```
      GENERAL AUTOMATION, INC,  ALL RIGHTS RESERVED
*****************************************************
*
*    PROGRAM NAME  FPH-21
*
*    MODEL NUMBER  8F021
*
*    PURPOSE       FORTRAN PHASE-21
*
*    PROGRAMMER    DICK WALLMANN
*
*****************   REVISION LIST   **************
*
*    RV DATE     SCO    BY  REASON FOR CHANGE
*    -- -------- ----- --- ----------------------
*
*    01 11/16/70 NONE  RPH INITIAL RELEASE
*
*****************************************************
*****************************************************
       HDNG    MPX FORTRAN ** COMPILATION ERRORS
*****************************************************
*STATUS-VERSION 1, MODIFICATION 0
*
*FUNCTION/OPERATION-
*    * LISTS ANY ERROR THAT WERE DETECTED DURING
*      THE COMPILATION PROCESS,
*    * REARRANGING THE STATEMENT STRING IF THERE
*      WERE NO ERRORS DETECTED,
*
*ENTRY POINTS-
*    * NEQ  PHASE IS LOADED BY PHASE 20 VIA ROLRX,
*           EXECUTION IS BEGUN AT LABEL NEQ
*
*INPUT-
*    * STATEMENT STRING
*    * SYMBOL TABLE
*    * FCOM
*
*OUTPUT-
*    * PRINTED ERROR MESSAGES IF ANY
*    * STATEMENT STRING
*    * SYMBOL TABLE
*    * FCOM
*
*    * SUBROUTINES
*      ROLRX
*
*EXITS-
*    * NORMAL-
*          PHASE 22 IS LOADED BY PHASE 21 VIA
*          ROLRX AND CONTROL IS PASSED TO IT.
*    * ERROR-
*          NO ERRORS ARE DETECTED IN THIS PHASE
*          ONLY THAT THEY ARE RECOGNIZED AND
*          PROCESSED ACCORDINGLY.  THE EXIT IS THE
*          SAME AS NORMAL.
*
*TABLES/WORK AREAS-
```

```
60  *    * STATEMENT STRING
61  *    * SYMBOL TABLE
62  *    * FCOM
63  *    * BUF+120, A 120 WORD MESSAGE BUILDING AND
64  *      OUTPUTTING AREA.
65  *
66  *ATTRIBUTES-NONE
67  *
68  *NOTES-NONE
69  **********************************************************
70          ABS  REF CORE
71  *
72  *       SYSTEM AND FORTRAN EQUATES
73  *
74  MEMRY EQU     #7FFF CORE      MAXIMUM CORE SIZE
75  PHSIZ EQU     4*320           MAXIMUM PHASE SIZE
76  OVERL EQU     MEMRY-PHSIZ     PHASES 2-29 START
77  FCOM  EQU     OVERL-22        FORTRAN COMM. TABLE
78  PHNTB EQU     FCOM-56         PHASE TABLE
79  ROLRX EQU     PHNTB-50        INTERPHASE CALL
80  AREA  EQU     OVERL+3*320+100 PRINT DATA ADDRESS
81  PRINT EQU     AREA+1          PRINT ENTRANCE
82  *
83          ORG   FCOM       FORTRAN COMM AREA
84  *
85  SOFS  BSS     1          START OF STRING
86  EOFS  BSS     1          END OF STRING
87  SOFST BSS     1          START OF SYMBOL TABLE
88  SOFNS BSS     1          START OF NON-STMNT NOS.
89  SOFXT BSS     1          SIZE OF WORK AREA
90  SOFGT BSS     1          SIZE OF CONSTANT AREA
91  EOFST BSS     1          END OF SYMBOL TABLE
92  COMON BSS     1          HIGH CORE COMMON ADDR
93  CSIZE BSS     1          SIZE OF COMMON
94  ERROR BSS     1          OVERLAP ERROR
95  *                   BIT 15 OVERLAP ERROR
96  *                   BIT 14 OTHER ERROR
97  FNAME BSS     1          PROGRAM NAME WD 1
98          BSS   1          PROGRAM NAME WD 2
99  SORF  BSS     1          SUBR (-) OR FUNC (+)
100 CCWD  BSS     1          CONTROL CARD WD
101 *                   BIT 15 TRANSFER TRACE
102 *                   BIT 14 ARITHMETIC TRACE
103 *                   BIT 13 EXTENDED PRECISION
104 *                   BIT 12 LIST SYMBOL TABLE
105 *                   BIT 11 LIST SUBPROGRAM NAMES
106 *                   BIT 10 LIST SOURCE PROGRAM
107 *                   BIT  9 ONE WORD INTEGERS
108 IOCS  BSS     1          IOCS CONTROL CARD WORD
109 *
110 *         SEE PHASE ONE FOR BIT PATTERNS
111 *
112 DFCNT BSS     1          DEFINE FILE COUNT
113 *
114 *
115 LCOMN BSS     2          INSKEL COMMON
116 *
117 ICCER BSS     2          IOCS CONTROL CARD ERROR
118 *
119         BSS   2          SYSTEM LOADER USE
```

```
120  *
121  *                    END OF FORTRAN COMMUNICATION
122  *                    AREA
123  ***************************************************
124  *                THE SWITCHES USED IN PHASE 21 FOLLOW
125  *                IF POSITIVE, THE SWITCH IS TRANSFER
126  *                IF ZERO, THE SWITCH IS NORMAL N
127  *                SWITCH OUTFL
128  *                  N  HEADER NOT PRINTED
129  *                  T  HEADER ALREADY PRINTED
130  *
131  *
132  *
133  *                OUTPUT STRING ERROR MESSAGES
134  *                PHASE
135  *
136  ***************************************************
137         HDNG     MPX FORTRAN ** COMPILATION ERRORS
138         ORG      OVERL      PHASE ENTRY
139  NEQ    BSC  L   ENT        ENTRY, GO INITIALIZE
140  *
141  *
142  WDCNT  DC       *-*        WORD COUNT FOR PRINT LINE
143  BUF    DC       *-*         START OF OUTPUT AREA
144  *
145  *
146  *                INITIALIZE TRANSFERVECTOR
147  ENT    LDX  L3 ZERO         LOAD WORK AREA POINTER
148         STX  L3 NEQ         SAVE FOR RE-INITIALIZATION
149  *
150  *                CHECK FOR OVERLAP ERROR
151         LD   L  ERROR
152         BSC  L  E2020,E     BR IF OVERLAP ERROR
153  *
154  *                MAKE SIZE OF WORK AREA EVEN ADDRESS
155         LD   L  SOFXT       LOAD WORK AREA START ADDR
156         BSC     E           SKIP IF EVEN
157         A     3 ONE-Z       ADJUST TO AN EVEN ADDRESS
158         STO  L  SOFXT       STORE IN WORK AREA POINTER
159  *
160  *
161  *                SET HEADER TEST FOR
162  *                .UNREFERENCED STATEMENT
163  *
164         LDX  L2 TEXT5       INITIALIZE TO PRINT PROPER
165         STX  L2 HTES1 1     *HEADER
166         LD    3 ZERO-Z
167         STO   3 OUTFL-Z     ZERO TO HEADER OUTP FLAG
168  *
169  *                DELETE EQUIVALENCE STATEMENTS
170  *                FROM STRING
171  *
172  MOVST  LDX   I1 SOFS        INITIALIZE POINTERS
173         LDX   I2 SOFS
174  *
175  *                INITIALIZE MOVING OF ONE STATEMENT
176  MOVS1  LD     1 0           LOAD STATEMENT ID WORD
177         AND    3 HF800-Z    MASK TO LEAVE ID CODE
178         S        HA800      TEST FOR EQUIVALENCE STMNT
179         BSC  L  MOVS3,Z     BRANCH IF NOT EQUIVALENCE
```

```
180   *
181   *                 EQUIVALENCE STMT-ENCOUNTERED
182   *                 TEST IF IT CONTAINS ERROR-MARK
183   *
184           LD     1 1          TEST IF EQUIVALENCE WITH
185           S        H0FFF      *ERROR MARK.
186           BSC   L  MOVS2, -   BRANCH IF WITH ERROR MARK
187   *
188   *                 DELETE ONE EQUIVALENCE-STMT
189           BSI      MOVIN      MOVE I/P PT TO NEXT STMNT
190           MDX      MOVS1      GO CHECK NEXT STATEMENT
191   *
192   *                 EQUIVALENCE WITH ERRORMARK
193   MOVS2   LD     1 2          MOVE ERROR NO.
194           STO    2 1
195           BSI      MOVIN      MOVE I/P PT TO NEXT STMNT
196           LD       HA008      OUTPUT ERROR ID
197           STO    2 0
198           MDX    2 2          INCR OUTPUT POINTER
199           MDX      MOVS1      GO CHECK NEXT STATEMENT
200   *
201   *
202   *                 STATEMENT OTHER THAN EQUIVALENCE
203   *
204   *
205   *                 TEST IF END-STMT
206   *
207   MOVS3   LD     1 0          LOAD STMNT ID WORD
208           AND    3 HF800-Z    MASK TO LEAVE ID CODE
209           S        H1000      TEST FOR END STATEMENT
210           BSC   L  MOVS5, -   BRANCH IF END STATEMENT
211   *
212   *                 MOVE ONE STATEMENT UNALTERED TO
213   *                 OUTPUT STRING
214   *
215           LD     1 0          LOAD STMNT ID WORD
216           AND    3 H07FC-Z    MASK TO LEAVE WORD COUNT
217           STO      MCT        STORE AS MOVE COUNT
218   *
219   MOVS4   LD     1 0          LOAD WORD FROM I/P STRING
220           STO    2 0          STORE ONTO O/P STRING
221           MDX    1 1          INCREMENT STRING POINTERS
222           MDX    2 1
223           MDX   L  MCT,-4     DECREMENT MOVE COUNT
224           MDX      MOVS4      BR IF NO SKIP
225           MDX      MOVS1      BR IF STATEMENT MOVED
226   *
227   *                 END-STATEMENT ENCOUNTERED
228   MOVS5   LD     1 0          MOVE END STATEMENT
229           STO    2 0
230   *
231   *                 SET NEW ENDOFSTRING ADDR
232           MDX    2 1          INCR OUTPUT POINTER
233           STX   L2 EOFS       STORE AS NEW END OF STRING
234   *
235   *
236   *                 BR TO PROGRAM OUTSIDE PRINT AREA
237   *
238           BSC   L  A1011      BR OUTSIDE PRINT AREA
239   *
```

```
240  HA800 DC       /A800     CONSTANT
241  H1000 DC       /1000     CONSTANT
242  H0FFF DC       /0FFF     CONSTANT
243  HA008 DC       /A008     CONSTANT
244  *
245  *              SUBROUTINE
246  *              MOVE INPUT POINTER TO NEXT STMT
247  *
248  MOVIN DC       0         LINK
249        LD    1  0         LOAD STMNT ID WORD
250        AND   3  H07FC-Z   MASK TO LEAVE WORD COUNT
251        SRA      2         RIGHT JUSTIFY
252        STO      MOVI1 1   STORE IN INSTRUCTION
253  MOVI1 MDX   L1 *-*       MOVE INPUT POINTER
254        BSC   I  MOVIN      RETURN
255  MCT   DC       0         WD CNT FOR MOVING A STMNT
256  *
257  *
258        ORG      BUF 120    FROM HERE ON IS NOT OVER-
259  *                          LAYED IN THIS PHASE
260  *
261  *              SEARCH FOR UNREFERENCED STATEMENTS
262  *              INITIALIZE SYMBOL TABLE POINTER
263  *
264  A1011 BSI  L   BLKPA      BLANK PRINT AREA
265        LD   L   SOFST      INITLZ SYMBOL TABLE PT
266        STO   3  STPTR-Z
267        S    L   EOFST      TEST FOR SYMBOL TBL ENTRIES
268        BSC  L   E1011,     BRANCH IF NONE
269        LDX  I1  SOFST      GET POINTER VALUE
270  *
271  A1012 LD     1  0         LOAD SYM TBL ID WORD
272        AND    3  H0200-Z   TEST IF STMNT NO
273        BSC       -         SKIP IF STATEMENT NUMBER
274        MDX       A1013     BR IF NOT
275        LD     1  0
276        AND    3  H0020-Z   TEST IF REFERENCED
277        BSC  L    A1013,Z   BR IF REFERENCED
278        LD     1  1         LOAD NAME WORD 1
279  *
280  *              TEST HEADER AND PRINT
281        BSI  L    MNAME,Z   BR IF NOT GENERATED LABEL
282  *
283  A1013 BSI  L    INCSP     INCR SYM TBL POINTER
284        DC       A1012     BR IF NOT END OF SYM TBL
285        BSI    3 HFILL-Z   PRINT IF BUFFER HALF FILLED
286  *
287  *              SET HEADER TEST FOR
288  *              UNDEFINED VARIABLES
289  A1021 LDX  L2   TEXT6     INITIALIZE TO PRINT PROPER
290        STX  L2   HTES1 1   *HEADER.
291        LD     3  ZERO-Z
292        STO    3  OUTFL-Z   SET HEADER INDICATOR
293  *
294  *
295        LDX  I1   SOFST     INITLZ SYMBOL TABLE PT
296        STX  L1   STPTR
297  *
298  *              SEARCH FOR UNDEFINED VARIABLES
299  A1022 LD     1  0         LD SYM TBL ID WORD
```

```
300          AND   3 HA7FC-Z  MASK TO LEAVE DEFINE BITS
301   *
302   *                SKIP IF UNDEFINED VARIABLE
303          BSC     Z        SKIP IF UNDEFINED
304   *
305          MDX     A1023    BR IF NOT
306   *
307   *                IF NOT, TEST FOR HEADER
308   *                MOVE NAME TO PRINT AREA
309   *
310          BSI   3 MNAME-Z  MOVE NAME TO PRINT AREA
311   *
312   *
313   *                INSERT ERRORSWITCH TO PREVENT
314   *                FINAL OUTPUT
315   *
316          BSI   3 ERRSW-Z  SET ERROR SWITCH
317   *
318   A1023  BSI   L  INCSP    INCR SYM TBL POINTER
319          DC      A1022    BR IF NOT END OF SYM TBL
320          BSI   3 HFILL-Z  PRINT IF BUFFER HALF FILLED
321   *
322   *
323   *                SEARCH FOR ERROR-STATEMENTS ON STRIN
324   *
325   E1011  SLA     0          ENTRY FROM A1011
326   *
327   *                SET HEADER FOR
328   *                .INVALID STATEMENTS,
329   *
330          LDX   L2 TEXT7    INITIALIZE TO PRINT PROPER
331          STX   L2 HTES1 1  *HEADER,
332   *
333   *                INITIALIZATION FOR SCAN OF STRING
334          LDX   I1 SOFS     INITIALIZE INPUT POINTER
335          LDX   I2 SOFS     INITIALIZE OUTPUT POINTER
336   *
337          LD    3 ZERO-Z    ZERO OUT
338          STO   3 STLAB-Z   *STORED LABEL,
339          STO   3 STCNT-Z   *STORED COUNT,
340          STO   3 OUTFL-Z   *AND HEADER FLAG
341   *
342   *
343   *                PROCESS ONE STATEMENT IN SEARCH FOR
344   *                ERROR STATEMENTS
345   *
346   E1021  LD      1 0        LOAD STMNT ID WORD
347          STO     2 0        STORE ON OUTPUT STRING
348          AND     3 IDMSK-Z  MASK TO SAVE STMNT ID
349          STO     3 STSID-Z  SAVE ID
350          LD      1 0        LOAD STMNT ID WORD
351          AND     3 NRMSK-Z  EXTRACT NORM
352          STO     3 NORM-Z   SAVE NORM
353   *
354   *                DO NOT COUNT *CALL FIO*
355   *                  AND -DOTEST- STATEMENTS
356   *
357          LD      3 STSID-Z
358          S       3 HD800-Z  TEST FOR FIO CALL
359          BSC   L  E102A, -  BR IF FIO CALL
```

```
360          LD     3 STSID-Z
361          S      3 HC000-Z  TEST FOR DO TEST
362          BSC  L   E102A, -  BR IF DO TEST
363   *
364   *                INCREMENT STATEMENT COUNT TO
365   *                LAST ENCOUNTERED LABELED STATEMENT
366   *
367          MDX  L   STCNT,1   INCR STMNT COUNT
368   *
369   *
370   *                STORE OUTPUT POINTER
371   *                FOR POSSIBLE DELETION OF THE
372   *                STATEMENT,IF IT IS AN
373   *                ERROR-STATEMENT
374   *
375   E102A STX  L2  STOOI     STORE OUTPUT POINTER
376   *
377          LD     1 0
378          BSC      E        SKIP IF STMT HAS NO LABEL
379          MDX      E1022    BR IF IT HAS LABEL
380          MDX      E1024    BR IF NO LABEL
381   *
382   *                IGNORE STATEMENT NUMBER
383   *                IF IT IS GENERATED BY THE
384   *                COMPILER  HAS NAME   0
385   *
386   E1022 LD     1 1         GET NAME FR SYMBOL TABLE
387          BSI    3 GNAME-Z  *
388          LD     3 NAME-Z   *
389          BSC  L   E1023, -  BR IF NAME   0
390   *
391   *
392   *                STORE STATEMENT NUMBER
393   *                AND INITIALIZE  REFERENCE COUNT TO
394   *                THAT STATEMENT
395          LD     1 1         SAVE STMNT NUMBER
396          STO    3 STLAB-Z
397          LD     3 ZERO-Z    ZERO TO STORED COUNT
398          STO    3 STCNT-Z
399   E1023 MDX    1 1         INCREMENT STRING POINTERS
400          MDX    2 1
401          MDX  L   NORM,-4   DECREMENT WORD COUNT
402   E1024 LD     3 STSID-Z   LOAD STMNT ID
403          S      3 ERRID-Z  TEST FOR ERROR ID
404          BSC      -        SKIP IF NOT ERROR-STMT
405          MDX      E1051    BR IF ERROR STMNT
406   *
407   *                TEST IF DATA-STATEMENT
408   *
409   DTA11 LD     3 STSID-Z   LOAD STMNT ID
410          S      3 HF800-Z  TEST IF DATA STMNT
411          BSC  L   E1031,Z  BR IF NOT DATA-STMT
412   *
413   *                CHECK THAT DATASTATEMENT DOES NOT
414   *                CONTAIN COMMON VARIABLES
415   *
416          LD     3 NORM-Z   STORE NORM
417          STO      STONO
418          STX    1 DTA21 1  SAVE POINTER
419   *
```

```
420  *                MOVE POINTER TO NEXT DATAGROUP IF AN
421  *
422  DTA12 MDX    1 1          MOVE POINTER
423        MDX  L  STONO,-4
424  DTA13 LD     1 0           INSERT CONSTANT LENGTH
425        AND      H0007      *
426        STO      CONSC      *
427  *
428  *                MOVE POINTER PAST THE CONSTANT
429  *
430  DTA14 MDX    1 1          INCR STRING POINTER
431        MDX  L  STONO,-4 DECR WD CNT AND CON LENGTH
432        MDX  L  CONSC,-1 SKIP WHEN CON MV COMPLETE
433        MDX      DTA14       LOOP BACK
434  *
435  *                MOVE POINTER TO NEXT NAME
436  *
437  DTA15 MDX    1 1          INCR STRING POINTER
438        MDX  L  STONO,-4 DECR WORD COUNT
439  *
440  *                TEST IF NAME IN COMMON
441  *
442        LD     1 0          GET NAME FROM SYM TBL
443        BSI    3 GNAME-Z    *
444        LD     3 GNAM4-Z    GET SYM TBL ID WORD
445        SLA      2
446        BSC  L  DTAER,Z     BR IF IN COMMON
447  *
448  *                MOVE POINTER TO NEXT NAME IF ANY
449  *
450        LD     1 0          LOAD NAME POINTER
451        SLA      1
452        BSC  L  DTA17,-     BR IF NO DISPLACEMENT AFTER
453        MDX    1 1          INCR STRING POINTER
454        MDX  L  STONO,-4 DECR WORD COUNT
455  DTA17 LD       STONO
456        SRA      3
457        BSC  L  DTA21, -  BRANCH END OF STATEMENT
458        LD     1 1
459        BSC  L  DTA12,-   BRANCH END OF VARIABLES
460        MDX      DTA15     GO MOVE TO NEXT NAME
461  *
462  *                RESTORE POINTER, BRANCH TO OUTPUT
463  *                DATASTATEMENT UNALTERED
464  *
465  DTA21 LDX   L1 *-*        RESTORE POINTER
466        MDX      E1031      BR TO O/P DATA STMNT
467  *
468  *                REPLACE DATASTATEMENT BY AN
469  *                ERRORSTATEMENT
470  *
471  *
472  *                MOVE POINTER TO NEXT STATEMENT
473  *
474  DTAER MDX    1 1          INCR STRING POINTER
475        MDX  L  STONO,-4 DECR CNT, SKIP WHEN ZERO
476        MDX      DTAER      LOOP TO MOVE POINTER
477  *
478        MDX    1 -2         DECR POINTER BY 2
479        LD       ERID       BUILD AN ERROR STMNT
```

```
480          STO   1 0          STORE ONTO I/P STRING
481          LD      ERTYP      LOAD ERROR TYPE CODE
482          STO   1 1          STORE ONTO I/P STRING
483          MDX     E1051      BR TO PRINT ERROR STMNT
484   *
485   STONO  DC      0          STORED NORM
486   CONSC  DC      0          CONSTANT LENGTH
487   DUPLC  DC      0          DUPLICATION FACTOR
488   ERTYP  DC      81         ERROR TYPE CODE
489   ERID   DC      /A008      ERROR ID
490   H0007  DC      /0007      CONSTANT
491   *
492   *              MOVE STATEMENT UNALTERED TO OUTPUT
493   *              STRING
494   *
495   E1031  LD    1 0          LOAD WORD FROM I/P STRING
496          STO   2 0          STORE ONTO O/P STRING
497          MDX   1 1          INCR STRING POINTERS
498          MDX   2 1
499          MDX   L NORM,-4    COUNTDOWN
500          MDX     E1031      BR IF NOT END OF STMNT
501          LD    3 STSID-Z    LOAD STORED ID
502          S     3 ENDID-Z    TEST IF END OF STRING ID
503          BSC   L E2011, -   BRANCH IF END OF STRING
504          MDX     E1021      BR IF NOT
505   *
506   *              PRINT ONE ERROR MESSAGE
507   *
508   *              SET ERRORSWITCH TO PREVENT
509   *              FINAL OUTPUT
510   *
511   E1051  LD    3 TWO-Z
512          BSI   3 ERRSW-Z    SET ERROR SWITCH ON
513   *
514   *
515   *              TEST IF HEADER IS PRINTED
516   *
517          BSI   3 HTEST-Z    PRINT HEADER IF FIRST TIME
518   *
519   *              MOVE ERROR MESSAGE TO PRINT AREA
520          BSI   3 TOPAB-Z    OUTPUT BLANK
521          LD    3 C-Z
522          BSI   3 TOPAU-Z    OUTPUT C
523   *
524   *              ERROR TYPE TO PRINT AREA
525          LD    1 1          LOAD ERROR CODE
526          BSI   3 CON3D-Z    GO OUTPUT AS 3 DECML DIGITS
527          MDX   L PAP,-3     BACK UP PRINT AREA POINTER
528          BSI   3 TOPAB-Z    BLANK OUT LEADING ZERO
529   E1052  LD    3 STLAB-Z    LOAD STORED LABEL
530          BSC     Z          SKIP IF STORED LABEL IS ZER
531          MDX     E1061      BR IF NOT
532   *
533   *              EBC ZEROS TO
534   *                      LABEL SPACE IN PRINT AREA
535   *
536          LDD   3 FZERO-Z    FIVE ZEROS
537          STD   3 NAME-Z     ZERO OUT NAME
538          MDX     E1061 1    GO OUTPUT BLANK NAME
539   *
```

```
*                LAST ENCOUNTERED STATEMENT NUMBER
*                TO PRINT AREA
*
E1061 BSI    3 GNAME-Z   PUT LAST STMNT NO. IN NAME
      STX    1 STXR1 1   SAVE I/P STRING POINTER
      LDX   L1 NAME-1    SET POINTER TO NAME-1
      LD     3 PAR31-Z
      STO    3 PAP-Z     SET PRNT AREA PT TO 31 POS
      BSI    3 MNAME-Z   LAST STMNT NO. TO PRNT AREA
STXR1 LDX   L1 ***       RESTORE I/P STRING POINTER
      LD     3 STCNT-Z   LOAD REFERENCE COUNT
      BSC   L  E1073, -  BR IF REFERENCE COUNT
*                          ZERO
*
*                STATEMENT NUMBER REFERENCE COUNT
*                TO PRINT AREA
*
      MDX   L  PAP,-2    DECR PRINT AREA PT BY 2
      LD     3 PLUS-Z
      BSI    3 TOPAU-Z   OUTPUT
      LD     3 STCNT-Z
      BSI   L  CON3D     OUTPUT REFERENCE COUNT
E1073 LDX    2 0
      LDX    3 -13       SET TO OUTPUT 26 CHARS
LOOPP LD    L3 TEXT8 13  ERROR AT STATEMENT NUMBER
      CRT      8
      SLA      8
      STO   L2 BUF 5      OUTPUT LEFT CHARACTER
      SLT      16
      STO   L2 BUF 6      OUTPUT RIGHT CHARACTER
      MDX    2 2
      MDX    3 1          SKIP WHEN MSG COMPLETE
      MDX      LOOPP      LOOP UNTIL ALL OUTPUT
      LDX   L3 Z
      BSI    3 MFILL-Z   PRINT ERROR MESSAGE
E1082 MDX    1 2          INCREM POINTER
      LDX   I2 STOOI      RESTORE X2  TO DELETE ERROR
*                           STATEMENT
      BSC   L  E1021      GO PROCESS NEXT STATEMENT
*
*
E2011 SLA      0          ENTRY FROM E1031 RTN
*
*                INSERT END OF STRING
*                IN COMMUNIC AREA
*
      STX   L2 EOFS       SET END OF STRING INTO FCOM
*
*
*                TEST IF OVERLAP ERROR
*
      LD    L  ERROR      LOAD ERROR SW FROM FCOM
      BSC   L  E2020,E    BR IF OVERLAP ERROR
      MDX      E2021      BR IF NOT
*
*                PRINT OVERLAP ERROR MESSAGE
E2020 BSI   L  BLKPA      BLANK PRINT AREA
      LDX   L2 TEXT3      SET UP OVERLAP MESSAGE
      STX   L2 HTES1+1     MODIFIES WD CT ADDR
      LD     3 ZERO-Z     FORCE MESSAGE OUTPUT
```

```
600         STO   3 OUTFL-Z
601         BSI   3 HTEST-Z  PRINT OVERLAP MESSAGE
602   ********************************************************
603   *
604   *          TEST IF OUTPUT IS TO BE SUPPRESSED
605   *
606   ********************************************************
607   E2021 LD    L ICCER      GET UTAPE/UDISK ERROR WD
608         BSC   L UTUDM,Z    BR IF ON OTHERWISE SEE IF
609         LD    L ERROR      OTHER ERRORS HAVE OCCURRED
610         BSC   L R1011,+-   EXIT ROUTINE IF NOT
611   *
612   *               PRINT OUTPUT SUPPRESSED MSG
613   *
614   UTUDL LDX   L2 TEXT4     GET EBC MSG AND WD CT
615         STX   L2 HTES1+1   PUT IT IN ROUTINE
616         LD     3 ZERO-Z    GET A CHEAP ZERO INTO OUTF
617         STO    3 OUTFL-Z   SO THAT HTEST WILL PRINT
618         BSI    3 HTEST-Z   GO SET UP TO PRINT
619         BSC   L  EXIT      EXIT TO CALL NEXT PHASE
620   *
621   UTUDM LDX   L2 TEXT2     GET UTAPE AND UDISK MSG
622         STX   L2 HTES1+1   PUT IT IN ROUTINE
623         LD     3 ZERO-Z    GET A CHEAP ZERO INTO OUTF
624         STO    3 OUTFL-Z   SO THAT HTEST WILL PRINT
625         BSI    3 HTEST-Z   GO SET UP TO PRINT IT
626         LD     3 TWO-Z     GET A TWO INTO ACC AND
627         STO   L  ERROR     SET ERROR ON FOR OUTPUT 1,
628         MDX       UTUDL    BR BACK TO PRT OTHER MSG
629   *
630   ********************************************************
631   *               REARRANGE THE STRING BY PLACING
632   *               FORMAT-STATEMENTS AND ARITH
633   *               STMT-FUNCTION-STATEMENTS IN THE
634   *               LOW-ADDRESS END OF THE STRING
635   *
636   R1011 LD    L  SOFS
637         STO    3 PTR-Z     INITIALIZE POINTER
638         LD    I  PTR
639         AND    3 HF800-Z   EXTRACT ID
640         STO    3 STSID-Z   SAVE ID
641         LD     3 ZERO-Z    SET PASS SWITCH FOR
642         STO      SWITC     *FIRST PASS
643   *
644   R1012 LD     3 STSID-Z   TEST FOR END STATEMENT
645         S      3 ENDID-Z
646         BSC   L  D1099, -  BR IF END-STATEMENT
647         LD       SWITC     LOAD PASS SWITCH
648         BSC   L  D1015, -  BR IF FIRST PASS
649   *
650   *               SECOND PASS
651   *
652   *               TEST IF FORMAT OR ARITH-STMT-FUNCT
653   *
654         LD     3 STSID-Z   TEST FOR FORMAT STATEMENT
655         S        FORMT
656         BSC   L  R1013, -  BR IF FORMAT
657         LD     3 STSID-Z   TEST FOR ARITH STMNT FUNC
658         S      3 HD000-Z
659         BSC   L  R1013, -  BR IF ARITH STMNT FUNC
```

```
660   *
661   *                NOT IN CARD SYSTEM
662   *
663           MDX      R1021     GO TEST IF END STMNT
664   *
665   *                FIRST PASS
666   *
667   D1015   LD     3 STSID-Z   TEST FOR DEFINE FILE STMNT
668           S        DFILE
669           BSC  L   R1013, -  BR IF DEFINE FILE
670           S        DATA      TEST FOR DATA STMNT
671           BSC  L   R1013, -  BR IF DATA STMNT
672   *
673   *
674   *                IF SUBPROGRAM, ALSO DISREGARD
675   *                LOADER-OVERLAY STATEMENTS
676   *
677           LD     3 STSID-Z   LOAD STORED ID
678           S      3 HD800-Z   TEST FOR INTERVAL O/P FMT
679           BSC      Z         SKIP IF POSSIBLE LDR STMNT
680           MDX      R1021     BR IF NOT
681           LD   L   SORF      TEST IF SUBR OR FUNCTION
682           BSC      -         SKIP IF SUBPROGRAM
683           MDX      R1021     BR IF NOT
684   *
685   R1013   BSI      MOVP      MOVE POINTER TO NEXT STMT
686           MDX      R1012     GO CHECK IF END STATEMENT
687   *
688   R1021   LD     3 PTR-Z     LOAD POINTER
689           STO    3 RALO-Z    INSERT REARRANGEMENT AREA
690   *                            LOW ADDRESS
691   R1022   BSI      MOVP      MOVE POINTER TO NEXT STMNT
692           S      3 ENDID-Z   TEST IF END STATEMENT
693           BSC  L   D1099, -  BR IF END-STATEMENT
694   *
695   *                NOT IN CARD SYSTEM
696   *
697           LD       SWITC     TEST PASS SWITCH
698           BSC  L   D1026,Z   BR IF SECOND PASS
699   *
700   *                FIRST PASS
701   *
702   D1025   LD     3 STSID-Z   TEST,FIRST PASS
703           S        DFILE     TEST IF DEFINE FILE STMNT
704           BSC  L   R1023, -  BR IF DEFINE FILE
705           S        DATA      TEST IF DATA STATEMENT
706           BSC  L   R1023, -  BR IF DATA STATEMENT
707           MDX      R1022     GO MOVE TO NEXT STATEMENT
708   *
709   *                SECOND PASS
710   *
711   D1026   SLA      0         TEST, SECOND PASS
712   *
713           LD     3 STSID-Z
714           S        FORMT     TEST IF FORMAT STATEMENT
715           BSC  L   R1023, -  BR IF FORMAT STATEMENT
716           LD     3 STSID-Z
717           S      3 HD000-Z   TEST IF ARITH STMNT FUNCT
718           BSC  L   R1022,Z   BR IF NOT ARITH STMNT FUNCT
719   *
```

```
720  *              FORMAT OR ARITH STMT FUNCT
721  *              ENCOUNTERED, PREPARE REARRANGEMENT
722  *
723  R1023 LD     3 NORM-Z    INSERT ROTATE COUNT
724        STO     R1031 1
725        A      3 PTR-Z     INSERT REARR-AREA HIGH ADR
726        STO     R1034 1
727        STO     R1035 1
728        S      3 ONE-Z     INSERT REARR-AREA HIGH ADDR
729        STO     R1033 1
730        A      3 ONE-Z
731        S      3 RALO-Z    INSERT REARR-AREA LENGTH
732        STO     R1032 1     NEGATIVE
733        LD     3 0
734        S       R1032 1    SET LENGTH POSITIVE
735        STO     R1032 1
736  *
737  *              ROTATE REARRANGEMENT AREA
738  R1031 LDX   L2 *-*         INITIALIZE ROTATE COUNT
739  R1032 LDX   L1 *-*         INITIALIZE .ROTATE ONE.
740  *
741  *              ROTATE REARR AREA BY ONE WORD
742  R1033 LD    L  *-*         REARR AREA HIGH ADDR
743        RTE      16
744  R1034 LD    L1 *-*         REARR AREA HIGH ADDR   1
745        RTE      16
746  R1035 STO   L1 *-*         REARR AREA HIGH ADDR   1
747        MDX    1 1
748        MDX      R1034      BR IF .ROTATE ONE. NOT COMP
749  *
750        MDX    2 -1         ROTATE COUNT
751        MDX      R1032      IF NOT ZERO, BR TO .ROTATE
752  *                            ONE.
753  *
754  *              ROTATE COMPLETED
755  *
756        LD     3 RALO-Z    INCR REARR AREA LOW ADDR
757        A      3 NORM-Z      BY LENGTH OF FORMAT STMT
758        STO    3 RALO-Z
759        STO    3 PTR-Z     REARR-AREA LOW ADDR TO PTR
760        MDX      R1022     GO MOVE TO NEXT STATEMENT
761  *
762  FORMT DC         /6000   CONSTANT
763  *
764  *              NOT IN CARD SYSTEM
765  *
766  D1099 LD         SWITC   LOAD PASS SWITCH
767        BSC   L    EXIT,Z  BR IF SECOND PASS
768        MDX   L    SWITC,1 CHANGE PASS SWITCH
769        LD     3 RALO-Z    REARRANGEMENT AREA LOW
770        STO    3 PTR-Z     *ADDRESS TO POINTER
771        LD       LINK      SET LINK ADDR IN SUBROUTINE
772        STO      MOVP
773        MDX      MOVP1     BR TO SUBR
774  *
775  LINK  DC         R1012   CONSTANT
776  DFILE DC         /F000   STMT ID DEFINE FILE
777  DATA  DC         /F800-/F000 CONSTANT
778  SWITC DC         0       PASS SWITCH
779  *
```

```
780   *
781   *      MOVE POINTER TO NEXT STATEMENT
782   *                STORE STATEMENT-ID AND NORM
783   *                STATEMENT-ID TO A-REGISTER
784   MOVP   DC        0          LINK
785          LD    I   PTR        GET ID WORD THIS STATEMENT
786          AND    3  H07FC-Z    EXTRACT NORM
787          SRA       2          RIGHT JUSTIFY
788          A      3  PTR-Z      ADVANCE PT TO NEXT STMNT
789          STO    3  PTR-Z
790   MOVP1  LD    I   PTR        GET ID WORD NEXT STATEMENT
791          AND    3  H07FC-Z    EXTRACT AND STORE NORM
792          SRA       2
793          STO    3  NORM-Z
794          LD    I   PTR
795          AND    3  HF800-Z    EXTRACT AND STORE ID
796          STO    3  STSID-Z
797          BSC   I   MOVP       RETURN
798   ********************************************************
799   *
800   *           COMPILATION STATUS MESSAGES
801   *
802   ********************************************************
803   *
804   TEXT2  DC        13          WORD COUNT
805          EBC       .UTAPE/UDISK BOTH SPECIFIED.
806   TEXT3  DC        16          WORD COUNT
807          EBC       .PROGRAM LENGTH EXCEEDS CAPACITY .
808   TEXT4  DC        13          WORD COUNT
809          EBC       .OUTPUT HAS BEEN SUPPRESSED.
810   TEXT5  DC        12          WORD COUNT
811          EBC       .UNREFERENCED STATEMENTS .
812   TEXT6  DC        10          WORD COUNT
813          EBC       .UNDEFINED VARIABLES .
814   TEXT7  DC        9           WORD COUNT
815          EBC       .INVALID STATEMENTS.
816   TEXT8  EBC       . ERROR AT STATEMENT NUMBER.
817   *
818   ********************************************************
819   *
820   *                SUBROUTINE
821   *                INCREMENT SYMBOL TABLE POINTER
822   *
823   INCSP  DC        0           LINK
824          LD    I   STPTR       GET ID WD IN SYMBOL TABLE
825          AND    3  H1800-Z    EXTRACT DIMENSION BITS
826          BSC       Z          SKIP IF NOT DIMENSIONED
827          LD     3  HFFFD-Z    MINUS THREE
828          A      3  HFFFD-Z
829          A      3  STPTR-Z    POSITION SYMBOL TB POINTER
830          STO    3  STPTR-Z    *PAST DIMENSIONED VARIABLE
831          S      L  EOFST       TEST FOR END OF SYMBOL TBL
832          BSC    L  INCS2,      BR IF END OF SYMBOL TABLE
833          LD     I  INCSP       IF NOT,RETURN TO ADDRESS
834   *                            *SPECIFIED IN LINK
835          STO       INCS1 1     MODIFY EXIT
836          LDX    I1 STPTR       POINTER TO X1
837   INCS1  BSC    L  *-*         RETURN
838   INCS2  MDX    L  INCSP,1     IF END OF SYMT,RETURN TO
839   *                               ADDR   LINK   1
```

```
840           BSC  I  INCSP     RETURN
841   *
842   *                SUBROUTINE
843   *                CONVERT BINARY VALUE TO
844   *                THREE DECIMAL DIGITS
845   *
846   CON3D   DC       0         LINK
847           SRT      16        SHIFT ARGUMENT TO EXTENSION
848           D        3 C100-Z  GET FIRST DIGIT
849           BSI      GOSTO     STORE FIRST DIGIT
850           LD       3 Z-Z
851           D        3 C10-Z   GET SECOND DIGIT
852           BSI      GOSTO     STORE SECOND DIGIT
853           SLT      16        GET THIRD DIGIT
854           BSI      GOSTO     STORE THIRD DIGIT
855           BSC  I   CON3D     RETURN
856   GOSTO   DC       0         LINK
857           SLA      8         LEFT JUSTIFY DIGIT
858           OR       3 HF000-Z CONVERT TO EBC
859           STO  I   PAP       STORE INTO PRINT AREA
860           MDX  L   PAP,1     INCREMENT PR AREA POINTER
861           BSC  I   GOSTO     RETURN
862   *
863   *                SUBROUTINE
864   *                HEADER TEST
865   *                PRINTS HEADER TO ERROR
866   *                MESSAGES IF FIRST MESSAGE
867   HTEST   DC       ***       LINK TO CALLING POINT 1
868           LD       3 OUTFL-Z LD OUTPUT FLAG
869           BSC  L   HTES3,Z   EXIT ROUTINE IF FLAG IS ON
870           MDX  L   OUTFL,1   ELSE SET FLAG ON AND CONT
871           LD       3 PAPIN-Z LD INIT ADDR OF PRINT AREA
872           STO  L   AREA      STORE ADDR OF AREA HERE
873           LD       3 D60-Z   LD A WORD COUNT AND STORE
874           STO  L   WDCNT     IT IN BUF+1
875           BSI  L   PRINT     PUT A BLANK LINE ABOVE HDR
876   *
877   *
878   HTES1   LDX  I2  ***       LD A WORD COUNT IN XR2
879   *                          *MODIFIABLE
880           STX  L2  WDCNT     STORE IN IN WORD COUNT
881           SLT      32        ZERO THE A AND THE Q
882           LDX  3   0         ZERO XR3
883           LD       HTES1 1   LD THE ADDR OF MSG WD COUN
884           A        ONE       INCREASE IT BY ONE
885           STO      * 1       PUT IT AWAY IN HTES2 1
886   *
887   HTES2   LD   L   ***       LD THE MSG WORD
888           SRT      8         PUT RIGHT HALF IN THE Q
889           SLA      8         LEFT JUSTIFY THE LEFT HALF
890           STO  L3  BUF       STORE UNPACKED WORD
891           SLT      16        PUT RIGHT HALF BACK IN A
892           STO  L3  BUF 1     STORE UNPACKED WORD
893           MDX  3   2         INCREMENT XR3 BY TWO
894           MDX  L   HTES2+1,1 INCREMENT MSG ADDR BY ONE
895           MDX  2   -1        DECREMENT WORD COUNT
896           MDX      HTES2     GO UNPACK ANOTHER MSG WORD
897   *
898           LDX  L3  Z         RESTORE XR3 TO TBL DISP
899           LD   3   PAPIN-Z   LD INT ADDR OF OUTPUT AREA
```

```
900           STO  L  AREA       STORE II IN AREA
901           BSI  L  PRINT      GO PRINT THE MESSAGE
902     *
903           BSI   3 BLKPA-Z    GO PUT EBC BLANKS IN BUF
904     HTES3 BSC  I  HTEST      EXIT THROUGH LINK WORD
905     *
906     *               GET SYM.T. NAME OF LABEL
907     GNAME DC        0        LINK
908           AND       GNAM3    EXTRACT DISPLACEMENT
909           STO       GNAM1 1  SAVE
910           LD   L    SOFST    LOAD START OF SYMBOL TB ADD
911           S         GNAM1 1  COMPUTE SYMBOL TBL ENT ADDR
912           S         GNAM1 1  *
913           S         GNAM1 1  *
914           A      3  THREE-Z  *
915           STO       GNAM0 1  STORE ID WORD ADDRESS
916           A      3  ONE-Z
917           STO       GNAM1 1  STORE FIRST NAME WORD ADDR
918           A      3  ONE-Z
919           STO       GNAM2 1  STORE SECOND NAME WORD ADDR
920     GNAM0 LD   L    ***      LOAD ID WORD
921     *                        *MODIFIABLE
922           STO       GNAM4    STORE ID WORD
923     GNAM1 LD   L    ***      LOAD FIRST NAME WORD
924     *                        *MODIFIABLE
925           STO       NAME     STORE FIRST NAME WORD
926     GNAM2 LD   L    ***      LOAD SECOND NAME WORD
927     *                        *MODIFIABLE
928           STO       NAME 1   STORE SECOND NAME WORD
929           BSC  I    GNAME    RETURN
930     NAME  DEC       0        2 WORD STORAGE FOR NAME
931     *                        *MODIFIABLE
932     GNAM3 DC        /07FF    CONSTANT
933     GNAM4 DC        0        SYMBOL TABLE ID WORD
934     *                        *MODIFIABLE
935     *
936     H07FC DC        /07FC    CONSTANT
937     HF800 DC        /F800    CONSTANT
938     RALO  DC        0        REARR AREA LOW ADDR
939     PTR   DC        0        POINTER
940     *
941     *
942     C10   DC        10       CONSTANT
943     C100  DC        100      CONSTANT
944     C120  DC        120      CONSTANT
945     HFFFD DC        /FFFD    CONSTANT
946     H0008 DC        /0008    CONSTANT
947     H0020 DC        /0020    CONSTANT
948     PAP   DC        BUF      OUTPUT AREA POINTER
949     PAPIN DC        BUF      INITIAL VALUE OF PAP
950     H1800 DC        /1800    CONSTANT
951     STOOI DC        0        STORED OUTPUT INDEX
952     HA7FC DC        /A7FC    CONSTANT
953     H0200 DC        /0200    CONSTANT
954     ZERO  DC        0        CONSTANT
955     Z     EQU       ZERO     TRANSFER VECTOR PT ADDR
956     ONE   DC        /0001    CONSTANT
957     TWO   DC        /0002    CONSTANT
958     THREE DC        /0003    CONSTANT
959     FOUR  DC        /0004    CONSTANT
```

```
 960  STLAB DC       0          STORED LABEL
 961  STCNT DC       0          STORED COUNT
 962  *                          NUMBER OF STATEMENTS
 963  *                         FROM LAST LABEL
 964  OUTFL DC       0          OUTPUT FLAG
 965  STSID DC       0          STORED STMNT-ID
 966  NORM  DC       0          STORED NORM
 967  IDMSK DC       /F800      MASK FOR EXTR STMNT-ID
 968  NRMSK DC       /07FC      MASK FOR EXTR NORM
 969  ERRID DC       /A000      ERROR ID
 970  ENDID DC       /1000      END ID
 971        BSS  E   0
 972  FZERO DC       /30C3      CONSTANT FOR FIVE ZEROS
 973        DC       /0C30      *
 974  STPTR DC       0          SYMBOL TABLE POINTER
 975  HD000 DC       /D000      CONSTANT
 976  HD800 DC       /D800      CONSTANT
 977  H4000 DC       /4000      CONSTANT
 978  HF000 DC       /F000      CONSTANT
 979  HC000 DC       /C000
 980  PAR31 DC       BUF 31     CONSTANT
 981  H3F00 DC       /3F00      CONSTANT
 982  PLUS  DC       /4E00      CONSTANT
 983  C     DC       /C300      CONSTANT
 984  D60   DC       60         CONSTANT
 985  PCNT  DC       *-*        PRINT COUNT
 986  *
 987  *                         SET ERROR SWITCH IN COMMUN AREA
 988  *                         TO PREVENT FINAL OUTPUT
 989  *
 990  ERRSW DC       0          LINK
 991  *
 992        LD   L   ERROR      SET ERROR SWITCH
 993        OR     3 TWO-Z      *
 994        STO  L   ERROR      *
 995        BSC  I   ERRSW      RETURN
 996  *
 997  *                         SUBROUTINE
 998  *                         MOVE NAME TO PRINT AREA
 999  *
1000  MNAME DC       0          LINK
1001        BSI    3 HTEST-Z    TEST IF HEADER PRINTED
1002        BSI      TOPAB      BLANK TO PRINT AREA
1003        LD     1 1          GET FIRST CHARACTER
1004        BSI      TOPA       OUTPUT FIRST CHARACTER
1005        LD     1 1          GET SECOND CHARACTER
1006        SLA      8
1007        BSI      TOPA       OUTPUT SECOND CHARACTER
1008        LD     1 2          GET THIRD CHARACTER
1009        RTE      16
1010        LD     1 1
1011        SLT      12
1012        BSI      TOPA       OUTPUT THIRD CHARACTER
1013        LD     1 2          GET FOURTH CHARACTER
1014        SLA      2
1015        BSI      TOPA       OUTPUT FOURTH CHARACTER
1016        LD     1 2          GET FIFTH CHARACTER
1017        SLA      8
1018        BSI      TOPA       OUTPUT FIFTH CHARACTER
1019        BSI      TOPAB      OUTPUT ONE BLANK
```

```
1020          BSI      TOPAB     OUTPUT ANOTHER BLANK
1021    *
1022    *                INCREMENT PRINT COUNT
1023          LD     3 PCNT-Z    INCREMENT PRINT COUNT
1024          A      3 H0008-Z   *BY 8
1025          STO    3 PCNT-Z
1026    *
1027    *     TEST IF LINE FULL
1028          S      3 C120-Z    SEE IF LINE FULL
1029          BSC    I MNAME, Z  RETURN IF LINE NOT FULL
1030          BSI    3 HFILL-Z   PRINT THIS LINE
1031          BSC    I MNAME     RETURN
1032    *
1033    *                SUBROUTINE
1034    *                CHAR IN A-REG TO PRINT AREA UNCHANGE
1035    *
1036    TOPAU DC         0         LINK
1037          STO        STOCH     SAVE CHARACTER
1038    TOPAX LD         STOCH     LOAD CHARACTER
1039          STO    I   PAP       SET INTO PR AREA UNCHANGED
  40          MDX    L   PAP,1     INCR PRINT AREA POINTER
1041          BSC    I   TOPAU     RETURN
1042    *
1043    *                SUBROUTINE
1044    *                CONVERT CHAR IN A-REG INTO
1045    *                EBC-CODE, THEN MOVE TO PRINT AREA
1046    *
1047    TOPA  DC         0         LINK
1048          AND    3 H3F00-Z   MASK TO LEAVE CHARACTER
1049          BSC    L TOPA2, -  BR IF BLANK
1050    *
1051    *                NOTE
1052    *                IF OTHER SPECIAL CHARACTERS THAN
1053    *                BLANK ARE EXPECTED, TESTING SHOULD
1054    *                BE HERE
1055          OR     3 HC000-Z   CONVERT TO EBCDIC
1056    TOPA1 STO        STOCH     STORE CHARACTER
1057          LD         TOPA      SET UP TO RETURN FROM
1058          STO        TOPAU     *TOPAU
 059          MDX        TOPAX     GO OUTPUT CHARACTER
1060    TOPA2 LD       3 H4000-Z   LOAD EBCDIC BLANK
1061          MDX        TOPA1     GO OUTPUT
1062    *
1063    TOPAB DC         0         LINK
1064          SLA        16        LOAD DUMMY BLANK
1065          BSI        TOPA      GO OUTPUT A BLANK
1066          BSC    I   TOPAB     RETURN
1067    STOCH DC         *-*       STORED CHARACTER
1068    *
1069    *
1070    *                SUBROUTINE
1071    *                BLANK TO PRINT AREA
1072    *
1073    BLKPA DC         0         LINK
1074          LD       3 H4000-Z   LOAD EBCDIC BLANK
1075          LDX      3 -120      SET UP TO BLANK PRINT AREA
1076    BLKP1 STO      L3 BUF 120    PUT EBC BLANK IN EACH WORD
1077          MDX      3 1           MOVE WORD POINTER
1078          MDX        BLKP1       LOOP BACK UNTIL ALL BLANKE
1079          LDX    L3 Z          SET TRANSFER VECTOR POINTER
```

```
1080          BSC  I  BLKPA     RETURN
1081  *
1082  *               TEST IF HALF FILLED LINE
1083  *               IN PRINT AREA
1084  HFILL   DC      ***       LINK
1085          LD     3 PCNT-Z   LOAD PRINT COUNT
1086          BSC  L  XHF, -     RETURN IF ZERO
1087          LD     3 PAPIN-Z  LOAD PRINT AREA ORIGIN
1088          STO    3 PAP-Z    RESET PRINT AREA POINTER
1089          STO  L  AREA      SET MESSAGE ADDRESS
1090          LD     3 D60-Z
1091          STO  L  WDCNT     SET WORD COUNT
1092          BSI  L  PRINT     PRINT LINE
1093          BSI     BLKPA     BLANK PRINT AREA
1094          SLA     16         ZERO THE ACCUMULATOR
1095          STO    3 PCNT-Z   ZERO PRINT COUNT
1096  XHF     BSC  I  HFILL     RETURN
1097  *
1098  *
1099  EXIT    BSI  L  ROLRX      CALL DOWN PHASE 22
  00          DC      22                NEXT PHASE NUMBER
1101          BSS     OVERL-*+320*3  PHASE-21 PATCH AREA
1102          END     NEQ
```