     2,CNODE     ;RELEASE NODE
 40525    6107         FREENODE
 40526  102400         SUB     0,0         ;CLEAR NODE POINTER
 40527   40371         STA     0,CNODE
 40530  125014         SKZ     1,1         ;SKIP IF NO ERROR
 40531    6267         RUNERROR            ;ERROR DETECTED BY SUBROUTINE
 40532  123110         46*K+SE
 40533    2246         JMP     @.EDST

 40534     677 .LCALT:         INFO+LCALT;Address of INFO cell containing CALL table address


 40535    6267 CALLE: RUNERROR            ;Subroutine number or name not implemented
 40536  122110           44*K+SE

          141         .LOC  LREV-.  ;PATCH SPACE (OVERLAP CHECK) ***

                    ; *** PATCHES FOR SOURCES 1 THRU 4 ONLY ***

                    .EOT  ; "RUN" R9.0 SOURCE #4
```

```
                ;                              << SI = R9ORUNS5A; BO = 18/A.RUN.8053! >>
                ; "RUN" SOURCE #5 OF 7 TAPES FOR "IRIS" R9.0

        40700           .LOC    LREV        ; TAPES 5 & 6 ALSO USED IN "RUNMAT"


40700       0           0
40701   54777 FIXS:    STA     3,.-1       ;FIX SUBSCRIPT OR CHANNEL #
40702    6121           FIX                 ;NUMBER IS IN AC(1)
40703  102240           ADCOR   0,0         ;Number out of range - force sign negative
40704  101015           SNZ     0,0         ;NEGATIVE ?
40705    2773           JMP     @FIXS-1     ;   NO
40706    6267           RUNERROR            ;   YES
40707  116110           34*K+SE


40710   24213 LUVD:     LDA     1,C336      ;LOOK UP VARIABLE DEFINITION
40711  146033           SLS     2,1         ;REAL VARIABLE ?
40712     417           JMP     LUDV        ;   NO
40713   24052           LDA     1,C177      ;   YES
40714  147530           ANDZL   2,1         ;CHANGE VAR# TO DISPLACEMENT
40715   30366           LDA     2,.BUS
40716   21001           LDA     0,VDT.,2
40717  113000           ADD     0,2         ;A(VARIABLE DEFINITION TABLE)
40720  133000           ADD     1,2         ;A(THIS VARIABLE'S ENTRY)
40721   21000           LDA     0,0,2       ;A0 = FIRST NAME WORD
40722  145000           MOV     2,1         ;A1 = A(FIRST NAME WORD)
40723   31001           LDA     2,1,2       ;A2 = DISPLACEMENT TO VARIABLE
40724  101113           SSN     0,0         ;ARRAY ?
40725    1400           JMP     0,3         ;   NO
40726  151400           INC     2,2         ;   YES, POINT TO FIRST ELEMENT
40727  151400           INC     2,2
40730    1400           JMP     0,3

40731  132540 LUDV:     SUBOL   1,2         ;DUMMY VARIABLE "&" OR "%"
40732  151120           MOVZL   2,2
40733   20111           LDA     0,DFV-1
40734  113000           ADD     0,2
40735   20002           LDA     0,C300
40736    1400           JMP     0,3

40737  102401 DFA:      SUB     0,0,SKP     ;RESULT TO "&" VARIABLE 337

40740   20024 DFP:      LDA     0,C4        ;RESULT TO "%" VARIABLE 336
40741   30103           LDA     2,DFV-1
40742  113000           ADD     0,2
40743     411           JMP     DFN
```

```
                ;                               << SI = R90RUNS5A;  BO = 18/A.RUN.8053! >>
 40744       31         DFS.
 40745    20000 DFV:    LDA     0,C10          ;RESULT TO DUMMY VARIABLE 200
 40746    30776          LDA     2,DFV-1
 40747   113000          ADD     0,2
 40750    34356          LDA     3,.BUS
 40751    21401          LDA     0,VDT.,3
 40752   117000          ADD     0,3
 40753    51401          STA     2,1,3
 40754   102400 DFN:    SUB     0,0            ;DUMMY FUNCTIONS
 40755    40277          STA     0,EVPF
 40756    24024          LDA     1,C4
 40757    34356          LDA     3,.BUS
 40760   173000          ADD     3,2
 40761     6120          DECIMAL
 40762    14275          DSZ     I
 40763    14275          DSZ     I
 40764     2401          JMP     @.+1
 40765    41675          EV3J

 40766        0          0                      ;A(VARIABLE STORAGE)
 40767        0          0                      ;SIZE
 40770    54356 ALOC:    STA     3,TS5          ;ASSIGN LOCATION FOR VARIABLE
 40771    44775          STA     1,ALOC-2
 40772   103112          ADDL#   0,0,SZC        ;STRING VARIABLE ?
 40773      412          JMP     ALOC1          ;   YES
 40774    34356          LDA     3,.BUS         ;   NO
 40775    31424          LDA     2,FLAG.,3
 40776    34003          LDA     3,C3
 40777   173100          AND     3,2            ;CURRENT PRECISION SWITCH SETTING
 41000   155300          MOVS    2,3
 41001   175220          MOVZR   3,3
 41002   175220          MOVZR   3,3
 41003   163000          ADD     3,0
 41004      410          JMP     ALOC2
 41005    24303 ALOC1:   LDA     1,TS2
 41006   125014          SKZ     1,1            ;2ND SUBSCRIPT
 41007     6267          RUNERROR               ;   YES
 41010   114110          30*K+SE
 41011    24302          LDA     1,TS1          ;   NO
 41012    34024          LDA     3,C4           ;ALLOW ROOM FOR TERMINATOR
 41013   167221          ADDZR   3,1,SKP
 41014   145400 ALOC2:  INC     2,1
 41015    44752          STA     1,ALOC-1       ;VARIABLE SIZE
 41016    34356          LDA     3,.BUS
 41017    31407          LDA     2,NVS.,3
 41020    44424          STA     1,ALOCO-1
 41021   147000          ADD     2,1
 41022    35410          LDA     3,EUS.,3
 41023   136103          SLE     1,3            ;SUFFICIENT SPACE ?
 41024     4421          JSR     ALOCO          ;   NO
 41025    34356          LDA     3,.BUS         ;   YES
 41026    45407          STA     1,NVS.,3       ;UPDATE .NVS
```

```
                        ;                                  << SI = R90RUNS5A;  BD = 18/A.RUN.8053! >>

 41027  34737           LDA     3,ALOC-2
 41030  41400           STA     0,0,3       ;UPDATE VDT ENTRY
 41031  51401           STA     2,1,3
 41032  34366           LDA     3,.BUS
 41033 173000           ADD     3,2
 41034 176400           SUB     3,3
 41035  55000           STA     3,0,2       ;CLEAR VARIABLE STORAGE AREA
 41036 151400           INC     2,2
 41037  14730           DSZ     ALOC-1
 41040    775           JMP     .-3
 41041  34725           LDA     3,ALOC-2
 41042  31401           LDA     2,1,3
 41043   2306           JMP     @TS5


 41044      0           0                          ;SIZE REQUIRED FOR VARIABLE
 41045  20777 ALOCO:    LDA     0,..-1             ;SETUP FOR ASKM
 41046  40370           STA     0,RRSZ
 41047  30717           LDA     2,ALOC-2
 41050  21000           LDA     0,0,2
 41051  24107           LDA     1,APM              ;RESET ARRAY BIT, PRECISION BITS
 41052 123400           AND     1,0
 41053  41000           STA     0,0,2
 41054 102400           SUB     0,0
 41055  41001           STA     0,1,2              ;RESET LOCATION
 41056   6267           RUNERROR                   ;Report storage overflow trap
 41057 101610           3*K+AE


 41060  77477 APM:      77477
```

```
                ;                                   << SI = R90RUNS5A; BD = 18/A.RUN.8053! >>
41061         0         0
41062     40302 ADIM:   STA     0,TS1       ;ASSIGN DIMENSIONS TO VARIABLE
41063     44303         STA     1,TS2
41064     54304         STA     3,TS3
41065    125015         SNZ     1,1         ;SINGLE SUBSCRIPT ?
41066       411         JMP     ADIM1       ;   YES
41067    115400         INC     0,3         ;   NO
41070    125202 ADML:   MOVZR   1,1,SZC     ;MULTIPLY SUBSCRIPTS
41071    163023         ADDZ    3,0,SNC
41072    175102         MOVZL   3,3,SZC
41073      6267         RUNERROR            ;ARRAY TOO LARGE
41074    152610         125*K+AE
41075    125014         SKZ     1,1
41076       772         JMP     ADML
41077    101423 ADIM1:  INCZ    0,0,SNC     ;FIRST SUBSCRIPT TOO LARGE ...
41100    101112         SSP     0,0         ;OR NEGATIVE ?
41101      6267         RUNERROR            ;   YES
41102    153210         126*K+AE
41103     40317         STA     0,TSX6      ;TOTAL # OF ELEMENTS
41104      4604         JSR     LUVD
41105    151015         SNZ     2,2         ;STORAGE ASSIGNED ?
41106       450         JMP     ADIML       ;   NO
41107    103112         ADDL#   0,0,SZC     ;STRING VARIABLE ?
41110      6267         RUNERROR            ;   YES
41111    134510         71*K+SE
41112    101113         SSN     0,0         ;DIMENSIONED ARRAY ?
41113      4204         JSR     ERRSY       ;   NO
41114      4501         JSR     ADIMN       ;   YES
41115     40306         STA     0,TS5
41116     34356         LDA     3,.BUS
41117     25104         LDA     1,UPS.,3
41120    167000         ADD     3,1
41121     44740         STA     1,ADIM-1
41122     25407         LDA     1,NVS.,3
41123     21401         LDA     0,VDT.,3
41124    117000         ADD     0,3
41125    175400 ADIM2:  INC     3,3         ;SEARCH FOR FOLLOWING VARIABLE STORAGE
41126    175400         INC     3,3
41127     20732         LDA     0,ADIM-1
41130    162432         SGR     3,0         ;END OF DEFINITION TABLE ?
41131       407         JMP     .+7         ;   YES
41132     21400         LDA     0,0,3       ;   NO, GET POINTER
41133    112433         SLE     0,2         ;IS IT BEFORE THIS ARRAY ...
41134    106133         SLE     0,1         ;OR AFTER CLOSEST YET FOUND ?
41135       770         JMP     ADIM2       ;   YES
41136    105000         MOV     0,1         ;   NO, SAVE POINTER
41137       766         JMP     ADIM2

41140    146400         SUB     2,1         ;AREA RESERVED FOR THIS ARRAY
41141     20306         LDA     0,TS5
41142    106133         SLE     0,1         ;SPACE REQUIRED > INITIAL DIM ?
41143      6267         RUNERROR            ;   YES, CAN'T DIM LARGER THAN ORIGINAL
41144    113510         27*K+SE
41145     34356         LDA     3,.BUS      ;   NO
41146    173000         ADD     3,2
```

SLS TWM 07/14/88

(E) #38 8/18/88

(1) x

```
                        ;                               << SI = R90RUNS5A;  BO = 18/A.RUN.8053! >>
41147   20302 ADIMS: LDA     0,TS1      ;STORE DIMENSIONS
41150   24303        LDA     1,TS2
41151   41376        STA     0,-2,2     ;# OF ROWS
41152   45377 ADIMA: STA     1,-1,2     ;# OF COLS OR # BYTES
41153   34366        LDA     3,.BUS
41154  172400        SUB     3,2        ;DISPLACEMENT
41155    2304        JMP     @TS3

41156    4512 ADIML: JSR     ALOC       ;LOCATION NOT ASSIGNED, ASSIGN IT
41157  151400        INC     2,2        ;MAKE ROOM FOR ONE DIMENSION
41160  103113        ADDL#   0,0,SNC    ;STRING VARIABLE ?
41161     405        JMP     ADIMD      ;   NO, ARRAY
41162   24302        LDA     1,TS1
41163   34366        LDA     3,.BUS
41164  173000        ADD     3,2
41165     765        JMP     ADIMA

41166  151400 ADIMD: INC     2,2        ;MAKE ROOM FOR 2ND DIMENSION
41167  101100        MOVL    0,0
41170  101240        MOVOR   0,0        ;SET "ARRAY" BIT
41171   41400        STA     0,0,3
41172    4123        JSR     ADIMN      ;COMPUTE # DATA WORDS
41173  145000        MOV     2,1
41174   40550        STA     0,ALOCO-1
41175  107000        ADD     0,1        ;NEXT ADDRESS BEYOND DATA
41176   34356        LDA     3,.BUS
41177  173000        ADD     3,2        ;CONVERT TO ABSOLUTE
41200   35310        LDA     3,EUS.,3
41201  136003        SLS     1,3        ;ENOUGH STORAGE SPACE ?
41202    4513        JSR     ALOCO      ;   NO, TRY TO GET MORE
41203   34366        LDA     3,.BUS     ;   YES
41204   45407        STA     1,NVS.,3   ;UPDATE .NVS
41205  167000        ADD     3,1
41206  155000        MOV     2,3
41207  102400        SUB     0,0
41210   41400        STA     0,0,3      ;CLEAR NEW STORAGE AREA
41211  175400        INC     3,3
41212  166002        SGE     3,1
41213     775        JMP     .-3
41214     733        JMP     ADIMS      ;   NO
```

```
                        ;                                  << SI = R90RUNS5A;  BO = 18/A.RUN.8053! >>
41215   54306 ADIMN:STA     3,TS5       ;COMPUTE # WORDS FOR ARRAY
41216   34052         LDA     3,C300
41217  163700         ANDS    3,0         ;PRECISION IN TOP 2 BITS
41220   40305         STA     0,TS4
41221   34317         LDA     3,TSX6      ;NUMBER OF ELEMENTS
41222  101122         MOVZL   0,0,SZC     ;PRECISION 2 OR 3 ?
41223  163000         ADD     3,0         ;   YES
41224  101122         MOVZL   0,0,SZC     ;PRECISION 1 OR 3 ?
41225  163000         ADD     3,0         ;   YES
41226  163000         ADD     3,0         ;COMPENSATE FOR ORIGIN ZERO
41227    2306         JMP     @TS5


41230   54124 VLOC:  STA     3,EVAX-1    ;LOCATE VARIABLE
41231    4132         JSR     EVEX
41232     413         JMP     VLOCS       ;   STRING
41233   10421         ISZ     EVAX-1      ;   NUMERIC WILL SKIP RETURN
41234   20356         LDA     0,.BUS
41235  151014         SKZ     2,2         ;LITERAL, OR . . .
41236  142432         SGR     2,0         ;OUTSIDE OF USER AREA ?
41237    4204         JSR     ERRSY       ;   YES
41240   20357         LDA     0,.EUS
41241  142433         SLF     2,0
41242    4204         JSR     ERRSY       ;   YES
41243   20221         LDA     0,C355      ;   NO
41244     405         JMP     VLOCS+4

41245  113003 VLOCS:ADD     0,2,SNC     ;STRING VARIABLE
41246  176001         ADC     3,3,SKP     ;   TWO SUBSCRIPTS
41247  136100         SUB     1,3         ;A3 := 0 ???
41250  106400         SUB     0,1
41251   44310         STA     1,TS7       ;DIMENSION
41252   50307         STA     2,TS6       ;LOCATION (SEE BELOW)
41253    2401         JMP     @EVAX-1

                    ;NOTE:  THE "LOCATION" RETURNED BY VLOC IN A2
                    ;       AND IN TS6 IS AN ABSOLUTE WORD ADDRESS
                    ;       FOR A NUMERIC VARIABLE OR A RELATIVE
                    ;       BYTE ADDRESS FOR A STRING VARIABLE.


                    ;ENTRIES IN EL (EVALUATOR LIST) STACK:
                    ;    0 - 100    LITERAL INTEGER CONSTANT 0 - 40 DECIMAL
                    ; 101 - 377     VALUE IS IN EN STACK (DISP. FROM EN-101)
                    ; 400 - 177776    IN USER AREA (LITERAL OR VARIABLE) *
                    ; 177777        VALUE IS IN DA

                    ;   *  BITS 15,14 = PRECISION-1,
                    ;      BITS 13-0  = DISP. FROM BUS


                    ;*****************************************************
                    ;* NOTE:  14-BIT DISP. LIMITS BASIC PARTITION TO 16K WORDS.  *
                    ;*****************************************************
```

```
                ;                                   << SI = R90RUNS5A;  BO = 18/A.RUN.8053! >>
41254       0           0
41255   54777 EVAX:   STA     3,,-1       ;EVALUATE ALGERRAIC EXPRESSION
41256    4405           JSR     EVEX
41257    4206           JSR     ERRST       ;STRINGS NOT ALLOWED
41260  151014           SKZ     2,2         ;HAS A VARIABLE LOC BEEN ASSIGNED ?
41261    2773           JMP     @EVAX-1     ;   YES
41262    4204           JSR     ERRSY       ;   NO, SYNTAX ERROR


41263  126401 EVEX:   SUB     1,1,SKP     ;EVALUATE EXPRESSION
41264  126000 EVSU:   ADC     1,1         ;EVALUATE SUBSCRIPTS
41265   44274           STA     1,EVSW
41266   54301           STA     3,TS
41267   24201           LDA     1,.EL       ;INITIALIZE STACK POINTERS
41270   44275           STA     1,I
41271   24202           LDA     1,.EN
41272   44276           STA     1,J
41273  102400 EV0:    SUB     0,0         ;INITIALIZE FLAGS & EL PAD
41274   40277           STA     0,EVPF
41275   40300           STA     0,EVLF
41276   30214           LDA     2,C340      ;"PAD" CHARACTER
41277   52275 EV1:     STA     2,@I        ;PUSH ONTO EL STACK
41300   10275 EV2:     ISZ     I
41301   20275           LDA     0,I
41302   24202           LDA     1,.EN
41303  106033           SLS     0,1         ;EL STACK OVERFLOW ?
41304    6267           RUNERROR            ;   YES
41305  112510           25*K+SE
41306    6264 EV3:     NEXTBYTE            ;   NO, GET NEXT TOKEN
41307   34275           LDA     3,I         ;PUT BYTE ONTO STACK
41310   51400           STA     2,0,3
41311   25777           LDA     1,-1,3      ;L[I-1]
41312   20214           LDA     0,C340
41313  142033           SLS     2,0         ;L[I] = OPERATOR ?
41314     444           JMP     EV0         ;   YES
41315  122002           SGE     1,0         ;L[I-1] = OPERATOR ?
41316    4204           JSR     ERRSY       ;   NO
41317   20064           LDA     0,C377
41320  122003           SLS     1,0
41321    4204           JSR     ERRSY       ;   NO
41322   20053           LDA     0,C200      ;   YES
41323  142033           SLS     2,0         ;L[I] = VARIABLE ?
41324     754           JMP     EV2         ;   YES
41325   20051           LDA     0,C100      ;   NO
41326  142402           SGR     2,0         ;L[I] = INTEGER CONSTANT ?
41327     750           JMP     EV1         ;   YES
41330   20571           LDA     0,C174      ;   NO
41331  142003           SLS     2,0         ;L[I] = F.P. CONSTANT ?
41332     407           JMP     EVC         ;   YES
41333   20176           LDA     0,C166      ;   NO
41334  142415           SNE     2,0         ;L[I] = LEN FUNCTION ?
41335   10300           ISZ     EVLF        ;   YES
41336   10275           ISZ     I           ;   NO
41337   30200           LDA     2,C376      ;MUST BE NUMERIC FUNCTION
41340     737           JMP     EV1         ;STORE FUNCTION PSEUDO-OPERATOR
```

```
                ;                                << SI = R90RUNS5A; BO = 18/A.RUN.8053! >>
41341   34356 EVC:   LDA     3,.BUS     ;FLOATING POINT CONSTANT
41342   25403        LDA     1,PBC.,3
41343  125620        INCZR   1,1        ;RELATIVE WORD LOCATION
41344  112400        SUB     0,2        ;NUMBER TYPE
41345  121000        MOV     1,0
41346  147000        ADD     2,1
41347  125520        INCZL   1,1        ;NEXT PROGRAM BYTE ADDRESS
41350   45403        STA     1,PBC.,3
41351  151220        MOVZR   2,2        ;SHIFT NUMBER TYPE TO TOP BITS
41352  151200        MOVR    2,2
41353  151200        MOVR    2,2
41354  113000        ADD     0,2        ;COMBINE WITH REL. LOCATION (A2=A(#))
41355   20055        LDA     0,C400
41356  113000        ADD     0,2
41357     720        JMP     EV1        ;STORE THE RELATIVE LOCATION OF CONSTANT

41360   20224 EVO:   LDA     0,C362     ;L[I] = OPERATOR
41361  142414        SEQ     2,0        ;L[I] = "(" ?
41362     416        JMP     EV5        ;   NO
41363   34064        LDA     3,C377     ;   YES
41364  122003        SLS     1,0        ;L[I-1] = ANY OF THE FOLLOWING ?
41365  136103        SLE     1,3        ;( [ , - + / * ^
41366     406        JMP     .+6
41367   20226        LDA     0,C366
41370  122414        SEQ     1,0
41371  106015        ADC#    0,1,SNR
41372    4204        JSR     ERRSY      ;   NO
41373     705        JMP     EV2        ;   YES

41374   20214        LDA     0,C340
41375  122414        SEQ     1,0        ;L[I-1] = PAD ?
41376    4204        JSR     ERRSY      ;   NO
41377     701        JMP     EV2        ;   YES

41400   20053 EV5:   LDA     0,C200     ;L[I] = OPERATOR, BUT NOT "("
41401  122002        SGE     1,0        ;L[I-1] = VARIABLE ?
41402     404        JMP     .+4
41403   20214        LDA     0,C340
41404  122002        SGE     1,0
41405     436        JMP     EVV        ;   YES
41406   20051        LDA     0,C100     ;   NO
41407  122402        SGR     1,0        ;L[I-1] = INTEGER ?
41410     522        JMP     EVP        ;   YES
41411   20054        LDA     0,C377     ;   NO
41412  122403        SLE     1,0        ;L[I-1] = OPERAND ?
41413     517        JMP     EVP        ;   YES
41414   20224        LDA     0,C362     ;   NO
41415  122414        SEQ     1,0        ;L[I-1] = "(", "[", OR PAD ?
41416  106015        ADC#    0,1,SNR
41417     404        JMP     .+4
41420   20214        LDA     0,C340
41421  122414        SEQ     1,0
41422    4204        JSR     ERRSY      ;   NO
41423   20177        LDA     0,C370     ;   YES
```

```
                ;                                   << SI = R90RUNS5A;  BO = 18/A.RUN.8053! >>
41424 142414            SEQ     2,0             ;L[I] = "-" ?
41425    405            JMP     .+5             ;   NO
41426 102400            SUB     0,0             ;   YES
41427  42275            STA     0,@I
41430  10275            ISZ     I
41431    616            JMP     EV1

41432 112015            ADC#    0,2,SNR         ;L[I] = "+" ?
41433    653            JMP     EV3             ;   YES
41434  30275            LDA     2,I             ;   NO
41435  24231            LDA     1,.EL
41436 132004            ADC     1,2,SZR         ;FIRST BYTE ?
41437   4204            JSR     ERRSY           ;   NO
41440  36275            LDA     3,@I            ;   YES
41441  10301            ISZ     TS
41442   2301            JMP     @TS             ;FIRST BYTE WAS TERMINATOR

41443  20284 EVV:       LDA     0,C362          ;L[I-1] = VARIABLE
41444 112015            ADC#    0,2,SNR         ;SUBSCRIPT FOLLOWS ? ((A2) IS A [ OR ( )
41445    633            JMP     EV2             ;   YES
41446 131000            MOV     1,2             ;   NO
41447  20274            LDA     0,EVSW
41450 101015            SNZ     0,0             ;EVALUATING SUBSCRIPTS ?
41451    405            JMP     .+5             ;   NO
41452  20231            LDA     0,.EL           ;   YES
41453 101400            INC     0,0
41454 116015            ADC#    0,3,SNR         ;FIRST OPERAND ?
41455    432            JMP     EVR             ;   YES
41456   6251            JSR     @.LUVD          ;   NO, LOOK UP DEFINITION
41457 151014            SKZ     2,2             ;LOCATION ASSIGNED ?
41460    406            JMP     .+6             ;   YES
41461 103112            ADDL#   0,0,SZC         ;STRING VARIABLE ?
41462   6267            RUNERROR                ;   YES
41463 114110            31*K!NOP
41464   6434            JSR     @.ALOC          ;   NO
41465    436            JMP     EVL
```

```
                ;                                   << SI = R90RUNS5A; BO = 18/A.RUN.8053! >>

41466 101113            SSN     0,0         ;MATRIX VARIABLE ?
41467 103113            ADDL#   0,0,SNC     ;   NO, STRING ?
41470    433            JMP     EVL         ;   YES    NO
41471  34275            LDA     3,I         ;          YES
41472  20231            LDA     0,.EI
41473 101400            INC     0,0
41474 116015            ADC#    0,3,SNR     ;FIRST OPERAND ?
41475    412            JMP     EVR         ;   YES, THAT'S OK
41476  14275            DSZ     I           ;   NO
41477  20176            LDA     0,C166      ;TOKEN CODE FOR "LEN"
41500  34275            LDA     3,I         ;STACK POINTER
41501  25776            LDA     1,-2,3      ;PREVIOUS TOKEN
41502 106415            SNE     0,1         ;WAS IT "LEN"?
41503    404            JMP     EVR         ;   YES, EVERYTHING OK
41504  25775            LDA     1,-3,3      ;PREVIOUS TOKEN
41505 106414            SEQ     0,1         ;WAS IT "LEN"?
41506    4206           JSR     ERRST       ;   NO, STRING ERROR
41507 102400 EVR:       SUB     0,0         ;   YES
41510 126400            SUB     1,1
41511  34366            LDA     3,.BUS
41512  15403            DSZ     PBC.,3      ;RESTORE "NEXTBYTED" BYTE
41513  34274            LDA     3,EVSW
41514 175014            SKZ     3,3         ;EVALUATING SUBSCRIPTS ?
41515   2301            JMP     @TS         ;   YES
41516   2401            JMP     @.+1        ;   NO
41517  42011 EVSR
```

```
                        ;                                  << SI = R9ORUNS5A; BO = 18/A.RUN.8053! >>
41520   407/0 .ALOC:ALOC
41521     174 C174: 174
41522     370 C370: 370

41523   24065 EVL:  LDA     1,C400    ;LOCATE VARIABLE STORAGE
41524  133000       ADD     1,2
41525   24062       LDA     1,C300
41526  107/00       ANDS    0,1
41527  133000       ADD     1,2       ;"LOCATION" OF VARIABLE VALUE
41530   34275       LDA     3,I
41531   517/7       STA     2,-1,3
41532   34275 EVP:  LDA     3,I       ;PROCESS LIST
41533   21400       LDA     0,0,3
41534   24230       LDA     1,C376
41535  106415       SNE     0,1       ;L[I] = FUNCTION PSEUDO-OPERATOR ?
41536     522       JMP     EVD       ;   YES
41537   25776       LDA     1,-2,3    ;   NO
41540  101200       MOVZR   0,0
41541  125200       MOVZR   1,1
41542  106103       SLE     0,1       ;P(L[I]) > P(L[I-2]) ?
41543     515       JMP     EVD       ;   YES
41544   20755       LDA     0,C174    ;   NO
41545  122032       SGE     1,0       ;P(L[I-2]) < 6 ?
41546    4204       JSR     ERRSY     ;   YES
41547   30052       LDA     2,C177    ;EVALUATE L[I-2] CENTERED TRIPLE)
41550  146400       SUB     2,1
41551   317/7       LDA     2,-1,3
41552  151415       INC#    2,2,SNR   ;SECOND OPERAND IN DA ?
41553     423       JMP     EVP1      ;   YES
41554  125015       SNZ     1,1       ;L[I-2] = FUNCTION PSEUDO-OP ?
41555     404       JMP     .+4       ;   YES
41556   31775       LDA     2,-3,3    ;   NO
41557  151415       INC#    2,2,SNR   ;FIRST OPERAND IN DA ?
41560     413       JMP     EVPO+1    ;   YES
41561   20277       LDA     0,EVPF    ;   NO
41562  101015       SNZ     0,0       ;ANYTHING IN DA ?
41563     407       JMP     EVPO      ;   NO
41564    6241       JSR     @.EVQ     ;   YES
41565   317/5       LDA     2,-3,3
41566   25776       LDA     1,-2,3
41567   20230       LDA     0,C376
41570  122415       SNE     1,0       ;FUNCTION ?
41571   317/7       LDA     2,-1,3    ;   YES
41572    6237 EVPO: JSR     @.EVG     ;GET FIRST OPERAND
41573   317/7       LDA     2,-1,3    ;<< ENTRY FROM ABOVE
41574  102400       SUB     0,0
41575     403       JMP     EVP1+2
```

```
                        ;                                   << SI = R9ORUNS5A; BO = 18/A.RUN.8053! >>
41576  31775 EVP1:  LDA     2,-3,3    ;SECOND OPERAND IS IN DA
41577  20030        LDA     0,C10
41600  25776        LDA     1,-2,3    ;<< ENTRY FROM EVPO
41601  34230        LDA     3,C376
41602 136115        SNE     1,3       ;FUNCTION ?
41603   2477        JMP     @.FUNC    ;   YES
41604  24065        LDA     1,C400    ;LOCATE OTHER OPERAND
41605 146033        SLS     2,1       ;OPERAND IN USER AREA ?
41606    415        JMP     EVP2      ;   YES
41607  24051        LDA     1,C100    ;   NO
41610 146133        SLE     2,1       ;INTEGER CONSTANT ?
41611    405        JMP     .+5       ;   NO
41612  52276        STA     2,@J      ;   YES
41613  30276        LDA     2,J
41614 126100        SUB     1,1
41615    421        JMP     EVP3

41616  24377        LDA     1,D.EN    ;OPERAND IS IN EN STACK
41617 132400        SUB     1,2
41620  24025        LDA     1,C5
41621    415        JMP     EVP3

41622    374        374
41623 151100 EVP2:  MOVL    2,2       ;OPERAND IS IN USER AREA
41624 126560        SUBCL   1,1
41625 151100        MOVL    2,2
41626 125100        MOVL    1,1       ;NUMBER TYPE
41627 151220        MOVZR   2,2
41630 151220        MOVZR   2,2       ;RELATIVE LOCATION
41631  36004        LDA     3,@PIB
41632 173000        ADD     3,2       ;ABSOLUTE LOCATION
41633  34045        LDA     3,C400    ;ADJUST FOR VARIABLE INDICATOR
41634 172400        SUB     3,2
41635 125100        INC     1,1
41636 107000 EVP3:  ADD     0,1       ;ADD INVERSE FLAG TO NUMBER TYPE
41637  34275        LDA     3,I
41640  21776        LDA     0,-2,3
41641  34761        LDA     3,EVP2-1
41642 116415        SNE     0,3       ;OPERATOR = "^" ?
41643   2436        JMP     @.EXPN    ;   YES
41644  34276        LDA     3,C366    ;   NO
41645 162400        SUB     3,0
41646   6120        DECIMAL           ;CALL ARITHMETIC ROUTINE
41647 152000 EVP4:  ADC     2,2
41650  50277        STA     2,EVPF    ;FLAG "RESULT IN DA"
41651  34275        LDA     3,I
41652  51775        STA     2,-3,3
41653  31400        LDA     2,0,3     ;COPY BACK NEXT OPERATOR
41654  51776        STA     2,-2,3
41655  14275 EVPI:  DSZ     I
41656  14275        DSZ     I
41657    653        JMP     EVP
```

```
                ;                                   << SI = R9ORUNS5A; 80 = 18/A.RUN.8053! >>
41660  25400 EVD:  LDA     1,0,3      ;DISCONTINUE PROCESSING LIST
41661  20641       LDA     0,C370
41662 122033       SLS     1,0        ;P(I) < 6 ?
41663   2415       JMP     @.EV2      ;   NO
41664  20226       LDA     0,C366     ;   YES
41665 122414       SEQ     1,0        ;L[I] = ")" ?
41666    416       JMP     EV6        ;   NO
41667  25776       LDA     1,-2,3     ;   YES
41670  20224       LDA     0,C362
41671 122414       SEQ     1,0        ;L[I-2] = "(" ?
41672    424       JMP     EV7        ;   NO
41673  25777       LDA     1,-1,3     ;   YES
41674  45776       STA     1,-2,3     ;DROP THE PARENTHESES
41675  14275 EV3J: DSZ     I
41676   2401       JMP     @.+1
41677  41306       EV3

41700  41300 .EV2: EV2
41701  42457 .EXPN:EXPN
41702  42317 .FUNC:FUNC

41703    364 C364: 364


41704 106015 EV6:  ADC#    0,1,SNR    ;L[I] = "]" ?
41705    430       JMP     EV5        ;   YES
41706  20775       LDA     0,C364     ;   NO
41707 122414       SEQ     1,0        ;L[I] = "," ?
41710    405       JMP     .+5        ;   NO
41711  25776       LDA     1,-2,3     ;   YES
41712 122014       ADC#    1,0,SZR    ;L[I-2] = "[" ?
41713   4204       JSR     ERRSY      ;   NO
41714   2764       JMP     @.EV2      ;   YES

41715  25776       LDA     1,-2,3
41716  20214 EV7:  LDA     0,C340
41717 122414       SEQ     1,0        ;L[I-2] = PAD ?
41720   4204       JSR     ERRSY      ;   NO
41721  31777       LDA     2,-1,3     ;   YES
41722 151414       INC#    2,2,SZR    ;RESULT IN DA ?
41723   6207       JSR     @.EVC      ;   NO, GET IT
41724  34366       LDA     3,.BUS
41725  21413       LDA     0,UFC.,3
41726  35412       LDA     3,UFS.,3
41727 116403       SLE     0,3        ;EVALUATING A USER'S FUNCTION ?
41730    335       JMP     EVF        ;   YES
41731  36275       LDA     3,@I       ;   NO, GET TERMINATOR BYTE
41732  10301       ISZ     TS
41733   2301       JMP     @TS        ;SKIP RETURN, RESULT IS IN DA
```

```
                ;                                   << SI = R90RUNS5A;  BO = 18/A.RUN.8053! >>
41734     401           401
41735   25776 EVS:      LDA     1,-2,3      ;SUBSCRIPT EVALUATION
41736   20745           LDA     0,C364
41737  122415           SNE     1,0         ;DOUBLE ?
41740     406           JMP     EVS0        ;   YES
41741  122004           ADC     1,0,SZR     ;SINGLE ?
41742    6142           TRAPFAULT           ;   NO, NCOD ALLOWED BAD CLOSURE !?
41743  116410           35*K!NOP
41744   41400           STA     0,0,3       ;   YES
41745     407           JMP     EVS1

41746   31777 EVS0:     LDA     2,-1,3      ;<< FROM EVS
41747    6515           JSR     @.EVCS
41750    6245           JSR     @.FIXS      ;SECOND SUBSCRIPT
41751   14275           DSZ     I
41752   14275           DSZ     I
41753   46275           STA     1,@I
41754   14275 EVS1:     DSZ     I           ;<< FROM EVS
41755   32275           LDA     2,@I
41756    6506           JSR     @.EVCS      ;FETCH THE SUBSCRIPT
41757    6245           JSR     @.FIXS      ;FIRST SUBSCRIPT
41760   46275           STA     1,@I        ;REPLACE VARIABLE POINTER WITH VARIABLE
41761   14275           DSZ     I           ;POP THE STACK
41762   14275           DSZ     I
41763   32275           LDA     2,@I
41764   24274           LDA     1,EVSW
41765  125015           SNZ     1,1         ;EVALUATING SUBSCRIPTS ?
41766     405           JMP     EVS2
41767   34275           LDA     3,I         ;   YES, DONE
41770   21402           LDA     0,2,3
41771   25403           LDA     1,3,3
41772    2301           JMP     @TS

41773    6251 EVS2:     JSR     @.LUVD      ;   NO  << FROM EVS1
41774   24062           LDA     1,C300
41775  107700           ANDS    0,1
41776   44305           STA     1,TS4
41777  101112           SSP     0,0         ;ARRAY ?
42000     520           JMP     EVA         ;   YES
42001  103113           ADDL#   0,0,SNC     ;NUMERIC VARIABLE ?
42002     507           JMP     EVN         ;   YES
42003  151015           SNZ     2,2         ;STRING.   IS LOCATION ASSIGNED ?
42004    6267           RUNERROR            ;   NO
42005  114610           31*K+AE
42006   34275           LDA     3,I         ;   YES
42007   21402           LDA     0,2,3       ;FIRST SUBSCRIPT
42010   25403           LDA     1,3,3       ;SECOND SUBSCRIPT
42011   34366 EVSR:     LDA     3,.BUS      ; << ENTRY FROM EVR
42012  157000           ADD     2,3         ;A(STRING)
42013   35100           LDA     3,0,3       ;STRING DIMENSION
42014   54322           STA     3,TSF
42015  151520           INCZL   2,2         ;B(STRING) REL
```

```
                        ;                                   << SI = R90RUNS5A;  BD = 18/A.RUN.8053! >>

42016  100404                NEG     0,0,SZR     ;FIRST SUBSCRIPT ZERO ?
42017  100000                COM     0,0         ;   NO, DECREMENT IT
42020  125005                MOVZ    1,1,SNR     ;SECOND SUBSCRIPT ZERO ? (CLEAR CARRY)
42021  165040                MOVO    3,1         ;   YES, SUBSTITUTE DIMENSION (SET CARRY)
42022  106002                SGE     0,1         ;   NO
42023  136403                SLE     1,3
42024    6267        EVSED:  RUNERROR            ;SUBSCRIPT > DIMENSION
42025  116110                34*K+SE
42026   34300                LDA     3,EVLF
42027  175014                SKZ     3,3         ;LEN FUNCTION FLAG SET ?
42030     403                JMP     EVSD0
42031   34322                LDA     3,TSF       ;   NO, RECOVER STRING'S DIMENSION
42032    2301                JMP     @TS

42033  143000        EVSD0:  ADD     2,0         ;   YES
42034   40322                STA     0,TSF       ;B(STRING) + 1ST SUBSCRIPT
42035  147000                ADD     2,1
42036   44323                STA     1,TSF1      ;B(STRING) + 2ND SUBSCRIPT
42037   14275        EVSD1:  DSZ     I           ;POP STACK
42040   32275                LDA     2,@I
42041   34224                LDA     3,C362
42042  156414                SEQ     2,3         ;L[I] = "(" ?
42043     407                JMP     EVSD2       ;   NO
42044    6264                NEXTBYTE            ;   YES
42045   34206                LDA     3,C366
42046  156415                SNE     2,3         ;MATCHING ")" ?
42047     770                JMP     EVSD1       ;   YES
42050    6267                RUNERROR            ;   NO, BAD PAREN CLOSURE
42051  105510                13*K+SE

42052   34230        EVSD2:  LDA     3,C376
42053  156414                SEQ     2,3         ;L[I] = FUNCTION PSEUDO OP ?
42054    4206                JSR     ERRST       ;   NO, ILLEGAL STRING USAGE
42055   14275                DSZ     I           ;   YES, POP THE STACK
42056   32275                LDA     2,@I
42057   34176                LDA     3,C166
42060  156414                SEQ     2,3         ;L[I] = "LEN" FUNCTION ?
42061    4206                JSR     ERRST       ;   NO, ILLEGAL STRING USAGE
42062    2401                JMP     @.+1        ;   YES
42063   42452                LENF

42064   42217    .EVGS:      EVGS
                             BSTR: 0    ; B(STRING)

                        .EOT    ;  "RUN" R9.0 SOURCE #5
```

SAA 10/23/82

```
STA  2, BSTR ;PRESERVE B(STRING)
LDA  2, -EL  ;(-) #EL ONTO STACK
LDA  3, I    ;(-) I ONTO STACK
ADCZ 2, 3, SNR ;IS EVALUATOR STACK EMPTY?
JMP  EVRET   ;   NO, RETURN TO CALLER
LDA  3, EVSW ;   YES
COM# 3,3,SZR ;EVALUATING CONSTANT?
JSR  ERRST   ;   NO, GENERATE ERROR
LDA  2, BSTR ;   YES, RETRIEVE B(STRING)
```

SAA 10/23/82

```
                ;                              << SI = R92RUNS6A; BO = 18/A.RUN.8053! >>
                ; "RUN" SOURCE #6 OF 7 FOR "IRIS" R9.0

42065  26275 EVF:   LDA     1,@I        ;EVALUATING INTERPRETIVE FUNCTION
42066  20215         LDA     0,C342
42067 106654         SUBOR#  0,1,SZR     ;End of definition?
42070   6267         RUNERROR            ;   NO
42071 115610         33*K+AE
42072  34366         LDA     3,.BUS      ;   YES
42073  31413         LDA     2,UFC.,3    ;POP USER FUNCTION STACK
42074  20025         LDA     0,C5
42075 112400         SUB     0,2
42076  51413         STA     2,UFC.,3
42077 173000         ADD     3,2         ;RESTORE RETURN BYTE ADDRESS
42100  25004         LDA     1,4,2
42101  45403         STA     1,PBC.,3
42102 102000         ADC     0,0         ;FLAG "RESULT IN DA"
42103  40277         STA     0,EVPF
42104  34275         LDA     3,I         ;CLOSE UP EL STACK
42105  41774         STA     0,-4,3
42106  14275         DSZ     I
42107   2401         JMP     @.+1
42110  41655         EVPI

42111  20032 EVN:    LDA     0,C12       ;NUMERIC VARIABLE WITH SUBSCRIPTS
42112  34275         LDA     3,I         ;(MUST AUTO-DIMENSION)
42113  25403         LDA     1,3,3
42114  31400         LDA     2,0,3
42115 125014         SKZ     1,1         ;SINGLE SUBSCRIPT (10 X 0) ?
42116 105000         MOV     0,1         ;   NO, DIMENSION 10 X 10
42117   6203         JSR     @.ADIM
42120  34275 EVA:    LDA     3,I         ;ARRAY
42121  51400         STA     2,0,3
42122  20366         LDA     0,.BUS
42123 113000         ADD     0,2
42124  21376         LDA     0,-2,2
42125  25377         LDA     1,-1,2
42126  31402         LDA     2,2,3
42127 142433         SLE     2,0         ;FIRST SUBSCRIPT > DIM ?
42130   4674         JSR     EVSE0       ;   YES
42131  21403         LDA     0,3,3       ;   NO
42132 106433         SLE     0,1         ;SECOND SUBSCRIPT > DIM ?
42133   4671         JSR     EVSED       ;   YES
42134 125400         INC     1,1         ;   NO
42135 125242         MOVZR   1,1,SZC     ;FORM PRODUCT OF SUBSCRIPTS
42136 143000         ADD     2,0
42137 151120         MOVZL   2,2
42140 125014         SKZ     1,1
42141    774         JMP     .-4
42142  30305         LDA     2,TS4       ;MULTIPLY BY NUMBER LENGTH
42143 151122         MOVZL   2,2,SZC
42144 113000         ADD     0,2
```

```
                ;                                  << SI = R92RUNS6A; BO = 18/A.RUN.8053! >>
42145 151122            MOVZL   2,2,SZC
42146 113000            ADD     0,2
42147 113000            ADD     0,2
42150  25400            LDA     1,0,3       ;ADD RESULT TO ARRAY LOCATION
42151 133000            ADD     1,2
42152  20065            LDA     0,C400      ;FLAG AS DISPLACEMENT FROM ((@PIB))
42153 113000            ADD     0,2
42154  20305            LDA     0,TS4       ;ADD NUMBER TYPE INDICATOR
42155 113000            ADD     0,2
42156 102400            SUB     0,0         ;CLEAR "OPERAND IN DA" FLAG
42157  40277            STA     0,EVPF
42160   2244            JMP     @.EV1       ;PUT "OPERAND" IN LIST

42161    607            INFO+LEPS.
42162  54166 EVQ:       STA     3,DB        ;SAVE (DA) IN NUMBER STACK
42163 102400            SUB     0,0
42164  40277            STA     0,EVPF
42165  34275            LDA     3,I
42166  30276            LDA     2,J
42167  24026            LDA     1,C6
42170 147000            ADD     2,1
42171  22770            LDA     0,@EVQ-1
                ;LOOKS LIKE THE FOLLOWING TEST IS NOT SAFE
                ;ALLOWS CONTINUATION EVEN IF J=LEPS
                ;RENNY BOSCH
42172 122433            SLE     1,0         ;NUMBER STACK OVERFLOW ?
42173   6267            RUNERROR            ;   YES
42174 112510            25*K+SE
42175  44276            STA     1,J         ;   NO
42176  20377            LDA     0,D.EN
42177 143000            ADD     2,0
42200 174400 EVQ1:      NEG     3,3         ;FIND "OPERAND IN DA" FLAG
42201 174000            COM     3,3
42202  24231            LDA     1,.EL
42203 166032            SGE     3,1
42204   6142            TRAPFAULT           ;NO -1 IN LIST !?
42205 116410            35*K!NOP
42206  25400            LDA     1,0,3
42207 125414            INC#    1,1,SZR
42210    770            JMP     EVQ1
42211  41400            STA     0,0,3       ;OPERAND POINTER INTO EL STACK
42212 102400            SUB     0,0
42213  24025            LDA     1,C5
42214   6120            DECIMAL             ;STORE ACCUMULATOR IN EN STACK
42215  34275            LDA     3,I
42216   2166            JMP     @DB

42217  20277 EVGS:      LDA     0,EVPF      ;GET SUBSCRIPT
42220 126400            SUB     1,1
42221  44277            STA     1,EVPF
42222 151414            INC#    2,2,SZR     ;IS ANYTHING ELSE . . .
42223 101015            SNZ     0,0         ;NOW IN DA ?
42224    406            JMP     EVG         ;   NO
42225  50147            STA     2,DB+1      ;   YES
42226  54170            STA     3,DB+2
42227   4733            JSR     EVQ         ;SAVE IN EN
```

```
42230  30167        LDA     2,DB+1
42231  34170        LDA     3,DB+2
```

```
                ;                                   << SI = R92RUNS6A; BO = 18/A.RUN.8053! >>
42232  151415  EVG:   INC#    2,2,SNR   ;GET OPERAND INTO DA
42233    1400         JMP     0,3       ;   IT'S ALREADY THERE
42234   54304         STA     3,TS3
42235   24065         LDA     1,C400
42236  146003         SLS     2,1       ;OPERAND IN USER AREA ?
42237     435         JMP     EVG3      ;   YES
42240   24051         LDA     1,C100    ;   NO
42241  146133         SLE     2,1       ;INTEGER CONSTANT ?
42242     422         JMP     EVG1      ;   NO
42243   24002         LDA     1,C2      ;   YES, FLOAT IT
42244   34012         LDA     3,C20
42245  156003         SLS     2,3       ;INTEGER < 10 ?
42246     404         JMP     EVG0      ;   NO
42247  153120         ADDZL   2,2       ;   YES
42250  153120         ADDZL   2,2
42251  126520         SUBZL   1,1
42252  141300  EVG0:  MOVS    2,0
42253   40160         STA     0,DA      ;LEFT-JUSTIFY MANTISSA
42254  102400         SUB     0,0
42255   40161         STA     0,DA+1    ;ZERO REST OF MANTISSA
42256   40162         STA     0,DA+2
42257   40163         STA     0,DA+3
42260   44164         STA     1,DAC     ;EXPONENT IS 1 OR 2
42261   40165         STA     0,DAS     ;SIGN IS POSITIVE
42262    6131         LOADDA
42263     406         JMP     EVG2

42264   24377  EVG1:  LDA     1,D.EN    ;OPERAND IS IN EN STACK
42265  132400         SUB     1,2
42266  102520         SUBZL   0,0
42267   24025         LDA     1,C5
42270    6120         DECIMAL           ;LOAD DA FROM STACK
42271  152000  EVG2:  ADC     2,2
42272   34275         LDA     3,I
42273    2304         JMP     @TS3

42274  132400  EVG3:  SUB     1,2       ;OPERAND IS IN USER AREA
42275  151120         MOVZL   2,2       ;ISOLATE # TYPE
42276  126560         SUBCL   1,1
42277  151100         MOVL    2,2
42300  125100         MOVL    1,1
42301  151220         MOVZR   2,2
42302  151220         MOVZR   2,2       ;RELATIVE LOCATION
42303   36004         LDA     3,@PIB
42304  173000         ADD     3,2       ;ABSOLUTE LOCATION
42305   50303         STA     2,TS2
42306  125400         INC     1,1       ;NUMBER TYPE
42307   44302         STA     1,TS1
42310  102520         SUBZL   0,0
42311    6120         DECIMAL           ;LOAD PACKED OPERAND
42312   24302         LDA     1,TS1     ;# TYPE
42313   30303         LDA     2,TS2     ;A(OPERAND)
42314   34275         LDA     3,I       ;EL STACK POINTER
42315    2304         JMP     @TS3
```

```
                        ;                                  << SI = R92RUNS6A; BO = 18/A.RUN.8053! >>
42316     132           132
42317   34275 FUNC:     LDA     3,I        ;FUNCTION, ANY NUMERIC TYPE
42320   25775           LDA     1,-3,3
42321   34775           LDA     3,FUNC-1
42322  136402           SGR     1,3        ;USER-DEFINED FUNCTION ?
42323     467           JMP     UFN        ;   YES
42324   34176           LDA     3,C166     ;   NO
42325  136033           SLS     1,3        ;LEGAL FUNCTION USAGE ?
42326     403           JMP     FUNCE
42327   34212           LDA     3,C136
42330  136432           SGR     1,3
42331    6267 FUNCE:    RUNERROR           ;   NO
42332  116510           35*K+SE
42333    4432           JSR     FUNCB      ;   YES

42334   42520           ERRFN   ;  137
42335   42604           INTF    ;  140
42336   42634           ABSF    ;  141
42337   42625           SGNF    ;  142
42340   43127           RNDF    ;  143
42341      52           PSQRF   ;  144
42342      51           PLOGF   ;  145
42343   40053           PEXPF   ;  146
42344   40054           PSINF   ;  147
42345   60055           PCOSF   ;  150
42346      50           PATNF   ;  151
42347      56           PTANF   ;  152
42350   43120           DETF    ;  153
42351   40745           DFV     ;  154
42352   40737           DFA     ;  155
42353   40740           DFP     ;  156
42354   42600           FRAF    ;  157
42355   42636           MANF    ;  160
42356   43111           CHRF    ;  161
42357   43063           IXRF    ;  162
42360   42560           SPCF    ;  163
42361   42573           NOTF    ;  164
42362   42641           CHFF    ;  165

42363     144 CF144:    144
42364     152 CF152:    152
```

```
                                                    << SI = R92RUNS6A;  BO = 18/A.RUN.8053! >>
42365   44322 FUNCB: STA     1,TSF
42366  137000        ADD     1,3         ;BRANCH ON FUNCTION
42367   31641        LDA     2,-137,3
42370    6107        STORDA
42371  155122        MOVZL   2,3,SZC     ;INTERPRETIVE FUNCTION ?
42372     430        JMP     SFN         ;   YES
42373   24322        LDA     1,TSF       ;   NO
42374   40322        STA     0,TSF
42375   34766        LDA     3,CF144
42376  136002        SGE     1,3
42377     411        JMP     RUNF
42400   34764        LDA     3,CF152
42401  136433        SLE     1,3
42402     406        JMP     RUNF
42403   50403        STA     2,.+3
42404  126400        SUB     1,1
42405    6101        CALL
42406       0        0
42407    2240        JMP     @.EVP4

42410  126400 RUNF:  SUB     1,1
42411    1000        JMP     0,2
```

```
                ;                                << SI = R92RUNS6A;  B0 = 18/A.RUN.8053! >>

42412  34366 UFN:   LDA     3,.BUS    ;USER'S FUNCTION
42413  31411         LDA     2,UFT.,3
42414 173000         ADD     3,2       ;.UFT
42415 133000         ADD     1,2
42416  21277         LDA     0,-101,2
42417 101015         SNZ     0,0       ;IS FUNCTION DEFINED ?
42420   6267         RUNERROR           ;  NO
42421 117110         36*K+SE
42422  34366 SFN:   LDA     3,.BUS    ;  YES  << SYSTEM FUNCTION ENTRY
42423  31413         LDA     2,UFC.,3  ;USER FUNC STK COUNT
42424  25414         LDA     1,FNS.,3  ;FOR-NEXT STK
42425 146033         SLS     2,1       ;FUNCTIONS NESTED TOO DEEP ?
42426   6267         RUNERROR           ;  YES
42427 117610         37*K+AE
42430  24025         LDA     1,C5      ;  NO
42431 147000         ADD     2,1       ;STEP STACK POINTER
42432  45413         STA     1,UFC.,3  ;SET NEXT STACK POINTER
42433 173000         ADD     3,2       ;ABSOLUTE STACK POINTER
42434  25403         LDA     1,PBC.,3
42435  45004         STA     1,4,2     ;SAVE RETURN BYTE ADDRESS
42436  41403         STA     0,PBC.,3  ;SET FUNCTION BYTE ADDRESS
42437 141000         MOV     2,0
42440 162400         SUB     3,0       ;REL A(FUNCTION FRAME)
42441  25401         LDA     1,VDT.,3
42442 137000         ADD     1,3
42443  41401         STA     0,1,3     ;DUMMY VARIABLE DISPLACEMENT
42444  20062         LDA     0,C300
42445  41400         STA     0,0,3
42446  34275         LDA     3,I
42447  21400         LDA     0,0,3
42450  41776         STA     0,-2,3
42451 102400         SUB     0,0       ;STORE DUMMY VARIABLE VALUE
42452  24024         LDA     1,C4      ;AT (A2)
42453   6120         DECIMAL
42454  14275         DSZ     I
42455   2401         JMP     @.+1      ;EVALUATE THE FUNCTION
42456  41273         EVO
```

```
                ;                                   << SI = R92RUN86A; BO = 18/A.RUN.8053! >>
42457    6101 EXPN:  CALL                           ;Exponentiation "B^X"
42460     147            PPWR
42461    2240         JMP     @.EVP4

42462   14300 LENF:   DSZ     EVLF                 ;STRING LENGTH FUNCTION
42463  100010         NOP
42464   20277         LDA     0,EVPF
42465  101014         SKZ     0,0                  ;ANYTHING IN DA ?
42466    6241         JSR     @.EVQ                ;   YES, STACK IT
42467   24323         LDA     1,TSF1               ;B(STRING + 2ND SS)
42470    6144         XGETBYTE
42471   50324         STA     2,TSF2               ;SAVE LAST BYTE
42472  102400         SUB     0,0
42473   40325         STA     0,TSF3               ;RESET BYTE COUNT
42474    6145         XPUTBYTE                     ;STORE TEMPORARY TERMINATOR
42475   24322         LDA     1,TSF
42476    6144 LENFL:  XGETBYTE                     ;GET A BYTE
42477  151015         SNZ     2,2                  ;LAST BYTE ?
42500     404         JMP     .+4                  ;   YES
42501   10325         ISZ     TSF3                 ;BUMP BYTE COUNT
42502  125100         INC     1,1
42503     773         JMP     LENFL

42504   20324         LDA     0,TSF2
42505   24323         LDA     1,TSF1
42506    6145         XPUTBYTE                     ;RESTORE SAVED BYTE
42507  102400         SUB     0,0
42510   24325         LDA     1,TSF3
42511    6122         FLOAT                        ;FLOAT BYTE COUNT
42512  152000         ADC     2,2
42513   50277         STA     2,EVPF               ;SET "RESULT IN DA" FLAG
42514    2244         JMP     @.EV1
```

```
                ;                                  << SI = R92RUNS6A; BO = 18/A.RUN.8053! >>
42515 177757            177777?EMOD1    ;Ones-complement of IF ERR 1 flag to clear flag
42516     20            EMOD1
42517    114            114

42520  30366 ERRFN:LDA     2,.BUS
42521  25063           LDA     1,STPBC,2;"ERROR" Function
42522   6144           XGETBYTE        ;Load statement code of current statement
42523  347/4           LDA     3,ERRFN-1
42524 156404           SUB     2,3,SZR  ;"IF" STATEMENT ?
42525    443           JMP     ERRFU    ;  NO
42526  34366           LDA     3,.BUS
42527  31403           LDA     2,PBC.,3 ;Load current PBC
42530 151220           MOVZR   2,2      ;Convert to word pointer
42531 173000           ADD     3,2
42532  31377           LDA     2,-1,2   ;  and load last word of IF expression
42533  25360           LDA     1,FL4.,3;Load current IF ERR mode flag (and others)
42534  20761           LDA     0,ERRFN-3
42535 107400           AND     0,1      ;Clear IF ERR mode
42536  20067           LDA     0,C1000
42537 142003           SLS     2,0      ;Is the IF ERR mode 0 or 1? (check last token of exprsn)
42540    430           JMP     ERRFU    ;  No, Error in function usage
42541  20064           LDA     0,C377
42542 143400           AND     2,0      ;Isolate lower byte in A0
42543 112404           SUB     0,2,SZR  ;Check upper byte, Is it Mode 1?
42544  30752           LDA     2,ERRFN-2;  Yes, Load IF ERR 1 mode flag bit
                                        ;  No, Leave A2 as zero
42545 147000           ADD     2,1      ;Add (OR) in new IF ERR 1 mode bit
42546  45460           STA     1,FL4.,3;And update FLAG4
42547  25403           LDA     1,PBC.,3;Load current PBC (May point at error code)
42550  30215           LDA     2,C342
42551 142015           ADC#    2,0,SNR  ;Was expression terminated by Stmt Follows Token?
42552    404           JMP     ERRF1    ;  Yes, Set Error Pointer (ERBP.) to current PBC
42553 142414           SEQ     2,0      ;  No, Was it an End of Statement (Line) Token?
42554    414           JMP     ERRFU    ;  No, Error - illegal function usage
42555 126400           SUB     1,1      ;    Yes, Reset Error Pointer
42556  45452 ERRF1:STA     1,ERBP.,3
42557   2247           JMP     @.NXLN



42560  34121 SPCF:  LDA     3,FIX&377 ;FIX SPC NUMBER
42561   5777           JSR     -1,3
42562    406           JMP     ERRFU
42563 155000           MOV     2,3
42564  30366           LDA     2,.BUS
42565  55100           STA     3,TSU.5,2;Save H.O. word
42566   6101           CALL
42567  40016           SPECIAL
42570   6267 ERRFU:RUNERROR
42571 116510           35*K+SE
42572   2240           JMP     @.EVP4
```

```
                ;                                  << SI = R92RUNS6A;  BO = 18/A.RUN.8053! >>
42573 105000 NOTF:  MOV     0,1            ;"NOT" FUNCTION
42574  20035        LDA     0,C15
42575 125015        SNZ     1,1
42576 101400        INC     0,0
42577    424        JMP     FDEC


42600  20042 FRAF:  LDA     0,C20          ;FRACTIONAL PORTION FUNCTION
42601    422        JMP     FDEC           ;DBRK


42602 100001        100001                 ;ALSO USED BY SIGNAL
42603  42602        .-1
42604  20042 INTF:  DOBREAK                ;INTEGER FUNCTION
42605   6120        DECIMAL                ;DBRK
42606  20165        LDA     0,DAS
42607 101014        SKZ     0,0            ;IS THERE AN INTEGER PORTION?
42610    404        JMP     DOINT          ;   YES
42611  20035 STZRO: DOSZERO
42612   6120        DECIMAL                ;SZRO
42613    405        JMP     DOZER

42614 102520 DOINT: DOLOAD                 ;<<FROM INTF
42615 126520        DTSINTEGER
42616  30765        LDA     2,INTF-1       ;ROUND UP
42617   6120        DECIMAL
                                           ;ADD DB TO DA (DA IS SET TO 0.0)
42620  20003 DOZER: DOADD
42621  24025        DTUNPACKED
42622  30176        LDA     2,.DB
42623   6120 FDEC:  DECIMAL                ;<< FROM FRAF, NOTF
42624   2240        JMP     @.EVP4


42625 101015 SGNF:  SNZ     0,0            ;SIGN FUNCTION
42626   2240        JMP     @.EVP4
42627  30165        LDA     2,DAS
42630  20036        DOSONE
42631   6120        DECIMAL                ;SONE
42632   6137        STORDA
42633 145000        MOV     2,1

42634  44165 ABSF:  STA     1,DAS          ;ABSOLUTE VALUE FUNCTION
42635    402        SKIP

42636  44164 MANF:  STA     1,DAC          ;MANTISSA PORTION FUNCTION
42637   6131        LOADDA                 ;<< FROM IXRF OR IXR1
42640   2240        JMP     @.EVP4
```

```
                ;                                << SI = R92RUNS6A;  BO = 18/A.RUN.8053! >>
42641    6245 CHFF:  JSR     @.FIXS     ;CHANNEL FUNCTIONS
42642   20530        LDA     0,D100
42643    6115        BINDIVIDE
42644  121000        MOV     1,0        ;Position Channel # for check to follow
42645   54322        STA     3,TSF      ;SAVE COMMAND NUMBER (0, 1 or 2)
42646    6101        CALL
42647  100002        CHKCHANNEL
42650    6267        RUNERROR
42651  130510        61*K+SE
42652   50521        STA     2,CHFF2-1;Save channel address
42653   20322        LDA     0,TSF
42654  105005        MOV     0,1,SNR    ;NUMBER OF RECORDS FUNCTION ?
42655     533        JMP     CHFF1      ;   YES
42656   25003        LDA     1,STS,2
42657  101224        MOVZR   0,0,SZR    ;CURRENT RECORD NUMBER FUNCTION ?
42660     514        JMP     CHFF2      ;   NO
42661   34510        LDA     3,C20K     ;   YES
42662  167414        AND#    3,1,SZR    ;CONTIGUOUS ?
42663     410        JMP     CHFF0      ;   YES
42664   35004        LDA     3,FSZ,2    ;   NO
42665   20405        LDA     0,VFMMSK
42666  107414        AND#    0,1,SZR    ;Non-data file or in OPENMAINT mode?
42667   35002        LDA     3,CBN,2,   ;   Yes, Use current block nbr as record number
42670  165000        MOV     3,1
42671     543        JMP     FLOTP

42672   14000 VFMMSK:       14000       ;Not-Formatted and OPENMAINT status bits in CHANNEL.STS

42673   21007 CHFF0:LDA     0,CLP,2    ;This file open as
42674  101213        SKO     0,0        ; a Polyfile ??
42675     406        JMP     CHF0A      ;   No
42676   25002        LDA     1,CRN,2    ;   Yes, get record number (lower 16 bits) and
42677   31006        LDA     2,CRO,2    ;get record number (upper 16 bits).
42700  151415        INC#    2,2,SNR    ;Record # = -1 ?
42701    2454        JMP     @XCHF4     ;   Yes, return the -1
42702     555        JMP     CH1P3      ;   No, Do double precision float
```

```
                ;                              << SI = R92RUNS6A; BO = 18/A.RUN.8053! >>
                ;NON POLYFILE - GET LAST RECORD ACCESSED
                ;COMPUTES THE NUMBER OF THE LAST RECORD
                ;ACCESSED IN A CONTIGUOUS OR INDEXED FILE
                ;ON ENTRY : (A2)=A(CHANNEL)

 42703   25032 CHF0A: LDA     1,CBN,2   ;FETCH CURRENT DISPLACEMENT INTO FILE BLKS<<FROMCHFF0
 42704   21005        LDA     0,WPR,2   ;# WORDS PER RECORD
 42705  123032        ADDZ#   1,0,SZC   ;HAS FILE EVER BEEN ACCESSED?
 42706    2447        JMP     @XCHF4    ;   NO, RETURN A -1
 42707    4455        JSR     CHFFLT    ;   YES, FLOAT CBN
 42710   24002        DTF2WORD          ;LDA 1,C2
 42711    4445        JSR     CHFSTO    ;STORE FLOATED CBN IN TSF
 42712   24005        LDA     1,C400    ;Pick up size of a block
 42713    4451        JSR     CHFFLT    ;and float it.
 42714   30253        LDA     2,.TSF    ;POINT AT CBN
 42715   24002        DTF2WORD          ;LDA 1,C2 (ALLOW 10 DIGIT PRECISION)
 42716   20005        DOMULTIPLY        ;LDA 0,C5
 42717    6120        DECIMAL;6120 (COMPUTE CBN*256)
 42720   24003        DTF3WORD          ;THEN MOVE RESULT TO TSF
 42721    4435        JSR     CHFSTO    ;AS 3 WORD PERSISION
 42722   20006        LDA     0,C777    ;GET DISPLACEMENT TASK
 42723   30150        LDA     2,CHFF2-1;RECOVER A (CHANNEL)
 42724   25003        LDA     1,STS,2   ;PICK UP FILES CURRENT BYTE POINTER
 42725  107620        ANDZR   0,1       ;ISOLATE THE DISPLACEMENT INTO CURRENT BLOCK(IN WORDS)
 42726    4436        JSR     CHFFLT    ;FLOAT IT IN DA
 42727   30253        LDA     2,.TSF    ;POINT TO (CBN*256)
 42730   20003        DOADD   ;LDA      0,C3
 42731   24003        DTF3WORD          ;LDA 1,C3
 42732    6120        DECIMAL;6120 [CBNX256]+DISPL
 42733   30253        LDA     2,.TSF    ;POINT TO TSF1 AGAIN
 42734   24003        DTF3WORD          ;SAVE ((CBN*256)+DISPL INT BLK)
 42735    4421        JSR     CHFSTO
 42736   30135        LDA     2,CHFF2-1;RECOVER CHANNEL ADDR.
 42737   25005        LDA     1,WPR,2   ;AND PICK UP WORDS/REC
 42740    4424        JSR     CHFFLT    ;FLOAT IT IN DA
 42741   20024        DODIVIDE          ;LDA 0,C4
 42742   24003        LDA     1,C13     ;3 WORD REVERSE DIVIDE
 42743   30253        LDA     2,.TSF    ;POINT AT TSF
 42744    6120        DECIMAL;[(CBN*256)+DISPL]/WPR
                ;AT THIS TIME 8 DA CONTAINS THE RECORD #
                ;WHICH WAS LAST ACCESSED ACCORDING TO THE CHANNEL IN FORMAT
 42745   20042        DOBREAK;LDA 0, C20
 42746    6120        DECIMAL;CHECK FOR OFF-BOUNDARY ACCESS
 42747   34160        LDA     3,DA
 42750  175014        SKZ     3,3       ;ON RECORD BOUNDARY?
 42751    6142        TRAPFAULT         ;   NO, COMPLAIN
 42752  133410        67*K!NOP
 42753    2401        JMP     @.+1      ;   YES, MOVE VALUE INTO
 42754   42611        STZRO

 42755   43115 XCHF4: CHRF4
```

```
                ;                              << SI = R92RUN86A:  BO = 18/A.RUN.8053! >>

                ; STORE DA IN TSF1 THROUGH TSF5
                ; ON ENTRY A1 CONTAINS THE NUMBER TYPE
                ; PACKS THE NORMALIZED NUMBER AT DA IN TSF THRU TSF5

42756  54405 CHFST: STA     3,CHFSR   ;REMEMBER CALLER
42757  30253         LDA     2,.TSF    ;POINT TO TSF
42760 102400         DOSTORE
42761   6120         DECIMAL
42762   2401         JMP     @CHFSRX   ;RETURN (DA) IS NOW IN TSF THRU TSF5

42763      0 CHFSR: 0        ;RETURN ADDRESS STORED


                ;FLOAT THE 16 BIT INTEGER IN A1 IN DA WITH SIGN POSITIVE

42764  54777 CHFFL: STA     3,CHFSR   ;REMEMBER CALLER
42765 102400         SUB     0,0       ;FORCE SIGN POSITIVE
42766   6122         FLOAT
42767   2774         JMP     @CHFSRX   ;AND RETURN

42770  10000 C10K:  10000
42771  20000 C20K:  20000
42772    144 D100:  144

42773      0         0       ;CHANNEL ADDRESS DEPOSITED HERE
42774 101234 CHFF2: MOVZR#  0,0,SZR   ;CHF(X) where X >= 400 ?
42775   6267         RUNERROR          ;   Yes, error
42776 116110         34*K!SE
42777  20066         LDA     0,C777    ;DISPLACEMENT INTO RECORD FUNCTION
43000 107403         AND     0,1,SNC   ;Mask for displacement, Record length function ?
43001    403         JMP     CHPF2     ;   No
43002  25005         LDA     1,WPR,2   ;   Yes
43003    431         JMP     FLOTP
```

```
                        ;                                  << SI = R92RUNS6A;  BO = 18/A.RUN.8053! >>
43004   21007 CHPF2:LDA     0,CLP,2
43005  101212         SKE     0,0         ;Open as a Polyfile ?
43006   25004         LDA     1,CRD,2     ;  Yes, get current record displacement
43007     425         JMP     FLOTP

                    ;Function to return number of data records in file.

43010   21000 CHFF1:LDA     0,FLU,2     ;Get LU of file from DFT
43011  101112         SSP     0,0         ;Device file?
43012     422         JMP     FLOTP       ;  Yes - done.
43013    6102         FLAGCHECK           ;  No - check channel mode
43014   10003         STS!SKIPZ
43015    4000         4000                ;Channel in Maintenance Mode?
43016    6247         RUNERROR            ;  Yes - can't honor this request
43017  147410         117*K!NOP           ;"Illegal Device Operation !! "
43020   30753         LDA     2,CHFF2-1;Get A(DFT)
43021   21000         LDA     0,FLU,2     ;Get LU and
43022   25001         LDA     1,FDA,2     ;RDA of file's header block
43023   30010         LDA     2,.BSA
43024    6135         READBLOCK
43025   30746         LDA     2,CHFF2-1;Get A(DFT)
43026    6102         FLAGCHECK
43027   10007         CLP!SKIPZ
43030       1         PFM                 ;File open in Polyfile Mode?
43031     406         JMP     CHF1P       ;  Yes - go sum all data volumes
43032   30010         LDA     2,.BSA      ;  No - just use NRCD from header
43033   25016         LDA     1,NRCD,2;Return number of records
43034  102400 FLOTP:SUB     0,0         ;Make sign positive
43035    6122 FFLOT:FLOAT
43036    2240         JMP     @.EVP4

43037  152400 CHF1P:SUB     2,2         ;Clear
43040   50323         STA     2,TSF1      ;  low
43041   50324         STA     2,TSF2      ;  and high record count
43042   24010 CHFP2:LDA     1,.BSA      ;A(Master volume header block)
43043  102620         SUBZR   0,0         ;Volume type 4 (Data)
43044    6101         CALL                ;Scan for next data volume
43045  100001         VOLFIND
43046     407         JMP     CH1P2       ;No more volumes
43047   20323         LDA     0,TSF1      ;Low count
43050   35001         LDA     3,BNR,2     ;# of records in volume
43051  163022         ADDZ    3,0,SZC     ;Add in # records in this volume
43052   10224         ISZ     TSF2        ;  and handle overflow from l.o. word
43053   40323         STA     0,TSF1      ;Save low order count
43054     766         JMP     CHFP2

43055   24323 CH1P2:LDA     1,TSF1      ;L.O.  Count
43056   30324         LDA     2,TSF2      ;H.O  Count
43057  102400 CH1P3:SUB     0,0         ;Force positive sign
43060   34122         LDA     3,FLOAT&377
43061    5777         JSR     -1,3        ;Double precision float
43062    2240         JMP     @.EVP4
```

```
                        ;                               << SI = R92RUNS6A; BO = 18/A.RUN.8053! >>
 43063    6121  IXRF:   FIX                             ;INTEGER EXPONENT OF RADIX FUNCTION
 43064     415          JMP     IXR1
 43065  131422          INCZ    1,2,SZC
 43066     413          JMP     IXR1
 43067  101015          SNZ     0,0                     ;NEGATIVE ARGUMENT ?
 43070     403          JMP     .+3                     ;   NO
 43071  130400          NEG     1,2                     ;   YES
 43072  151400          INC     2,2
 43073   20036          LDA     0,C16
 43074    6120          DECIMAL                         ;SONE
 43075    6137          STORDA
 43076   50164          STA     2,DAC
 43077    6131  IXRDN:  LOADDA
 43100    2240          JMP     @.EVP4

 43101  111000  IXR1:   MOV     0,2
 43102   20037          LDA     0,C17
 43103    6120          DECIMAL                         ;SPIN
 43104    6137          STORDA
 43105   20023          LDA     0,CM400
 43106  151014          SKZ     2,2
 43107   40164          STA     0,DAC
 43110     767          JMP     IXRDN


 43111   24164  CHRF:   LDA     1,DAC                   ;CHARACTERISTIC PORTION FUNCTION
 43112  135102          MOVL    1,3,SZC
 43113  124400          NEG     1,1
 43114  102561          SUBCL   0,0,SKP
 43115  126520  CHRF4:  SUBZL   1,1
 43116     717          JMP     FFLOT
```

```
                        ;                                  << SI = R92RUNS6A; BO = 18/A.RUN.8053! >>
 40117       25                 DET.
 40120    30356  DETF:  LDA     2,.BUS     ;DETERMINANT VALUE FUNCTION
 40121    20776          LDA     0,.-2
 40122   113000          ADD     0,2
 40123    24024          LDA     1,C4
 40124   102520          SUBZL   0,0
 40125     6120  JFDEC: DECIMAL
 40126     2240          JMP     @.EVP4

 40127    34366  RNDF:  LDA     3,.BUS     ;Random Number Function
 40130    25445          LDA     1,RNDI.,3;Load current "SEED"
 40131   102520          SUBZL   0,0        ;Load function nbr to generate random nbrs?
 40132     6101          CALL               ;And then do so
 40133      150            PRAND
 40134    34366          LDA     3,.BUS
 40135    45445          STA     1,RNDI.,3;Save new "SEED"
 40136     2240          JMP     @.EVP4     ;Random Nbr is in DA


          241           .LOC    EL-.        ;PATCH SPACE (OVERFLOW CHECK)

                ;*************************************************
                ;* NOTE:  FOR PATCHES TO TAPES #5 AND #6 ONLY.  *
                ;*************************************************

               .EOT   ;  "RUN" R9.0 SOURCE #6
```

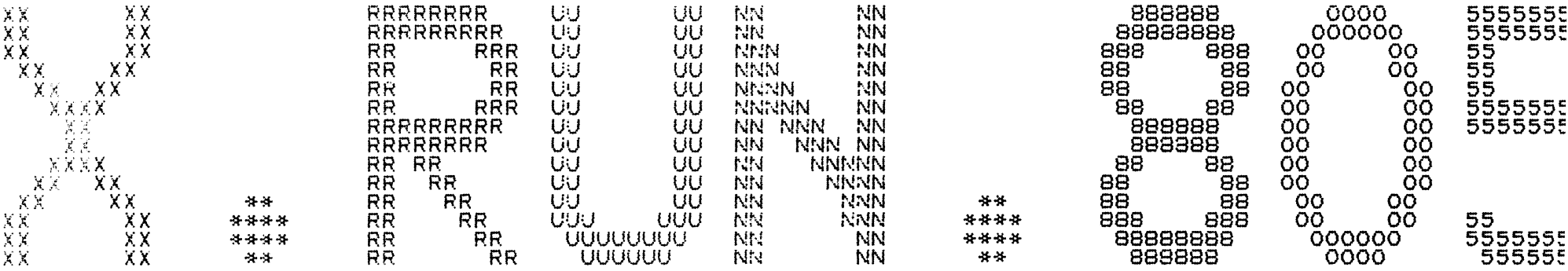
```
                ;                              << SI = R9ORUNS7A;  BO = 18/A.RUN.8053! >>

                ;"RUN" SOURCE #7 OF 7 FOR "IRIS" R9.0

                ;PICO-N TEST7:  TEST PICO-N FAULT FLAG #2
                ;4-10-80
                ;   May 24, 1983  (3221)  TWM  Convert to R8.2
                ; EDITED FOR R9.0 (4275) BY MLR

                ;PERFORMED EVERY 1 TO 4TH TIME THE 'RUN' PROCESSOR
                ;IS LOADED FROM THE 'RUN' FILE.
                ;IF THE FAULT FLAG IS ON, RESET ERROR BRANCHING &
                ;SET THE ESCAPE FLAG.
                ;THE PICO-N FAULT FLAG #2 IS SET BY 'PNTEST6B'.

        43400           .LOC    EL

43400 1777/7            -1
43401  447/7 PNT7:      STA     1,.-1
43402  30100            LDA     2,.INFO
43403  21041            LDA     0,TSC.,2
43404  30003            LDA     2,C3
43405 113100            AND     0,2             ;BITS 0-3
43406 103100            ADDL    0,0
43407 103100            ADDL    0,0
43410 101300            MOVS    0,0
43411  24003            LDA     1,C3
43412 123100            AND     1,0             ;BITS 4-7
43413 112404            SUB     0,2,SZR         ;TIME TO CHECK THE FLAG ?
43414    413            JMP     T7X             ;   NO
43415  30100            LDA     2,.INFO         ;   YES
43416  31067            LDA     2,.CTT.,2       ;A(CRST)
43417  31002            LDA     2,2,2           ;A(FLSHX)
43420  31377            LDA     2,-1,2          ;FAULT FLAG
43421 151113            SSN     2,2             ;FLAG SET ?
43422    405            JMP     T7X             ;   NO
43423 102400            SUB     0,0             ;   YES
43424  30046            LDA     2,.BUS
43425  41062            STA     0,ERBP.,2       ;RESET ERROR BRANCHING
43426  50073            STA     2,ESCF          ;SET ESCAPE FLAG
43427   2401 T7X:       JMP     @.+1            ;RETURN
43430  36450            QT.1

                        .END    ;"RUN" R9.0 SOURCE #7
```

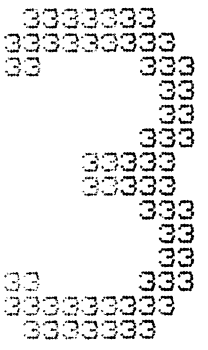
ABREL	33446	ABSF	42634	ACNB	32554	ADIM	41062	ADIM1	41077		
ADIM2	41125	ADIMA	41152	ADIMD	41166	ADIML	41156	ADIMN	41215		
ADIMS	41147	ADML	41070	AE	100210	ALGEX	6257	ALOC	40770		
ALOC0	41045	ALOC1	41005	ALOC2	41014	APEND	34711	APM	41060		
ATF	350	ATFH	34716	BADST	33460	BDMSK	32523	BDSET	32524		
BILDF	37602	BINDI	6115	BINMU	6116	BPI	16	BRDEF	200		
BSACF	75	BSCRV	134261	BUMPU	6117	C10	30	C100	51		
C1000	67	C10K	42770	C11	31	C114	33057	C12	32		
C125	211	C13	33	C130	33061	C136	212	C14	34		
C143	36765	C15	35	C16	36	C160	174	C163	175		
C166	176	C17	37	C17CK	21	C171	177	C173	33060		
C174	41521	C177	52	C1777	70	C2	2	C20	42		
C200	53	C2000	71	C205	54	C20K	42771	C215	55		
C240	56	C244	57	C254	36130	C260	60	C271	61		
C3	3	C30	210	C300	62	C307	32522	C334	63		
C336	213	C340	214	C342	215	C343	33043	C344	216		
C347	217	C351	220	C352	34413	C353	34414	C355	221		
C357	34415	C360	222	C361	223	C362	224	C364	41703		
C365	225	C366	226	C37	43	C370	41522	C371	37720		
C375	227	C376	230	C377	64	C4	24	C40	44		
C400	65	C4000	72	C5	25	C52	36127	C6	26		
C600	100	C7	27	C77	50	C774C	22	C777	66		
CALL0	40406	CALL	6101	CALL2	40404	CALL3	40424	CALLC	40363		
CALLD	40521	CALLE	40535	CALLN	40475	CALLO	40376	CALLS	40440		
CALLX	40512	CECOD	36343	CERRF	33554	CESF	37010	CF144	42363		
CF152	42364	CH1P2	43055	CH1P3	43057	CHAER	35376	CHAI1	35403		
CHAI2	35433	CHAI3	35434	CHAI4	35435	CHAI5	35436	CHAI6	35437		
CHAI7	35326	CHANN	6106	CHAPR	36644	CHFOA	42703	CHF1P	43037		
CHFF	42641	CHFFO	42673	CHFF1	43010	CHFF2	42774	CHFFL	42764		
CHFP2	43042	CHFSR	42763	CHFST	42756	CHKSC	36032	CHNCK	35424		
CHNNX	35422	CHPF2	43004	CHRF	43111	CHRF4	43115	CKEM	32703		
CLAL1	40010	CLALL	40004	CLCHN	40021	CLCNT	40022	CLOSF	37734		
CLOSX	37751	CM400	23	CNODE	371	CPYFN	37756	CSEN	32352		
CSENT	32363	CTCEN	32203	CTLC	32505	CTLC2	32514	CVSP	32415		
CVSP1	32467	CVSP2	32501	CVSP3	32477	D100	42772	DA	160		
DAC	164	DAS	165	DATAP	6110	DB	166	DBA	41		
DBC	172	DBS	173	DECIM	6120	DEC.	21	DEF	34642		
DETF	43120	DET.	25	DFA	40737	DFN	40754	DFP	40740		
DFS.	31	DFTCA	34106	DFV	40745	DIM	35253	DIM1	35277		
DIM2	35316	DIM3	35321	DMCAL	34110	DOINT	42614	DOZER	42620		
DPS1	34717	DPS2	34720	DQUEU	6105	DSC.	22	DWC.	20		
D.BSC	201	D.EN	377	D.MAT	202	EDSTX	33573	EL	43400		
EMODE	20	EN	43450	END	33717	ENDA	33731	ENDA0	32602		
ENDA1	33732	ENDAA	33727	ENDST	33566	ERBP.	52	ERLN.	53		
ERPBO	365	ERR	32603	ERR2	32645	ERR4	32700	ERR6	34144		
ERRF	76	ERRF1	42556	ERRFN	42520	ERRFU	42570	ERRN.	51		
ERRST	206	ERRSY	204	ESCF	73	ESCP	37023	ESCP1	37046		
ESCP2	37047	ESCPB	37024	ESTOP	37065	ETSF	74	EUS.	10		
EV0	41273	EV1	41277	EV2	41300	EV3	41306	EV3J	41675		
EV5	41400	EV6	41704	EV7	41716	EVA	42120	EVAX	41255		
EVC	41341	EVCHB	33147	EVCHC	33167	EVCHD	33134	EVCHE	33141		
EVCHF	33210	EVCHG	33211	EVCHS	33062	EVCHX	33063	EVCPE	33103		
EVD	41660	EVDON	34631	EVEX	41263	EVF	42065	EVG	42232		
EVG0	42252	EVG1	42264	EVG2	42271	EVG3	42274	EVGS	42217		
EVIN	40260	EVL	41523	EVLF	300	EVN	42111	EVO	41360		
EVP	41532	EVPO	41572	EVP1	41576	EVP2	41623	EVP3	41636		
EVP4	41647	EVPF	277	EVPI	41655	EVQ	42162	EVQ1	42200		
EVR	41507	EVS	41735	EVSO	41746	EVS1	41754	EVS2	41773		

EVSD0	42033	EVSD1	42037	EVSD2	42052	EVSE	34605	EVSE1	34626
EVSED	42024	EVSR	42011	EVSU	41264	EVSW	274	EVV	41443
EXLIN	33542	EXPN	42457	EXPRE	6260	EXST1	33627	EXST2	33622
EXST5	33653	EXST6	33713	EXSTM	33605	FDEC	42623	FDEFS	200
FDLIM	37424	FDPDN	34012	FDPGE	34006	FDPLC	33757	FDPNX	33773
FDPRT	34021	FDSTP	37425	FESIZ	14	FESX	34566	FFLOT	43035
FINDL	6123	FINMR	36770	FIVRD	0	FIX	6121	FIXS	40701
FL1.	54	FL2.	56	FL3.	57	FL4.	60	FLAGC	6102
FLAG.	24	FLOAT	6122	FLOTP	43034	FLUS.	36462	FNC	37412
FNDS	32243	FNDS1	32245	FNDS2	32247	FNDS.	33470	FNLIM	2
FNPBC	6	FNPLC	1	FNS	177	FNSPR	13	FNSS	10
FNSTP	7	FNS.	14	FONXT	37406	FOR	37203	FOR1	37214
FOR2	37224	FOR3	37227	FOR5	37275	FOREN	37375	FORER	37400
FORLD	37413	FORN	37234	FORPB	37423	FORS1	37341	FORSC	37352
FORSX	37344	FORSL	37350	FORST	37414	FRAF	42600	FREEN	6107
FSC.	15	FSESZ	37337	FUNC	42317	FUNCB	42365	FUNCE	42331
GBUMP	40115	GCP	36275	GCP1	36300	GETBY	6124	GFLAG	36537
GIBP	36535	GO1	36553	GO2	36563	GO3	36601	GO4	36606
GO5	36625	GO6	36636	GOGO	6262	GOGOQ	36540	GOGO.	36541
GOSUB	34214	GOT3	36536	GOTO	34230	GOTO1	34256	GOTO2	34273
GOTO3	34276	GOTO4	34301	GOTOL	34303	GSC.	17	GSS	337
GSSS	10	GSS.	16	I	275	IF	34327	IF1	34352
IF1A	34355	IF2	34416	IFC	34472	IFD	34527	IFE	34540
IFF	34535	IFFIX	34306	IFGB	34560	IFL	34512	IFLW	343
IFQ	34434	IFR	34460	IFREI	34326	IFS	34447	IFS2	34451
IFT	34377	IFX	34552	IITAC	35773	INBYT	6125	INDON	36211
INEND	36221	INLEN	64	INP2	36014	INP2A	36017	INP2B	36046
INP2L	36064	INP2P	36072	INP2T	36105	INP3	36114	INP3B	36125
INP4	36131	INPBT	36263	INPL1	36156	INPLI	36142	INPMN	36024
INPNL	36252	INPS2	36060	INPSF	36054	INPST	36205	INPTS	36224
INPUT	36000	INSTB	6126	INTEG	6263	INTF	42604	IOCAL	34103
IOP	6	ISA2D	6127	ISA2L	6130	ITAC	345	IXR1	43101
IXRDN	43077	IXRF	43063	J	276	JFDEC	43125	JFLTO	151
JPRND	35647	JPRTF	35610	KILL	37772	LDONE	35036	LENF	42462
LENFL	42476	LET	34661	LEIS	34721	LETS1	35056	LETS3	35203
LETS4	35244	LETSA	34745	LFISC	35073	LETSD	35076	LETSS	34733
LETSV	35156	LETSV	35037	LEIU6	35174	LETV1	35136	LETV2	35151
LETVB	35104	LETX	34676	LOADD	6131	LREV	40700	LUDV	40731
LUVSB	40710	MANF	42636	MAT	36655	MAT1	36671	MAT13	36674
MATSB	36742	MATRW	36732	MATS	36755	MOST	37073	MOST1	37112
MOST2	37115	MOST3	37151	MOST5	37171	MOST6	37176	MOSTA	37126
MOSTB	37133	MOVST	35441	MTOST	6261	MXPRC	4	NEXT	37427
NEXTO	37435	NEXT1	37450	NEXT2	37501	NEXT3	37510	NEXT4	37517
NEXT5	37515	NEXTB	6264	NEXTI	37527	NOTBS	32323	NOTF	42573
NOTTO	35031	NVS.	7	NXLIN	33536	NXSTM	33562	NXTIO	37546
NXTIL	37554	NXTI2	37572	ON	34147	ONDO	34204	ONSKP	34175
OPENF	37601	OPNBC	37726	OPNCC	37723	OPNCM	37640	OPNDN	37667
OPNGO	37623	OPNMN	37662	OPNNE	37675	OPNNF	37706	OPNNX	37610
OPNOL	37646	OPNSC	37625	OUTBY	6132	OUTTE	6133	PARMT	477
PBCS	347	PBC.	3	PCTBL	400	PIB	4	PLC.	2
PNT7	43401	PNT7T	36447	PRDON	35666	PREVN	477	PRINT	35466
PRNDS	35605	PRNS1	35740	PRNT3	35700	PRNTA	35501	PRNTB	35652
PRNTC	35667	PRNTD	35552	PRNTE	35526	PRNTL	35710	PRNTN	35614
PRNTR	35650	PRNTS	35732	PRNTT	35751	PRTCF	35762	PRTN1	35631
PRTN2	35634	PRTNF	35531	PRTPC	35726	PRTSC	35727	PSTS.	55
PTCF1	35770	PTEOL	35716	PUIBY	6134	QB1	36331	QB2	36345
QBRTN	351	QBYTE	6265	QRYT.	36322	QCHAR	6103	QDAT	36432
QPAT	36417	QPT1	36430	QSP	36360	QSPO	36376	QSP1	36413

QST	36434	QTXT.	6266	QT.1	36450	QT.2	36457	QT..	36442
QUEUE	6104	READA	33471	READB	6135	READD	33472	REEAD	33230
RELJM	6136	RES	447	REST	32225	RESTR	32227	REVNO	456
REV	0	RFBA	272	RJSR	6136	RLBA	273	RMCON	35440
RNDF	43127	RNDGC	36344	RNDI.	45	RNDM	34633	RNMRT	36766
RODBP.	23	RRSZ	370	RSVTB	476	RTNBC	34072	RTNBD	34107
RTNBE	34112	RTNBK	34064	RTNER	34140	RTNS1	34124	RTNSF	34133
RTNSK	34131	RTP	7	RTRN	34024	RTRN1	34052	RUN1	32217
RUNS	32235	RUNER	6267	RUNF	42410	RUNRV	1000	RUP	5
RWIB	33240	RWIBO	33252	RWIB1	33277	RWIB2	33304	RWIB3	33274
RWIC	33341	RWIC1	33351	RWIC2	33360	RWIC3	33371	RWIC4	33377
RWICS	33404	RWICE	33413	RWIQ	33315	RWIQ2	33306	RWIRT	33427
RWIS	33331	RWIS1	33334	RWIS2	33336	SATUP	420	SBA	40
SBP	34713	SCDCA	34147	SE	100110	SER	34710	SERCL	40344
SETSD	6271	SFN	42422	SFS	34715	SGN2A	40171	SGNF	42625
SGNLA	40030	SGNSD	40043	SIG1A	40121	SIGN1	40063	SIGN2	40165
SIGN3	40200	SIGNE	40252	SIGNL	40024	SIGNN	40231	SIGNS	40241
SIGNV	40130	SIGNX	40216	SKRST	32713	SKSCS	33044	SKSGO	32777
SKSNL	33012	SKSNO	32742	SKSNS	32721	SKSSF	33030	SKSSR	33026
SKSST	33053	SKST1	32756	SKST2	32764	SKSTM	32720	SKSTS	32772
SLT	457	SLT.	5	SNO	32560	SNOW	33755	SPCF	42560
SPINP	6146	SRCH	40271	SRCH1	40324	SS	34714	STINP	6140
STINT	6147	STOP	33746	STORD	6137	STOUT	6141	STPBC	63
STPRC	24	STZRO	42611	SVIB	32240	SWOOP	455	SWPI	403
SWPI2	32306	SWPO	457	SWPOK	402	T1	340	T2	342
T3	344	T4	346	T7X	43427	TASKQ	15	TBYTE	34712
TRAPF	6142	TS	301	TS1	302	TS2	303	TS3	304
TS4	305	TS5	306	TS6	307	TS7	310	TSC.H	40061
TSE	330	TSE1	331	TSE2	332	TSE3	333	TSE4	334
TSE5	335	TSE6	336	TSE7	337	TSF	322	TSF1	323
TSF2	324	TSF3	325	TSF4	326	TSF5	327	TSN	353
TSN1	354	TSN10	363	TSN11	364	TSN2	355	TSN3	356
TSN4	357	TSN5	360	TSN6	361	TSN7	362	TSNS	11
TSN.	103	TSP	340	TSP1	341	TSP10	350	TSP11	351
TSP12	352	TSP2	342	TSP3	343	TSP4	344	TSP5	345
TSP6	346	TSP7	347	TSU.	73	TSU.1	74	TSU.2	75
TSU.3	76	TSU.4	77	TSU.5	100	TSU.6	101	TSU.7	102
TSX	311	TSX1	312	TSX10	321	TSX2	313	TSX3	314
TSX4	315	TSX5	316	TSX6	317	TSX7	320	UFC.	13
UFN	42412	UFS	146	UFSS	5	UFS.	12	UFT	114
UFT.	11	UPS.	4	UVS.	6	VBLDF	37717	VBLDT	37716
VDT	1	VEMD1	33445	VFLDP	36767	VFMMS	42672	VLOC	41230
VLOCS	41245	VMINU	35611	WAIT	36640	WR1TE	33235	WRITB	6143
WRTN	352	XCHF4	42755	XFL1	36507	XFL1A	36517	XFL2	36530
XFLUS	6270	XFLU.	36467	XGETB	6144	XNDCH	40020	XOB	347
XOBE	62	XOBS	100	XOB.	61	XPUTB	6145	XSGNE	40062
.ABA	14	.ADIM	233	.ALOC	41520	.BPS	77	.BRKP	150
.BSA	10	.BUS	366	.CESF	234	.CSEN	235	.CTCE	37007
.DA	174	.DA3	175	.DB	176	.DB3	177	.EDST	246
.EL	231	.EN	232	.EUS	367	.EV1	244	.EV2	41700
.EVCH	236	.EVCS	40270	.EVG	237	.EVGS	42064	.EVP4	240
.EVQ	241	.EVSE	242	.EVSU	243	.EXLN	32237	.EXPN	41701
.EXST	375	.FDPL	37426	.FIXS	245	.FLTO	152	.FMRD	372
.FNXT	37336	.FUNC	41702	.GCP	35725	.GGOQ	36341	.HBA	11
.HXA	12	.IEND	36102	.INFO	100	.INP2	36103	.INTR	111
.ITAC	36104	.LCAL	40534	.LCM	114	.LETX	35103	.LTVS	34707
.LUTB	35154	.LUVD	251	.MOST	34706	.NRET	112	.NXLN	247
.NXST	374	.OPNB	37725	.OPNC	37721	.PRCF	35612	.PRMT	203

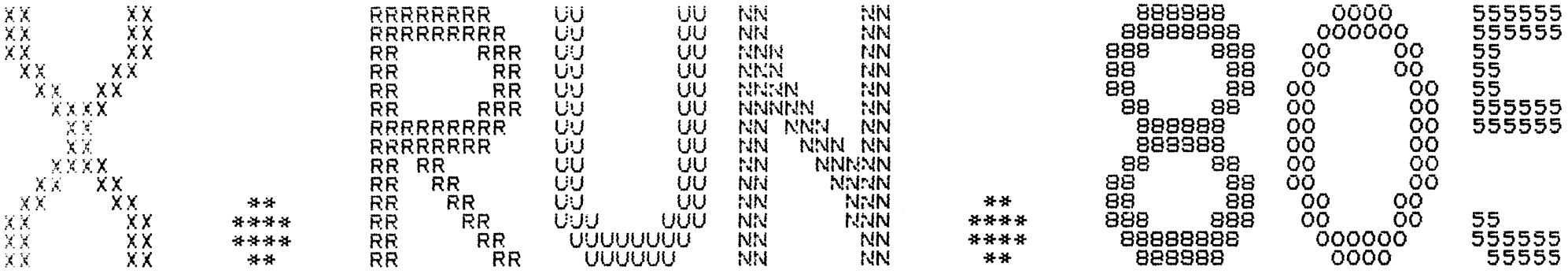
```
---

.PRTS   35613    .PRTT   35706    .PTBL      65    .RUN2     373    .RWIC   36764
.SKRS   37340    .SKST     250    .SRET     113    .SSA       13    .TBYT   37072
.TIME   40060    .TSE      254    .TSF      253    .TSN      255    .TSX      252
.VLOC     256    .WAIT   36342
```

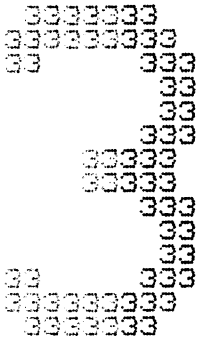




File: 18/X.RUN 8053 Printed: MAR 8, 1988 1:13:20
File size: 104 blocks Copy 1 of 1

Printed for RELEASE CONTROLL





File: 18/X.RUN.8053 Printed: MAR 8, 1988 1:13:28
File size: 101 blocks Copy 1 of 1

Printed for RELFASE CONTROLL

***** J O B S T A T I S T I C S *****

4	TOTAL # DUPLICATE KEYS
0	TOTAL # DIR. RE-CRCS
3,027	TOTAL # KEYS INSERTED
0	TOTAL # ASSEMBLY ERRS

.ADIM 3.030: 52.056 112.037

.ALOC 105.041 107.006:

.BSA 124.024 124.031 124.040

.BUS 6.043: 8.011 9.035 10.034 10.049 10.063 14.012
16.009 17.011 17.036 17.048 18.008 18.015 18.017
19.013 19.025 19.045 20.060 21.040 22.007 22.022
22.037 23.026 23.035 26.047 27.009 28.045 28.053
30.009 30.021 30.031 30.048 31.009 31.015 32.012
32.023 32.030 32.035 33.009 33.023 34.009 34.033
35.021 36.034 36.050 37.014 37.021 37.052 38.022
38.037 40.028 40.045 41.031 42.025 42.047 43.018
43.032 43.041 43.054 44.038 45.014 46.022 50.008
52.029 52.038 53.006 53.034 54.018 54.031 56.024
56.035 57.055 58.007 58.009 58.024 59.011 59.017
59.052 61.030 62.034 62.049 63.035 63.050 65.033
65.042 66.041 66.047 68.009 69.008 69.025 70.013
70.030 71.048 72.035 73.040 74.024 74.028 74.042
74.057 77.010 77.032 78.007 78.039 78.043 79.026
79.033 79.043 79.054 79.056 81.008 81.056 82.009
82.029 84.041 90.016 90.024 91.025 92.022 92.031
95.018 96.008 97.030 98.010 98.017 98.031 98.049
98.056 99.010 100.035 100.059 101.010 101.019 101.031
101.036 102.024 104.006 106.027 109.047 110.053 112.015
112.040 117.007 117.015 119.010 119.016 119.050 126.007
126.015 126.020 127.042

.CERF 3.031: 28.043 32.033 59.010

.CSFN 3.032: 28.029 28.034 84.017 84.024 84.051 86.014
86.044

.CTCL 74.007: 74.016

.CTL. 127.036

.DB 120.035

.EDST 3.041: 26.053 26.062 45.016 48.051 49.042 52.061
59.020 62.033 65.007 65.025 84.033 86.019 86.049
88.034 89.024 89.030 91.033 92.025 94.012 96.024

.EL 3.028: 53.020 68.019 103.020 105.016 105.031 106.012
113.040

.EN 3.029: 103.022 103.031

.EUS 6.044: 8.024 96.009 102.028

```
.EV1        3.039:  113.017  118.037
.EV2      109.009  109.023:  109.038
.EVCH       3.033:   56.015   72.016   72.022
.EVCS      93.006:   93.007
.EVG        3.034:  107.051  109.046
.EVGC     110.018  110.025  111.048:
.EVP4       3.035:  116.024  118.008  119.056  120.037  120.041  120.053
          124.035  124.057  125.019  126.013  126.022
.EVO        3.036:  107.045  118.014
.EVSC       3.037:   50.015   51.013   53.016   56.033   86.027
.EVSU       3.038:   52.040
.EXLN      10.036   10.059   10.060:
.EXPN     108.045  109.024:
.EXSI       6.057:   19.034   37.045   37.050   38.043   41.042   41.045
           42.008   42.013   42.019   79.013   79.021   82.014
.FDPL      79.059   80.026:
.FIXD       3.040:   60.024  110.019  110.026  121.006
.FMRD       6.054:
.FNXT      77.006   79.015:
.FUNC     108.011  109.025:
.GCP       59.056:   59.057   62.042
.GGGG      66.046   66.055:
.IEND      62.014   63.006:
.INFO      12.007  127.024  127.035
.INPS      63.007:   64.047
.ITAC      63.008:   65.006
.LOAL      95.026   95.045   96.026:
```

```
.LETX      50.006:  50.044
.LTVB      46.042   46.045   46.061:
.LUTD      50.051:  51.052
.LUVD       3.044:  52.044   72.030  105.035  110.039
.MOST      46.060:  48.010   48.048
.NXLN       3.042:  41.044   42.014   42.020   53.014  119.042
.NXSI       6.056:  10.042   46.029   47.042   52.012   73.054   79.048
           81.060
.OPNB      84.048   85.017:
.OPNC      84.009   85.011:
.PRCT      57.025   57.060:
.PRMT       2.054:
.PRTC      57.010   57.061:
.PRTT      57.019   59.038:
.PTBL       7.007=  49.015   49.020   58.014   58.022
.RDX       13.017   13.045
.RUN2       6.055:
.RWIC      71.053   72.056   73.028:
.SVRS      78.041   79.017:  79.049
.SWST       3.043:  19.032   37.007   37.046   38.041   54.029   79.031
.TBYT      75.032:  75.044
.TIME      88.037:  89.012   89.016
.TSE        3.050:
.TSF        3.049:  94.023  122.020  122.031  122.035  122.043  123.012
.TSN        3.051:  27.037
.TSX        3.048:  26.041   26.045
```

.TXTF 10.019 12.025 17.044 34.023 34.049 63.041 65.048

.VLOC 3.052: 26.020 30.028 46.037 48.029 64.013 79.041
80.009 90.013 94.031

.WAIT 66.052 66.056:

ABN. 65.017 74.015

ABREL 28 047 28.054 29.018:

ABSF 115.023 120.048:

ACNB 4.023 17.036:

ADIM 3.030 100.007: 100.038 100.044

ADIM1 100.011 100.020:

ADIM2 100.042: 100.050 100.052

ADIMA 101.009: 101.021

ADIMD 101.017 101.023:

ADIML 100.027 101.014:

ADIMN 100.033 101.027 102.006:

ADIMO 101.006: 101.045

ADML 100.013: 100.019

AE 37.062 39.015 53.012 53.053 73.014 77.038 79.061
81.020 95.029 95.048 99.033 100.017 100.023 110.049
112.014 117.020

AFP. 12.016

AFSET 9.043

ALGEX 4.007= 24.016 36.024 38.013 41.018 45.010 60.022
78.012 78.020 78.032 92.049 95.039

ALLOL 34.032

ALOC 52.052 98.027: 98.028 98.048 99.007 99.015 99.017
99.025 101.014 107.006

ALOCC 98.051 98.055 99.023: 101.029 101.035

ALOC1 98.030 98.040:

ALOC2 98.039 98.047:

APEND 47.011: 47.032 47.037 47.044

APK 99.027 99.036:

ATF 17.028 56.013 61.028 66.012= 67.011 67.056 67.062
68.007

ATFH 47.019: 48.024 48.033

BADST 30.009: 33.020

BDMSK 17.006: 17.014

BDSET 4.038 14.034 17.010:

BILDF 33.034 83.014:

BINDI 121.008

BINMU 72.048 72.052

BNR 124.046

BPI 12.017

BPS 10.006 34.042

BSCRV 8.040

BUILD 84.050

BUMPU 28.052 32 049 73.017 89.029 89.037

C10 67.041 98.007 108.007

C100 20.047 33.016 103.050 104.046 108.015 114.012

C1000 119.024

C10K 123.027:

C11 21.013 27.025 29.021 51.051 58.013 71.051 95.011

C114 22 028 22.049:

C12 17.054 72.054 112.031

```
C125        3.009:   11.027   20.050   54.034   79.037
C13       122.042
C130       20.053   22.051:
C136        3.010:   33.017   53.009   72.012  115.015
C14        21.016   30.033   67.035   69.036   70.039   81.043   95.012
C143       73.031:  74.041
C15       120.007
C16        78.028  125.014
C166       62.054  103.056  106.017  111.042  115.012
C17        59.024  125.022
C173       21.019   22.050:
C174      103.053  107.008:  107.029
C177       21.022   66.017   67.021   97.028  107.032
C2         41.020   78.053   81.042  114.015
C20       114.016  120.013
C200       18.032   19.027   53.023   72.025  103.047  104.040
C204      121.022  123.028:
C215       34.028   53.025   59.006   64.031   65.015   65.037
C240       49.032   59.032   67.014
C254       63.033:  64.026
C3         28.048   81.032   88.009   89.006   89.026   91.018   92.027
           98.033  127.026  127.031
C30         3.008:  28.027   94.019
C300       11.030   97.047  102.007  107.014  110.040  117.033
C307       16.021   16.024:
C336        3.011:  52.023   72.026   97.025
```

```
C340        3.012:  103.027  103.039  104.035  104.043  104.056  109.041

C342        3.013:   20.057   21.008   21.046   22.018   32.018   36.025
           37.008   37.011   38.014   46.046   49.006   50.020   57.032
           59.049   62.031   64.017   65.023   78 033   92.038   94.044
          112.011  119.035

C343       22.031   22.033:

C344        3.014:  41.028   47.052   57.017   57.049   72.009   78.013

C347        3.015:  51.007   56.029   78.025

C351        3.016:  21.025   27.006   42.022   44.034   46.043   57.013
           57.038   63.018

C352       40.038   41.050:

C353       41.051:  42.006

C355        3.017:  41.039   46.019   47.025   50.012   83.042  102.031

C357       40.040   41.052:  42.016

C360        3.018:  24.011   24.044   25.013   25.029   56.041   57.027
           62.027   62.063   66.031

C361        3.019:  26 051   30.053   43.021   43.045   44.019   47.028
           47.034   52.058   65.026   66.027   72.058   75.045   83.034
           83.051   84.030   86.016   86.046   88.014   90.028   94.037
           95.034   95.040

C362        3.020:  57.043  104.022  104.052  105.024  109.014  111.027

C364      109.027: 109.032  110.008

C365        3.021:  57.020   57.044   62.038

C366        3.022: 104.029  108.046  109.010  111.031

C37        84.013

C370      104.059  107.009: 109.007

C371       84.038   85.009:

C375        3.023:  21.028   21.033   57.023   57.039   60.033   62.016
           62.021   66.020

C376        3.024:  48.013  103.060  107.021  107.048  108.009  111.037
```

```
C377        11.024   23.031   29.019   38.031   66.034   67.053   83.026
            86.007   93.044  103.044  104.025  104.049  119.027

C4          24.025   26.034   52.018   67.036   90.035   97.053   98.016
            98.045  117.039  126.010

C40         67.048

C4GO         7.007  104.018  107.012  108.012  108.037  113.011  114.009
           122 018

C5          26.046   41.015   41.021   51.010   51.048  108.025  112.017
           113.049  114.036  117.021

C52         63.010   63.031:

C6          50.034   64.022   90.008  113.026

C7          49.016   58.021

C77         24.032   24.035   28.022

C774C       17.021   18.020

C777       122.026  123 035

CAL2P       94.033:  94.060

CALL         7.014    9.042   12.046   16.014   17.056   19.015   23.044
            26.057   28.050   34.031   45.012   53.029   53.043   56.018
            71.018   71.038   86.042   88.030   89.027   90.036   91.015
            93.011   93.050   96.013  116.022  118 006  119.052  121.011
           124.042  126.018

CALL2       94.031:  94.041

CALLO       94.032   94.049:

CALLC       93.009   94.014:  96.007

CALLP       95.032   95.051   96.014:

CALLE       95.031   95.044   95.050   96.030:

CALLN       95.015   95.039:

CALLO       94.025:  94.030

CALLS       33.040   95.009:

CALLX       95.037   96.007:
```

```
CBN       121.028  122.011
CECOD      66.057:  67.025
CERRF      31.008   31.029:  31.043   32.017
CESF        3.031   74.009:
CF144     115.045: 116.014
CF152     115.046: 116.017
CH1P2     124.044  124.052:
CH1P3     121.041  124.054:
CHAER      53.031   53.049:
CHAI1      53.033   54.007:
CHAI2      54.037   54.039:
CHAI3      54.007   54.040:
CHAI4      54.022   54.030   54.041:
CHAI5      54.011   54.042:
CHAI6      54.008   54.043:
CHAIN      33.039   53.006:
CHANL      27.043   84.015   84.022   84.049   86.012   87.015
CHAPR      69.015   70.020   71.046:
CHFOA     121.036  122.011:
CHF1P     124.030  124.037:
CHFF      115.043  121.006:
CHFPO     121.024  121.034:
CHFF1     121.018  124.013:
CHFF2     121.015  121.021  122.027  122.038  123.032: 124.021  124.026
CHFFL     122.015  122.019  122.030  122.040  123.022:
CHFP2     124.040: 124.050
```

```
CHFSR    123.011  123.015  123.017: 123.022  123.025
CHFST    122.017  122.025  122.037  123.011:
CHKCH     23.045   56.019   93.012  121.012
CHKSC     62.023   62.026:
CHM1      34.039   73.015
CHNCK     54.027   54.031:
CHNNX     54.029:  54.036
CHPFP    123.037  124.006:
CHRF     115.039  125.031:
CHRF4    122.056  125.035:
CIA       17.057
CKEM      18.029   19.019   19.044:  19.049   19.050   34.018   56.009
CLAL1     87.013:  87.019
CLALL     86.009   87.009:
CLCHN     87.010   87.013   87.014   87.023:
CLCNT     87.012   87.018   87.024:
CLOSE     86.013   87.016
CLOSF     33.036   86.006:  86.018
CLOSX     86.019:  87.020
CLP       23.047   93.036  121.034  124.006  124.028
CM400      8.018   27.031   28.020  125.025
CNODE      6.046:   7.021   18.010   18.014   93.016   93.056   94.008
          94.011   94.017   96.010   96.017   96.020
CPLU      12 008
CPYRT     84.007   84.046   86.026:  86.031   86.032   86.033   86.036
          86.037   86.040
CRD      124.008
```

```
CRN       121.037
CRO       121.038
CSEN        3.032   13.007:
CSENT      13.007   13.019:
CTCEN      10.011:  74.007
CTLC       10.011   16.009:  16.018   16.020   16.022
CTLC2      16.012   16.017:
CVSP       10.033   14.008:  14.030   15.019   16.013   54.042
CVSP1      15.022:  15.031
CVSP2      15.008   15.032:
CVSP3      15.020   15.030:
D.BSC       2.051:  34.037
D.EN        6.061:  108.023  113.036  114.033
D.MAT       2.052:  73.011
D100      121.007   123.029:
DA        114.023   114.025  114.026  114.027  122.049
DAC       114.028   120.051  125.017  125.027  125.031
DAS       114.029   120.021  120.042  120.048
DB        113.021   113.052  113.061  113.062  113.064  113.065
DBA        73.020
DEC        11.033    30.049   73.044
DECIN      26.042    30.043   30.047   41.017   41.023   46.054   48.038
           49.017    50.036   51.012   51.050   51.054   58.015   58.023
           64.024    64.039   78.019   78.029   78.056   80.020   81.035
           81.040    81.047   82.025   92.021   92.035   94.007   98.019
          108.048   113.050  114.037  114.055  117.040  120.020  120.025
          120.031   120.036  120.044  122.023  122.034  122.044  122.048
          123.014   125.015  125.023  126.012
DEF        33.046    46.015:
```

```
DELET     86.043
DET.     126.006
DETF     115.033  126.007:
DFA       97.051:  115.035
DFN       97.056   98.014:
DFP       97.053:  115.036
DFS.      98.006
DFT.      34.038   73.010
DFV       97.045   97.054   98.007:   98.008   115.034
DIM       33.055   52.017:  52.060
DIM1      52.020   52.031   52.038:
DIM2      52.043   52.046   52.054:
DIM3      52.035   52.051   52.057:
DIREC     93.030
DOADD    120.033  122.032
DOBRE    120.019  122.047
DODIV    122.041
DOINT    120.023  120.028:
DOLOA    120.028
DOMUL    122.022
DOSGN    120.043
DOSTO    123.013
DOSZE    120.024
DOZER    120.026  120.033:
DPS1      47.020:   48.026   48.039
DPS2      47.021:   48.028   48.041
```

DSC. 10.051 11.013 11.016 11.017 73.046

DSPS 12.018

DTF2W 122.016 122.021

DTF3W 122.024 122.033 122.036

DTSIN 120.029

DTUNP 120.034

DWC. 10.053 11.011 11.022 30.022 30.037 30.040 73.042

EDSTX 32.017: 46.057

EL 3.028 126.025 127.020

EMOD1 16.007 29.009 119.006 119.007

EN 3.029 6.061

END 19.007 31.019 31.045 32.043 33.047 34.019: 34.020
54.041

ENDA 34.030: 34.054

ENDAO 18.006: 19.017

ENDA1 18.006 34.031:

ENDAA 34.021 34.028:

ENDST 3.041 30.056 32.012:

ERBP. 16.010 18.031 39.018 65.034 74.025 74.045 119.041
127.043

ERLN. 18.026 74.053

ERPBC 6.041: 18.028 39.020 74.048

ERR 4.032 18.007:

ERR2 18.030 18.035 19.008: 19.029 19.033 19.037

ERR4 19.020 19.036:

ERR6 36.029 37.061: 40.026

ERRF 7.019 10.032 18.036 19.022 31.029 39.031 54.010

```
ERRF1     119.037  119.041:
ERRFN     115.021  119.010:  119.013  119.022  119.030
ERRFU     119.015  119.026  119.039  119.048  119.054:
ERRN.      18.022   74.049
ERROR      19.016
ERRST       2.059:  30.029   77.007   79.042   81.007   90.014  103.009
          106.024  111.039  111.044
ERRSV       2.056:  24.015   24.049   25.017   25.031   26.016   32.022
           33.041  33.044   36.027   38.016   41.030   42.024   42.044
           43.025  46.021   46.040   46.049   47.027   47.056   48.030
           50.016  50.022   51.009   51.014   52.027   52.055   53.017
           56.034  56.043   57.053   62.029   62.065   65.028   66.033
           72.023  72.029   77.009   78.015   78.035   83.033   83.053
           86.028  88.016   88.019   90.012   90.019   91.011   92.040
           93.008  93.019   93.024   93.027   94.042   94.046   94.051
           95.036  95.042  100.032  102.027  102.030  103.012  103.043
          103.046  104.032  104.037  104.058  105.018  107.031  109.037
          109.043
ESCF       74.009   74.013   74.021   74.023  127.044
ESCP       10.010   74.021:
ESCP1      16.008   74.034   74.041:
ESCP2      16.006   74.042:
ESCP5      74.019   74.022:
ESTOP      74.027   74.039   74.057:
ETSF       32.045   84.035
EUS         8.022    9.039   98.053  101.033
EV0       103.024:  117.043
EV1         3.039  103.028:  103.052  103.061  104.020  105.011
EV2       103.029:  103 049  104.033  104.038  105.026  109.023
EV3       103.035:  105.014  109.021
EV3J       98.023  109.019:
```

```
EV5       104.024  104.040:
EV6       109.012  109.030:
EV7       109.016  109.041:
EVA       110.044  112.038:
EVAX        4.008  102.020  102.023  102.040  103.007:  103.011
EVC       103.055  104.006:
EVCHB      24.024   24.034:
EVCHC      24.046   24.053:
EVCHD      24.013   24.023:
EVCHE      24.020   24.028:   24.037    25.041
EVCHF      25.015   25.025:
EVCHG      25.011   25.029:
EVCHS      23.023:  93.006
EVCHX       3.033   23.024:   26.009    26.015
EVCPE      23.040:  25.022   25.024
EVD       107.023  107.028  109.006:
EVDON      44.047   44.052:
EVEX        4.011  102.021  103.008  103.016:
EVF       109.051  112.010:
EVG         3.034  113.060  114.006:
EVG0      114.018  114.022:
EVG1      114.014  114.033:
EVG2      114.031  114.038:
EVG3      114.011  114.042:
EVG9      111.048  113.055:
EVIN        4.020   92.048:   92.055
```

```
EVL       105.042  106.010  107.012:
EVLF        5.013: 103.026  103.058  111.015  118.010
EVN       110.046  112.031:
EVO       103.041  104.022:
EVP       104.048  104.051  107.019: 108.057
EVP0      107.041  107.044  107.051:
EVP1      107.036  107.054  108.006:
EVP2      108.014  108.029: 108.043
EVP3      108.021  108.026  108.040:
EVP4        3.035  108.049:
EVPF        5.012:  98 015  103.025  107.042  108.050  112.024  113.016
          113.023  113 055  113.057  118.012  118.036
EVPI      108.055: 112.029
EVQ         3.036  113.021: 113.028  113.063
EVQ1      113.038: 113.046
EVR       105.034  106.015  106 021  106.025:
EVS       109.031  110.007:
EVS0      110.010  110.017:
EVS1      110.015  110.023:
EVS2      110.033  110.039:
EVSD0     111.017  111.021:
EVSD1     111.025: 111.033
EVSD2     111.029  111.037:
EVSE        3.037   42.043   43.024   44.022   44.031:  44.037   44.052
           44.053   47.045   47.055
EVSE1      44.033   44.049:
EVSED     111.013: 112.046  112.049
```

```
EVSR      106.033  110.053:
EVSU        3.038  103.017:
EVSW        5.009: 103 018  105.028  106.029  110.031
EVV       104.045  105.024:
EXLIN      10.060   31.015:  38.012   40.032
EXPN      109.024  118.006:
EXPRC       4.010=  26.028   41.006   44.032   46.041   57.009
EXST1      18.039   32.047   33.009:  54.039   74.055
EXST2      32.039   32.045:
EXST5      33.022   33.029:
EXST6      33.029   34.008:
EXSTK       6.057   31.024   31.046   32.030:
FBA.       71.026
FDA        34.039   73.015  124.023
FDEC      120.010  120.014  120.036:
FDLIM      78.022   78.051   80.024:  81.045
FDPDN      35.034   35.049:
FDPGE      35.039   35.044:
FDPLC      35.020:  35.057   36.037   37.055   80.026
FDPNX      35.032:  35.043   35.047
FDPRT      35.053   35.056:
FDSTP      78.049   80.025:  81.030
FESIZ      77.027   79.016   81.013
FESX       42.037   43.035   44.014:  44.026   44.027
FFLOT     124.034: 125.036
FINMR       6.054   73.040:
```

FIVRD 77.017 77.028 78.006 81.013

FIX 24.018 36 028 38.017 92 051 95.043 97.016 119.046
125.006

FIXS 3.040 97.015: 97.019

FL1. 15.006

FL3. 17.029 18.019 34.035 64.009 74.044

FL4. 16.017 119.021 119.033

FLAG. 11.037 11.041 14.029 14.033 17.012 19.046 30.032
32.036 52.030 52.034 73.049 73.053 98.032

FLAGC 12.012 74.036 124.016 124.027

FLOA 48.034 50.048 91.024 93.055 118.034 123.024 124.034
124.055

FLOTN 121.030 123.039 124.009 124.015 124.033:

FLU 124.013 124.022

FLUS. 4.014 68.036:

FLW. 12.013 69.019 70.024 74.037

FNC 77.014 77.026 77.035 78.037 78.044 79.007 79.046
80.008 80.011: 80.015 81.012 81.021 81.026 81.044
81.049 82.008 82.045

FNDS 11.007: 11.038 30.019

FNDS. 30.019: 30.020 30.051

FNDS1 11.010: 11.029

FNDS2 11.008 11.012:

FNLIM 80.024 82.048

FNPBC 78.048 82.012

FNPLC 78.046 82.010

FNS. 14.019 77.012 81.010 117.017

FNSTP 78.038 79.008 80.025 81.050 82.019 82.051

FONXT 79.015 80.006: 81.006
FOR 33.049 77.006:
FOR1 77.015: 77.022
FOR2 77.019 77.024:
FOR3 77.027: 77.031
FOR5 78.027 78.032:
FOREN 79.032 79.050 79.053:
FORER 78.042 79.056:
FORLD 78.052 80.013: 81.031
FORN 77.016 77.032:
FORPB 79.028 79.053 80.023:
FORS1 79.010 79.019:
FORSC 79.029 79.033: 79.051
FORSX 79.012 79.026:
FORSL 79.031: 79.039
FORST 78.023 78.050 80.014:
FRAF 115.037 120.013:
FREEN 18.012 94.009 94.016 96.018
FSC. 14.020 78.011 80.006 81.058
FSESZ 77.020 77.024 78.008 79.016: 81.016 81.024
FSZ 121.025
FUNC 109.025 115.007: 115.009
FUNCB 115.019 116.006:
FUNCE 115.014 115.017:
GBUMP 89.034: 89.036 90.038
GCP 59.056 66.016:

```
GCP1        66.019:   66.030
GETBY       13.009    68.026
GFLAG       70.008:   70.012   71.027
GIBP        70.006:   70.029   70.031   70.044   70.047   71.007
GO1         70.017    70.022:
GO2         70.030:   70.048
GO3         70.036    70.044:
GO4         70.043    71.006:
GO5         70.034    71.022:
GO6         71.020    71.031:
GOOO         4.016=   10.030   28.044   32.048   63.049   68.038   73.008
            89.015    89.035
GOOO.        4.017    70.011:
GOOOQ       66.055    70.010:
GOSUB       33.031    39.011:
GOT3        70.007:   70.022   70.027   71.030   71.032
GOTO        33.030    38.052   39.031:
GOTO1       40.007:   40.025
GOTO2       40.016    40.021:
GOTO3       40.019    40.024:
GOTO4       40.013    40.028:
GOTOL       40.030:
GSC.        14.018    36.010   36.015   39.011   39.023
GSB.        14.017    36.011   77.033
I            5.010:   98.020   98.021  103.021  103.028  103.029  103.030
           103.036   103.059  105.009  105.010  105.015  105.019  106.011
           106.016   106.018  107.017  107.019  108.041  108.051  108.055
           108.056   109.019  109.052  110.020  110.021  110.022  110.023
```

I (CONID) 110.024 110.027 110.028 110.029 110.030 110.034
110.050 111.025 111.026 111.040 111.041 112.010 112.025
112.027 112.032 112.038 113.024 113.051 114.039 114.058
115.007 117.035 117.041

IF 33.042 41.006:

IF1 41.012 41.025:

IF1A 41.028: 42.041 43.039

IF2 41.038 42.006:

IFC 42.053: 43 016 43.027

IFD 43.012 43.023 43.031:

IFE 43.008 43.041:

IFF 43.037: 43.050

IFFIX 40.036: 41.011 42.040

IFGB 43.006 43 029 44.007:

IFL 42.052 42.057 43.017:

IFLW 61.010= 62.007 62.012 62.045 62.048 63.037 64.015
64.021 64.043 64.046

IFQ 41.009 42.022:

IFR 42.043: 43.047

IFRET 40.036 40.043 40.044 40.051 40.053:

IFS 41.007 42.034:

IFS2 42.032 42.036:

IFT 41.016 41.022 41.047: 51.006

IFX 42.036 43.026 43.052:

IITAC 61.020: 62.019 62.041 62.057 63.008 63.013 63.021

INDON 64.040: 65.031

INEND 63.006 65.006:

INFO 2.038 9.056 87.022 88.037 96.026 113.020

INLEN 17.030 64.010

INP2 62 012: 63.007 64.042

INP2A 62.015: 62.037 62.067 63.029

INP2B 62.018 62 038:

INP2L 62.040 62 054:

INP2P 62.060: 63.016

INP2T 62.056 63.010:

INP3 63.012 63.018:

INP3B 62.043 63.024 63.028:

INP4 63.020 63.035:

INPBT 65.036 65.042:

INPL1 63.047 64.013:

INPLI 63.039 63.045: 65.052

INPMN 62.020: 62.025

INPNL 64.025 64.035 65.033:

INPS2 62.036 62.049: 62.066

INPSF 62.045: 63.028 63.044

INPST 64.036: 65.039

INPTS 64.014 65.010: 65.021

INPUT 33.057 61.027:

INSTB 75.058

INTEG 4.019= 23.037 24.049 25.018 62.058 63.014 66.019
83.037 83.045 86.010 92.007

INTF 92.017 115.022 120.019: 120.030

IOBRE 10.026

IOCAL 10.025

ITAC 61.011= 61.020 62.011 63.045 64.030 64.040 65.030
65.046
IXRL 125.007 125.009 125.021:
IXRDN 125.018: 125.028
IXRF 115.040 125.006:
J 5.011: 103.023 108.018 108.019 113.025 113.035
JFDEC 126.012:
JPRND 58.035: 60.020 60.029 60.041
JPRTF 57.057: 59.054
KILL 33.033 86.040: 86.048
LBA. 69.038 70.041
LCALT 96.026
LCFNB 95.049
LCFNM 95.030
LDONL 47.036 48.051:
LENF 111.046 118.010:
LENFL 118.022: 118.027
LEPS. 113.020
LET 33.053 46.037:
LETS 46.038 47.023:
LETS1 49.022: 49.037
LETS3 51.025 51.029:
LETS4 51.055 52.006:
LETSA 47.039 47.043:
LETSC 49.031 49.035:
LETSD 49.034 49.038:

LETSS 47.033: 48.015 48.043 48.049
LETSU 49.008 51.007: 51.011 51.049
LETSV 47.046 49.006:
LETU6 51.021: 51.027
LETV1 50.027 50.034:
LETV2 50.037 50.046:
LETV3 46.061 50.008: 50.014
LFTX 46.050: 50.006
LINKP 53.030
LOAD0 8.036 26.043 81.041 114.030 120.052 125.018
LOADU 7.015
LREV 96.033 97.010
LUDV 97.027 97.043:
LUVD 3.044 97.025: 100.025
MANF 115.038 120.051:
MAT 33.054 72.008: 72.011
MAT1 72.014 72.021:
MAT1D 72.018 72.024:
MAT2 72 017 72.049 73.007:
MATRW 72.054: 73.025
MATS 72.043 73.019:
MODE0 93.031
MOST 46.060 65.009 75.034:
MOST1 75.047 75.050: 76.025
MOST2 75.053: 76.041
MOST3 76.026: 76.053

MOST5 76.021 76.028 76.043:

MOST6 75.061 76.014 76.049:

MOSTA 75.056 76.007:

MOSTB 75.060 76.012:

MOVEW 71.019

MOVST 53.024 55.014: 55.029 86.022

MTOST 4.013= 19.012 34.019 34.030 34.047 53.015 56.021

MXPRC 78.038 79.008 81.050

NDCH. 87.022

NEXT 33.050 81.006:

NEXTO 81.012: 81.022

NEXT1 81.015 81.024:

NEXT2 81.049: 82.060

NEXT3 81.056: 82.007

NEXT4 81.055 82.008: 82.053

NEXT5 81.052 82.006:

NEXTD 4.022= 20.045 20.056 21.007 21.032 21.038 21.045
22.044 26.050 26.061 27.012 42.028 43.020 43.044
44.018 44.041 46.015 46.017 46.018 47.024 47.050
48.011 50.011 50.019 52.017 52.022 52.057 56.032
56.040 59.022 59.041 59.044 59.058 60.010 60.032
60.039 62.015 62.020 62.026 62.030 63.022 65.022
72.008 72.024 72.057 79.040 83.025 84.029 84.036
86.006 86 045 94.052 95.033 103.035 111.030

NEXTI 81.029 82.018: 82.031

NOP 30.016 37.010 43.014 67.018 87.017 105.040 110.013
113.043 118.011 122.052 124.020

NOTBS 8.039 12.024:

NOTF 115.042 120.006:

NOTTO 47.054 48.045:

NRCD	124.032						
NVS.	9.036	15.012	96.011	98.050	98.057	100.039	101.037
NXLIN	3.042	31.008:	32.020	33.051	33.052		
NXSTM	6.056	31.043:	31.044				
NXTIO	82.027	82.034:					
NXTI1	82.032	82.035	82.040:				
NXTI2	82.050	82.054:					
OBP.	69.035	70.038	71.025				
OCC.	14.011	58.032	58.034	59.026	60.026	67.017	67.033
	67.040	67.044	67.047	67.050	67.061		
ON	33.043	38.013:	38.042				
ONDO	38.034	38.045:					
ONSKP	38.018	38.021	38.037:				
OPEN	84.016						
OPENF	33.035	83.013:					
OPENM	84.023						
OPNBC	84.045	85.017	85.019:				
OPNCC	83.021	83.049	84.010	84.014	84.021	85.011	85.014:
OPNCM	83.041	83.044	83.051:				
OPNDN	84.018	84.028:	84.052				
OPNCC	83.028	83.037:					
OPNMN	84.012	84.020:					
OPNNE	83.057	84.035:					
OPNNF	84.040	84.045:					
OPNNX	83.025:	84.032					
OPNOL	84.007:						

```
OPNSC      83.035   83.039:
OUTBY      34.029   69.045   70.046
OUTTE      12.024   17.043   17.058   34.022   34.048
PARMT       2.054    9.051:
PATNF     115.031
PBC.       17.037   17.038   18.037   19.031   20.061   21.041   21.044
           22.017   22.023   22.026   22.038   22.041   23.029   23.036
           26.048   27.010   30.010   31.023   32.013   32.024   32.026
           32.031   33.012   33.014   35.030   36.019   36.036   36.055
           37.015   37.040   37.054   38.023   38.038   38.040   38.046
           39.035   41.032   41.035   42.026   42.048   43.019   43.034
           43.043   43.055   44.039   46.027   50.009   50.010   52.039
           53.007   54.026   54.032   56.026   57.056   59.053   62.035
           62.050   63.036   65.044   74.046   78.040   78.047   79.034
           79.044   79.058   81.059   82.013   83.030   84.042   95.009
           95.019   95.025  104.007  104.013  106.028  112.022  117.025
          117.027  119.017  119.034
PBCS       61.012=  62.051   65.043
PCOSF     115.030
PCTBL       7.008:   8.029    8.033
PCX.       67.032
PCXMC      67.034
PEXPF     115.028
PFM       124.029
PFSEA      93.033
PIB         8.010    8.016   12.015   94.056  108.035  114.049
PLC.       14.014   17.049   18.023   22.008   22.009   22.010   31.010
           31.011   31.016   32.040   35.056   36.051   37.022   37.023
           37.028   37.029   37.030   37.036   40.030   54.019   74.029
           74.050   78.045   82.011
PLOGF     115.027
PNT7      127.023:
PNT7T      68.022:
```

```
PPWR      118.007
PRAND      45.013  126.019
PRDON      59.014   59.020:
PREVN       9.052:
PRINT      33.058   56.012:  56.014   56.044   56.045
PRNOO      57.037   57.042   57.047   57.052   57.054:
PRNS1      60.012:  60.019
PRNT3      59.031:  60.030
PRNTA      56.016   56.020   56.023:
PRNTB      59.008:  59.051
PRNTC      57.029   59.022:
PRNTD      57.026:  58 035   59.036   59.046   60.016
PRNTE      56.039   56.044:
PRNTL      57.015   59.041:  59.043
PRNTN      57.012   58.006:
PRNTR      57.034   59.006:
PRNTS      57.061   60.006:
PRNTT      59.038   60.022:
PRTCF      57.060   60.032:  60.037
PRTN1      58.012   58.021:
PRTN2      58.018   58.024:
PRTNF      57.009:  57.057
PRTPC      57.022   59.057:
PRTSC      57.031   59.058:
PSINF     115.029
PSORF     115.026
```

PSTS. 17.013 39.039

PTANT 115.032

PTCF1 60.035 60.039:

PTEOL 59.048: 59.060

PUTBY 53.026 55.024 86.035

QB1 66.046: 66.053

QB2 66.045 66.051 67.009:

QBRTN 66.013= 66.040 68.013

QBYT. 4.026 66.039:

QBYTE 4.025= 56.046 59.007 59.015 59.033 59.040 60.017
60.036 62.024 63.025 66.018 66.026 66.035 68.029

QDAT 67.055 68.006:

QPAT 67.013 67.053:

QPT1 67.023 67.058 67.062:

QSP 67.016 67.021:

QSP0 67.027 67.031 67.035:

QSP1 67.043 67.048:

QST 67.019 67.051 67.063 68.008:

QT. 4.029 68.016:

QT.1 68.024: 68.030 127.046

QT.2 68.028 68.032:

QTXT. 4.028= 10.028 63.040 65.047

RDE. 67.028

READA 30.020: 30.024

READB 124.025

READD 26.006 30.021: 30.055

READI 26.007 27.044
REEAD 26 008: 26.010 27.040 33.059
RESET 12.013
REST 10.041: 33.056
RESTR 10.035 10.041 10.049: 54.043
REV. 8.013
REVNO 8.012 8.040:
RFBA 5.007: 8.028 17.025 58.029 69.010 69.023 69.030
70.015 70.028 71.009 71.022 71.047 74.058
RLBA 5.008: 8.035 66.043 66.049
RMCON 53.040 54.044:
RNDF 115.025 126.015:
RNDGC 66.058: 67.029
RNDI. 15.010 45.015 126.016 126.021
RNDM 33.038 45.010:
RNMRT 73.016 73.033:
ROBP. 17.026 58.028 66.042 66.048 68.010 68.011 69.009
69.027 69.031 70.014 70.032 71.006 71.011 71.023
71.049 71.050 74.059
RRSZ 6.045: 7.020 19.009 99.024
RSVTB 9.041 9.046:
RTNBC 37.007: 37.018
RTNBD 37.013 37.021:
RTNBE 37.019 37.025:
RTNBK 36.050: 37.033
RTNER 37.034 37.047 37.052:
RTNSI 37.027 37.036:

RTNSF 37.046: 37.049

RTNSX 36.041 37.044:

RTRN 33.048 36.010: 36.021

RTRN1 36.023 36.032:

RUN1 10.031:

RUN2 6.055 6.058 10.058:

RUNER 2.056 2.059 4.031= 13.014 23.040 24.028 24.051
26.059 30.026 31.032 36.013 37.061 39.014 39.033
52.010 52.048 53.011 53.052 58.016 61.022 62.061
65.040 66.023 72.032 73.013 77.037 79.060 81.019
83.047 88.032 92.010 92.052 93.013 93.052 95.028
95.047 96.022 96.030 97.020 98.042 99.032 100.016
100.022 100.029 100.057 103.033 105.039 110.048 111.013
111.034 112.013 113.033 115.017 117.013 117.019 119.054
121.013 123.033 124.019

RUNF 116.016 116.019 116.026:

RUNRV 2.040= 9.052

RUP 12.011 14.009 34.036 58.031 59.025 60.025 65.016
67.009 69.018 69.034 70.023 70.037 71.024 73.009
74.014 74.035 88.027

RWIB 26.011 26.017: 26.053

RWIBO 26.019 26.028:

RWIB1 26.049: 27.026

RWIB2 26.054: 72.060

RWIBC 26.022 26.033 26.046:

RWIC 26.049 27.028: 73.028

RWIC1 27.037: 28.055

RWIC2 27.041 27.044:

RWIC3 27.048 28.007:

RWIC4 27.052 28.008 28.013: 28.025

RWICF 28.011 28.019:

```
RWICC     27.045   28.027:
RWIQ      26.031   27.006:
RWIQ2     26.056:  27.008
RWIRT     28.036   28.042
RWIS      26.029   27.019:
RWIS1     26.026   27.022:
RWIS2     27.017   27.024:
SATUP      7.022    8.009:  54.040
SBA       30.013   38.028   95.022
SBP       47.016:  47.047   48.008   48.016   48.045
SCOPE     12.047   16.015   53.044
SE         2.057    2.060   13.015   23.041   24.029   24.052   26.060
          30.027   31.033   36.014   39.034   52.011   52.049   58.017
          61.023   62.062   65.041   66.024   72.033   83.048   88.033
          92.011   92.053   93.014   93.053   96.023   96.031   97.021
          98.043  100.030  100.058  103.034  111.014  111.035  113.034
         115.018  117.014  119.055  121.014  123.034
SEARC     93.032
SER       47.010:  47.031
SERCL     93.043   93.051:
SETBL      4.037=  10.027   10.058
SFN      116.011  117.015:
SFS       47.018:  47.049   48.019
SGN2A     90.041   91.013:
SGNF     115.024  120.040:
SGNLA     88.011:  88 023
SGNSD     88.022:  90.010
SIG1A     89.008   90.006:
```

SIGN1	88.021	89.006:					
SIGN2	90.030	91.009:					
SIGNC	91.020:	91.032					
SIGNE	88.039	89.010	90.034	91.012	92.037:		
SIGNL	33.032	88.007:	88.028	90.039			
SIGNN	91.017	92.018:	92.020				
SIGNO	91.029	92.024	92.027:				
SIGNV	90.007	90.013:	90.032				
SIGNX	88.011	88.024	89.009	90.033	92.006:		
SIGPA	28.051	88 031	89.028	90.037	91.016		
SKIPZ	74.037	124.017	124.028				
SKRST	20.037:	31.042	79.017				
SKSCS	21.018	21.024	22.035:				
SKSGO	20.049	21.038:					
SKSNL	20.052	21.010	21.049	22.007:			
SKSNO	20.040 22.047	20.055	20.059	21.007:	21.030	21.036	22.042
SKSNC	20.045:	22.030					
SKSOF	21.012	21.048	22.022:				
SKSSR	22 019:	22.032					
SKSSY	21.027	22.044:	22.046				
SKST1	21.015	21.019:					
SKST2	21.021	21.025:					
SKSTM	3.043	20.037	20.043:	22.013	22.019		
SKSTO	21.032:	21.035					
SLT	10.050	14.013	32.041	35.022	37.031	39.045	53.035

```
SNO        17.042:   19.023    34.056
SNOW       34.053    34.056:
SPCF      115.041   119.046:
SPECI     119.053
SPINP      64.012
SRCH       33.037    93.007:
SRCH1      93.029    93.035:
SS         47.017:   47.048    48.009    48.046
STINP      34.041
STOP       33.045    34.047:   74.061
STORD       9.010    26.038    41.026    78.036    78.057    81.036    81.048
           116.009   120.045   125.016   125.024
STOUT      12.045    19.024    19.036    69.041    71.031
STPBC      18.027    19.030    32.032    33.013    37.053    39.017    74.047
           79.027    79.035    79.055    79.057   119.011
STPRC      80.019
STS       121.019   122.028   124.017
STZRO     120.024:  122.054
SVTB        9.046    10.062:
SWOOP       7.016     8.015     8.039:
SWPI        7.012:   10.008
SWPI2       7.024    12.007:
SWPO        9.006:   10.009
SWPOK       7.011:    7.013     7.018     9.007
SZP.        8.017
T1         66.008=   66.039    67.010    68.008    68.036    68.040
T2         66.009=   69.007    69.012    69.032
```

```
T3          66.010=  66.016   66.036   69.024   69.026   69.043   69.046

T4          66.011=  68.017   68.024   68.025   68.032

T7X        127.034  127.040  127.045:

TBYTE       46.050   46.055   47.015:  47.029   47.033   47.051   48.012
            48.031   75.032

TRAPF        8.027   30.015  110.012  113.042  122.051

TS           5.019:   7.012   12.010   12.019   14.008   15.032   17.042
            17.060   19.018   28.042   28.049   39.051   40.018   40.022
            53.018   55.017   55.018   75.038   76.024   78.021   78.024
            78.030   86.029   93.015   93.035   93.049  103.019  105.020
           105.021  106.031  109.053  109.054  110.037  111.019

TS1          5.020:   8 009    8.037   30.030   30.052   39.052   40.023
            46.016   46.024   50.023   50.043   53.019   55.028   86.030
            98.044  100.007  101.006  101.018  114.053  114.056

TS2          5.021:  50.031   50.038   51.020   51.021   51.022   53.027
            55.015   55.020   55.023   55.030   98 040  100.008  101.007
           114.051  114.057

TS3          5.022:  50.029   50.039   51.016   51.029   51.041   52.007
            55.016   55.025   75.040   75.053   76.026   76.043  100.009
           101.012  114.008  114.040  114.059

TS4          5.023:  49.019   49.029   49.035   50.017   50.024   50.035
            51.043   52.006   55.014   55.022   55.031  102.009  110.042
           112.056  113.013

TS5          5.024:  49.021   49.022   49.023   50.018   50.025   50.040
            53.022   53.028   75.041   76.008   76 022   76.039   76.050
            98.027   99.019  100.034  100.055  102.006  102.016

TS6          5.025:  11.007   11.015   11.042   30.046   46.053   48.025
            48.037   48.040   49.010   50.051   51.037   64.038   75.050
            76.016   76.017   76.033   76.044   78.018   78.055   81.034
            81.039   82.018   82.040   82.044  102.039

TS7          5.026:  17.010   17.033   30.045   46.052   47.040   48.027
            48.036   48.042   49.012   51.033   52.021   52.024   64.037
            76.019   76.034   76.038   78.016   78.054   80.017   81.027
            81.033   81.038   90.020  102.038

TSC         88.037  127.025

TSC.H       88.038:  89.018
```

TSE	3.050	6.008:	9.006	9.044	36.018	36.035	37.016
	38.027	38.045	51.015	51.019	51.036	51.053	86.023
	86.025	88.008	88.012	90.040	91.013	91.020	91.030
	92.014	94.014	94.047				
TSE1	6.009:	36.033	36.039	36.042	37.009	37.025	37.048
	51.018	51.026	51.030	51.045	51.056	52.009	88.010
	88.022	90.025	90.031	91.009	91.019	91.026	91.031
	94.049	94.054					
TSE2	6.010:	88.013	90.015	92.008	92.037	94.018	94.033
	94.034	94.035	94.036				
TSE3	6.011:	92.006	92.015				
TSE4	6.012:	94.021	94.040				
TSE5	6.013:						
TSE6	6.014:	35.024	35.042	35.044			
TSE7	6.015:	20.039	20.046	22.027	35.029	35.041	35.046
	35.049						
TSF	3.049	5.042:	110.056	111.018	111.022	116.006	116.012
	116.013	118.021	121.010	121.016			
TSF1	5.043:	111.024	118.015	118.030	124.038	124.045	124.049
	124.052						
TSF2	5.044:	118.017	118.029	124.039	124.048	124.053	
TSF3	5.045:	118.019	118 025	118.033			
TSF4	5.046:						
TSF5	5 047:						
TSN	3.051	6.029:	24.034	27.046	27.051	28.012	28.024
	86.023	93.040					
TSN1	6.030:	24.053	93.046				
TSN10	6.037:	23.043	23.050	23.051	24.030	25.033	25.034
	26.025	28.014	62.008	63.015	64.011	73.023	
TSN11	6.038:	23.025	25.009	25.025	28.032	28.040	62.009
	62.059	64.007					
TSNP	6 031:	24.039	25.036	27.032	27.035	28.019	29.018

TSN3	6.032:	27.030					
TSN4	6.033:	23.027	23.033	23.034	25.038	25.040	27.011
	27.015	27.029	29.024	29.026	72.039	72.055	73.019
TSN5	6.034:	23.038	26.056	27.042	56.017	93.010	
TSN6	6.035:	23.028	26.017	27.038			
TSN7	6.036:	24.017	24.021	27.028	28.017	66.028	86.015
	92.050	92.054					
TSP	6.017:	10.062	10.063	43.015	43.053	66.008	
TSP1	6.018:	42.027	42.031	43.017	43.056	59.009	71.046
	71.054	88.006	88.017	88.026	91.014		
TSP10	6.025:	66.012					
TSP11	6.026:	66.013					
TSP12	6.027:	44.040	44.044	66.014			
TSP2	6.019:	41.019	41.025	41.027	42.046	43.013	44.007
	44.014	44.023	66.009				
TSP3	6.020:	42.049	43.033	43.042	47.023	49.038	51.059
	57.016	57.026	57.054	58.006	59.023	59.045	59.048
	59.059	60.011	60.023	60.040	61.010	89.011	89.022
TSP4	6.021:	42.053	43.009	43.031	43.036	43.052	66.010
TSP5	6.022:	42.038	42.050	42.055	43.010	43.049	44.017
	44.024	49.011	49.026	49.027	49.039	59.031	59.034
	60.007	60.018	61.011	89.013	89.017		
TSP6	6.023:	42.045	44.010	44.011	66.011		
TSP7	6.024:	49.013	49.036	60.009	60.012	60.013	61.012
TSU	18.009	18.016	19.014	19.026	56.036	58.010	58.027
	73.041	92.023					
TSU.1	56.037	58.026	73.043				
TSU.2	17.032	19.011	56.025	59.012	59.019	61.032	69.013
	70.018						
TSU.3	73.045	90.027	91.028				
TSU.4	17.024	33.024	38.048	40.046			

TSU.5 119.051

TSU.6 73.047

TSX 3.048 5.030: 75.034 76.032 76.047 78.017

TSX1 5.031: 83.018 83.031 83.038 84.008 84.028 84.047

TSX10 5.038: 25.035 28.009 40.037 40.039 41.010 41.013
41.036 42 039 42.042

TSX2 5.032: 51.039 51.058 75.039 76.012 76.029 76.040

TSX3 5.033: 83.016 83.039 83.055

TSX4 5.034: 75.035 75.048 76.030

TSX5 5.035: 26 037 26.040 26.044 75.036 75.054 76.007
76.036

TSX6 5.036: 48.017 75.037 76.010 100.024 102.010

TSX7 5.037: 48.021 75.043 76.052

UFC. 14.016 33.011 109.048 112.016 112 019 117.016 117.023

UFN 115.011 117.007:

UFS. 14.015 33.010 109.049

UFT. 14.021 46.023 117.008

UNLOC 26.058

UPS. 15.015 100.036

UVS. 15.011

VBLDF 84.043 85.008:

VBLDT 84.037 85.007:

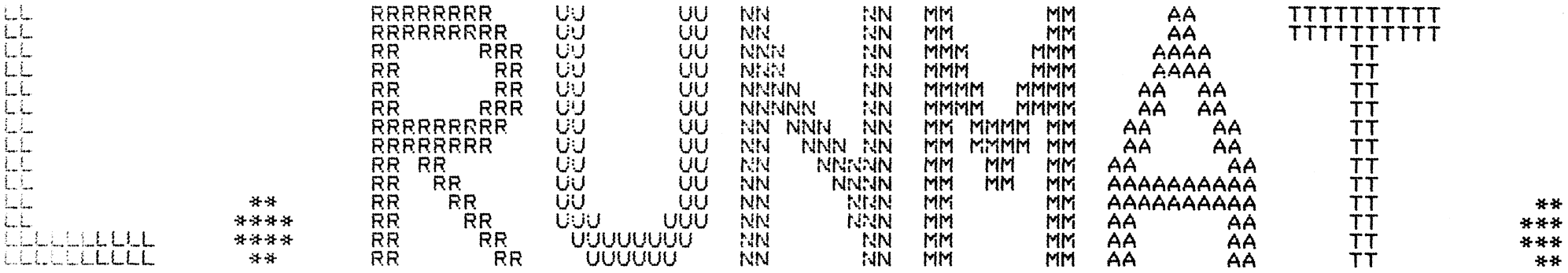
VDT. 11.012 15.013 22.011 31.017 35.025 39.046 54.020
97.031 98.011 100.040 117.030

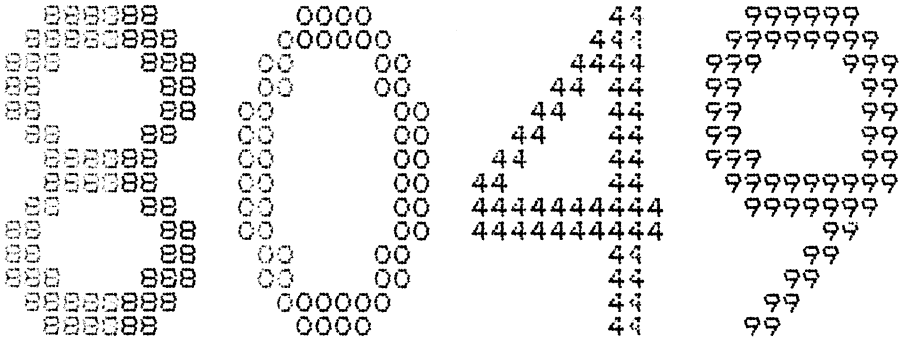
VEMD1 29.009:

VFLDF 73.036: 73.050

VFMKC 121.026 121.032:

```
VLOC        3.052  102 020:
VLOCS     102.022  102.032  102.034:
VMINU      57.048   57.059:
VOLFI     124.043
WAIT       66.056   68.039   69.022   69.042   71.037:
WONA       71.039
WPR       122 012  122.039  123.038
WRITE      26.014:  33.060
WRITI      26.013   56.010   72.020
WRTN       66.014=  71.037   71.040
XCHF4     121.040  122 014  122.056:
XFL1       69.025:  69.047
XFL1A      69.029   69.034:
XFL2       69.040   69.043:
XFLU.       4.035   69.007:
XFLUC       4.034=  49.014   58.008   59.016   61.029   63.048   68.037
           73.007   89.014
XGETB      17.039   23.030   32.016   33.015   36.020   42.054   44.012
           44.016   48 007   49.024   50.030   51.023   51.042   53.008
           54.033   55.019   56.027   60.014   69.044   70.045   75.051
           76.011   79.036   95.010  118.016  118.022  119.012
XNOCH      87.011   87.022:
XOR         8.025   39.012   71.014
XORE        8.030
XPUTB      28.016   49.028   49.041   50.033   50.042   51.046   51.061
           52 008   65.012   68.012   76.018   76.046  118.020  118.031
XSGNE      88.025   88.039:
```





File: 18/L.RUNMAT.8049 Printed: MAR 8, 1988 1:18:34
File size: 332 blocks Copy 1 of 1

Printed for RELEASE CONTROLL