```
@                 @@@@@@@   @@@@@    @@@@@    @@@@@   @@@@@@@
@                 @        @     @  @     @  @     @        @
@                 @        @     @  @        @     @       @
@                 @@@@@    @     @   @@@@@    @@@@@       @
@                 @        @     @        @  @     @     @
@         @@@     @        @     @  @     @  @     @    @       @@@
@@@@@@@   @@@     @@@@@@@   @@@@@    @@@@@    @@@@@     @       @@@

 @@@@@    @@@@@    @@@@@      @
@     @  @     @  @     @    @@
@     @        @  @     @   @ @
 @@@@@        @    @@@@@      @
    @     @@@@        @       @
   @     @           @        @
  @      @@@@@@@    @         @@@@@
```

```
Spool Queue Line #:  33
IRIS LU/Filename   :  18/L.EOS87.9291

           Printed on/at :  FEB  6, 1990  12:43:03
           For Group/User:  0 , 1
               On Port No:  5

Print control parameters :
    Printer Class code    :  0
    Form Code/paper type  :  ?
    Print Priority (0-9)  :  5
    Starting Page Number  :   1
    This is copy number   :  1
    Keep file (Y/N)       :  Y
    Notify User when done:  N
    Comments, optional    :  For R9.5  RELSE CN
```

```
.EOT    ;"R9xJCL.EOS87" FOR IRIS R9.x
.EOT
.EOT    ;"DSUBDEFS" FOR IRIS
        .END
```

```
;       Batchfile:  R95JCL.EOS87

;
;       ; D = 9291
;
;       -R95DEFSPZ
;       -R95DSUBDEFSD
;       R92DSBEOS87SA
;
                     .EOT   ;"R9xJCL.EOS87" FOR IRIS R9.xx
```

```
;                          << SI = R92DSBEOS87SA; BO = 18/A.EOS87.9291! >>
;LAST EDITED BY RDC FOR R9.0
;change mode 6 to return not found as FLAG=0 otherwise FLAG=1


                               ;CALL87.S - The special LINPOS manipulation discsub

;       Author: JPMH
;       Date:   8-Mar-84
;
             14          REVIS   =      14

;       This discsub provides all manipulation required for the array LINPOS
;       used by the new TYPIST editor.
;
;       The calling sequence is
;               CALL 87,MODE,LINE,BLOCK,BYTE,FLAG,LINPOS$
;                       LINPOS is stored as a string since arrays are not
;       permitted as parameters to discsubs.
;
;       MODE = 0 - return revision
;            = 1 - initialize last line in use
;            = 2 - set pointers
;            = 3 - read pointers
;            = 4 - make space for pointers
;            = 5 - remove pointer entry
;            = 6 - search for previous falgged entry
;            = 7 - count the number of previous rulers and return it
;                  in block

              1          .TXTM    1                 ;set byte sex

         105400          .LOC     LE087             ;the address to assemble as

  105400     165 ENTRY:  EOS87                        ;the IRIS name
  105401       3         START-ENTRY                ;standard discsub linkage
  105402  177416         ENTRY-DSBEND

  105403   50460 START:  STA      2,APT             ;remember address of pointer table       2
  105404   50460         STA      2,APTTM           ;the stepping argument table pointer
  105405   25013         LDA      1,13,2            ;number type of LINPOS$                 1
  105406  125123         MOVZL    1,1,SNC           ;check that its string
  105407     464         JMP      ERR               ;it was not
  105410  125220         MOVZR    1,1               ;restore without the string type        1
  105411  125220         MOVZR    1,1               ;size in words not bytes                1
  105412   44457         STA      1,LINSZ           ;the dimensioned size
```

```
;                     << SI = R92DSBEOS87SA; BO = 18/A.EOS87.9291! >>
   105413  25012         LDA     1,12,2          ;the address of linpos                     1
   105414  44456         STA     1,LINPS         ;remember it
   105415   4420         JSR     PICK            ;go and get the MODE                      ?1??
   105416 101004         MOV     0,0,SZR         ;PICK returned the sign in A0, check
   105417    454         JMP     ERR             ; +ve otherwise error
   105420  44413         STA     1,MODE          ;note it
   105421   4431         JSR     CONT            ;continue but set the vector table           3
                                                 ; address in A3

   105422     52 VECTOR: REVNO-VECTOR            ;displacement to the various routines
   105423     73         C87V1-VECTOR            ; is stored in this vector table
   105424     76         C87V2-VECTOR
   105425    144         C87V3-VECTOR
   105426    202         C87V4-VECTOR
   105427    231         C87V5-VECTOR
   105430    252         C87V6-VECTOR
   105431    316         C87V7-VECTOR

   105432     10 VSIZE:  .-VECTOR                ;size of the vector table

   105433      0 MODE:   0                       ;the called mode


                 !!!!!!

   105434      0 RETS:   0                       ;return address for subroutine

          105435 PICK:   ;this routine returns the value of the next numeric parameter
                         ; a fatal error is generated if the magnitude of the value is
                         ; not in the range 0..65535
   105435  54777         STA     3,RETS          ;return address
   105436  30426         LDA     2,APTTM         ;the stepping argument table pointer        2
   105437  25001         LDA     1,1,2           ;the number type of the parameter          1
   105440 125132         MOVZL#  1,1,SZC         ;check its numeric
   105441    432         JMP     ERR             ;NO
   105442  31000         LDA     2,0,2           ;the address                                2
   105443 102520         SUBZL   0,0             ;A0=1 is instruction to LOAD              0
   105444   6120         DECIMAL                 ;load the parameter into DA               ????
   105445   6121         FIX                     ;FIX DA into A0 and A1                    01??
   105446    425         JMP     ERR             ;non-skip if bad number
   105447  10415         ISZ     APTTM           ;step the temporary pointer
   105450  10414         ISZ     APTTM           ; for the 2 word entry
   105451   2763         JMP     @RETS           ;we are done

                 !!!!!!

          105452 CONT:   ;A3 has the address of the vector table
   105452  20761         LDA     0,MODE          ;what command we are to execute           0
   105453  24757         LDA     1,VSIZE         ;what is the maximum allowed               1
   105454 106433         SLE     0,1             ;check its in range
```

```
;                           << SI = R92DSBEOS87SA;  BO = 18/A.EOS87.9291! >>
  105455    416             JMP      ERR                 ;its not
  105456 171000             MOV      3,2                 ;second copy of the table address          2
  105457 117000             ADD      0,3                 ;address of the required vector             3
  105460  35400             LDA      3,0,3               ;value of the vector                        3
  105461 173000             ADD      3,2                 ;absolute address of the required rtne     2
  105462   5000             JSR      0,2                 ;gp to it with A3 pointing at APT           3

  105463      0 APT:        0                            ;pointer to argument table
  105464      0 APTTM:      0                            ;stepping version of above
  105465      0 BLOCK:      0                            ;values passed to and from BASIC
  105466      0 BYTE:       0
  105467      0 FLAG:       0
  105470      0 LINE:       0

  105471      0 LINSZ:      0                            ;size of array
  105472      0 LINPS:      0                            ;address of array

              0 APT.        =        0                   ;relative to special table
              1 APTT.       =        APT.+1
              2 BLOC.       =        APTT.+1
              3 BYTE.       =        BLOC.+1
              4 FLAG.       =        BYTE.+1
              5 LINE.       =        FLAG.+1
              6 LINS.       =        LINE.+1
              7 LINP.       =        LINS.+1

                !!!!!!

  105473   2112 ERR:        JMP      @.NRET              ;error return


         105474 REVNO:      ;return the revision number of this routine
  105474  24403             LDA      1,RNUM              ;the revision number                        1
  105475   4403             JSR      SAVA1               ;save as LINE parameter                  ????
  105476    421             JMP      EXIT                ;we are done

  105477     14 RNUM:       REVIS                        ;revision of the routine

                !!!!!!

         105500 SAVA1:      ;save contents of accumulator 1 into the next parameter A1
  105500 102400             SUB      0,0                 ;tell float that we are +ve              0
  105501  54440 SAVAA:      STA      3,RETS3             ;save return address
  105502   6122             FLOAT                        ;float A1 as +ve into DA                 ????
  105503  30761             LDA      2,APTTM             ;the temporary pointer                     2
  105504  25001             LDA      1,1,2               ;the number type                          1
  105505 125112             MOVL#    1,1,SZC             ;is it numeric
  105506    765             JMP      ERR                 ;NO
  105507  31000             LDA      2,0,2               ;the address of the parameter              2
  105510 102400             SUB      0,0                 ;instruct DECIMAL to store               0
```

```
;                        << SI = R92DSBEOS87SA; BO = 18/A.EOS87.9291! >>
    105511   6120                DECIMAL                    ;store DA into the parameter            ????
    105512  10752                ISZ      APTTM             ;step past this parameter
    105513  10751                ISZ      APTTM
    105514   2425                JMP      @RETS3            ;we are done

                    !!!!!!

           105515 C87V1:         ;set the last line in use to 1
    105515 102400                SUB      0,0               ;zero for setting                       0
    105516  42754                STA      0,@LINPS          ;zero the last line in use
    105517   2113 EXIT:          JMP      @.SRET            ;perform a non-error return

                    !!!!!!

           105520 C87V2:         ;set the pointers for line LINE
    105520   4422                JSR      RDALL             ;go and get LINE, BLOCK, BYTE and FLAG  ????
    105521  24747                LDA      1,LINE            ;the line specified                      1
    105522  22750                LDA      0,@LINPS          ;the last in use                        0
    105523 122033                SLS      1,0               ;specified line already in use
    105524  46746                STA      1,@LINPS          ;NO, so set it as the new maximum
    105525  20740 MOD2A:         LDA      0,BLOCK           ;the block number specified             0
    105526 101300                MOVS     0,0               ;set into high byte                     0
    105527 101120                MOVZL    0,0               ;shift left to allow FLAG in            0
    105530  24737                LDA      1,FLAG            ;whatever was input                      1
    105531 125220                MOVZR    1,1               ;copy flag into carry                    1  C
    105532  24734                LDA      1,BYTE            ;the byte address as a nine bit          1  C
    105533 123200                ADDR     1,0               ;add in byte, divide byte by 2 and      0
                                                            ; shift in the flag
    105534  24734                LDA      1,LINE            ;line we are setting
    105535  30735                LDA      2,LINPS           ;address of LINPOS                        2
    105536 133000                ADD      1,2               ;A2 address of this entry                 2
    105537  41000                STA      0,0,2             ;save the entry
    105540    757                JMP      EXIT              ;we are done

                    !!!!!!

    105541         0 RETS3:      0                          ;return address for subroutine

           105542 RDALL:         ;read LINE, BLOCK,BYTE and FLAG
    105542  54777                STA      3,RETS3           ;return address
    105543   4413                JSR      RDLIN             ;go and get the line                    ?1??
    105544   4671                JSR      PICK              ;BLOCK
    105545  44720                STA      1,BLOCK
    105546   4667                JSR      PICK
    105547  44717                STA      1,BYTE
    105550   4665                JSR      PICK              ;get thge input flag                    ?1??
    105551 102520                SUBZL    0,0               ;generate a 1                           0
    105552 125004                MOV      1,1,SZR           ;skip if A1 zero                         1
    105553 105000                MOV      0,1               ;set 1 since non zero                    1
                                 ;A1 = 0 or 1 as the flag
```

```
;                     << SI = R92DSBEOS87SA;  BO = 18/A.EOS87.9291! >>
   105554  44713         STA     1,FLAG          ;save the flag as 0 or 1
   105555   2764         JMP     @RETS3          ;we are done

                 !!!!!!

          105556 RDLIN:  ;read the line pointer
   105556  54407         STA     3,RETS2         ;save the return address
   105557   4656         JSR     PICK            ;get the parameter                      ?1??
   105560  34711         LDA     3,LINSZ         ;the bounds of the array                  3
   105561 136433         SLE     1,3             ;are we in range
   105562   2112         JMP     @.NRET          ;NO - error
   105563  44705         STA     1,LINE          ;save the value in line
   105564   6401         JSR     @RETS2          ;go back to the caller

   105565      0 RETS2:  0                       ;subroutine return address

                 !!!!!!

          105566 C87V3:  ;return BLOCK,BYTE,FLAG for the given line
   105566   4770         JSR     RDLIN           ;get the line number from input         ?1??
   105567  30703 MOD3A:  LDA     2,LINPS         ;where the array is                       2
   105570  21000         LDA     0,0,2           ;the last line in use                   0
   105571 122433         SLE     1,0             ;is the attempt beyond the end
   105572    425         JMP     BEYOND          ;YES
   105573 133000         ADD     1,2             ;set pointer for the line                 2
   105574  21000         LDA     0,0,2           ;get the word value                     0
   105575 126400         SUB     1,1             ;generate a zero for adding to           1
   105576 101120         MOVZL   0,0             ;move the flag bit into carry           0    C
                                                 ;  and multiply block and byte by 2
   105577 125100         MOVL    1,1             ;carry becomes lsb of A1                 1
   105600  44667         STA     1,FLAG          ;save as the flag
   105601  24415         LDA     1,C776          ;prepare to mask to byte address
   105602 107400         AND     0,1             ;remove the block part                   1
   105603  44663         STA     1,BYTE          ;save it
   105604 101300         MOVS    0,0             ;block number into ls byte              0
   105605  24064         LDA     1,C377          ;isolate the block                       1
   105606 107600         ANDR    0,1            ;                                         1
   105607   4671         JSR     SAVA1           ;output to user as BLOCK                ????
   105610  24656         LDA     1,BYTE          ;output BYTE                             1
   105611   4667         JSR     SAVA1           ;                                       ????
   105612  24655         LDA     1,FLAG          ;                                        1
   105613   4665         JSR     SAVA1           ;                                       ????
   105614    703         JMP     EXIT            ;we are done with no errors

   105615      0 TABLE:  0                       ;some routine use this as address of
                                                 ;  storage containing BLOCK, APT, ...


   105616    776 C776:   776                     ;constants used to mask byte address
```

```
;                     << SI = R92DSBEOS87SA;  BO = 18/A.EOS87.9291! >>
          105617 BEYOND: ;the specified line is beyond the end of the lines in use
   105617 102520         SUBZL   0,0              ;generate 1, we are going to return -1   0
   105620 105000         MOV     0,1              ; as the BLOCK, A0=sign, A1=magnitude    1
   105621   4660         JSR     SAVAA            ;save A1 with sign A0
   105622    675         JMP     EXIT             ;we are done


   105623    655 JSAVA1: JMP     SAVA1            ;stepping stone

                 !!!!!!

          105624 C87V4:  ;Make space for and insert values for LINE
   105624    401         JMP     .+1
   105625   4715         JSR     RDALL            ;read LINE,BLOCK,BYTE,FLAG              ????
   105626  30644         LDA     2,LINPS          ;where is the array                        2
   105627  21000         LDA     0,0,2            ;current last line in use                0
   105630  24640         LDA     1,LINE           ;get the line that we are to insert       1
   105631 122433         SLE     1,0              ;if the inserted line is in range
   105632    417         JMP     M4DON            ;its beyond the end so don't need to
                                                  ; shuffle
   105633  24636         LDA     1,LINSZ          ;dimension of the array                   1
   105634 106033         SLS     0,1              ;skip if it fits
   105635    636         JMP     ERR
   105636 113000         ADD     0,2              ;A2 is address of last in use              2
   105637 101400         INC     0,0              ;modify so that we generate the one      0
                                                  ;  extra when we subtract to find the
                                                  ;  number to move
   105640  24630         LDA     1,LINE           ;the line number that we are to create    1
   105641 106400         SUB     0,1              ;the number to move NEGATED               1
   105642  21000 M4LP:   LDA     0,0,2            ;get the entry to move                   0
   105643  41001         STA     0,1,2            ;move it
   105644 150400         NEG     2,2              ;;increment the pointer to move            2
   105645 151400         INC     2,2              ;; by a kludge method                      2
   105646 150400         NEG     2,2              ;;                                         2
   105647 125404         INC     1,1,SZR          ;reduce the number to count               1
   105650    772         JMP     M4LP             ;we are not done yet
   105651  12621 M4DON:  ISZ     @LINPS           ;increment the last line in use
   105652    653         JMP     MOD2A            ;set the block, byte and flag and finish

                 !!!!!!

          105653 C87V5:  ;Delete a pointer
   105653   4703         JSR     RDLIN            ;get the line number                    ?1??
   105654  30616         LDA     2,LINPS          ;the address of the array                  2
   105655  21000         LDA     0,0,2            ;the current number in use               0
   105656 101015         MOV#    0,0,SNR          ;check that its not zero
   105657    614         JMP     ERR              ;it was so we are in trouble
   105660 122415         SUB#    1,0,SNR          ;are we removing the last
   105661    410         JMP     M5LST            ;YES, so no shuffle required
   105662 133000         ADD     1,2              ;start at the line pointed at              2
```

```
;                            << SI = R92DSBEOS87SA;  BO = 18/A.EOS87.9291! >>
   105663 106400         SUB     0,1               ;the number to shuffle NEGATED         1
   105664  21001 M5LP:   LDA     0,1,2             ;get the next one                     0
   105665  41000         STA     0,0,2             ;save it moved
   105666 151400         INC     2,2               ;we are ready for the next              2
   105667 125404         INC     1,1,SZR           ;reduce the number to move             1
   105670    774         JMP     M5LP              ;we are not done so loop
   105671  16601 M5LST:  DSZ     @LINPS            ;reduce the number in use,
   105672    401         JMP     .+1               ;in case there was not a skip
;            JMP      EXIT              ;we are done
   105673   2113 JEXIT:  JMP     @.SRET            ;should be a JMP EXIT but there is
                                                   ; an addressing error and this saves code

                 !!!!!!

          105674 C87V6:  ;search backwards for an entry with the flag set
   105674  54721         STA     3,TABLE           ;where all the specials are
   105675   4661         JSR     RDLIN             ;go and read the required line number   ?1??
   105676 102520         SUBZL   0,0               ;generate a one                        0
   105677 122432         SGR     1,0               ;check that line is greater than 1
   105700    415         JMP     NOTFND            ;it was not, so we will find none
   105701  34714         LDA     3,TABLE           ;where specials are                      3
   105702  31407         LDA     2,LINP.,3         ;the address of the array               2
   105703 133000         ADD     1,2               ;the address of this line entry         2
   105704 112400         SUB     0,2               ;reduce by 1 as we look at previous line 2
   105705 124400         NEG     1,1               ;number of lines to check NEGATED       1
   105706  50453         STA     2,PTR87           ;save it so we can use indirection
   105707  22452 M6LP:   LDA     0,@PTR87          ;get the next value to check           0
   105710 101132         MOVZL#  0,0,SZC           ;is the flag bit set
   105711    407         JMP     FOUND             ;we found the bit
   105712  14447         DSZ     PTR87             ;check the next previous, dont worry
                                                   ; about 0
   105713 125404         INC     1,1,SZR           ;one less to check                      1
   105714    773         JMP     M6LP              ;since we are not done, loop
          105715 NOTFND: ;we found no flagged entries
   105715 126420         SUBZ    1,1               ;generate a zero                        1
   105716   4412         JSR     SAVLN             ;save the result in line
   105717    754         JMP     JEXIT             ;we are done

          105720 FOUND:  ;we found a flagged entry
   105720  24441         LDA     1,PTR87           ;where we were when we found it         1
   105721  21407         LDA     0,LINP.,3         ;where the array starts                0
   105722 106400         SUB     0,1               ;convert displacement to a line no      1
   105723  45405         STA     1,LINE.,3         ;so that MOD3 can find it
   105724   4404         JSR     SAVLN             ;set it into return LINE
   105725  34670         LDA     3,TABLE           ;where things are stored                 3
   105726  25405         LDA     1,LINE.,3         ;get LINE into A1 for MOD3a            1
   105727    640         JMP     MOD3A             ;get and return BLOCK, BYTE and FLAG

          105730 SAVLN:  ;save the A1 value as LINE for return
   105730  54635         STA     3,RETS2           ;return address
```

```
;                            << SI = R92DSBEOS87SA; BO = 18/A.EOS87.9291! >>
   105731  34664          LDA      3,TABLE         ;where all the special stuff is          3
   105732  31400          LDA      2,APT.,3        ;theC87Vginal pointer                   2
   105733 151400          INC      2,2             ;step past the MODE                     2
   105734 151400          INC      2,2
   105735  51401          STA      2,APTT.,3       ;save so that next op occurs on LINE
   105736  34627          LDA      3,RETS2         ;so SAVA1 knows where to return to       3
   105737    664          JMP      JSAVA1          ;use the SAVA1 routine to return it

                 !!!!!!

          105740 C87V7:   ;count the number of previous rulers
   105740  54655          STA      3,TABLE         ;where the special table is              3
   105741 102400          SUB      0,0             ;there are none yet                   0
   105742  41402          STA      0,BLOC.,3       ;BLOCK is used for return
   105743   4613          JSR      RDLIN           ;what line to count from              ?1??
   105744  34651          LDA      3,TABLE         ;restore the pointer                     3
   105745 124400          NEG      1,1             ;the number to count NEGATED           1
   105746 125405          INC      1,1,SNR         ;don't coun this line                  1
   105747    407          JMP      M7DON           ;we are at line one so there can
                                                   ; be none
   105750  11407 M7LP:    ISZ      LINP.,3         ;use LINPS as the pointer
   105751  23407          LDA      0,@LINP.,3      ;get the next entry                   0
   105752 101132          MOVZL#   0,0,SZC         ;is flag bit set
   105753  11402          ISZ      BLOC.,3         ;YES so increment the count
   105754 125404          INC      1,1,SZR         ;reduce the number to count            1
   105755    773          JMP      M7LP            ;since we are not done, loop
   105756  25402 M7DON:   LDA      1,BLOC.,3       ;get the value we have counted         1
   105757   4644          JSR      JSAVA1          ;save and exit
   105760    713          JMP      JEXIT

   105761      0 PTR87:   0                        ;pointer used for stepping through array

   105762 105762 DSBEND:  .                        ;the end address

               0       .ERR    ENTRY+400<.;OVERFLOW CHECK

                          .END
```

```
---

APT   105463    APTTM 105464    APTT.      1    APT.       0    BEYON 105617
BINDI   6115    BINMU   6116    BLOCK 105465    BLOC.      2    BPI       16
BSACF     75    BUMPU   6117    BYTE  105466    BYTE.      3    C10       30
C100      51    C1000     67    C11       31    C12       32    C13       33
C14       34    C15       35    C16       36    C160     174    C163     175
C166     176    C17       37    C170K     21    C171     177    C177      52
C1777     70    C2         2    C20       42    C200      53    C2000     71
C205      54    C215      55    C240      56    C244      57    C260      60
C271      61    C3         3    C300      62    C334      63    C37       43
C377      64    C4        24    C40       44    C400      65    C4000     72
C5        25    C6        26    C600     100    C7        27    C77       50
C774C     22    C776  105616    C777      66    C87V1 105515    C87V2 105520
C87V3 105566    C87V4 105624    C87V5 105653    C87V6 105674    C87V7 105740
CALL    6101    CHANN   6106    CM400     23    CONT  105452    DA       160
DAC      164    DAS      165    DATAP   6110    DB       166    DBA       41
DBC      172    DBS      173    DECIM   6120    DFTCA  34106    DMCAL  34110
DQUEU   6105    DSBEN 105762    ENTRY 105400    ERR   105473    ERRF      76
ESCF      73    ETSF      74    EXIT  105517    FINDL   6123    FIX     6121
FLAG  105467    FLAGC   6102    FLAG.      4    FLOAT   6122    FOUND 105720
FREEN   6107    GETBY   6124    HALTS   6153    INBYT   6125    INSTB   6126
IOCAL  34103    IOP        6    ISA2D   6127    ISA2L   6130    JEXIT 105673
JFLTO    151    JSAVA 105623    LACNT   4000    LAFSE  13000    LALCO  47400
LALLO   1400    LATOE  36000    LBAKU 106000    LBILD   5000    LBUIL   4400
LCALL  75000    LCHAN  41000    LCHFL  30000    LCHSU  61000    LCLEA   7400
LCLOS   7000    LCLPY  76000    LCNVA  11400    LCNVD  12000    LCOMM  33400
LDALC   2000    LDALL   1000    LDB7A 114000    LDB7B 114400    LDB7C 115000
LDB7D 115400    LDB7E 116000    LDB7F 116400    LDB7G 117000    LDB7H 117400
LDB7I 120000    LDEKE  52400    LDELE   3400    LDIRE  50400    LDLTP  20400
LDREN  37400    LDSB1    400    LDSB2  22400    LDSB3  47000    LDSB4  65000
LDSB5  77000    LDSB6 106400    LDSB7 113400    LECHO  37000    LEO87 105400
LERRO  23000    LFAUL    400    LFFIL   2400    LFIXD  57400    LFNDC 112000
LFNDL  20000    LFOFI  17000    LGETR  10000    LGHOP 107400    LGHOS 107000
LGMUX  16000    LHCON  17400    LIBCA  44400    LIBEN  45000    LIBTR  45400
LIDAT 103000    LINE  105470    LINE.      5    LINPS 105472    LINP.      7
LINSZ 105471    LINS.      6    LLINK  35400    LLOAD  34400    LLOGI  32000
LLUIN 112400    LMAPB  73000    LMDEO  65000    LMDE1  66000    LMDE5  71400
LMRC3  56400    LMRFH  57000    LMRFI  54000    LMTAP  55400    LMTAS  54400
LMTFP  56000    LMTFY  60400    LMTNX  55000    LMTPL  60000    LOADD   6131
LOPEN   6000    LOPNM  13400    LPATQ 110000    LPEXP  23400    LPFAB  72000
LPFLN  73400    LPFNA   3000    LPFRL  72400    LPFSE  67000    LPFSH  70000
LPFSX  70400    LPLOG  24400    LPPWR  33000    LPRAN  36400    LPRCO  71000
LPSIN  25400    LPSQR  22400    LPTAN  25000    LQIBF  63400    LQICL  63000
LQIOP  62400    LRDFH  26400    LRDIS  31400    LRDSE 110400    LREDC  50000
LREDI  11000    LREDM  14000    LREDP  74000    LRENA  15000    LREOP  53000
LRESO  42000    LRWIT 113000    LRWMB  14400    LRWSX 111400    LS105  77000
LS152 102000    LS153 101000    LS154 100400    LS156 101400    LS157 100000
LSAVE  43000    LSAVP  43400    LSEAB  64000    LSEAR  51000    LSETF  40000
LSHUF  52000    LSIGP  12400    LSING  40400    LSMCS 106400    LSPEC  27000
LSTRI  32400    LSYSC  30400    LTP01 102400    LTP03 104000    LTP04 104400
LTP05 105000    LVMUX  42400    LWRIT  47000    LXMIN  62000    M4DON 105651
M4LP  105642    M5LP  105664    M5LST 105671    M6LP  105707    M7DON 105756
M7LP  105750    MOD2A 105525    MOD3A 105567    MODE  105433    NOTFN 105715
OUTBY   6132    OUTTE   6133    PIB        4    PICK  105435    PTR87 105761
PUTBY   6134    QCHAR   6103    QUEUE   6104    RDALL 105542    RDLIN 105556
READB   6135    RELJM   6136    RETS  105434    RETS2 105565    RETS3 105541
REVIS     14    REVNO 105474    RJSR    6136    RNUM  105477    RTP        7
RUP        5    SAVA1 105500    SAVAA 105501    SAVLN 105730    SBA       40
SCDCA  34147    SPINP   6146    START 105403    STINP   6140    STINT   6147
```

```
STORD   6137   STOUT   6141   TABLE 105615   TASKQ     15   TRAPF   6142
VECTO 105422   VSIZE 105432   WRITB   6143   XGETB   6144   XPUTB   6145
.ABA      14   .BPS      77   .BSA      10   .DA      174   .DA3     175
.DB      176   .DB3     177   .FLTO    152   .HBA      11   .HXA      12
.INFO    100   .INTR    111   .LCM     114   .NRET    112   .SRET    113
.SSA      13
```

```
@     @           @@@@@@@   @@@@@    @@@@@    @@@@@   @@@@@@@
 @   @            @        @     @  @     @  @     @        @
  @ @             @        @     @  @        @     @       @
   @              @@@@@    @     @   @@@@@    @@@@@       @
  @ @             @        @     @        @  @     @     @
 @   @     @@@    @        @     @  @     @  @     @    @        @@@
@     @    @@@    @@@@@@@   @@@@@    @@@@@    @@@@@     @        @@@

 @@@@@    @@@@@    @@@@@      @
@     @  @     @  @     @    @@
@     @        @  @     @   @ @
 @@@@@        @    @@@@@      @
    @     @@@@        @       @
   @     @           @        @
  @      @@@@@@@    @       @@@@@



Spool Queue Line #:  34
IRIS LU/Filename  :  18/X.EOS87.9291

           Printed on/at :  FEB  6, 1990  12:44:49
           For Group/User:  0 , 1
               On Port No:  5

Print control parameters :
    Printer Class code    :  0
    Form Code/paper type  :  ?
    Print Priority (0-9)  :  5
    Starting Page Number  :   1
    This is copy number   :  1
    Keep file (Y/N)       :  Y
    Notify User when done:  N
    Comments, optional    :  For R9.5  RELSE CN
```

```
@     @           @@@@@@@   @@@@@    @@@@@    @@@@@   @@@@@@@
 @   @            @        @     @  @     @  @     @        @
  @ @             @        @     @  @        @     @       @
   @              @@@@@    @     @   @@@@@    @@@@@      @
  @ @             @        @     @        @  @     @     @
 @   @     @@@    @        @     @  @     @  @     @    @        @@@
@     @    @@@    @@@@@@@   @@@@@    @@@@@    @@@@@     @        @@@

 @@@@@    @@@@@    @@@@@      @
@     @  @     @  @     @    @@
@     @        @  @     @   @ @
 @@@@@       @    @@@@@      @
    @     @@@@        @       @
   @     @           @        @
  @      @@@@@@@    @       @@@@@


Spool Queue Line #:  34
IRIS LU/Filename   :  18/X.EOS87.9291

           Printed on/at :  FEB  6,  1990   12:44:59
           For Group/User:  0 ,  1
               On Port No:  5

Print control parameters :
    Printer Class code    :  0
    Form Code/paper type  :  ?
    Print Priority (0-9)  :  5
    Starting Page Number  :   1
    This is copy number   :  1
    Keep file (Y/N)       :  Y
    Notify User when done:   N
    Comments, optional    :  For R9.5  RELSE CN
```

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

  1   TOTAL # DUPLICATE KEYS
  0   TOTAL # DIR. RE-ORGS
224   TOTAL # KEYS INSERTED
  0   TOTAL # ASSEMBLY ERRS
```

```
.ERR       9.040
.NRET      4.034    6.016
.SRET      5.016    8.015
APT        2.047    4.013:
APT.       4.023=   4.024    9.007
APTT.      4.024=   4.025    9.010
APTTM      2.048    3.037    3.046    3.047    4.014:   4.050    5.007
           5.008
BEYON      6.029    7.006:
BLOC.      4.025=   4.026    9.019    9.029    9.032
BLOCK      4.015:   5.026    5.048
BYTE       4.016:   5.031    5.050    6.039    6.044
BYTE.      4.026=   4.027
C377       6.041
C776       6.037    6.054:
C87V1      3.016    5.013:
C87V2      3.017    5.020:
C87V3      3.018    6.024:
C87V4      3.019    7.017:
C87V5      3.020    7.047:
C87V6      3.021    8.020:
C87V7      3.022    9.016:
CONT       3.012    3.052:
DECIM      3.043    5.006
DSBEN      2.045    9.038:
ENTRY      2.043:   2.044    2.045    9.040
```

```
EOS87      2.043
ERR        2.051    3.010    3.040    3.045    4.006    4.034:   4.053
           7.028    7.052
EXIT       4.040    5.016:   5.038    6.048    7.010
FIX        3.044
FLAG       4.017:   5.029    6.006    6.036    6.046
FLAG.      4.027=   4.028
FLOAT      4.049
FOUND      8.034    8.044:
JEXIT      8.015:   8.042    9.034
JSAVA      7.013:   9.012    9.033
LEO87      2.040
LINE       4.018:   5.022    5.034    6.017    7.022    7.033
LINE.      4.028=   4.029    8.048    8.051
LINP.      4.030=   8.027    8.046    9.026    9.027
LINPS      3.007    4.021:   5.015    5.023    5.025    5.035    6.026
           7.020    7.042    7.049    8.012
LINS.      4.029=   4.030
LINSZ      2.054    4.020:   6.014    7.026
M4DON      7.024    7.042:
M4LP       7.035:   7.041
M5LP       8.007:   8.011
M5LST      7.054    8.012:
M6LP       8.032:   8.038
M7DON      9.024    9.032:
M7LP       9.026:   9.031
MOD2A      5.026:   7.043
```

```
MOD3A       6.026:    8.052
MODE        3.011     3.026:    3.053
NOTFN       8.025     8.039:
PICK        3.008     3.033:    5.047     5.049     5.051     6.013
PTR87       8.031     8.032     8.035     8.045     9.036:
RDALL       5.021     5.044:    7.019
RDLIN       5.046     6.011:    6.025     7.048     8.022     9.020
RETS        3.031:    3.036     3.048
RETS2       6.012     6.018     6.020:    8.055     9.011
RETS3       4.048     5.009     5.042:    5.045     6.007
REVIS       2.016=    4.042
REVNO       3.015     4.037:
RNUM        4.038     4.042:
SAVA1       4.039     4.046:    6.043     6.045     6.047     7.013
SAVAA       4.048:    7.009
SAVLN       8.041     8.049     8.054:
START       2.044     2.047:
TABLE       6.050:    8.021     8.026     8.050     9.006     9.017     9.021
VECTO       3.015:    3.016     3.017     3.018     3.019     3.020     3.021
            3.022     3.024
VSIZE       3.024:    3.054
```