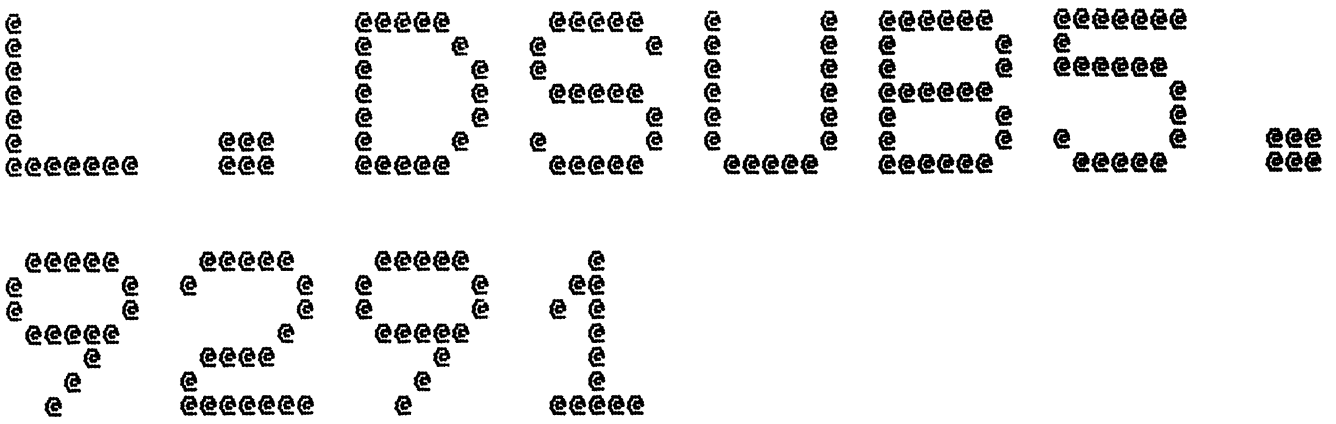
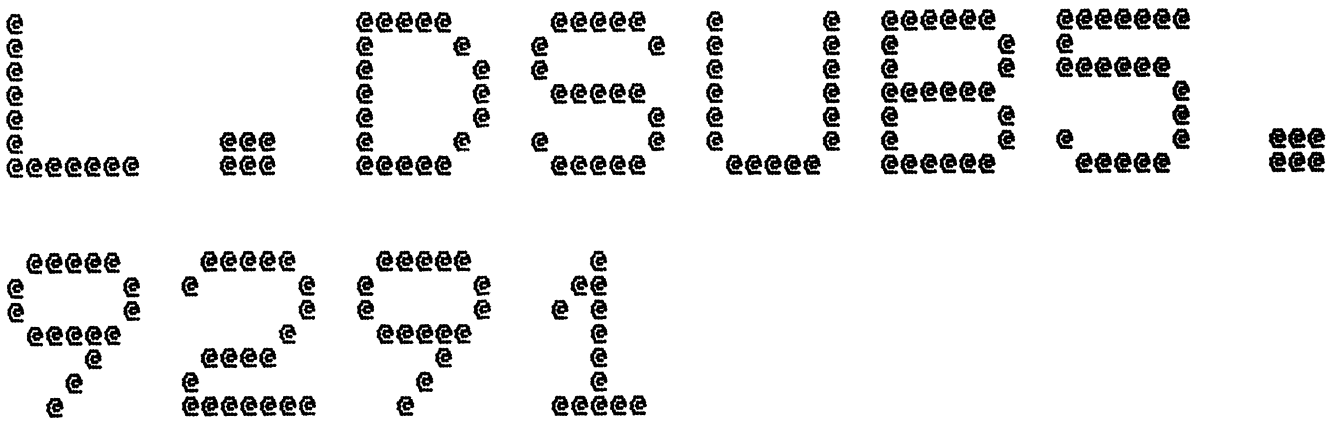


Spool Queue Line #: 27
IRIS LU/Filename : 18/L.DSUB5.9291

Printed on/at : FEB 6, 1990 12:29:11
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 #: 27
IRIS LU/Filename : 18/L.DSUB5.9291

Printed on/at : FEB 6, 1990 12:29:18
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   ;"RxxJCL.DSUB5"
.EOT
.EOT   ;"DSUBDEFS" FOR IRIS
.EOT   ;  DISCSUBS 5 TITLE PAGE
.EOT   ;R9.0 "DISCSUBS" GROUP 5, SOURCE 1
.EOT   ;R9.0 "DISCSUBS" GROUP 5, SOURCE 2
.END
```

```
ASM 18/A.DSUB5.9291!,@18/L.DSUB5.9291!,B050,-B051,B052
FEB  6, 1990    9:58:35


;       Batchfile:   R95JCL.DSUB5

;
;       ; D = 9291
;       ; NAME = DSUB5
;       ; TYPE = 33400
;
;       -R95DEFSPZ
;       -R95DSUBDEFSD
;       R92DSUB5SA
;       R92DS5STYLUS1B
;       R92DS5STYLUS2B
;       *.END
;
                    .EOT   ;"RxxJCL.DSUB5"
```

```
;                        << SI = R92DSUB5SA;  BO = 18/A.DSUB5.9291! >>

;"DISCSUBS" == DISC RESIDENT SUBROUTINES FOR "IRIS"
;  June 17, 1982  (2168)   WAM
;  EDIT by MLR (4275)
;   EDITED 12-FEB-86 BY RDC TO REMOVE STULUS
;   EDITED 16-JUN-87 BY RDC TO REPLACE STULUS

;                      All Rights Reserved
;         Copyright (C) FOR 8.3 1981, POINT 4 Data Corporation
;         Copyright (C) 1983, POINT 4 Data Corporation
;         Copyright (C) 1986, POINT 4 Data Corporation
;        This document contains secret and confidential
;       information of POINT 4 Data Corporation.  It may
;      not be reproduced, used, or disclosed without the
;     prior written permission of POINT 4 Data Corporation

                1       .TXTM  1


;  GROUP #5  -- WORD PROCESSING SUPPORT   - RESTRICTED

            77000 .LOC  LDSB5

     77000  77000       LS105      ;(105)  ST151 (151)
     77001 100000       LS157      ;(157)  ST106 (106)  ST107 (107)
     77002 100400       LS154      ;(154)  ST155 (155)
     77003 101000       LS153      ;(153)
     77004 101400       LS156      ;(156)
     77005 102000       LS152      ;(152)
     77006 102400       LTP01      ;(160)
     77007 103000       LIDAT      ;(161)
     77010 104000       LTP03      ;(162)
     77011 104400       LTP04      ;(163)
     77012 105000       LTP05      ;(164)
     77013 105400       LE087      ;(165)
     77014 106000       LBAKUP     ;(173)

;  GROUP #1   -- "IRIS" SYSTEM COMPONENTS
;  GROUP #2   -- "BASIC" COMPONENTS          - RESTRICTED
;  GROUP #3   -- DATA FILE EXTENSIONS
;  GROUP #4   -- POLYFILE COMPONENTS
;  GROUP #5   -- WORD PROCESSING SUPPORT     - RESTRICTED
;  GROUP #6   -- "SMBASIC" COMPONENTS        - RESTRICTED

                  .EOT  ; DISCSUBS 5 TITLE PAGE
```

; CALL 89 is dsub 105

```
;                        << SI = R92DS5STYLUS1B; BO = 18/A.DSUB5.9291! >>
;  PROGRAM IS USED BY THE EDITOR PACKAGE TO PRINT
;  OUT THE LINE IN THE JUSTIING, FLAG FOR UNDERLINING,
;  FLAG FOR REVERSE PRINTING, AND THE DESIRED WIDTH
;  OF THE OUTPUT IN NO. OF CHARS. USING 12 CHAR./IN.
;  THE UNDER. FLAG SHOULD BE 0 UPON INITAL ENTRY AND
;  NOT CHANGED AFTER THAT. THE REV. PRINT FLAG SHOULD
;  BE 0 UPON INITAL ENTRY AND SET BACK TO 0 WHENEVER
;  A UNFILLED LINE IS PRINTED AND THE CARRIAGE RETURNS
;  TO THE LEFT MARGIN. THE FORMAT OF THE CALL IS AS:
;  CALL ---,A$,B$,C$,W,X,Y,Z
;   A$=STRING TO SUBR., EXCESS CHARS. FROM SUBR.
;   B$=VOID TO SUBR.  , STRING FROM SUBR.
;   C$=STRING OF VARIABLE CODES
;   W =FLAG FOR UNDERLINING
;   X =FLAG FOR REV. PRINTING
;   Y =DESIRED NO. OF CHAR.
;   Z =MIN. HORZ. MOTION INDEX

          77000 .LOC     LS105;BLOCKS 87,88

  77000     105 DSB87:   ST105           ;DISCSUB NBR
  77001       5          S105X-DSB87       ;ENTRY POINT
  77002   40151          ST151
  77003      60          EDITX-DSB87
  77004  177034          DSB87-DSE88

  77005   44436 S105X:   STA      1,AC1.T    ;SAVE AC1
  77006   40434          STA      0,ACO.T
  77007   50435          STA      2,AC2.T
  77010   25001          LDA      1,1,2      ;GET 1ST VAR TYPE
  77011  125112          MOVL#    1,1,SZC    ;MAKE SURE THAT IT IS A NUM.
  77012    2112          JMP      @.NRET     ;  IT'S NOT, SO ERR RETURN
  77013   31000          LDA      2,0,2      ;GET ADDR. OF 1ST NUM.
  77014  102520          SUBZL    0,0        ;LOAD +1 INTO ACO
  77015    6120        DECIMAL
  77016    6121        FIX
  77017    2112          JMP      @.NRET     ;TOO LARGE A NUM., SO ERR RETURN
  77020   20032          LDA      0,C12      ;GET MAX. NO. OF CALLS ACCES.
  77021  106032          SGE      0,1        ;MAKE SURE IS WITHIN RANGE
  77022    2112          JMP      @.NRET     ;  NOT, SO ERR RETURN
  77023    4422          JSR      CTBLE      ;GET CORE ADDR. OF TABLE
  77024  137000          ADD      1,3        ;ADD CALL DISP.
  77025   21777          LDA      0,-1,3     ;GET DISCSUB NO.
  77026  101015          SNZ      0,0        ;MAKE SURE IS NOT ZERO (WOULD BE ERROR)
  77027    2112          JMP      @.NRET
  77030   40407          STA      0,CALTO    ;STORE DISCSUB NO. IN PATH AHEAD
  77031   20411          LDA      0,ACO.T    ;RE-LOAD ACO
  77032   24411          LDA      1,AC1.T
  77033   30411          LDA      2,AC2.T
  77034  151400          INC      2,2        ;MOVE AC2 TO IGNORE 1ST VAR.
  77035  151400          INC      2,2
  77036    6101          CALL
  77037       0 CALTO:   0                   ;WILL BE REPLACED WITH LOOKED-UP DISCSUB NO.
  77040    2112          JMP      @.NRET     ;ON RETURN HERE - ERROR RETURN TO PROGRAM
  77041    2113          JMP      @.SRET     ;GOOD RETURN FROM HERE.
```

```
;                        << SI = R92DS5STYLUS1B; BO = 18/A.DSUB5.9291! >>
77042       0 AC0.T:   0                    ;TEMP CELL FOR AC0
77043       0 AC1.T:   0                    ;TEMP CELL FOR AC1
77044       0 AC2.T:   0                    ;TEMP CELL FOR AC2

77045    5400 CTBLE:   JSR      0,3         ;GET ACTUAL ADDR. OF NEXT CELL
77046     157          ST157                  ;DISC. NO. CALL 1
77047   20106          ST106                  ;DISC. NO. CALL 2
77050   40151          ST151                ;DISC. NO. CALL 3 (EXT.)
77051     154          ST154                  ;DISC. NO. CALL 4
77052     155          ST155                  ;DISC. NO. CALL 5
77053     153          ST153                  ;DISC. NO. CALL 6
77054     156          ST156                    ;DISC. NO. CALL 7
77055       0          0                    ;DISC. NO. CALL 8
77056       0          0                    ;DISC. NO. CALL 9
77057       0          0                    ;DISC. NO. CALL 10

77060   50576 EDITX:   STA      2,NAI1       ; SET UP FOR AUTO ADDRESSING OF ==>
77061    4577          JSR      TABLE      ; GET POINTER TO TABLE
77062   54575          STA      3,NAI2       ; SET UP FOR AUTO ADDRESSING OF TABLE
77063   10574        ISZ     NAI2
77064     401        401
77065   52572          STA      2,@NAI2      ; SAVE POINTER TO CALL VAR. TABLE
77066   24003          LDA      1,C3       ; GET LOOP COUNTER
77067   44572          STA      1,CNT50
77070   35000          LDA      3,0,2      ; GET 1ST STRING ADDR.
77071   26004          LDA      1,@PIB     ; GET REQ. USER STORAGE POINTER
77072     404          JMP      AL1
```

```
;                        << SI = R92DS5STYLUS1B;  BO = 18/A.DSUB5.9291! >>
    77073  10563 ALPHA: ISZ     NAI1
    77074    401         401
    77075  36561           LDA       3,@NAI1      ; GET NEXT STRING ADDR.
    77076 136540 AL1:     SUBOL    1,3           ; SUBTRACT OUT REG.  USER STORAGE POINTER
    77077  10560         ISZ     NAI2
    77100    401         401
    77101  56556           STA       3,@NAI2      ; AND SAVE
    77102  10554         ISZ     NAI1
    77103    401         401
    77104  36552           LDA       3,@NAI1      ; GET NEXT STRING DIM.
    77105 175123           MOVZL     3,3,SNC     ; IS MSB = 1 ??
    77106   2112           JMP       @.NRET      ;   NO  - ERROR RETURN
    77107 175220           MOVZR     3,3         ;   YES - RESTORE WITH MSB = 0
    77110  10547         ISZ     NAI2
    77111    401         401
    77112  56545           STA       3,@NAI2      ; AND SAVE
    77113  14546           DSZ       CNT50       ; DECR.  COUNTER
    77114    757           JMP       ALPHA       ; AND CONTINUE WITH NEXT VAR.
    77115  24552           LDA       1,ADRC1     ;GET ADDR.  OF STRING C
    77116   6144         XGETBYTE
    77117  50551           STA       2,LEADN     ;AND SAVE AS LEAN-IN CHAR.
    77120  24024           LDA       1,C4        ; GET LOOP COUNTER
    77121  44540           STA       1,CNT50
    77122  10534 NUMER: ISZ     NAI1
    77123    401         401
    77124  32532           LDA       2,@NAI1      ; GET NEXT VARIABLE ADDR.
    77125  10531         ISZ     NAI1
    77126    401         401
    77127  26527           LDA       1,@NAI1      ; GET NEXT VARIABLE PREC.
    77130 125132           MOVZL#    1,1,SZC     ; IS MSB = 0 ??
    77131   2112           JMP       @.NRET      ;   NO  - ERROR RETURN
    77132 102520           SUBZL     0,0         ; SET ACO = 1
    77133   6120         DECIMAL
    77134   6121         FIX
    77135 100010           NOP
    77136  10521         ISZ     NAI2
    77137    401         401
    77140  46517           STA       1,@NAI2      ; AND SAVE BIN.  NUMBER
    77141  14520           DSZ       CNT50       ; DECR.  COUNTER
    77142    760           JMP       NUMER       ; AND CONTINUE LOOP
    77143  14531           DSZ       VAR4        ;DECR.  HMI BY 1
    77144 102441           SUBO      0,0,SKP     ; CLEAR ACO
    77145   2112           JMP       @.NRET      ; WAS ONLY 1 BEFORE, SO ERROR RETURN
    77146  40513           STA       0,CNT50     ; NO.  OF CHAR.  NOT COUNTING "_"
    77147  40532           STA       0,CNT51     ; BYTE DISP.  INTO STRING
    77150  40532           STA       0,CNT52     ; POSITION OF 1ST " " IN LAST GROUP
    77151  40532           STA       0,CNT53     ; FLAG THAT LAST CHAR.  WAS NON-" "
    77152  40523         STA     0,NULLF    ;INIT NULL FLAG
    77153  20520         LDA     0,VAR3     ;LENGTH DESIRED
    77154 101015         SNZ     0,0        ;IS IT ZERO NOW ?
    77155 101400         INC     0,0        ;   YES, TREAT SAME AS ONE
    77156  40515         STA     0,VAR3
    77157    402           SKIP
    77160  10521           ISZ       CNT51       ;INCR.  CHAR POINTER
```

```
;                        << SI = R92DS5STYLUS1B;  BO = 18/A.DSUB5.9291!  >>
    77161   20500 COUNT:   LDA      0,CNT50     ;  GET NO.  OF CHAR.  COUNTED
    77162   24511          LDA      1,VAR3      ;  GET NO.  OF CHAR.  WANTED
    77163  106433          SUBZ#    0,1,SNC     ;  IS AC0 <= AC1 ??
    77164     441          JMP      CHLEN       ;   NO   - DONE,  NOW CHANGE LENGTH
    77165   20514          LDA      0,CNT51     ;  GET NEXT CHAR.  POINTER
    77166   24476          LDA      1,DIMA      ;  GET DIM.  OF STRING
    77167  122432          SUBZ#    1,0,SZC     ;  IS DIM.>CHAR.  POINTER ??
    77170    2112          JMP      @.NRET      ;   NO   - NOT ENOUGH CHAR.
    77171   24472          LDA      1,ADRA      ;  GET ADDR.  OF STRING A
    77172  107000          ADD      0,1         ;  GET BYTE POINTER NEXT CHAR.
    77173    6144        XGETBYTE
    77174   10505          ISZ      CNT51
    77175    4527          JSR      HALFT       ;  TEST IF SUB OR SUPER-SCRIPT
    77176     762          JMP      COUNT-1     ;   SHADOW PRINT,  DON'T COUNT
    77177     761          JMP      COUNT-1     ;   OVERSTRIKE,  DON'T COUNT
    77200     760          JMP      COUNT-1     ;   IT IS A SUB-SCRIPT,  DON'T COUNT EITHER
    77201     757          JMP      COUNT-1     ;   IT IS A SUPER-SCRIPT
    77202   34476          LDA      3,C337      ;  GET UNDERLINE CODE.
    77203  172415          SUB#     3,2,SNR     ;  IS CHAR.  UNDERLINE ??
    77204     755          JMP      COUNT       ;   YES - CONT.  WITH NEXT
    77205  151015          MOV#     2,2,SNR     ;  IS THIS CHAR.  A NULL ??
    77206     501          JMP      CNTIO         ;   YES - TREAT IT AS A SPACE
    77207   34056          LDA      3,C240      ;  GET SPACE CHAR.
    77210  172414          SUB#     3,2,SZR     ;  IS CHAR.  A SPACE ??
    77211     407          JMP      CNTS1
    77212   20471 CNTO:  LDA      0,CNT53    ;  GET FLAG
    77213  101015          MOV#     0,0,SNR     ;  DOES FLAG = 1 ??
    77214     406                   JMP      CNTS2       ;   NO   - JUMP OUT
    77215   20444          LDA      0,CNT50
    77216   40464          STA      0,CNT52     ;  SAVE CURR.  COUNT
    77217  102401          SUB      0,0,SKP     ;  ZERO AC0
    77220  102520 CNTS1:     SUBZL    0,0          ;  SET AC0 = +1
    77221   40462          STA      0,CNT53     ;  AND SAVE AS FLAG
    77222   10437 CNTS2:     ISZ      CNT50        ;  INCR.  CHAR.  COUNTER
    77223  151014          MOV#     2,2,SZR     ;  IF LAST CHAR.  WAS A NULL,  THEN DONE
    77224     735          JMP      COUNT
    77225   30432 CHLEN:   LDA      2,NAI2       ;  GET NEXT ADDR.  OF TABLE
    77226   24002          LDA      1,C2        ;  SET 2 IN AC1
    77227   20025          LDA      0,C5        ;  SET 5 (MUL.) IN AC0
    77230    6120        DECIMAL
    77231    6121        FIX
    77232  100010        NOP
    77233   44450        STA      1,CNT53
    77234   20446          LDA      0,CNT52;  GET CHAR.  POSITION OF LAST " "
    77235   20436 CHLN1: LDA      0,VAR3     ;  GET DISIRED WIDTH
    77236  100400        NEG      0,0        ;DECREMENT DESIRED WIDTH ...
    77237  100000        COM      0,0        ;...  BY ONE
    77240  101015        SNZ      0,0        ;IS IT ZERO NOW ?
    77241  101400        INC      0,0        ;   YES,  TREAT SAME AS ONE
    77242   30432        LDA      2,VAR4     ;  GET MIN.  HORZ.  INDEX
    77243    6116        BINMULTIPLY
```

```
;                        << SI = R92DS5STYLUS1B; BO = 18/A.DSUB5.9291! >>
 77244 105005            MOV     0,1,SNR   ; IS WIDTH DESIRED ZERO ??
 77245   2112            JMP     @.NRET    ;  YES - ERROR RETURN
 77246  20434            LDA     0,CNT52   ; GET NO. OF CHAR.
 77247 101015            MOV#    0,0,SNR   ; IS DIVISOR ZERO ??
 77250    500            JMP     MUSTH     ;  YES - TRY TO HYPHENATE.
 77251 100400            NEG     0,0       ;DECREMENT DESIRED WIDTH ...
 77252 100000            COM     0,0       ;...  BY ONE
 77253 101015            SNZ     0,0       ;IS IT ZERO NOW ?
 77254 101400            INC     0,0       ;  YES, TREAT SAME AS ONE
 77255    434            JMP     CHLN2
 77256      0 NAI1:  0    ;LOCAL VARIABLE
 77257      0 NAI2:  0    ;LOCAL VARIABLE

 77260   5400 TABLE: JSR     0,3       ;GET POINTERS TO TABLE
 77261      0 CNT50: 0     ;NOTE:
 77262      0 TABAD: 0     ;      THE FOLLOWING CELLS THRU
 77263      0 ADRA:  0     ;NULLF ARE IN A SPECIAL ORDER.
 77264      0 DIMA:  0     ;DO NOT REARRANG ! ! !
 77265      0 ADRB:  0
 77266      0 DIMB:  0
 77267      0 ADRC1: 0
 77270      0 LEADN: 0     ;1ST GETS DIM C$, GETS OVERLAID BY LEADIN
 77271      0 VAR1:  0
 77272      0 VAR2:  0
 77273      0 VAR3:  0
 77274      0 VAR4:  0
 77275      0 NULLF: 0             ;(0=> NO NULL ENCOUNTERED)
 77276  70000 C.7:      70000   ;  .7 STORED IN 2 WORDS
 77277    200 C.7S:     200     ;  2ND WORD OF C.7
 77300    337 C337:     337
 77301      0 CNT51:    0
 77302      0 CNT52:    0
 77303      0 CNT53:    0
 77304      0 TEMP:     0
 77305      0 TEMP1:    0
 77306      0 TEMP2:    0

 77307  54766 CNTIO: STA     3,NULLF   ;SET "NULL WAS ENCOUNTERED"
 77310    702         JMP     CNTO
 77311   6115 CHLN2: BINDIVIDE
 77312 175400         INC     3,3
 77313  54442         STA     3,SP1     ;THIS IS SPACING CODE FOR 1ST PART
 77314 122440         SUBO    1,0       ;SUB REMAIN FROM DIVISOR
 77315 101400         INC     0,0       ;SINCE WIDTH WAS DECREMENTED
 77316  40441         STA     0,SP2PO   ;AND SAVE DIVIDING LINE POSITION
 77317  54437         STA     3,SP2     ;THIS IS SPACING CODE FOR 2ND PART
 77320  10436         ISZ     SP2       ;INCR. CODE BY 1
 77321    437         JMP     READY

 77322      0 NAI5:  0     ;LOCAL VAR
 77323      0 NAI4:  0     ;LOCAL VAR
```

```
;                        << SI = R92DS5STYLUS1B; BO = 18/A.DSUB5.9291! >>
77324  20744 HALFT:   LDA      0,LEADN    ;SEE IF CHAR. IS A LEAD-IN CODE
77325 112414           SUB#     0,2,SZR
77326   1404           JMP      4,3        ; IT ISN'T, SO SKIP PAST ALL VALID RETURNS
77327  54757           STA      3,TEMP2    ;SAVE RETURN ADDRESS
77330 125400           INC      1,1        ;INCR BYTE POINTER
77331   6144         XGETBYTE
77332  50753           STA      2,TEMP1    ;SAVE 2ND CHAR
77333  20024           LDA      0,C4       ;GET LOOP COUNTER
77334  40750           STA      0,TEMP     ;AND STORE
77335  24732           LDA      1,ADRC1     ;GET ADDR. OF STRING C
77336 125400           INC      1,1        ;INCR. TO NEXT POSITION
77337   6144         XGETBYTE
77340  20745           LDA      0,TEMP1    ;GET SOURCE CHAR
77341 112415           SUB#     0,2,SNR    ;DO THEY MATCH
77342   2744           JMP      @TEMP2
77343  10743           ISZ      TEMP2
77344  14740           DSZ      TEMP
77345    771           JMP      .-7
77346  30722           LDA      2,LEADN    ;RELOAD INITIAL CHAR.
77347   2737           JMP      @TEMP2     ;AND RETURN(HAS BEEN INCR. PAST ALL VALID RETURNS)

77350  24723 MUSTH:   LDA      1,VAR3     ; GET MAX LENGTH OF STRING
77351  44731           STA      1,CNT52    ; AND SAVE
77352    663         JMP     CHLN1

77353      0 CNT60: 0
77354    210 C210:  210
77355      0 SP1:     0
77356      0 SP2:     0
77357      0 SP2P0:   0

77360  20712 READY:   LDA      0,VAR2     ; GET REV. LINE FLAG
77361 101015           MOV#     0,0,SNR    ; SKIP IF FLAG ON
77362  14775           DSZ      SP2P0      ; DECR. CHAR. POINTER
77363 100010           NOP
77364 102440           SUB0     0,0        ; CLEAR AC0
77365  40766           STA      0,CNT60    ; ZERO COUNTERS
77366  40557           STA      0,CNT61
77367  40560           STA      0,CNT63
77370  40560           STA      0,CNT64
77371   4667           JSR      TABLE      ; GET POINTER TO TABLE
77372  54730           STA      3,NAI5      ; AND SAVE IN AN AUTO INCR.
77373   4576                   JSR     TABL2      ; GET POINTER TO 2ND TABLE
77374  54727           STA      3,NAI4      ; AND SAVE IN AN AUTO INCR.
77375  20034           LDA      0,C14      ; GET LOOP COUNTER OCT 14 DEC. 12
77376  40705           STA      0,CNT53
77377  10723 MOVE:  ISZ     NAI5
77400    401         401
77401  22721         LDA      0,@NAI5     ; GET NEXT WORD
77402  10721         ISZ     NAI4
77403    401         401
77404  42717           STA      0,@NAI4     ; AND SAVE IN NEXT LOC.
77405  14676           DSZ      CNT53      ; DECR. LOOP COUNTER
77406    771           JMP      MOVE       ; AND CONTINUE LOOP
77407  24571           LDA      1,.ADRC    ;GET ADDR OF STRING C
```

```
;                        << SI = R92DS5STYLUS1B; BO = 18/A.DSUB5.9291! >>
   77410  20025           LDA     0,C5        ;GET DISP. IN DESIRED
   77411 107000           ADD     0,1
   77412   6144         XGETBYTE
   77413  50566           STA     2,.UND2     ;AND SAVE
   77414  20666           LDA     0,CNT52     ; GET CHAR. POINTER
   77415  40531           STA     0,CNT62     ; AND SAVE WITHIN RANGE
   77416  24565           LDA     1,.VAR2     ; GET FLAG FOR REV. PRINTING
   77417 125014           MOV#    1,1,SZR     ; IS FLAG SET ??
   77420    404           JMP     BEGIN       ;  YES - DON'T ADD CODES NOW
   77421   4512           JSR     INSRT       ; ENTER CODES TO NEW STRING
   77422  24734           LDA     1,SP2       ; GET CODE FOR 2ND PART
   77423  44732           STA     1,SP1       ; AND STORE IN 1ST PART
   77424  24521 BEGIN:    LDA     1,CNT61     ; GET CURR. DISP. TO STRING A
   77425  20547           LDA     0,.ADRA     ; GET BEGINNING ADDR. OF A
   77426 107000           ADD     0,1         ; GET BYTE ADDR. OF CURR BYTE
   77427   6144         XGETBYTE
   77430   4674           JSR     HALFT       ; TEST IF SUPER OR SUB-SCRIPT
   77431    565           JMP     SHDOW       ; IT IS A SHADOW PRINT
   77432    565           JMP     OVSTK       ; IT IS AN OVERSTRIKE
   77433    575           JMP     SCRDW       ; IT IS A SUB-SCRIPT
   77434    567           JMP     SCRUP       ;  IT IS A SUPER-SCRIPT
   77435  10510           ISZ     CNT61       ; INCR. POINTER
   77436  34555           LDA     3,FBLNK     ; GET REQ. SPACE CODE
   77437  20056           LDA     0,C240      ; GET SPACE CODE
   77440 172415           SUB#    3,2,SNR     ; IS CURR. CHAR A REQ. SPACE ??
   77441 111000           MOV     0,2         ;  YES - REPLACE WITH REQ. SPACE
   77442  34540           LDA     3,.VAR1     ; GET UNDERLINE FLAG
   77443  20547           LDA     0,C337A     ; GET UNDERLINE CODE
   77444  24504           LDA     1,CNT64     ; GET FLAG THAT LAST WAS UNDERLINE CODE.
   77445 112414           SUB#    0,2,SZR     ; IS CURR. CHAR A UND. CODE ??
   77446    414           JMP     CONTS       ;  NO - SKIP NEXT ROUTINE
   77447 152440           SUBO    2,2         ; PUT ZERO IN AC2
   77450 175015           MOV#    3,3,SNR     ; IS FLAG ON ??
   77451 152520           SUBZL   2,2         ;  NO - PUT +1 IN AC2 TO TURN ON
   77452 125015           MOV#    1,1,SNR     ; IS FLAG ZERO
   77453    404           JMP     CONT4       ;  YES - SKIP TO NEXT
   77454 171222           MOVZR   3,2,SZC     ; IF EVEN (0 OR 2) LEAVE AS IS.
   77455 151400           INC     2,2         ; IF A ONE, INCR. TO TWO
   77456 151120           MOVZL 2,2
   77457  40471 CONT4:    STA     0,CNT64     ; SAVE NON-ZERO FLAG
   77460  50522           STA     2,.VAR1     ; AND SAVE AC2 AS FLAG
   77461    743           JMP     BEGIN       ; AND CONTINUE WITH NEXT CHAR.
```

```
;                        << SI = R92DS5STYLUS1B;  BO = 18/A.DSUB5.9291! >>
    77462  24671 CONTS: LDA     1,CNT60   ; GET NO. OF CHARS. NOT COUNT "-"
    77463  20463         LDA     0,CNT62   ; GET NO. OF CHARS. NEEDED
    77464 106432         SUBZ#   0,1,SZC   ; IS AC0 > AC1 ??
    77465    503         JMP     JFINI     ; NO - MUST BE DONE UND.LINING
    77466 175014         MOV#     3,3,SZR    ; IS THE UNDERLINE FLAG ON ??
    77467    404            JMP      FLGON      ;   YES - JUMP TO NEXT ROUTINE
    77470 141000            MOV      2,0        ; MOVE BYTE JUST READ IN
    77471   4432            JSR      INSER      ; AND PUT BYTE IN STRING B
    77472    415            JMP      CONT2      ; AND CONTINUE WITH TESTING
    77473  34056 FLGON:     LDA      3,C240     ; GET A SPACE CODE
    77474 156415            SUB#     2,3,SNR    ; IS THE BYTE A SPACE ??
    77475    405            JMP      UNLIN      ;   YES - JUST UNDERLINE
    77476 141000            MOV      2,0        ; MOVE BYTE JUST READ IN
    77477   4424            JSR      INSER      ; AND PUT BYTE IN STRING B
    77500  20654            LDA      0,C210     ; GET BACKSPACE CHAR.
    77501   4422            JSR      INSER      ; AND PUT INTO STRING B
    77502  20510 UNLIN:     LDA      0,C337A    ; GET UNDERLINE CODE
    77503  34477            LDA      3,.VAR1    ; GET UNDERLINE FLAG
    77504 175213            MOVR#    3,3,SNC    ; WE KNOW ITS NON-ZERO, SO IF ITS EVEN
    77505  20474            LDA      0,.UND2    ;   IT MUST BE TWO
    77506   4415            JSR      INSER      ; AND PUT BYTE IN STRING B
    77507  24644 CONT2:     LDA      1,CNT60    ; GET POINTER TO THIS BYTE
    77510  10643            ISZ      CNT60      ; INCR. CHAR. COUNTER
    77511 102400            SUB      0,0        ;RESET LAST CHAR FLAG
    77512  40436            STA      0,CNT64
    77513  20644            LDA      0,SP2P0    ; GET POINTER TO SOMEWHERE
    77514 106414            SUB#     0,1,SZR    ; ARE THE POINTERS EQUAL ??
    77515    707            JMP      BEGIN      ; AND CONTINUE WITH NEXT CHAR.
    77516  20465            LDA      0,.VAR2    ; FLAG TO REV. THE STRING
    77517 101014            MOV#     0,0,SZR    ; IS THE STRING TO BE REV. ??
    77520    431            JMP      CONT3      ;   YES - ENTER INFO. BACKWARDS
    77521   4412            JSR      INSRT      ;   NO  - ENTER CONTROL INFO.
    77522    702            JMP      BEGIN      ; AND CONTINUE WITH NEXT CHAR.
    77523  30424 INSER:     LDA      2,CNT63    ; GET POINTER TO NEXT POS.
    77524  24453            LDA      1,.DIMB    ; GET DIM. OF STRING B
    77525 132432            SUBZ#    1,2,SZC    ; IS THERE ENOUGH ROOM ??
    77526   2112            JMP      @.NRET     ;   NO - ERROR RETURN
    77527  24447            LDA      1,.ADRB    ; GET BEGINNING OF STRING B
    77530 147000            ADD      2,1        ; GET BYTE ADDR. OF CURR BYTE
    77531  10416            ISZ      CNT63
    77532   2145            JMP      @XPUTBYTE&377; STORE BYTE, WILL RETURN TO CALLER
    77533  54437 INSRT:     STA      3,RETR1    ; SAVE RETURN ADDR.
    77534  20454            LDA      0,C233     ; GET 1ST CODE
    77535   4766            JSR      INSER      ; AND STORE
    77536  20453            LDA      0,C237     ; GET 2ND CODE
    77537   4764            JSR      INSER      ; AND STORE
    77540  20615            LDA      0,SP1      ; GET VAR.
    77541  30053            LDA      2,C200     ; GET BIT 8 = 1
    77542 143000            ADD      2,0        ; SET BIT 8 = 1 OF CHAR.
    77543  34427            LDA      3,RETR1    ; RETURN DIRECTLY TO CALLING ROUTINE
    77544    757            JMP      INSER      ; AND STORE

    77545      0 CNT61: 0
    77546      0 CNT62: 0
    77547      0 CNT63: 0
    77550      0 CNT64: 0
```

```
;                              << SI = R92DS5STYLUS1B;  BO = 18/A DSUB5.9291! >>
    77551   4404 CONT3:   JSR     INSR2       ; ENTER CODES BACKWARDS
    77552  24604          LDA     1,SP2       ; GET CODE FOR 2ND PART
    77553  44602          STA     1,SP1       ; AND STORE IN 1ST PART
    77554    650          JMP     BEGIN       ; AND CONTINUE WITH NEXT CHAR.
    77555  20600 INSR2:   LDA     0,SP1       ; GET SPACING VAR.
    77556  54414          STA     3,RETR1     ; SAVE RETURN ADDR.
    77557  30053          LDA     2,C200      ; GET BIT 8 = 1
    77560 143000          ADD     2,0         ; SET BIT 8 = 1 OF CHAR.
    77561   4742          JSR     INSER       ; AND STORE
    77562  20427          LDA     0,C237
    77563   4740          JSR     INSER       ; STORE 2ND CODE
    77564  20424          LDA     0,C233      ; GET 1ST CODE
    77565  34405          LDA     3,RETR1     ; SO WILL RETURN DIRECTLY
    77566    735          JMP     INSER       ; AND STORE 1ST CODE

    77567    635 JBEGI:   JMP     BEGIN       ;ELEVATOR TO BEGINNING
    77570    461 JFINI:   JMP     FINIS       ;ELEVATOR TO END

    77571   5400 TABL2:   JSR     0,3     ;NOTE:
    77572      0 RETR1:   0                 ;      THE FOLLOWING CELLS THRU
    77573      0 .TABA:   0                 ;.NULF ARE IN A SPECIAL ORDER.
    77574      0 .ADRA: 0                   ;DO NOT REARRANGE ! ! !
    77575      0 .DIMA: 0
    77576      0 .ADRB:   0
    77577      0 .DIMB:   0
    77600      0 .ADRC:   0
    77601      0 .UND2:   0
    77602      0 .VAR1:   0
    77603      0 .VAR2:   0
    77604      0 .VAR3:   0
    77605      0 .VAR4:   0
    77606      0 .NULF: 0
    77607      0 CD2:   0
    77610    233 C233:    233
    77611    237 C237:    237
    77612    337 C337A:   337
    77613    340 FBLNK:   340
    77614    304 C304:    304
    77615    325 C325:    325

    77616 100010 SHDOW:   NOP     ;BLANK ROUTINE RIGHT NOW.
    77617 100010 OVSTK:   NOP     ;BLANK ROUTINE RIGHT NOW.
    77620  10725 COMM:    ISZ     CNT61       ;INCR POINTER PAST
    77621  10724          ISZ     CNT61       ;LEADIN AND CODE CHAR.
    77622    745          JMP     JBEGI       ;AND CONT. WITH NEXT CHAR.
    77623  34760 SCRUP:   LDA     3,.VAR2     ; GET REV. LINE FLAG
    77624 175014          MOV#    3,3,SZR     ; IS FLAG SET ??
    77625    417          JMP     SCRUB       ;  YES - USE DOWN CODE BACKWARDS
    77626  24766          LDA     1,C304      ; GET A 'D' CODE
    77627    405          JMP     SCRD2       ; AND PUT IN STRING
    77630  34753 SCRDW:   LDA     3,.VAR2     ; GET REV. LINE FLAG
    77631 175014          MOV#    3,3,SZR     ; IS FLAG SET ??
    77632    414          JMP     SCRDB       ;  YES - USE UP CODE BACKWARDS
    77633  24762          LDA     1,C325      ; GET A 'U' CODE
    77634  20754 SCRD2:   LDA     0,C233      ; GET AN ESC CODE
```

```
;                    << SI = R92DS5STYLUS1B; BO = 18/A.DSUB5.9291! >>

77635  44752 SCRD3:  STA     1,CD2       ; SAVE 2ND CODE FOR LATER
77636   4665        JSR    INSER        ; MOVE 1ST CODE INTO STRING
77637  20750         LDA     0,CD2       ; RECOVER 2ND CODE
77640   4663         JSR     INSER       ; MOVE 2ND CODE INTO STRING
77641 102400         SUB     0,0         ; RESET LAST CHAR FLAG
77642  40706         STA     0,CNT64
77643    755         JMP     COMM        ; AND CONTINUE WITH NEXT CHAR.
77644  20751 SCRUB:  LDA     0,C325      ; GET A 'U' CODE
77645    402         JMP     SCRDB+1
77646  20746 SCRDB:  LDA     0,C304      ; GET A 'D' CODE
77647  24741         LDA     1,C233      ; GET AN ESC CODE
77650    765         JMP     SCRD3       ; AND PUT CODES INTO STRING BACKWARDS

77651 102440 FINIS:  SUBO    0,0         ; PUT NULL BYTE IN AC0
77652   4651         JSR     INSER       ; PUT BYTE AT END OF STRING
77653  24730         LDA     1,.VAR2     ; GET REV. PRINT FLAG
77654 125015         MOV#    1,1,SNR     ; IS STRING TO BE REVERSED ??
77655    414         JMP     FINI2       ; NO JUMP TO NEXT ROUTINE
77656  14671         DSZ     CNT63       ; REMOVE NULL BYTE
77657   4676         JSR     INSR2       ; YES - ADD END CODES
77660 102440         SUBO    0,0         ; PUT NULL BYTE IN AC0
77661   4642         JSR     INSER       ; PUT BACK ON END OF STRING
77662  30711         LDA     2,.TABA     ; GET ADDR OF CALL VAR. TABLE
77663 151400         INC     2,2
77664 151400         INC     2,2         ; INCR. TO 2ND ENTRY (RETURN STRING)
77665   6101         CALL
77666    107           ; REVERSE LINE CALL
77667   2112         JMP     @.NRET      ; ERROR RETURN
77670 102441         SUBO    0,0,SKP
77671 102520 FINI2:  SUBZL   0,0         ; SET AC0 = 1
77672  40711         STA     0,.VAR2     ; AND SAVE AS REV. FLAG
77673  30701         LDA     2,.ADRA     ; GET BEG. ADDR. OF STRING
77674  24651         LDA     1,CNT61     ; GET NEXT BYTE TO READ
77675 147000         ADD     2,1         ; GET BYTE ADDR OF NEXT BYTE
77676   6144        XGETBYTE
77677  24056         LDA     1,C240      ; GET SPACE CODE
77700 132414         SUB#    1,2,SZR     ; IS THE BYTE A SPACE ??
77701    403         JMP     FINI1       ;  NO - JUMP TO NEXT ROUTINE
77702  10643         ISZ     CNT61       ; INCR. POINTER
77703    770         JMP     FINI2+2     ; CONTINUE LOOP
```

```
;                        << SI = R92DS5STYLUS1B; BO = 18/A.DSUB5.9291! >>
    77704  30670 FINI1:   LDA     2,.ADRA     ; GET BEG. ADDR A
    77705  20640          LDA     0,CNT61     ; GET NEXT BYTE TO MOVE
    77706  24667          LDA     1,.DIMA     ; GET TOTAL LENGTH OF A
    77707  34677        LDA     3,.NULF
    77710 175015        SNZ     3,3         ;DID VAR3 LENGTH CONTAIN A NULL ?
    77711    405        JMP     FINI5       ;   NO, RET LEFTOVER IN A$
    77712  24662        LDA     1,.ADRA     ;   YES, SET  A$ ...
    77713 102400        SUB     0,0         ;...  TO NULL STRING ...
    77714   6145        XPUTBYTE            ;...  TO SHOW ...
    77715    406        JMP     FINI3       ;...  NO LEFTOVER


    77716 143000 FINI5: ADD     2,0         ; GET START ADDR. OF MOVE
    77717 147000          ADD     2,1       ; GET END ADDR. OF MOVE
    77720   6101          CALL
    77721 100016          MOVBYTES          ; MOVE STRING DOWN TO BEGINNING
    77722 140000          140000  ; BOTH SOURSE AND DESTINATION RELATIVE
    77723  24657        FINI3: LDA     1,.VAR1    ; START CONVERTING FLAGS BACK
    77724 102440          SUBO    0,0
    77725   6122        FLOAT
    77726  30645          LDA     2,.TABA     ; GET ADDR. OF CALL VAR. TABLE
    77727  25007          LDA     1,7,2       ; GET PREC. OF VAR. 1
    77730  31006          LDA     2,6,2       ; GET ADDR. OF VAR. 1
    77731 102440          SUBO    0,0         ; PUT 0 IN AC0
    77732   6120        DECIMAL
    77733  24650          LDA     1,.VAR2     ; CONVERT REV. LINE FLAG BACK
    77734 102440          SUBO    0,0
    77735   6122        FLOAT
    77736  30635          LDA     2,.TABA     ; GET ADDR. OF CALL VAR. TABLE
    77737  25011          LDA     1,11,2      ; GET PREC. OF VAR. 2
    77740  31010          LDA     2,10,2      ; GET ADDR. OF VAR. 2
    77741 102440          SUBO    0,0         ; PUT ZERO INTO AC0
    77742   6120        DECIMAL
    77743   2113          JMP     @.SRET      ; AND GOOD RETURN TO PROGRAM

          77744 DSE88   =             .

              0 .ERR   DSB87+1000<.;BLOCK OVERFLOW TEST
```

```
;                    << SI = R92DS5STYLUS1B;  BO = 18/A.DSUB5.9291! >>


          100000              .LOC LS157

    100000    157 DSBSI:    ST157
    100001      7           STRIX-DSBSI
    100002  20106           ST106
    100003     10           FINDX-DSBSI
    100004    107           ST107
    100005    176           REVSX-DSBSI
    100006 177453           DSBSI-DSESI

;  PROGRAM TO FIND LOCATION OF SUBSTRING IN ANOTHER STRING
;  CALL SEQUENCE IS     CALL ---,A$,B$,X
;  ROUTINE FINDS STARTING LOCATION OF A$ IN B$ STARTING AT X+1
;  OUTPUT IS    X = LOC.    OR X = 0 IF NOT FOUND.

    100007 102521 STRIX:    SUBZL    0,0,SKP    ;  PUT +1 IN AC0
    100010 102440 FINDX:    SUBO     0,0        ;  ZERO AC0
    100011  40464           STA      0,CNT3
    100012  36004           LDA      3,@PIB     ;  LOAD REG. USER STORAGE POINTER
    100013  25000           LDA      1,0,2      ;  LOAD POINTER TO STRING A
    100014 166540           SUBOL    3,1        ;  SUBTRACT USERAGE STORAGE DISPLACEMENT
    100015  44447           STA      1,ADDA     ;  AND STORE
    100016  21002           LDA      0,2,2      ;  LOAD POINTER TO STRING B
    100017 162540           SUBOL    3,0        ;  SUBTRACT USER STORAGE DISPLACEMENT
    100020 100400           NEG      0,0
    100021 100000           COM      0,0        ; DECR. POINTER BY ONE
    100022  40443           STA      0,ADDB     ;  AND STORE
    100023  35001           LDA      3,1,2      ;  LOAD DIMENSION OF STRING A
    100024 175123           MOVZL    3,3,SNC    ;  TEST IF MSB = 1
    100025   2112           JMP      @.NRET     ;    ERROR RETURN
    100026 175220           MOVZR    3,3        ;    YES - RESTORE WITH MSB = 0
    100027 137000           ADD      1,3        ;  ADD BEGINNING ADDR.  TO DIM.
    100030  54437           STA      3,AD       ;  AND STORE
    100031  35003           LDA      3,3,2      ;  LOAD DIMENSION OF STRING B
    100032 175123           MOVZL    3,3,SNC    ;  TEST IF MSB = 1 ??
    100033   2112           JMP      @.NRET     ;    NO - ERROR RETURN
    100034 175220           MOVZR    3,3        ;    YES - RESTORE WITH MSB = 0
    100035 117000           ADD      0,3        ;  ADD BEGINNING ADDR.  TO DIM.
    100036  54432           STA      3,BD       ;  AND STORE
    100037  25005           LDA      1,5,2      ;  LOAD PREC.  OF X
    100040 125132           MOVZL#   1,1,SZC    ;  TEST IF MSB = 0 ??
    100041   2112           JMP      @.NRET     ;    NO - ERROR RETURN
    100042  44427           STA      1,CD       ;  AND SAVE
    100043 102520           SUBZL    0,0        ;  SET AC0 = +1
    100044  31004           LDA      2,4,2      ;  LOAD ADDRESS OF VAR.  X
    100045  50421           STA      2,ADRC     ;  AND STORE
```

```
;                    << SI = R92DS5STYLUS1B;  BO = 18/A.DSUB5.9291!  >>

100046   6120          DECIMAL
100047   6121          FIX
100050   2112             JMP      @.NRET
100051  30424             LDA      2,CNT3
100052 112015             ADC#     0,2,SNR    ;THE ONLY WAY TO NOT SKIP IS IF
100053 126400             SUB      1,1        ; AC0=0 AND AC2=1
100054  40422             STA      0,REVFL
100055  30410             LDA      2,ADDB     ; GET BEGINNING ADDR. OF B
100056 147000             ADD      2,1        ; ADD TO STARTING BYTE
100057  44413             STA      1,CNT      ; STORE STARTING LOC.
100060  24404             LDA      1,ADDA     ; GET BEGINNING ADDR. OF A
100061   6144          XGETBYTE
100062  50415             STA      2,CHARA    ; AC2 AND CHARA HAVE 1ST CHAR
100063    430             JMP      CONT       ; JUMP OVER STORAGE TABLE

100064      0 ADDA:    0
100065      0 ADDB:    0
100066      0 ADRC:    0
100067      0 AD:      0
100070      0 BD:      0
100071      0 CD:      0
100072      0 CNT:     0
100073      0 CNT1:    0
100074      0 CNT2:    0
100075      0 CNT3:    0
100076      0 REVFL:   0
100077      0 CHARA:   0
100100      0 CHARB:   0

100101   6144 CONT1: XGETBYTE          ;GET NEXT BYTE
100102  34775             LDA      3,CHARA    ; GET 1ST CHAR OF STRING A
100103 151015             MOV#     2,2,SNR    ; IS THIS BYTE A NULL ??
100104    422             JMP      NOTF       ;  YES - JUMP TO NOT FOUND
100105  20770             LDA      0,CNT3
100106  24770             LDA      1,REVFL
100107 107234             ADDZR#   0,1,SZR
100110 156405             SUB      2,3,SNR
100111 156415             SUB#     2,3,SNR
100112    417             JMP      MAYBE
100113  20763 CONT:       LDA      0,REVFL
100114 101014             MOV#     0,0,SZR
100115    404             JMP      .+4
100116  10754             ISZ      CNT
100117  34751             LDA      3,BD
100120    403             JMP      .+3
100121  14751             DSZ      CNT
100122  34743             LDA      3,ADDB
100123  24747             LDA      1,CNT
100124 136414             SUB#     1,3,SZR
100125    754             JMP      CONT1
100126  24737 NOTF:       LDA      1,ADDB     ;  YES - GET BEGINNING ADDR. OF B
100127  44743             STA      1,CNT      ; AND STORE AS LOC.
100130    434             JMP      FOUND      ; AND CONVERT BACK TO X
```

```
;                    << SI = R92DS5STYLUS1B; BO = 18/A.DSUB5.9291! >>

100131  24741 MAYBE:   LDA     1,CNT
100132  44741           STA     1,CNT1      ; STORE CURRENT COUNT INTO STRING B
100133 101014           MOV#    0,0,SZR     ; SKIP IF FLAG IS ZERO
100134    430           JMP     FOUND       ; CALL 1 - FOUND WHA WAS NEEDED.
100135  20727           LDA     0,ADDA      ; GET BEG. ADDR. STRING A
100136  40736           STA     0,CNT2      ; AND HOLD
100137  10734 MAY1:     ISZ     CNT1        ; INCR. POINTER TO STRING B
100140  10734           ISZ     CNT2        ; INCR. POINTER TO STRING A
100141  20726           LDA     0,AD        ; GET DIM. OF STRING A
100142  24732           LDA     1,CNT2      ; GET POINTER TO STRING A
100143 106032           ADCZ#   0,1,SZC     ; IS DIM. >= POINTER ??
100144    420           JMP     FOUND       ;   NO - MUST HAVE FOUND STRING
100145   6144         XGETBYTE
100146 151015           MOV#    2,2,SNR     ; IS THIS BYTE A NULL ??
100147    415           JMP     FOUND       ;  YES - MUST HAVE FOUND STRING
100150  50730           STA     2,CHARB     ;   NO - SAVE CHAR. UNTIL LATER
100151  20717           LDA     0,BD        ; GET DIM. OF STRING B
100152  24721           LDA     1,CNT1      ; GET POINTER TO STRING B
100153 106032           ADCZ#   0,1,SZC     ; IS DIM. >= POINTER ??
100154    737           JMP     CONT        ;   NO - CONTINUE WITH MAIN SEARCH
100155   6144         XGETBYTE
100156 151015           MOV#    2,2,SNR     ; IS THIS BYTE A NULL ??
100157    734           JMP     CONT        ;  YES - CONTINUE WITH MAIN SEARCH
100160  24720           LDA     1,CHARB     ; GET SAVED CHAR. FROM A
100161 132415           SUB#    1,2,SNR     ; ARE THE CHAR. THE SAME ??
100162    755           JMP     MAY1        ;  YES - CONTINUE WITH NEXT CHAR.
100163    730           JMP     CONT        ;   NO - CONTINUE WITH MAIN SEARCH

100164 102400 FOUND:    SUB     0,0         ; MOVE 0 TO AC0
100165  24705           LDA     1,CNT       ; LOAD BIN. COUNT TO AC1
100166  30677           LDA     2,ADDB      ; GET START ADDR. OF B
100167 146400           SUB     2,1         ; SUBTRACT OUT STRING DISP.
100170   6122         FLOAT
100171 102400           SUB     0,0         ; MOVE 0 TO AC0
100172  24677           LDA     1,CD        ; LOAD PREC. OF X
100173  30673           LDA     2,ADRC      ; LOAD WITH ADDR. OF X
100174   6120         DECIMAL
100175   2113           JMP     @.SRET      ; RETURN +1
```

```
;                    << SI = R92DS5STYLUS1B; BO = 18/A.DSUB5.9291! >>

; PROGRAM TO REVERSE A LINE OF CHARACTERS IN THE SAME STRING
; CALL SEQUENCE IS     CALL ---,A$
; THE ROUTINE THEN FINDS THE 1ST NULL CHARACTER AND
; REVERSES ALL THE PREVIOUS CHAR.  AND ADDS SPECIAL
; REVERSE LINE PRINTING CODES FOR THE DIABLO 1610/1620.

100176  35000 REVSX:   LDA     3,0,2       ; LOAD POINTER TO STRING
100177  26004           LDA     1,@PIB      ; GET REG. USER STORAGE POINTER
100200 136540           SUBOL   1,3         ; SUBTRACT REG. USER STORAGE AMOUNT
100201  54454           STA     3,ADDK      ; AND SAVE
100202  35001           LDA     3,1,2       ; LOAD DIM OF STRING
100203 175123           MOVZL   3,3,SNC     ; TEST IF MSB = 1 ??
100204   2112           JMP     @.NRET      ;   NO - ERROR RETURN

100205 175220           MOVZR   3,3         ;  YES - RESTORE WITH MSB = 0
100206  54450           STA     3,KD        ; AND SAVE
100207 126400           SUB     1,1         ; CLEAR AC
100210  44447           STA     1,CNT10     ; CLEAR TEMP CELLS
100211  44447           STA     1,CNT11
100212  20443           LDA     0,ADDK      ; GET ADDR.  OF STRING
100213  24444           LDA     1,CNT10
100214  20441 SIZE:     LDA     0,ADDK      ; GET ADDR.  OF STRING
100215 107000           ADD     0,1         ; GET DISP.  TO CURR BYTE
100216   6144         XGETBYTE
100217 151015           MOV#    2,2,SNR     ; IS BYTE ZERO ?? (END OF STRING)
100220    407           JMP     STRT1       ;  YES - JUMP TO FINISH SETUP
100221  10436           ISZ     CNT10       ;   NO - CONTINUE ON WITH SEARCH
100222  24435           LDA     1,CNT10     ; GET READ FOR END TEST
100223  34433           LDA     3,KD        ; LOAD DIM.  OF STRING
100224 136414           SUB#    1,3,SZR     ; IS IT THE END OF DIM SIZE ??
100225    767           JMP     SIZE        ;   NO - CONTINUE WITH SEARCH
100226   2112           JMP     @.NRET      ; ERROR - NO ROOM FOR REV.  CODES
100227  30430 STRT1:    LDA     2,CNT10     ; SET UP FOR IF ENOUGH ROOM
100230  24025           LDA     1,C5        ; NO.  OF CODES TO ADD
100231 147000           ADD     2,1
100232  34424           LDA     3,KD        ; LOAD AC3 WITH DIM.  OF STRING
100233 166432           SUBZ#   3,1,SZC     ; IS IT LESS THAN THE DIMENSION
100234   2112           JMP     @.NRET      ;   NO - ERROR BRANCH HOME
100235  24026           LDA     1,C6        ; NO.  OF LOOPS NEEDED
100236  44423           STA     1,CNT12     ; HOLD FOR COUNTING
100237   4425           JSR     REVCD       ; FIND LOCATION OF CODE TABLE
100240  54423           STA     3,NAI3      ;  AND SAVE IN A AUTO-ICR.
100241  14420 NEXT:     DSZ     CNT12       ; DECR.  LOOP COUNTER
100242    402           JMP     .+2
100243    430           JMP     STRT2       ; JUMP DONE AND START REV.
100244  24413           LDA     1,CNT10     ; GET NEXT BYTE POINTER
100245  20410           LDA     0,ADDK      ; GET BYTE ADDR.  OF STRING
100246 107000           ADD     0,1         ; ADD STARTING BYTE ADDR.
100247  10414         ISZ     NAI3        ; PSEUDO AUTO INCREMENT
100250 100010         NOP
100251  22412           LDA     0,@NAI3     ;  GET NEXT CHAR TO INSERT
100252   6145         XPUTBYTE
100253  10404           ISZ     CNT10       ; INCR.  BYTE POINTER
100254    765           JMP     NEXT        ; AND CONTINUE
```

```
;                    << SI = R92DS5STYLUS1B; BO = 18/A.DSUB5.9291! >>

100255       0 ADDK:    0
100256       0 KD:      0
100257       0 CNT10:   0
100260       0 CNT11:   0
100261       0 CNT12:   0
100262       0 CHARK:   0
100263       0 NAI3:  0                  ;PSEUDO AUTO INCREMENT CELL

100264    5400 REVCD:   JSR      0,3          ; SET AC3 TO POINTER TO TABLE
100265       0          0
100266     266          266                  ; CODES TO REV PRINT
100267     233          233                  ; CODES ARE ENTERED BACKWARDS
100270     210          210                  ; AND REVERSED AFTER ADDED TO
100271     212          212                  ; THE END OF STRING
100272       0          0

100273   14764 STRT2:   DSZ      CNT10        ; RESET POINTER TO FIRST NULL
100274   14763          DSZ      CNT10        ; RESET POINTER TO LAST NON-NULL
100275   24763 STRT:    LDA      1,CNT11      ; GET DISP. TO BYTE TO SWITCH
100276   30761          LDA      2,CNT10      ; TOTAL NO. OF CHARS.
100277  132400          SUB      1,2          ; GET DISP. OF BYTE TO SWITCH WITH
100300   50761          STA      2,CNT12      ; AND SAVE
100301  146432          SUBZ#    2,1,SZC      ; IS BYTE TO SWITCH >= TO SWITCH WITH
100302    2113          JMP      @.SRET       ;   YES - JUMP DONE
100303   20752          LDA      0,ADDK       ; GET BYTE ADDR OF STRING
100304  107000          ADD      0,1          ; GET BYTE ADDR OF CURR BYTE
100305    6144        XGETBYTE
100306   50754          STA      2,CHARK      ; AND HOLD
100307   20746          LDA      0,ADDK       ; GET BYTE ADDR. OF STRING
100310   24751          LDA      1,CNT12      ; GET BYTE POSITION OF 2ND BYTE
100311  107000          ADD      0,1          ; GET BYTE ADDR OF 2ND BYTE
100312    6144        XGETBYTE
100313   20747          LDA      0,CHARK      ; GET READY TO WRITE 1ST BYTE
100314   50746          STA      2,CHARK      ; HOLD 2ND BYTE
100315    6145        XPUTBYTE
100316   20737          LDA      0,ADDK       ; GET BYTE ADDR OF STRING
100317   24741          LDA      1,CNT11      ; GET BYTE POSITION OF 1ST BYTE
100320  107000          ADD      0,1          ; GET BYTE ADDR OF 1ST BYTE
100321   20741          LDA      0,CHARK      ; GET READY TO WRITE 2ND BYTE
100322    6145        XPUTBYTE
100323   10735          ISZ      CNT11        ; INCR. POINTER TO NEXT CHAR.
100324     751          JMP      STRT         ; AND CONTINUE

        100325 DSESI   = .          ; END OF "DISCSUBS" BLOCK #SI

             0 .ERR    DSBSI+400<.          ; BLOCK OVERFLOW TEST

               .EOT  ;R9.0 "DISCSUBS" GROUP 5, SOURCE 1
```

```
;                    << SI = R92DS5STYLUS2B; BO = 18/A.DSUB5.9291! >>

              4 PEDA = 4      ;******* Another KLUDGE to support STYLUS

;  PROGRAM IS USED TO EXPAND A STRING IN ORDER TO PUT
;  IN A BACKSPACE AND UNDERLINE FROM THE START IF X<>0 OR FROM THE 1ST UNDERLINE CHAR.
;  AND CONTINUE UNTIL THE NEXT UNDERLINE CHAR OR
;  END OF STRING.  IF THE END OF STRING COMES 1ST,
;  THE VAR.  X IS LEFT AT 1 FOR RE-ENTRY WITH THE
;  NEXT STRING.  IF THE STRING ISN'T DIM.  LONG ENOUGH
;  FOR THE EXTRA CHAR.  ADDED,  THE END CHAR'S ARE
;  DROPPED.  B$ IS THE STRING OF VARIABLE CODES.
;  THE FORMAT OF CALL IS AS   CALL ---, A$, B$, X

          100400          .LOC    LS154 ;BLOCK 89

  100400     154 DSB89:   ST154
  100401       5          UNLNX-DSB89
  100402     155          ST155
  100403     353          LNLHX-DSB89
  100404  177410          DSB89-DSE89

  100405   50525 UNLNX:   STA     2,CT022    ;  SAVE TEMP CELL
  100406   35000          LDA     3,0,2      ;  LOAD POINTER TO STRING
  100407   22004          LDA     0,@PIB     ;  GET REG.  USER STORAGE POINTER
  100410  116540          SUBOL   0,3        ;  SUBTRACT OUT DISPLACEMENT
  100411   54510          STA     3,ADRS     ;  AND SAVE
  100412   35001          LDA     3,1,2      ;  LOAD DIM.  OF STRING
  100413  175123          MOVZL   3,3,SNC    ;  TEST IF MSB = 1 ??
  100414    2112          JMP     @.NRET     ;   NO  - ERROR RETURN
  100415  175220          MOVZR   3,3        ;   YES - RESTORE WITH MSB = 0
  100416   54504          STA     3,SD       ;  AND SAVE
  100417   35003          LDA     3,3,2      ;  LOAD PREC.  OF B$
  100420  175133          MOVZL#  3,3,SNC    ;  TEST IF MSB = 0 ??
  100421    2112          JMP     @.NRET     ;   NO  - ERROR RETURN
  100422   25002          LDA     1,2,2      ;  LOAD ADDR.  OF B$
  100423  106540          SUBOL   0,1
  100424   34026          LDA     3,C6       ;GET LOOP COUNTER
  100425   54504          STA     3,CNT21    ;STORE
  100426    4512          JSR     TABL       ;GET ADDR.  OF TABLE
  100427   54471          STA     3,NAI6      ;SAVE,  USING AUTO-INCR.  CELL
  100430    6144 LOOP:    XGETBYTE           ;GET NEXT BYTE OF STRING
  100431   10467        ISZ     NAI6      ;PSEUDO AUTO INCREMENT
  100432  100010        NOP
  100433   52465          STA     2,@NAI6     ;AND STORE IN NEXT CELL
  100434  125400          INC     1,1        ;INCR.  TO NEXT BYTE
  100435   14474          DSZ     CNT21      ;DECR.  LOOP COUNTER
  100436     772          JMP     LOOP       ;AND CONT.  LOOP
  100437   30473          LDA     2,CT022    ;RELOAD AC2
  100440   25005          LDA     1,5,2      ;  LOAD PREC.  OF VAR.  X
  100441  125132          MOVZL#  1,1,SZC    ;  TEST IF MSB = 0 ??
```

```
;                        << SI = R92DS5STYLUS2B;  BO = 18/A.DSUB5.9291! >>
   100442   2112             JMP      @.NRET       ;  NO  - ERROR RETURN
   100443  44464             STA      1,TD         ; AND STORE
   100444  31004             LDA      2,4,2        ; LOAD POINTER TO VAR.
   100445  50461             STA      2,ADRT       ; AND SAVE
   100446 102520             SUBZL    0,0          ; GET +1 INTO AC0
   100447   6120          DECIMAL
   100450   6121          FIX
   100451   2112             JMP      @.NRET       ; ERROR RETURN
   100452  44451             STA      1,FLAG       ; STORE X INTO FLAG
   100453 102440             SUB0     0,0
   100454  40455             STA      0,CNT21      ; CLEAR COUNTER
   100455  40455             STA      0,CT022      ; CLEAR COUNTER
   100456    475             JMP      BGIN

   100457  54471 SHFT2:      STA      3,SHFTR      ; SAVE RETURN ADDR.
   100460  24451             LDA      1,CNT21      ; GET POINTER TO NEXT
   100461  44447             STA      1,CNT20      ; AND USE 2ND TEMP POINTER
   100462  10447             ISZ      CNT21        ; INCR.  MAIN BYTE POINTER
   100463  10446             ISZ      CNT21
   100464  20435 SHIFT:      LDA      0,ADRS       ; GET BYTE ADDRESS OF STRING
   100465  24443             LDA      1,CNT20      ; GET BYTE DISP.  OF NEXT
   100466 107000             ADD      0,1          ; GET BYTE ADDRESS OF NEXT
   100467   6144          XGETBYTE
   100470  20444             LDA      0,CHARU      ; SHIFT CHAR.  OUT OF STACK
   100471  34442             LDA      3,CHART      ; START SHIFTING STACK
   100472  54442             STA      3,CHARU
   100473  50440             STA      2,CHART
   100474  24434             LDA      1,CNT20      ; GET BYTE DISP.  OF THIS CHAR.
   100475   4414             JSR      STCHR        ; STORE CHAR.
   100476   2452             JMP      @SHFTR       ; END OF STRING, JUMP OUT
   100477 101015             MOV#     0,0,SNR      ; WAS LAST CHAR.  A NULL ??
   100500   2450             JMP      @SHFTR       ;  YES - JUMP OUT
   100501  10427             ISZ      CNT20        ; INCR BYTE POINTER
   100502    762             JMP      SHIFT        ; AND CONTINUE WITH SHIFT
   100503  20056 CSNT2:      LDA      0,C240       ; GET A SPACE CODE
   100504   6145          XPUTBYTE
   100505  30056             LDA      2,C240       ; FAKE ROU.  SO IT THINKS IT READ A SPACE
   100506    464             JMP      CSNT5        ; AND CONTINUE MAINSTREAM
   100507  24422 PUTCH:      LDA      1,CNT21      ; GET CURR.  BYTE POINTER
   100510  10421             ISZ      CNT21        ; INCR.  TO NEXT POSITION
   100511  30411 STCHR:      LDA      2,SD         ; GET STRING DIM.
   100512 146432             SUBZ#    2,1,SZC      ; IS DIM.  > CURR.  POS.  ??
   100513   1400             JMP      0,3          ;  NO - NON-SKIP RETURN
   100514  30405             LDA      2,ADRS       ; GET BEG.  STRING ADDR.
   100515 147000             ADD      2,1          ; GET BYTE ADDR.
   100516 175400             INC      3,3          ; TO DO A SKIP-RETURN
   100517   2145             JMP      @XPUTBYTE&377; STORE BYTE, WILL RETURN DIRECT TO CALLER
```

```
;                        << SI = R92DS5STYLUS2B;  BO = 18/A.DSUB5.9291! >>
100520       0 NAI6:  0                    ;PSEUDO AUTO INCREMENT CELL
100521       0 ADRS:    0
100522       0 SD:      0
100523       0 FLAG:    0                      ;  1 = UNDERLINE   0 = DON'T
100524     337 UNCD:    337                    ;  UNDERLINE CODE
100525     210 BKCD:    210                    ;  BACKSPACE CODE
100526       0 ADRT:    0
100527       0 TD:      0
100530       0 CNT20:   0
100531       0 CNT21: 0
100532       0 CTO22:   0
100533       0 CHART:   0
100534       0 CHARU:   0
100535     233 K233:    233
100536     304 K304:    304
100537     325 K325:    325
100540    5400 TABL:    JSR       0,3
100541     340 FBLKN:   340
100542       0 LEEDN:   0         ;GETS READ IN LEAD-IN.
100543       0 SHADO:   0         ;GETS SHADOW CODE.  (NOT USED BY SUBR.)
100544       0 OVERS:   0         ;GETS OVERSTRIKE CODE.  (NOT USED BY SUBR.)
100545       0 DWCD:    0         ;GETS DOWN OR SUBSCRIPT CODE
100546       0 UPCD:    0         ;GETS UP OR SUPERSCRIPT CODE
100547       0 UND2:    0         ;GETS DOUBLE UNDERLINE CHAR.
100550       0 SHFTR:   0

100551     706 JSHFT:   JMP       SHFT2       ;  ELEVATOR TO SHIFT 2 SUBROUTINE

100552   10757          ISZ       CNT21       ;  INCR.  TO NEXT POSITION
100553   24756 BGIN:   LDA      1,CNT21     ;  GET NEXT TEST POSITION
100554   20746          LDA       0,SD        ;  GET STRING DIMENSION
100555  106432          SUBZ#     0,1,SZC     ;  IS IT THE END OF STRING ??
100556     542          JMP       FINS1       ;   YES - FINISH UP
100557   20742          LDA       0,ADRS      ;  LOAD STRING ADDRESS
100560  107000          ADD       0,1         ;  ADD CURR.  DISPLACEMENT
100561    6144        XGETBYTE
100562  151015          MOV#      2,2,SNR     ;  IS IT A NULL CHAR ??
100563     535          JMP       FINS1       ;   YES - FINS1H UP
100564   20755          LDA       0,FBLKN     ;  GET FORCED BLANK CODE
100565  112415          SUB#      0,2,SNR     ;  IS THIS FORCED BLANK ??
100566     715          JMP       CSNT2       ;   YES - CONV.  TO SPACE
100567   20753          LDA       0,LEEDN     ;GET LEADIN CODE
100570  112415          SUB#      0,2,SNR     ;DO THEY MATCH
100571     422          JMP       FNDLN       ;  YES-JUMP TO FOUND LEADIN
100572   34731 CSNT5:   LDA       3,FLAG
100573   20731          LDA       0,UNCD      ;  TEST ON UNDERLINE CODE
100574  112414          SUB#      0,2,SZR     ;  IS IT AN UNDERLINE CODE ??
100575     501          JMP       CSNT        ;   NO   - CONTINUE
100576   24734          LDA       1,CTO22     ;  GET FLAG THAT LAST WAS UNDERLINE CODE
100577  152440          SUBO      2,2         ;  ZERO AC2
100600  175015          MOV#      3,3,SNR     ;   YES, TOGGLE FLAG.  IS IT SET ??
100601  152520          SUBZL     2,2         ;  SET AC2 = 1
100602  125015          MOV#      1,1,SNR     ;IS FLAG ZERO
100603     404          JMP       CSNT1       ;   YES - CONT ON.
100604  171222          MOVZR     3,2,SZC     ;  IF EVEN (0 OR 2) LEAVE AS IS.
```

```
;                        << SI = R92DS5STYLUS2B; BO = 18/A.DSUB5.9291! >>
   100605 151400            INC     2,2          ; IF A ONE, INC. TO TWO
   100606 151120            MOVZL   2,2
   100607  40723 CSNT1:     STA     0,CTO22      ;SAVE NON-ZERO FLAG
   100610  50713            STA     2,FLAG       ; STORE FLAG
   100611   4437            JSR     MOVE1        ; MOVE BYTES DOWN 1
   100612    741            JMP     BGIN         ; AND CONTINUE LOOP
   100613  44720 FNDLN:     STA     1,CHART      ;SAVE AC1
   100614  50720            STA     2,CHARU      ;ALSO SAVE AC2
   100615 125400            INC     1,1          ;INCR. AC1 TO NEXT BYTE IN STRING
   100616   6144          XGETBYTE
   100617  20727            LDA     0,UPCD       ;IS IT A SUPERSCRIPT
   100620 112415            SUB#    0,2,SNR
   100621    415            JMP     CSNT3        ; YES, SO HANDLE
   100622  20723            LDA     0,DWCD       ;IS IT A SUBSCRIPT
   100623 112415            SUB#    0,2,SNR
   100624    414            JMP     CSNT4        ; YES, SO HANDLE
   100625  20716            LDA     0,SHADO      ;IS IT A SHADOW PRINT
   100626 112415            SUB#    0,2,SNR
   100627    416            JMP     CSNT6        ; YES, SO DON'T UNDERLINE
   100630  20714            LDA     0,OVERS      ;IS IT AN OVERSTRIKE
   100631 112415            SUB#    0,2,SNR
   100632    413            JMP     CSNT6        ; YES, SO DON'T UNDERLINE
   100633  24700            LDA     1,CHART      ;NOT LEGAL, SO JUST PRINT AS IS
   100634  30700            LDA     2,CHARU      ;RETURN BOTH AC'S TO ORIGINAL
   100635    735            JMP     CSNT5        ;AND CONT MAINSTREAM.

   100636  20700 CSNT3:     LDA     0,K304       ; GET A -1/2 LINE FEED CODE
   100637    402            JMP     CSNT4+1
   100640  20677 CSNT4:     LDA     0,K325       ; GET A +1/2 LINE FEED CODE
   100641   6145          XPUTBYTE
   100642  24671            LDA     1,CHART      ;GET POINTER TO 1ST CHAR
   100643  20672            LDA     0,K233       ;GET 1ST CHAR
   100644   6145          XPUTBYTE
   100645 176400 CSNT6:     SUB     3,3
   100646  10663            ISZ     CNT21        ;INCR. POINTER
   100647    427            JMP     CSNT         ;AND RETURN TO NEXT CHAR.
   100650  54700 MOVE1:     STA     3,SHFTR      ; SAVE RETURN ADDR.
   100651  30660            LDA     2,CNT21      ; GET THIS BYTE ADDR.
   100652 141400            INC     2,0          ; GET NEXT BYTE ADDR.
   100653  24647            LDA     1,SD         ; GET STRING DIMENSION
   100654  34645            LDA     3,ADRS       ; GET BEG. BYTE ADDR.
   100655 163000            ADD     3,0          ; GET BYTE ADDR. NEXT BYTE
   100656 167000            ADD     3,1          ; GET BYTE ADDR. END BYTE
   100657 173000            ADD     3,2          ; GET BYTE ADDR. THIS BYTE
   100660   6101            CALL                 ; AND SHIFT STRING DOWN ONE
   100661 100016            MOVBYTES
   100662 140000            140000               ; SOURCE & DEST. ARE RELATIVE
   100663 151015            MOV#    2,2,SNR      ; WAS LAST CHAR. A NULL
   100664   2664            JMP     @SHFTR       ;  YES
   100665 105000            MOV     0,1          ; MOVE POINTER TO LAST CHAR. OF SOURCE TRANSFERED
   100666 102440            SUBO    0,0          ; PUT NULL IN AC0
   100667   6145          XPUTBYTE
   100670   2660            JMP     @SHFTR       ; AND RETURN
```

```
;                        << SI = R92DS5STYLUS2B; BO = 18/A.DSUB5.9291! >>
100671 175213 UNDLN:   MOVR#    3,3,SNC    ; IS NON-ZERO, SO IF EVEN, IT MUST BE 2
100672  20655           LDA      0,UND2     ; SO USE DOUBLE UNDERLINE CODE.
100673   4614           JSR      PUTCH      ; STORE CHAR.
100674    424           JMP      FINS1      ; END OF STRING, JUMP OUT
100675    656           JMP      BGIN       ; CONTINUE LOOP WITH NEXT CHAR.
100676 126400 CSNT:    SUB      1,1
100677  44633           STA      1,CTO22    ; RESET DOUBLE UNDERLINE TEST
100700  24635         LDA     1,K233
100701 132415         SNE     1,2        ;ESC SEQUENCE ?
100702    426         JMP     CSNT7      ;   YES
100703 175015           MOV#     3,3,SNR    ; IS THE FLAG SET ??
100704    646           JMP      BGIN-1     ;   NO  - CONTINUE WITH NEXT
100705  24056           LDA      1,C240     ;   YES - LOAD SPACE CODE
100706 146415           SUB#     2,1,SNR    ; IS CHAR. A SPACE ??
100707    762           JMP      UNDLN      ;   YES - REPLACE WITH UNDLN CODE
100710  10621           ISZ      CNT21      ; INCR POINTER TO NEXT CHAR.
100711 175213           MOVR#    3,3,SNC    ; IS NON-ZERO, SO IF EVEN, IT MUST BE 2
100712  20635           LDA      0,UND2     ; SO USE DOUBLE UNDERLINE CODE.
100713  40620           STA      0,CHART    ; SAVE AS 2ND CODE TO GO OUT
100714  20611           LDA      0,BKCD     ; GET BACKSPACE CODE
100715  40617           STA      0,CHARU    ; SAVE AS 1ST CODE TO GO OUT
100716   4633           JSR      JSHFT      ; SHIFT STRING, INSERTING THESE 2 CODES
100717    634           JMP      BGIN       ; AND CONTINUE MAIN LOOP
100720  24603 FINS1:   LDA      1,FLAG     ; CONV. BIN. FLAG TO X
100721 102440           SUBO     0,0        ; ZERO AC0
100722   6122         FLOAT
100723 102440           SUBO     0,0        ; ZERO AC0
100724  24603           LDA      1,TD       ; GET PREC. OF X
100725  30601           LDA      2,ADRT     ; GET ADDR. OF X
100726   6120         DECIMAL
100727   2113           JMP      @.SRET     ; AND RETURN +1
100730  10601 CSNT7: ISZ      CNT21      ;ON ESC SEQUENCE, NO UNDERLINE
100731  10600         ISZ      CNT21      ;STEP PASSED SPECIAL CONTROL CODES
100732    620         JMP      BGIN-1

100733     20         .BLK     20
```

```
;                       << SI = R92DS5STYLUS2B;  BO = 18/A. DSUB5. 9291! >>
;  ADJUST I/O BUFFER LENGTH TO AMOUNT SPECIFIED.
;
;          . DUSR    DFLT = 30 ;CHANGE TO 30 (SPARE) ON 7. 3

                57          FL3. =57;CELL IN PARTITION

  100753   25001 LNLHX:    LDA      1, 1, 2      ;GET DIMENSION OF VARIABLE
  100754  125132            MOVZL#   1, 1, SZC    ;CHECK FOR NUMERIC
  100755    2112            JMP      @. NRET     ;ERROR RETURN
  100756   31000            LDA      2, 0, 2      ;GET VARIABLE ADDRESS
  100757  102520            SUBZL    0, 0         ;SET ACCUM 0 TO 1
  100760    6120          DECIMAL
  100761    6121          FIX
  100762  100010            NOP
  100763  101014          SKZ      0, 0          ;IS SIGN POSITIVE?
  100764  124400          NEG      1, 1          ;NO,  (AC1 HAS ABS VALUE)
  100765   36004          LDA      3, @PIB       ;YES,  GET PARTITION POINTER
  100766   45457          STA      1, FL3. , 3   ;SAVE LENGTH IN PARTITION
  100767    2113            JMP      @. SRET     ;AND RETURN HOME.

          100770 DSE89     = .

               0 . ERR     DSB89+400<.     ;BLOCK OVERFLOW TEST
```

```
;                        << SI = R92DS5STYLUS2B;  BO = 18/A.DSUB5.9291!  >>

;DISCSUB 136  -  CALL 56
;  CALL 56 V$,R,D,C,A,S    OR  CALL 56,V$,R,D,C,A,S,B$;     V$-STRING VARIABLE
;     R -FINE BLOCK DISC ADDRS
;        ZERO ON INITIAL ENTRY - DISCSUB WILL SET R VALUE
;        TO BE RETURNED
;     D -DIRECTORY NUMBER
;     A -DATA RECORD STORAGE ARRAY
;     C -CHANNEL NUMBER OF OPENED FILE
;     S -RETURN STATUS
;     B$-KEY VALUES  (OPTIONAL) MUST BE AT LEAST 512 BYTES LONG.
;        DISPLACEMENT TO INDIVIDUAL KEYS IS AS:
;        V$=B$((2*(K1+1))*I+1)
;         WHERE  K1 IS KEY LENGTH IN WORDS
;         AND    I  IS ITEM NO. OF RECORD NO.  IN THE "A" ARRAY
;


          101000            .LOC LS153     ;BLOCK 90

    101000     153 DSB90:    ST153                 ;DISCSUB NBR
    101001       3           S153X-DSB90         ;ENTRY POINT
    101002  177400           DSB90-DSE90        ;END POINT

    101003   50576 S153X:    STA      2,VARPT    ;ARGUMENT POINTER
    101004  126400            SUB      1,1
    101005   44576            STA      1,DATAR    ;CLEAR DATA RCD SWITCH
    101006   44576            STA      1,NXTR     ;CLEAR BLK FOWARD CHAIN RCD NBR
    101007   44576            STA      1,MODE     ;CLEAR SWITCH
    101010   25015            LDA      1,15,2
    101011  125015            MOV#     1,1,SNR
    101012     420            JMP      D127D
    101013  125123            MOVZL    1,1,SNC
    101014    2112            JMP      @.NRET
    101015  125220            MOVZR    1,1
    101016   20066            LDA      0,C777
    101017  122432            SGR      1,0
    101020    2112            JMP      @.NRET
    101021   25017            LDA      1,17,2     ;GET MODE TYPE DIM.
    101022  125015            MOV#     1,1,SNR    ;MAKE SURE IT WAS SPECIFIED
    101023     405            JMP      D127Q
    101024   31016            LDA      2,16,2     ;GET VAR.  ADDR.
    101025    4512            JSR      JLDV       ;JUMP TO LoaD Variable.
    101026     513            JMP      D127U      ;TOO LARGE A VALUE
    101027   44556            STA      1,MODE     ;SAVE AS OP.  MODE
    101030   30551 D127Q:     LDA      2,VARPT    ;RELOAD AC2
    101031   25014            LDA      1,14,2
```

```
;                    << SI = R92DS5STYLUS2B;  BO = 18/A.DSUB5.9291!  >>

101032  44564 D127D:   STA     1,TADR
101033  21010         LDA     0,10,2      ;PUP ARRAY ADDRS
101034  40553         STA     0,SADR      ;SET START PAC ADDRS
101035  25007         LDA     1,7,2       ;PUP DIM VALUE
101036  31006         LDA     2,6,2       ;PUP CHANNEL ADRS
101037   4500         JSR     JLDV        ;LOAD VARIABLE
101040    501         JMP     D127U       ;ERR 21 RTN
101041 121000         MOV     1,0         ;SET UP FOR CALL
101042   6101         CALL
101043 100002         CHKCHANNEL          ;TEST CHANNEL FOR OPEN
101044    476         JMP     D127S       ;ERR 23 RTN-INVALID CHANNEL NBR

101045 125005         MOV     1,1,SNR     ;OPEN
101046    474         JMP     D127S       ;NO- ERR 23 RTN
101047  50537         STA     2,CADRS     ;SAVE CHANNEL ADRS
101050  30531         LDA     2,VARPT
101051  25003         LDA     1,3,2       ;DIM VALUE
101052  31002         LDA     2,2,2       ;ADDRS
101053   4464         JSR     JLDV        ;LOAD VARIABLE
101054    470         JMP     D127P       ;ERR 22 RTN
101055  44525         STA     1,DISCA     ;SAVE DISC ADDRS
101056 125004         MOV     1,1,SZR     ;FIRST READ
101057    431         JMP     D127H       ;  NO - DON'T ISSUE SEARCH
101060  30521 D127G:   LDA     2,VARPT
101061  25005         LDA     1,5,2       ;DIM VALUE
101062  31004         LDA     2,4,2       ;ADDRS POINTER
101063   4454         JSR     JLDV        ;LOAD FROM CORE TO AC1
101064    455         JMP     D127U       ;INVALID CHAR
101065  34003         LDA     3,C3        ;SEARCH MODE
101066 175300         MOVS    3,3
101067 167000         ADD     3,1         ;BUILD MODE-DIR NBR
101070  30516         LDA     2,CADRS     ;CHNL ADDRS
101071  20510         LDA     0,VARPT
101072   6101         CALL
101073  40061         SEARCH              ;FIRST FINE BLK READ
101074    444         JMP     JD27Y       ;ERR RTN NON SKIP
101075 125004         MOV     1,1,SZR     ;TEST STATUS
101076    442         JMP     JD27Y       ;NON VALID
101077  30502       LDA     2,VARPT   ;CONVERT REC # TO BINARY
101100   4461       JSR     D127X     ;LOAD TYPE AND ADDR
101101   4436       JSR     JLDV      ;LOAD DA AND FIX
101102    442       JMP     D127P     ;INVALID REC #, ERR 22
101103  44500       STA     1,DATAR   ;SAVE REC #
101104  34143       LDA     3,WRITB&377;GET RDA OF FINE BLOCK
101105  35776       LDA     3,-2,3    ;P.HXA
101106  25404       LDA     1,PEDA,3
101107  44473       STA     1,DISCA   ;SAVE RDA
101110  30476 D127H:   LDA     2,CADRS
101111  21000         LDA     0,FLU,2     ;PUP LU NBR
101112  30014         LDA     2,.ABA      ;BUFFER CORE ADRS
101113   6135         READBLOCK           ;READ FINE BLOCK
101114  21000         LDA     0,0,2
101115 101133         MOVZL#  0,0,SNC
101116    575         JMP     D127T       ;NOT A FINE BLOCK-ERR 25
```

```
;                        << SI = R92DS5STYLUS2B;  BO = 18/A.DSUB5.9291! >>
  101117  24472           LDA     1,MASK1
  101120 107705           ANDS    0,1,SNR
  101121    571           JMP     D127R       ;NO KEYS
  101122  44470           STA     1,NKEYS     ;NBR OF ACTIVE KEYS IN BLK PLUS NXT AVAIL
  101123  24064           LDA     1,C377
  101124 107405           AND     0,1,SNR
  101125    565           JMP     D127R       ;SIZE ERR - ZERO
  101126 120400           NEG     1,0
  101127  40465           STA     0,NSKEY
  101130 125400           INC     1,1
  101131  44462           STA     1,SKEY      ;KEY SIZE
  101132 151400           INC     2,2
  101133  21000           LDA     0,0,2
  101134  40450           STA     0,NXTR      ;SAVE NXT RCD NBR
  101135 151400           INC     2,2         ;POINT TO RCD NBR
  101136    472           JMP     D127M

  101137    561 JLDV:     JMP     LDV         ;ELEVATOR TO LDV
  101140    537 JD27Y:JMP     D127Y

  101141 102521 D127U:    SUBZL   0,0,SKP     ;ERR STATUS 21
  101142  20003 D127S:    LDA     0,C3        ;ERR STATUS
  101143    402           JMP     D1270
  101144  20002 D127P:    LDA     0,C2        ;ERR 22
  101145  24450 D1270:    LDA     1,C20D
  101146 107000           ADD     0,1         ;RETURN ERR STATUS GT 20
  101147 102400           SUB     0,0
  101150  40435           STA     0,MODE      ;CLEAR SWITCH SO WON'T WRITE TO DISC.
  101151   4554 D127E:    JSR     STV         ;JUMP TO STORE VARIABLE
  101152  25013           LDA     1,13,2      ;PUP DIM VALUE
  101153  31012           LDA     2,12,2      ;STATUS ADRS
  101154   5400           JSR     0,3         ;FINISH MOVING DA TO CORE
  101155  24427           LDA     1,NXTR      ;GET NEXT BLOCK NO.
  101156 125015           MOV#    1,1,SNR     ;MAKE SURE IS NON-ZERO
  101157    405           JMP     .+5         ;  ZERO,  SO SKIP
  101160   4545           JSR     STV         ;DO 1ST HALF OF STORE
  101161  25003 D127X:LDA     1,3,2
  101162  31002           LDA     2,2,2
  101163   5400           JSR     0,3         ;AND 2ND HALF
  101164  30415           LDA     2,VARPT
  101165  25021           LDA     1,21,2      ;GET VAR TYPE OF R1
  101166 125015           MOV#    1,1,SNR     ;WAS IT SPECIFIED (I.E.  NON-ZERO DIM)
  101167    406           JMP     .+6         ;  NO - DON'T USE
  101170  24412           LDA     1,DISCA     ;GET OLD DISC ADDR.
  101171   4534           JSR     STV         ;MOVE INTO DA
  101172  25021           LDA     1,21,2
  101173  31020           LDA     2,20,2      ;AND MOVE TO CORE
  101174   5400           JSR     0,3
  101175  20410           LDA     0,MODE      ;GET OP.  MODE
  101176 101015           MOV#    0,0,SNR     ;WAS IT MODE 1 OR 2??
  101177   2113           JMP     @.SRET      ;  NO - JUST RETURN WITHOUT WRITE TO DISC.
  101200    417         JMP     D127V
```

```
;                    << SI = R92DS5STYLUS2B; BO = 18/A.DSUB5.9291! >>

101201       0 VARPT:   0                    ;ADDRS OF USER ARGUMENT POINTER LIST
101202       0 DISCA:   0                    ;DISC ADDRS
101203       0 DATAR:   0                    ;REAL DISC RECORD NBR
101204       0 NXTR:    0                    ;RWD CHAIN RCD NBR
101205       0 MODE:    0                    ;TEMP CELL AND MODE OF OP.
101206       0 CADRS:   0                    ;CHANNEL ADDRS
101207       0 SADR:    0                    ;USER RCD ARRAY ADRS
101210       0 BLKA:    0                    ;FINE BLK ADRS
101211   77400 MASK1:   77400                ;MASK FOR NBR OF KEYS
101212       0 NKEYS:   0                    ;NBR OF ACTIVE KEYS THIS FILE
101213       0 SKEY:    0                    ;KEY LENGTH +1
101214       0 NSKEY:   0                    ;NEG. KEY LENGTH
101215      24 C20D:    24                   ;20D CONSTANT
101216       0 TADR:    0                    ;ADDR OF OPTIONAL STRING TO RECEIVE KEY'S

101217   24763 D127V: LDA     1,DISCA   ;GET READY TO WRITE BLOCK BACK TO DISC
101220   30766        LDA     2,CADRS   ;GET CHAN. ADDRS
101221   21000        LDA     0,FLU,2   ;GET LU.
101222   30014        LDA     2,.ABA    ;GET CORE ADDRS
101223    6143        WRITBLOCK         ;AND WRITE BLOCK
101224    2113        JMP     @.SRET    ;AND RETURN TO MAIN PROGRAM
```

```
;                        << SI = R92DS5STYLUS2B; BO = 18/A.DSUB5.9291! >>

101225  30763 D127L:  LDA     2,BLKA
101226  20765         LDA     0,SKEY
101227 113000         ADD     0,2
101230  50760 D127M:  STA     2,BLKA      ;SAVE BLK POINTER
101231  25000         LDA     1,0,2       ;PUP RCD NBR FROM FINE BLK
101232 124235         COMZR#  1,1,SNR     ;END OF ITEMS = -2
101233    447         JMP     D127K       ;YES SETUP FOR RETURN TO CALLER
101234  30747         LDA     2,DATAR
101235 151005         MOV     2,2,SNR     ;FIRST PASS
101236    403         JMP     D127N       ;NO
101237 132404         SUB     1,2,SZR     ;SEARCH 3 RCD MATCH
101240    417         JMP     D127J       ;NO-TEST NEXT  ITEM
101241  50742 D127N:  STA     2,DATAR
101242  30743         LDA     2,MODE      ;GET OP.  MODE
101243 151014         MOV#    2,2,SZR
101244    501         JMP     D127B       ;EITHER MODE 1 OR 2
101245   4460         JSR     STV         ;JUMP TO STORE VARIABLE
101246  25011         LDA     1,11,2      ;GET VAR.  TYPE
101247  30740         LDA     2,SADR      ;GET LOC.  TO STORE TO
101250 141000         MOV     2,0         ;ALSO PUT LOC.  IN AC0
101251 123000         ADD     1,0         ;INCR.  AC0 TO NEXT LOC.
101252  40735         STA     0,SADR      ;RE-SAVE NEW LOC.
101253   5400         JSR     0,3         ;FINISH MOVE OF DA TO CORE
101254  30742         LDA     2,TADR      ;WAS B$ SPEC.
101255 151014         MOV#    2,2,SZR     ;  NO - DON'T MOVE KEY VALUES
101256    512         JMP     D127A       ;MOVE KEY VALUE
101257  14733 D127J:  DSZ     NKEYS       ;FINISHED THIS BLK
101260    745         JMP     D127L       ;NO-BUMP TO NXT FIELD
101261  20026         LDA     0,C6        ;ERR 26
101262  24721         LDA     1,DATAR
101263 125004         MOV     1,1,SZR     ;NO MATCH FLWG SEARCH 3
101264    661         JMP     D127O       ;YES-ERR EXIT
101265  30722         LDA     2,SADR
101266  45000         STA     1,0,2
101267  45001         STA     1,1,2       ;SET END ITEM IN USER ARRAY
101270  24714         LDA     1,NXTR
101271   4434         JSR     STV         ;JUMP TO STORE VARIABLE
101272  25003         LDA     1,3,2       ;GET DIM VALUE
101273  31002         LDA     2,2,2       ;GET ADDR.  VALUE
101274   5400         JSR     0,3         ;FINISH SUBR.
101275 126400         SUB     1,1         ;CLEAR AC1
101276    653         JMP     D127E
```

```
;                        << SI = R92DS5STYLUS2B; BO = 18/A.DSUB5.9291! >>
   101277  20002 D127Y:   LDA     0,C2
   101300 106414          SUB#    0,1,SZR    ;END OF DIR
   101301    650          JMP     D127E      ;NO - ERR FLWG SEARCH MODE 3
   101302  20702 D127K:   LDA     0,NXTR
   101303 101014          MOV#    0,0,SZR    ;TEST VALID END OF ITEM
   101304    411          JMP     D127Z      ;ERR 29
   101305  30702          LDA     2,SADR     ;USER ARRAY
   101306  41000          STA     0,0,2
   101307  41001          STA     0,1,2      ;SET END ITEM
   101310  24002          LDA     1,C2       ;END OF DIR STATUS =2
   101311    640          JMP     D127E      ;EXIT
   101312 102401 D127R:   SUB     0,0,SKP    ;ERR 20
   101313  20025 D127T:   LDA     0,C5
   101314    631          JMP     D1270
   101315  20031 D127Z:   LDA     0,C11      ;ERR 29  NO END OF DIR ITEM
   101316    627          JMP     D1270

   101317      0          0                  ;SAVE RETURN ADDR.
   101320  54777 LDV:     STA     3,LDV-1    ;SAVE RETURN ADDR FROM LOAD VAR.
   101321 102520          SUBZL   0,0
   101322   6120        DECIMAL
   101323  34774          LDA     3,LDV-1    ;RETREIVE RETURN ADDR.
   101324   2121          JMP     @FIX&377      ;WILL RETURN DIRECT TO CALLER
   101325  54772 STV:     STA     3,LDV-1    ;SAVE RETURN ADDR FROM STORE VAR.
   101326 102400          SUB     0,0        ;MAKE SURE SIGN IS POS.
   101327   6122        FLOAT
   101330  30651          LDA     2,VARPT    ;RETREIVE TABLE POINTER
   101331   6766          JSR     @LDV-1     ;RETURN, SAVING NEXT ADDR.  TO FINISH
   101332 102400          SUB     0,0        ;FINALLY MOVE FROM DA TO CORE
   101333   2120          JMP     @DECIMAL&377;JUMP THIS WAY TO RETURNS DIRECT TO CALLER.
   101334  54763 D127C:   STA     3,LDV-1    ;SAVE RETURN ADDRESS
   101335 135000          MOV     1,3        ;SET UP TO USE AC ADDRESSING
   101336  25400          LDA     1,0,3      ;GET NEXT WORD
   101337  45000          STA     1,0,2      ;STORE NEXT WORD
   101340 151400          INC     2,2
   101341 175400          INC     3,3        ;INCR.  TO NEXT WORD
   101342 101404          INC     0,0,SZR    ;INCR.  COUNTER
   101343    773          JMP     .-5        ;NOT DONE SO CONT.  LOOP
   101344   2753          JMP     @LDV-1     ;RETURN WITH ACO = ZERO
   101345  30634 D127B:   LDA     2,VARPT
   101346  25011          LDA     1,11,2
   101347  30640          LDA     2,SADR
   101350 141000          MOV     2,0
   101351 123000          ADD     1,0
   101352  40635          STA     0,SADR
   101353   4745          JSR     LDV
   101354    736          JMP     D127R
   101355  30633          LDA     2,BLKA
   101356 125015          MOV#    1,1,SNR
   101357    733          JMP     D127R
   101360  45000          STA     1,0,2
   101361 151400          INC     2,2
   101362  20632          LDA     0,NSKEY
   101363  24633          LDA     1,TADR
   101364   4750          JSR     D127C
   101365 175400          INC     3,3
```

```
;                          << SI = R92DS5STYLUS2B; BO = 18/A.DSUB5.9291! >>
   101366   54630           STA     3,TADR
   101367     670           JMP     D127J       ;AND CONTINUE MAINSTREAM.
   101370   20624 D127A:    LDA     0,NSKEY
   101371   24617           LDA     1,BLKA
   101372  125400           INC     1,1
   101373    4741           JSR     D127C       ;MOVE KEY OF AC0 LENGTH FROM AC1 TO AC2
   101374   41000           STA     0,0,2       ;STORE ZERO INTO LAST WORD
   101375  151400           INC     2,2         ;MOVE POINTER TO NEXT WORD
   101376   50620           STA     2,TADR      ;AND SAVE
   101377     660           JMP     D127J       ;AND CONTINUE MAINSTREAM.

          101400 DSE90=.                        ;END OF DISCSUB

               0 .ERR   DSB90+400<.             ;BLOCK OVERFLOW TEST
```

```
;                        << SI = R92DS5STYLUS2B; BO = 18/A.DSUB5.9291! >>

;  PROGRAM IS USED BY THE LETTER QUALITY PRINTER PACKAGE TO
;  CALCULATE THE LENGTH OF LINE IN THE JUSTING, FLAG FOR UNDERLINING,
;  FLAG FOR REVERSE PRINTING, AND THE DESIRED WIDTH
;  OF THE OUTPUT IN NO. OF CHARS. USING 12 CHAR./IN.
;  THE UNDER. FLAG SHOULD BE 0 UPON INITAL ENTRY AND
;  NOT CHANGED AFTER THAT. THE REV. PRINT FLAG SHOULD
;  BE 0 UPON INITAL ENTRY AND SET BACK TO 0 WHENEVER
;  A UNFILLED LINE IS PRINTED AND THE CARRIAGE RETURNS
;  TO THE LEFT MARGIN. THE FORMAT OF THE CALL IS AS:
;  NOTE:THIS IS A NEW DISCSUB MADE FROM EDITJUST
;  CALL ---,A$,B$,C$,W,X,Y,Z
;   A$=STRING TO SUBR., EXCESS CHARS. FROM SUBR.
;   B$=VOID TO SUBR.  , STRING FROM SUBR.
;   C$=STRING OF VARIABLE CODES
;   W =FLAG FOR UNDERLINING
;   X =FLAG FOR REV. PRINTING
;   Y =DESIRED NO. OF CHAR.
;   Z =MIN. HORZ. MOTION INDEX

         101400            .LOC    LS156 ;BLOCK 91

 101400     156 DSB91:    ST156
 101401       3           EDITY-DSB91
 101402  177533           DSB91-DSE91

 101403   50554 EDITY:    STA     2,NAI7      ; SET UP FOR AUTO ADDRESSING OF ==>
 101404    4563            JSR     TBLE      ; GET POINTER TO TBLE
 101405   54553            STA     3,NAI8      ; SET UP FOR AUTO ADDRESSING OF TBLE
 101406   10552          ISZ     NAI8
 101407     401          401
 101410   52550            STA     2,@NAI8     ; SAVE POINTER TO CALL VAR.. TBLE
 101411   24003            LDA     1,C3      ; GET LOOP KOUNTER
 101412   44556            STA     1,KNT50
 101413   35000            LDA     3,0,2       ; GET 1ST STRING ADDR.
 101414   26004            LDA     1,@PIB      ; GET REG. USER STORAGE POINTER
 101415     404            JMP     AL2
 101416   10541 BETA:   ISZ     NAI7
 101417     401          401
 101420   36537          LDA     3,@NAI7     ; GET NEXT STRING ADDR.
 101421  136540 AL2:    SUBOL    1,3       ; SUBTRACT OUT REG. USER STORAGE POINTER
 101422   10536          ISZ     NAI8
 101423     401          401
 101424   56534            STA     3,@NAI8     ; AND SAVE
 101425   10532          ISZ     NAI7
 101426     401          401
```

```
;                        << SI = R92DS5STYLUS2B;  BO = 18/A.DSUB5.9291! >>
   101427  36530           LDA     3,@NAI7      ; GET NEXT STRING DIM.
   101430 175123           MOVZL   3,3,SNC    ; IS MSB = 1 ??
   101431   2112           JMP     @.NRET     ;  NO  - ERROR RETURN
   101432 175220           MOVZR   3,3        ;  YES - RESTORE WITH MSB = 0
   101433  10525         ISZ     NAI8
   101434    401         401
   101435  56523           STA     3,@NAI8     ; AND SAVE
   101436  14532           DSZ     KNT50      ; DECR. KOUNTER
   101437    757           JMP     BETA       ; AND CONTINUE WITH NEXT VAR..
   101440  24536           LDA     1,ADRC2     ;GET ADDR. OF STRING C
   101441   6144         XGETBYTE
   101442  50535           STA     2,LEIDN    ;AND SAVE AS LEAN-IN CHAR.
   101443  24024           LDA     1,C4       ; GET LOOP KOUNTER
   101444  44524           STA     1,KNT50
   101445  10512 NUMBR:  ISZ     NAI7
   101446    401         401
   101447  32510         LDA      2,@NAI7     ; GET NEXT VARIABLE ADDR.
   101450  10507         ISZ     NAI7
   101451    401         401
   101452  26505           LDA     1,@NAI7     ; GET NEXT VARIABLE PREC.
   101453 125132           MOVZL#  1,1,SZC    ; IS MSB = 0 ??
   101454   2112           JMP     @.NRET     ;  NO  - ERROR RETURN
   101455 102520           SUBZL   0,0        ; SET AC0 = 1
   101456   6120         DECIMAL
   101457   6121         FIX
   101460 100010           NOP
   101461  10477         ISZ     NAI8
   101462    401         401
   101463  46475           STA     1,@NAI8     ; AND SAVE BIN. NUMBER
   101464  14504           DSZ     KNT50      ; DECR. KOUNTER
   101465    760           JMP     NUMBR      ; AND CONTINUE LOOP
   101466  14515           DSZ     VAR.4      ;DECR. HMI BY 1
   101467 102441           SUBO    0,0,SKP    ; CLEAR AC0
   101470   2112           JMP     @.NRET     ; WAS ONLY 1 BEFORE, SO ERROR RETURN
   101471  40477           STA     0,KNT50    ; NO. OF CHAR. NOT KOUNTING "_"
   101472  40516           STA     0,KNT51    ; BYTE DISP. INTO STRING
   101473  40516           STA     0,KNT52    ; POSITION OF 1ST " " IN LAST GROUP
   101474  40516           STA     0,KNT53    ; FLAG THAT LAST CHAR. WAS NON-" "
   101475  40507           STA     0,NULFG    ;INIT NULL FLAG
   101476  20504           LDA     0,VAR.3    ;LENGTH DESIRED
   101477 101015           SNZ     0,0        ;IS IT ZERO NOW ?
   101500 101400           INC     0,0        ;  YES, TREAT SAME AS ONE
   101501  40501           STA     0,VAR.3
   101502    402           SKIP
   101503  10505           ISZ     KNT51          ;INCR. CHAR POINTER

   101504  20464 KOUNT:    LDA     0,KNT50    ; GET NO. OF CHAR. COUNTED
   101505  20503           LDA     0,KNT51    ; GET NEXT CHAR. POINTER
   101506  24474           LDA     1,VAR.3     ; GET DESIRED LENGTH
   101507 122432           SUBZ#   1,0,SZC    ; DONE ?
   101510    424           JMP     CNLEN      ;  YES
   101511  24461           LDA     1,ODR0     ; GET ADDR. OF STRING A
   101512 107000           ADD     0,1        ; GET BYTE POINTER NEXT CHAR.
   101513   6144         XGETBYTE
   101514  10474           ISZ     KNT51
   101515   4501           JSR     HOLFT      ; TEST IF SUB OR SUPER-SCRIPT
```

```
;                          << SI = R92DS5STYLUS2B; BO = 18/A.DSUB5.9291! >>
101516    765           JMP     KOUNT-1     ;  SHADOW PRINT, DON'T COUNT
101517    764           JMP     KOUNT-1     ;  OVERSTRIKE, DON'T COUNT
101520    763           JMP     KOUNT-1     ;  IT IS A SUB-SCRIPT, DON'T COUNT EITHER
101521    762           JMP     KOUNT-1     ;  IT IS A SUPER-SCRIPT
101522  34465           LDA     3,K337      ; GET UNDERLINE CODE.
101523 172415           SUB#    3,2,SNR     ; IS CHAR. UNDERLINE ??
101524    760           JMP     KOUNT       ;  YES - CONT. WITH NEXT
101525 151015           MOV#    2,2,SNR     ; IS THIS CHAR. A NULL ??
101526  54456         STA     3,NULFG     ;SET "NULL WAS ENKOUNTERED"
101527 102520 KNT1:     SUBZL   0,0         ; SET AC0 = +1
101530  40462           STA     0,KNT53     ; AND SAVE AS FLAG
101531  10437 KNT2:     ISZ     KNT50       ; INCR. CHAR. KOUNTER
101532 151014           MOV#    2,2,SZR     ; IF LAST CHAR. WAS A NULL, THEN DONE
101533    751           JMP     KOUNT
101534  30424 CNLEN:    LDA     2,NAI8       ; GET NEXT ADDR. OF TBLE
101535  24002           LDA     1,C2        ; SET 2 IN AC1
101536  20025           LDA     0,C5        ; SET 5 (MUL.) IN AC0
101537   6120         DECIMAL
101540   6121         FIX
101541 100010           NOP
101542  44450           STA     1,KNT53
101543  20446           LDA     0,KNT52     ; GET CHAR. POSITION OF LAST " "
101544  24424 CHLNI:  LDA     1,KNT50
101545 102400         SUB     0,0         ;SET SIGN POSITIVE
101546   6122         FLOAT   ;CONVERT KOUNT TO BCD
101547  30422         LDA     2,TOBOD     ;CALLING PARMS LIST
101550  25017         LDA     1,17,2      ;CHECK AFTER V4
101551 125112         SSP     1,1         ;IS V5 A STRING ?
101552   2112         JMP     @.NRET      ;   YES, RET BASIC ERROR 38
101553  31016         LDA     2,16,2      ;GET V5 ADDRESS
101554 102400         SUB     0,0         ;STORE BCD
101555   6120         DECIMAL             ;INTO V5
101556   2113         JMP     @.SRET      ;ALL DONE

101557      0 NAI7:   0         ;LOCAL VAR.
101560      0 NAI8:   0         ;LOCAL VAR.
101561  34417 CHLN3:  LDA     3,VAR.1     ;UNDERLINE SEEN ROUTINE
101562 175015         SNZ     3,3         ;TOGGLE VAR.1 ...
101563 176521         SUBZL   3,3,SKP     ;...  WHICH IS THE ...
101564 176400         SUB     3,3         ;...  UNDERLINE MODE
101565  54413         STA     3,VAR.1
101566    716         JMP     KOUNT
```

```
;                    << SI = R92DS5STYLUS2B; BO = 18/A.DSUB5.9291! >>

101567   5400 TBLE:   JSR      0,3        ; GET POINTERS TO TBLE
101570      0 KNT50:   0       ;NOTE:
101571      0 TOBOD:   0       ;      THE FOLLOWING CELLS THRU
101572      0 ODRO:    0       ;NULFG ARE IN SPECIAL ORDER.
101573      0 DIMO:    0       ;DO NOT REARRANGE ! ! !
101574      0 ODRB:    0
101575      0 DIMN:    0
101576      0 ADRC2:    0
101577      0 LEIDN:   0         ;1ST GETS DIM C$, GETS OVERLAID BY LEADIN
101600      0 VAR.1:    0
101601      0 VAR.2:    0
101602      0 VAR.3:    0
101603      0 VAR.4:    0
101604      0 NULFG: 0                  ;(0=> NO NULL ENKOUNTERED)
101605  70000 C..7:     70000     ; .7 STORED IN 2 WORDS
101606    200 C..7S:    200       ; 2ND WORD OF C..7
101607    337 K337:    337
101610      0 KNT51:   0
101611      0 KNT52:   0
101612      0 KNT53:   0
101613      0 TEMP.:    0
101614      0 TEMP3:   0
101615      0 TEMP4:   0

101616  20761 HOLFT:    LDA      0,LEIDN    ;SEE IF CHAR. IS A LEAD-IN CODE
101617 112414            SUB#     0,2,SZR
101620   1404            JMP      4,3        ; IT ISN'T, SO SKIP PAST ALL VALID RETURNS
101621  54774            STA      3,TEMP4    ;SAVE RETURN ADDRESS
101622 125400            INC      1,1        ;INCR BYTE POINTER
101623   6144         XGETBYTE
101624  50770            STA      2,TEMP3    ;SAVE 2ND CHAR
101625  20024            LDA      0,C4       ;GET LOOP KOUNTER
101626  40765            STA      0,TEMP.     ;AND STORE
101627  24747            LDA      1,ADRC2     ;GET ADDR. OF STRING C
101630 125400            INC      1,1        ;INCR. TO NEXT POSITION
101631   6144         XGETBYTE
101632  20762            LDA      0,TEMP3    ;GET SOURCE CHAR
101633 112415            SUB#     0,2,SNR    ;DO THEY MATCH
101634   2761            JMP      @TEMP4
101635  10760            ISZ      TEMP4
101636  14755            DSZ      TEMP.
101637    771            JMP      .-7
101640  30737            LDA      2,LEIDN    ;RELOAD INITIAL CHAR.
101641   2754            JMP      @TEMP4     ;AND RETURN(HAS BEEN INCR. PAST ALL VALID RETURNS)
101642  24740 MUSHT:     LDA      1,VAR.3     ; GET MAX LENGTH OF STRING
101643  44746            STA      1,KNT52    ; AND SAVE
101644    700         JMP     CHLNI

        101645 DSE91 =          .

             0 .ERR   DSB91+400<.;BLOCK OVERFLOW TEST
```

```
;                        << SI = R92DS5STYLUS2B; BO = 18/A.DSUB5.9291! >>

;SUBROUTINE IS USED TO POP ELEMENTS IN AN ARRAY EITHER UP OR DOWN ONE,
;AND INSERT OR DELETE ONE ELEMENT.  IT CAN ALSO BE USED TO CLEAR AN ARRAY
;IF SO DESIRED OR SET IT ALL EQUAL TO ONE VARIABLE.
;
;THE FORMAT OF THE CALL IS AS:
;        CALL    ,A,B,C,D,E
;WHERE   A IS THE ARRAY
;        B IS THE NO. OF ELEMENTS IN 'A'
;        C IS THE SHIFTED ELEMENT POINTER
;        D IS THE MODE  0 IS SET ALL EQUAL TO 'E'
;                       1 IS FOR SHIFTING FROM ELEM. TO BEG. AND INSERT AT ELEM.
;                       2 IS FOR SHIFTING FROM ELEM. TO END  AND INSERT AT ELEM.
;                       3 IS FOR SHIFTING FROM END TO ELEM.  AND DELETE
;        E IS THE VALUE INSERTED OR CONTSTANT USED FOR MODE 0.
;
          102000            .LOC    LS152 ;BLOCK 92

  102000     152 DSB92:    ST152
  102001       3           PUSHY-DSB92
  102002  177626           DSB92-DSE92

  102003    4452 PUSHY:    JSR     TABL3
  102004   54463           STA     3,NAI9
  102005   50447           STA     2,VARPY
  102006   50445           STA     2,AC2
  102007   20025           LDA     0,C5
  102010   40456           STA     0,CNT5
  102011   25001 LOOOP:     LDA     1,1,2
  102012   31000           LDA     2,0,2
  102013  125112           SSP     1,1
  102014    2112           JMP     @.NRET
  102015  102520           SUBZL   0,0
  102016    6120         DECIMAL
  102017    6121         FIX
  102020  100010           NOP
  102021   10446         ISZ     NAI9      ;PSEUDO AUTO INCREMENT
  102022  100010         NOP
  102023   46444           STA     1,@NAI9
  102024   10427           ISZ     AC2
  102025   10426           ISZ     AC2
  102026   30425           LDA     2,AC2
  102027   14437           DSZ     CNT5
  102030     761           JMP     LOOOP
  102031   30423           LDA     2,VARPY
  102032   25000           LDA     1,0,2
  102033   44423           STA     1,ADDRA
  102034   31001           LDA     2,1,2
  102035   50422           STA     2,SIZEA
  102036   20423           LDA     0,VARC
  102037    6116         BINMULTIPLY
  102040  123000           ADD     1,0
  102041   40423           STA     0,SHFPT
  102042   20416           LDA     0,VARB
```

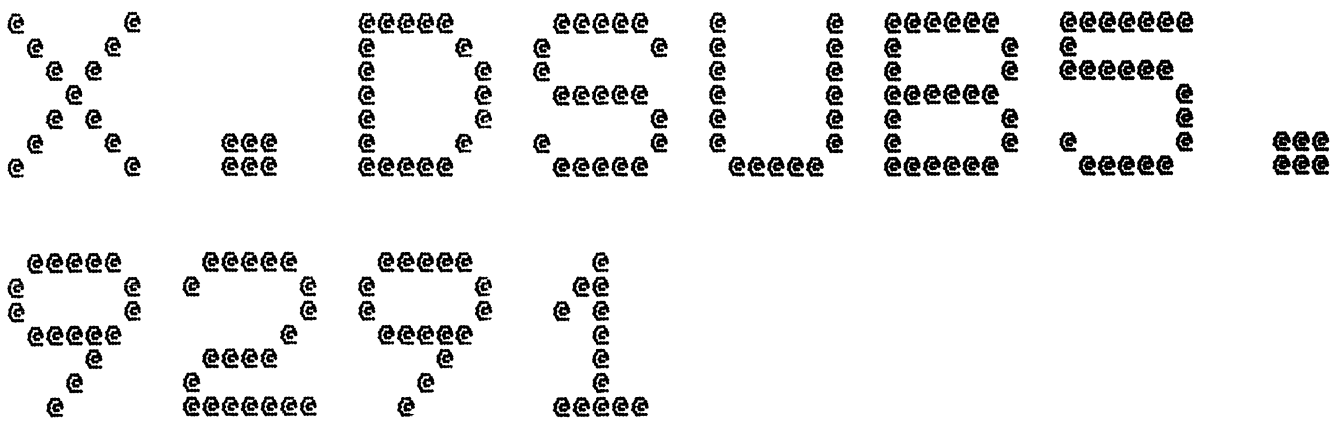
```
;                          << SI = R92DS5STYLUS2B; BO = 18/A.DSUB5.9291! >>
   102043 101015           MOV#    0,0,SNR
   102044   2112           JMP     @.NRET      ; CAN'T HAVE ZERO NO. OF ELEM.
   102045   6116         BINMULTIPLY
   102046 143000           ADD     2,0
   102047 123000           ADD     1,0
   102050  40415           STA     0,MAX
   102051  14414           DSZ     MAX
   102052    416           JMP     MAINL

;STORAGE TABLES

   102053      0 AC2:     0
   102054      0 VARPY:   0
   102055   5400 TABL3:    JSR     0,3
   102056      0 ADDRA:   0
   102057      0 SIZEA:   0
   102060      0 VARB:    0
   102061      0 VARC:    0
   102062      0 VARD:    0
   102063      0 VARE:    0
   102064      0 SHFPT:   0
   102065      0 MAX:     0
   102066      0 CNT5:    0
   102067      0 NAI9:  0        ;PSEUDO AUTO INCREMENT CELL

   102070  34772 MAINL:   LDA     3,VARD
   102071 175015           MOV#    3,3,SNR
   102072    454           JMP     FILL        ;FILL CONSTANT
   102073  20764           LDA     0,SIZEA
   102074  30770           LDA     2,SHFPT
   102075 126120           ADCZL   1,1
   102076 137005           ADD     1,3,SNR
   102077    410           JMP     MODE2
   102100 175113           SSN     3,3
   102101    413           JMP     MODE3
   102102  30754           LDA     2,ADDRA
   102103  24761           LDA     1,SHFPT
   102104 107000           ADD     0,1
   102105 143000           ADD     2,0
   102106    412           JMP     GETVR
   102107  24756 MODE2:    LDA     1,MAX
   102110 106400           SUB     0,1
   102111 113000           ADD     0,2
   102112  20752           LDA     0,SHFPT
   102113    405           JMP     GETVR
   102114  24751 MODE3:    LDA     1,MAX
   102115 135400           INC     1,3
   102116 116400           SUB     0,3
   102117 143001           ADD     2,0,SKP
   102120  34744 GETVR:    LDA     3,SHFPT
   102121  54745           STA     3,CNT5
```

```
;                          << SI = R92DS5STYLUS2B;  BO = 18/A.DSUB5.9291!  >>
   102122   6101            CALL
   102123 100015            MOVEWORDS
   102124 102400            SUB       0,0
   102125  40735 GET1:  STA     0,VARD
   102126 102400 WRTVR:     SUB       0,0
   102127  24730            LDA       1,SIZEA
   102130  30736            LDA       2,CNT5
   102131 151014            MOV#      2,2,SZR
   102132   6120          DECIMAL
   102133  14727            DSZ       VARD
   102134   2113            JMP       @.SRET
   102135  30731            LDA       2,CNT5
   102136  24721            LDA       1,SIZEA
   102137 133000            ADD       1,2
   102140  50726            STA       2,CNT5
   102141 102400            SUB       0,0
   102142  14716            DSZ       VARB
   102143 102520            SUBZL     0,0
   102144  40716            STA       0,VARD
   102145    761            JMP       WRTVR
   102146  30710 FILL:      LDA       2,ADDRA
   102147  50717            STA       2,CNT5
   102150 102520            SUBZL     0,0
   102151    754            JMP       GET1

          102152 DSE92      =.

               0 .ERR       DSB92+400<.     ;BLOCK OVERFLOW TEST

                 .EOT   ;R9.0 "DISCSUBS" GROUP 5, SOURCE 2
```

```
. END
```

AC0.T	77042	AC1.T	77043	AC2	102053	AC2.T	77044	AD	100067	ADDA	100064
ADDB	100065	ADDK	100255	ADRA	102056	ADRB	77263	ADRC	100066	ADRC1	77267
ADRC2	101576	ADRS	100521	ADRT	100526	AL1	77076	AL2	101421	ALPHA	77073
BD	100070	BEGIN	77424	BETA	101416	BGIN	100553	BINDI	6115	BINMU	6116
BKCD	100525	BLKA	101210	BPI	16	BSACF	75	BUMPU	6117	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	C20D	101215	C210	77354	C215	55
C233	77610	C237	77611	C240	56	C244	57	C260	60	C271	61
C3	3	C300	62	C304	77614	C325	77615	C334	63	C337	77300
C337A	77612	C37	43	C377	64	C4	24	C40	44	C400	65
C4000	72	C5	25	C6	26	C600	100	C7	27	C77	50
C774C	22	C777	66	CADRS	101206	CALL	6101	CALTO	77037	CD	100071
CD2	77607	CHANN	6106	CHARA	100077	CHARB	100100	CHARK	100262	CHART	100533
CHARU	100534	CHLEN	77225	CHLN1	77235	CHLN2	77311	CHLN3	101561	CHLNI	101544
CM400	23	CNLEN	101534	CNT	100072	CNT1	100073	CNT10	100257	CNT11	100260
CNT12	100261	CNT2	100074	CNT20	100530	CNT21	100531	CNT3	100075	CNT5	102066
CNT50	77261	CNT51	77301	CNT52	77302	CNT53	77303	CNT60	77353	CNT61	77545
CNT62	77546	CNT63	77547	CNT64	77550	CNTIO	77307	CNTO	77212	CNTS1	77220
CNTS2	77222	COMM	77620	CONT	100113	CONT1	100101	CONT2	77507	CONT3	77551
CONT4	77457	CONTS	77462	COUNT	77161	CSNT	100676	CSNT1	100607	CSNT2	100503
CSNT3	100636	CSNT4	100640	CSNT5	100572	CSNT6	100645	CSNT7	100730	CTO22	100532
CTBLE	77045	C.7	77276	C.7S	77277	C..7	101605	C..7S	101606	D127A	101370
D127B	101345	D127C	101334	D127D	101032	D127E	101151	D127G	101060	D127H	101110
D127J	101257	D127K	101302	D127L	101225	D127M	101230	D127N	101241	D127O	101145
D127P	101144	D127Q	101030	D127R	101312	D127S	101142	D127T	101313	D127U	101141
D127V	101217	D127X	101161	D127Y	101277	D127Z	101315	DA	160	DAC	164
DAS	165	DATAP	6110	DATAR	101203	DB	166	DBA	41	DBC	172
DBS	173	DECIM	6120	DFTCA	34106	DIMA	77264	DIMB	77266	DIMN	101575
DIMO	101573	DISCA	101202	DMCAL	34110	DQUEU	6105	DSB87	77000	DSB89	100400
DSB90	101000	DSB91	101400	DSB92	102000	DSBSI	100000	DSE88	77744	DSE89	100770
DSE90	101400	DSE91	101645	DSE92	102152	DSESI	100325	DWCD	100545	EDITX	77060
EDITY	101403	ERRF	76	ESCF	73	ETSF	74	FBLKN	100541	FBLNK	77613
FILL	102146	FINDL	6123	FINDX	100010	FINI1	77704	FINI2	77671	FINI3	77723
FINI5	77716	FINIS	77651	FINS1	100720	FIX	6121	FL3.	57	FLAG	100523
FLAGC	6102	FLGON	77473	FLOAT	6122	FNDLN	100613	FOUND	100164	FREEN	6107
GET1	102125	GETBY	6124	GETVR	102120	HALFT	77324	HALTS	6153	HOLFT	101616
INBYT	6125	INSER	77523	INSR2	77555	INSRT	77533	INSTB	6126	IOCAL	34103
IOP	6	ISA2D	6127	ISA2L	6130	JBEGI	77567	JD27Y	101140	JFINI	77570
JFLTO	151	JLDV	101137	JSHFT	100551	K233	100535	K304	100536	K325	100537
K337	101607	KD	100256	KNT1	101527	KNT2	101531	KNT50	101570	KNT51	101610
KNT52	101611	KNT53	101612	KOUNT	101504	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										

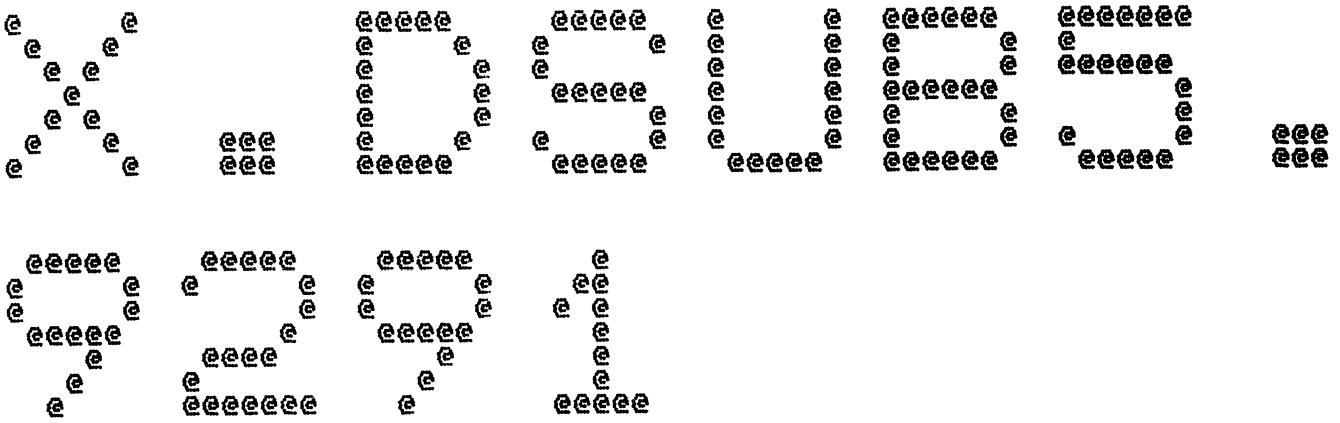
LDSB1	400	LDSB2	22400	LDSB3	47000	LDSB4	65000	LDSB5	77000
LDSB6	106400	LDSB7	113400	LDV	101320	LEADN	77270	LECHO	37000
LEEDN	100542	LEIDN	101577	LE087	105400	LERRO	23000	LFAUL	400
LFFIL	2400	LFIXD	57400	LFNDC	112000	LFNDL	20000	LFOFI	17000
LGETR	10000	LGHOP	107400	LGHOS	107000	LGMUX	16000	LHCON	17400
LIBCA	44400	LIBEN	45000	LIBTR	45400	LIDAT	103000	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	LNLHX	100753	LOADD	6131	LOOOP	102011	LOOP	100430
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	MAINL	102070
MASK1	101211	MAX	102065	MAY1	100137	MAYBE	100131	MODE	101205
MODE2	102107	MODE3	102114	MOVE	77377	MOVE1	100650	MUSHT	101642
MUSTH	77350	NAI1	77256	NAI2	77257	NAI3	100263	NAI4	77323
NAI5	77322	NAI6	100520	NAI7	101557	NAI8	101560	NAI9	102067
NEXT	100241	NKEYS	101212	NOTF	100126	NSKEY	101214	NULFG	101604
NULLF	77275	NUMBR	101445	NUMER	77122	NXTR	101204	ODRB	101574
ODRO	101572	OUTBY	6132	OUTTE	6133	OVERS	100544	OVSTK	77617
PEDA	4	PIB	4	PUSHY	102003	PUTBY	6134	PUTCH	100507
QCHAR	6103	QUEUE	6104	READB	6135	READY	77360	RELJM	6136
RETR1	77572	REVCD	100264	REVFL	100076	REVSX	100176	RJSR	6136
RTP	7	RUP	5	S105X	77005	S153X	101003	SADR	101207
SBA	40	SCDCA	34147	SCRD2	77634	SCRD3	77635	SCRDB	77646
SCRDW	77630	SCRUB	77644	SCRUP	77623	SD	100522	SHADO	100543
SHDOW	77616	SHFPT	102064	SHFT2	100457	SHFTR	100550	SHIFT	100464
SIZE	100214	SIZEA	102057	SKEY	101213	SP1	77355	SP2	77356
SP2PO	77357	SPINP	6146	STCHR	100511	STINP	6140	STINT	6147
STORD	6137	STOUT	6141	STRIX	100007	STRT	100275	STRT1	100227
STRT2	100273	STV	101325	TABAD	77262	TABL	100540	TABL2	77571
TABL3	102055	TABLE	77260	TADR	101216	TASKQ	15	TBLE	101567
TD	100527	TEMP	77304	TEMP1	77305	TEMP2	77306	TEMP3	101614
TEMP4	101615	TEMP.	101613	TOBOD	101571	TRAPF	6142	UNCD	100524
UND2	100547	UNDLN	100671	UNLIN	77502	UNLNX	100405	UPCD	100546
VAR1	77271	VAR2	77272	VAR3	77273	VAR4	77274	VARB	102060
VARC	102061	VARD	102062	VARE	102063	VARPT	101201	VARPY	102054
VAR.1	101600	VAR.2	101601	VAR.3	101602	VAR.4	101603	WRITB	6143
WRTVR	102126	XGETB	6144	XPUTB	6145	.ABA	14	.ADRA	77574
.ADRB	77576	.ADRC	77600	.BPS	77	.BSA	10	.DA	174
.DA3	175	.DB	176	.DB3	177	.DIMA	77575	.DIMB	77577
.FLTO	152	.HBA	11	.HXA	12	.INFO	100	.INTR	111
.LCM	114	.NRET	112	.NULF	77606	.SRET	113	.SSA	13
.TABA	77573	.UND2	77601	.VAR1	77602	.VAR2	77603	.VAR3	77604
.VAR4	77605								



Spool Queue Line #: 28
IRIS LU/Filename : 18/X.DSUB5.9291

Printed on/at : FEB 6, 1990 12:35:53
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 #: 28
IRIS LU/Filename : 18/X.DSUB5.9291

Printed on/at : FEB 6, 1990 12:36:03
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

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

1	TOTAL # DUPLICATE KEYS
0	TOTAL # DIR. RE-ORGS
1,243	TOTAL # KEYS INSERTED
0	TOTAL # ASSEMBLY ERRS

CHARK 18.012: 18.034 18.039 18.040 18.045

CHART 20.030 20.032 21.017: 22.012 22.028 22.036 23.024

CHARU 20.029 20.031 21.018: 22.013 22.029 23.026

CHKCH 26.016

CHLEN 6.009 6.042:

CHLN1 6.050: 8.029

CHLN2 7.015 7.045:

CHLN3 34.042:

CHLNI 34.028: 35.053

CNLEN 33.056 34.020:

CNT 15.016 15.028: 15.049 15.052 15.054 15.058 16.007
16.036

CNT1 15.029: 16.008 16.013 16.024

CNT10 17.024 17.027 17.033 17.034 17.039 17.052 17.059
18.009: 18.023 18.024 18.026

CNT11 17.025 18.010: 18.025 18.043 18.047

CNT12 17.046 17.049 18.011: 18.028 18.036

CNT2 15.030: 16.012 16.014 16.016

CNT20 20.022 20.026 20.033 20.038 21.014:

CNT21 19.043 19.051 20.016 20.021 20.023 20.024 20.044
20.045 21.015: 21.034 21.035 22.040 22.043 23.021
23.037 23.038

CNT3 14.027 15.010 15.031: 15.040

CNT5 36.033 36.048 37.028: 37.056 38.012 38.017 38.020
38.027

CNT50 4.029 5.022 5.028 5.044 5.049 6.006 6.034
6.039 7.020:

CNT51 5.050 5.059 6.010 6.017 7.036:

CNT52 5.051 6.035 6.049 7.008 7.037: 8.028 9.010

CNT53	5.052	6.031	6.038	6.048	7.038:	8.051	8.058
CNT60	8.031:	8.042	10.006	10.027	10.028		
CNT61	8.043	9.018	9.027	10.058:	11.048	11.049	12.039
	12.045	13.007					
CNT62	9.011	10.007	10.059:				
CNT63	8.044	10.039	10.045	10.060:	12.025		
CNT64	8.045	9.034	9.045	10.030	10.061:	12.012	
CNTIO	6.027	7.043:					
CNTO	6.031:	7.044					
CNTS1	6.030	6.037:					
CNTS2	6.033	6.039:					
COMM	11.048:	12.013					
CONT	15.020	15.046:	16.026	16.029	16.033		
CONT1	15.036:	15.056					
CONT2	10.014	10.027:					
CONT3	10.036	11.006:					
CONT4	9.041	9.045:					
CONTS	9.036	10.006:					
COUNT	6.006:	6.019	6.020	6.021	6.022	6.025	6.041
CSNT	21.053	22.041	23.011:				
CSNT1	21.059	22.008:					
CSNT2	20.040:	21.046					
CSNT3	22.018	22.032:					
CSNT4	22.021	22.033	22.034:				
CSNT5	20.043	21.050:	22.030				
CSNT6	22.024	22.027	22.039:				

CSNT7 23.015 23.037:
CTO22 19.027 19.053 20.017 21.016: 21.054 22.008 23.012
CTBLE 3.046 4.010:
D127A 29.032 31.008:
D127B 29.022 30.045:
D127C 30.036: 30.060 31.011
D127D 25.037 26.007:
D127E 27.034: 29.048 30.008 30.016
D127G 26.030:
D127H 26.029 26.054:
D127J 29.018 29.033: 31.007 31.015
D127K 29.013 30.009:
D127L 29.007: 29.034
D127M 27.021 29.010:
D127N 29.016 29.019:
D127O 27.028 27.030: 29.038 30.019 30.021
D127P 26.026 26.048 27.029:
D127Q 25.046 25.051:
D127R 27.008 27.012 30.017: 30.052 30.055
D127S 26.017 26.020 27.027:
D127T 26.060 30.018:
D127U 25.049 26.013 26.034 27.026:
D127V 27.057 28.023:
D127X 26.046 27.042:
D127Y 27.024 30.006:
D127Z 30.011 30.020:

```
TABL2      8.048   11.024:
TABL3     36.028   37.019:
TABLE      4.023    7.019:   8.046
TADR      26.007   28.021:  29.030   30.059   31.006   31.014
TBLE      32.033   35.007:
TD        20.007   21.013:  23.033
TEMP       7.039:   8.014    8.022
TEMP.     35.027:  35.039   35.047
TEMP1      7.040:   8.012    8.018
TEMP2      7.041:   8.009    8.020    8.021    8.025
TEMP3     35.028:  35.037   35.043
TEMP4     35.029:  35.034   35.045   35.046   35.050
TOBOD     34.031   35.009:
UNCD      21.010:  21.051
UND2      21.029:  23.007   23.023
UNDLN     23.006:  23.020
UNLIN     10.017   10.022:
UNLNX     19.022   19.027:
UPCD      21.028:  22.016
VAR.1     34.042   34.046   35.016:
VAR.2     35.017:
VAR.3     33.045   33.048   33.054   35.018:  35.051
VAR.4     33.037   35.019:
VAR1       7.028:
VAR2       7.029:   8.037
VAR3       5.054    5.057    6.007    6.050    7.030:   8.027
```

VAR4	5.046	6.055	7.031:				
VARB	36.059	37.022:	38.022				
VARC	36.055	37.023:					
VARD	37.024:	37.031	38.009	38.015	38.024		
VARE	37.025:						
VARPT	25.030	25.051	26.022	26.030	26.039	26.045	27.045
	28.008:	30.032	30.045				
VARPY	36.030	36.050	37.018:				
WRITB	26.050	28.027					
WRTVR	38.010:	38.025					
XGETB	5.025	6.016	8.011	8.017	9.008	9.021	12.041
	15.018	15.036	16.019	16.027	17.030	18.033	18.038
	19.046	20.028	21.041	22.015	33.016	33.059	35.036
	35.042						
XPUTB	10.046	13.014	17.058	18.041	18.046	20.041	20.052
	22.035	22.038	22.057				