

```

1 * GENERAL AUTOMATION, INC. ALL RIGHTS RESERVED
2 *****
3 *
4 * PROGRAM NAME RPH-12
5 *
6 * MODEL NUMBER 8F012
7 *
8 * PURPOSE FORTRAN PHASE-12
9 *
10 * PROGRAMMER DICK WALLMANN
11 *
12 ***** REVISION LIST *****
13 *
14 * RV DATE SCC BY REASON FOR CHANGE
15 * -----
16 *
17 * 01 11/16/70 NONE RPH INITIAL RELEASE
18 *
19 *****
20 *****
21 HDNG MPX FORTRAN ** ASCAN T
22 *****
23 *STATUS-VERSION 1, MODIFICATION 0
24 *
25 *FUNCTION/OPERATION-
26 * * EXAMINES ONLY THE ARITHMETIC, IF, CALL, AND
27 * STATEMENT FUNCTION STATEMENTS IN THE STMT
28 * STRING.
29 * * CHECKS SYNTAX OF STATEMENTS EXAMINED.
30 * * INSERTS A CALL OPERATOR BETWEEN THE SUBPROG
31 * NAME AND ITS DUMMY ARGUMENTS IN A CALL STMT
32 * * CHECKS STATEMENT FUNCTION CALLS, INCLUDING
33 * NESTED CALLS, FOR VALID NAMES AND THE
34 * CORRECT NUMBER OF PARAMETERS.
35 * * CHECKS TO SEE THAT VARIABLES ARE DEFINED
36 * CORRECTLY AND SETS DEFINED INDF IN SYM TBL.
37 * * CHECKS FOR VALID STATEMENT NUMBER REFERENCES
38 * IN IF STATEMENTS. REFERENCES TO FORMAT
39 * STATEMENT NUMBERS ARE INVALID.
40 * * CHECKS SYNTAX OF THE RECORD NUMBER
41 * EXPRESSION IN DISK READ/WRITE STATEMENTS.
42 * THE RIGHT PARENTHESIS IS CHANGED TO A COLON
43 * OPERATOR FACILITATING SCAN OF DISK READY
44 * WRITE STMT IN THE NEXT PHASE.
45 *
46 *ENTRY POINTS-
47 * * START-PHASE 12 IS LOADED BY PHASE 11 VIA A
48 * CALL TO POLRX. EXECUTION IS BEGUN
49 * AT LOCATION START.
50 *INPUT-
51 * * THE STATEMENT STRING
52 * * THE SYMBOL TABLE
53 * * FCOM
54 *
55 *OUTPUT-
56 * * THE STATEMENT STRING
57 * * THE SYMBOL TABLE
58 * * FCOM
59 *

```

```

60 *EXTERNAL REFERENCES-
61 *   * SUBROUTINES-
62 *     ROLRX
63 *   * OTHER FORTRAN PHASES-
64 *     NONE
65 *
66 *EXITS-
67 *   * NORMAL-
68 *     EXITS VIA A CALL TO THE ROLRX ROUTINE TO
69 *     READ IN PHASE 13.
70 *   * ERRORS-
71 *     OVERLAP-BYPASSES PROCESSING,SETS ERROR WORD
72 *     IN FCOM, CALLS PHASE 13 VIA ROLRX.
73 *     SYNTAX- ERRORS DETECTED BY THIS PHASE ARE
74 *     NUMBERS 36,37,38,39,40,41,42,43.
75 *
76 *TABLES/WORK AREAS-
77 *   * THE STATEMENT STRING
78 *   * THE SYMBOL TABLE
79 *   * THE FORTRAN COMMUNICATION AREA
80 *
81 *ATTRIBUTES-NONE
82 *
83 *NOTES-
84 *   THE SWITCHES USED BY THIS PHASE FOLLOW. IF
85 *   NON-ZERO, THE SWITCH IS TRANSFER T. IF ZERO,
86 *   THE SWITCH IS NORMAL N
87 *     SW1-DIMENSIONED NAME SUBSCRIPTED
88 *     T   NOT SUBSCRIPTED
89 *     SW4-SIGN ALLOWABLE
90 *     N   SIGN ALLOWABLE
91 *     SW5-POINTER INSIDE FUNCTION CALL
92 *     T   POINTER INSIDE FUNCTION CALL
93 *     SW7-DIMENSIONED NAME
94 *     T   NAME DIMENSIONED
95 *
96 *****
97 HDNG      MPX FORTRAN ** ASCAN I
98 ABS   REF CORE
99 *
100 *      SYSTEM AND FORTRAN EQUATES
101 *
102 MEMRY EQU   FFFF CORE      MAXIMUM CORE SIZE
103 PHSIZ EQU   4*320                MAXIMUM PHASE SIZE
104 OVERL EQU   MEMRY-PHSI7          PHASES 2-29 START
105 FCOM EQU    OVERL-22             FORTRAN COMM. TABLE
106 PHNTR EQU   FCOM-56              PHASE TABLE
107 ROLRX EQU   PHNTR-50             INTERPHASE CALL
108 *
109 *      FORTRAN COMMUNICATION AREA
110 *
111          ORG      FCOM          FORTRAN COMMUNICATION AREA
112 SOFS BSS     1                START OF STRING
113 EOFS BSS     1                END OF STRING
114 SOFST BSS    1                START OF SYMBOL TABLE
115 SCFNS BSS    1                START OF NON-STMNT NOS.
116 SOFXT BSS    1                START OF SUBSC TEMPS
117 SOFGT BSS    1                START OF GENERATED TEMPS
118 EOFST BSS    1                END OF SYMBOL TABLE
119 COMMON BSS   1                NEXT AVAILABLE COMMON

```

120	OSIZE	BSS	1	SIZE OF COMMON
121	ERROR	BSS	1	OVERLAP ERROR
122	FNAME	BSS	1	PROGRAM NAME WD 1
123		BSS	1	PROGRAM NAME WD 2
124	SORF	BSS	1	SUBR (-) OR FUNC (+)
125	CCWD	BSS	1	CONTROL CARD WORD
126	*			BIT 15 TRANSFER TRACE
127	*			BIT 14 ARITHMETIC TRACE
128	*			BIT 13 EXTENDED PRECISION
129	*			BIT 12 LIST SYMBOL TABLE
130	*			BIT 11 LIST SUBPROGRAM NAMES
131	*			BIT 10 LIST SOURCE PROGRAM
132	*			BIT 9 ONE WORD INTEGERS
133	*			BIT 8 PUNCH
134	*			BIT 7 NONPROCESS PROGRAM
135	IOCS	BSS	1	IOCS CONTROL CARD WORD
136	*			
137	*			SEE PHASE ONE FOR BIT PATTERNS
138	*			
139	DFCNT	BSS	1	DEFINED FILE COUNT
140	*			
141	LCOMN	BSS	2	SIZE OF INSKEL COMMON
142	*			
143	ICCR	BSS	2	IOCS CONTROL CARD ERROR
144	*			
145	*			
146		BSS	2	SYSTEM LOADER USE
147	*			END OF FORTRAN COMMUNICATION
148	*			AREA
149	*****			
150	*			
151		HDNG		MPX FORTRAN ** ASCAN I
152	*			CHECK FOR OVERLAP ERROR
153		ORG		OVERL PHASE ENTRY
154	*			
155	START	LD		ERROR IF NO OVERLAP ERROR THEN B
156		BSC	L	ORGIN, - TO MOVE STRING.
157	*			
158	*			READ AND GO TO NEXT PHASE
159	*			
160	OVERF	BSI	L	ROLEX CALL DOWN PHASE 17
161		OC		13 NEXT PHASE NUMBER
162	*			MOVE STRING NEXT TO SYMBOL TABLE
163	*			
164	ORGIN	LD		EOFS END OF STRING ADDRESS
165		S		SOFS START OF STRING ADDRESS
166		STO		* 1 ADDR OF NEXT INSTRUCTION
167		LDX	L3	*-* RANGE OF MOVE, MODIFIABLE
168		MDX	3	1 NO. OF WORDS INCLUSIVE
169		LDX	I2	EOFS END OF STRING ADDR
170		LDX	I1	EOFST END OF SYMBOL TABLE ADDRESS
171	ORGA1	LD	2	0 NEXT WORD TO MOVE
172		STO	1	2 MOVE STRING NEXT TO SYM TB
173		MDX	1	-1 DECR MOVE TO ADDRESS
174		MDX	2	-1 DECR MOVE FROM ADDRESS
175		MDX	3	-1 DECR NO. OF WORDS TO MOVE
176		MDX		ORGA1 CONTINUE
177		MDX	1	3 INCR TO EQUAL 1ST STMT PT
178	*			
179	*			INITIALIZE INPUT AND OUTPUT

```

180 *          STRING POINTERS
181 *
182     LDX  I2 SOFS      START OF STRING ADDR
183     MDX   2 -1       DECR BY 1
184     STX  L2 CKS2 1   STORE AS ASF TABLE ADDRESS
185     MDX   2 2
186     STX  L2 XY33 1   ASF TABLE REFERENCE
187     MDX   2 -1
188     STX  L2 CKS3 1   STORE FOR ASF ARGUMENT CNT
189 *
190 *          INITIALIZE TO SCAN STATEMENT
191 *
192 XY1  LD    1 0       STMT ID WORD
193     AND  S07FF      EXTRACT STMT WORD COUNT
194     SRA   2         RIGHT JUSTIFY
195     STO  L  XY27 1   STORE AS INSTRUCTION ADDR
196     STX  1 XT1      INDEX REG 1 TEMPORARY
197     A    XT1        ADDR OF STMT ID WORD
198     STO  L  NXTID   ADDR OF NEXT STMT ID WORD
199     STX  L2 IDSV2   INDEX REG 2 TEMPORARY
200     STX  1 IDSV1   INDEX REG 1 TEMPORARY
201     BSI  L  MOVE1   BR TO OUTPUT 1 ELEMENT
202     LD   1 -1       STMT ID WORD
203     EOR  S0001      EXTRACT AND REVERSE BIT 15
204     BSC  L  XY2,E   BR IF NOT A NUMBERED STMT
205     BSI  L  MOVE1   BR TO OUTPUT ONE ELEMENT
206     MDX  L  XY27 1,-1 ADJUST TO EXCLUDE STMT NO
207 *
208 *          CHECK FOR END STATEMENT
209 *
210 XY2  LD   L  *-*    STMT ID WD. MODIFIABLE
211 IDSV1 EQU  XY2 1   LABEL PREVIOUS ADDRESS
212     SRA  11        RIGHT JUSTIFY STMT TYPE
213     S    TEND      END STMT CONSTANT
214     BSC  L  XY5,2  BR IF NOT END STMT
215 *
216 *          REMOVE ASF TABLE IF PRESENT
217 *
218     MDX  L  CKAS 1,-1 DECR ASF TABLE COUNTER
219     MDX   * 1       BR IF ASF TABLE
220     MDX  XY34       BR IF NO ASF TABLE
221     STX  2 EOFS     END OF STRING ADDR
222     LD   SOFS      START OF STRING ADDR
223     A    L  CKAS 1   ASF TABLE COUNT
224     STO  * 1       ADDR OF NEXT INSTRUCTION
225     LDX  L1 *-*    MOVE FROM ADDRESS
226 *          *MODIFIABLE
227     LDX  I2 SOFS   MOVE TO ADDRESS
228     S    L  EOFS   END OF STRING ADDRESS
229     STO  * 1       ADDR OF NEXT INSTRUCTION
230     LDX  L3 *-*    RANGE OF MOVE. MODIFIABLE
231 MV1  LD   1 0      NEXT WORD TO MOVE
232     STO  2 0       REMOVE ARITH STMT FUNC TB
233     MDX  1 1       INCR MOVE FROM ADDR
234     MDX  2 1       INCR MOVE TO ADDR
235     MDX  3 1       INCR MOVE COUNTER
236     MDX  MV1      BR TO CONTINUE MOVE
237 XY34 MDX  2 -1     END OF STRING   XF2-1
238     STX  L2 EOFS   NEW END OF STRING ADDR
239     MDX  OVERF    LOOP

```

```

240 *
241 *          CONSTANTS AND WORK AREA
242 *
243 XT1  DC          0          INDEX REG 1 TEMPORARY
244 S07FF DC        /07FF      STMNT TYPE MASK CONSTANT
245 TEND  DC        /02        END STMNT CONSTANT
246 ERNO  DC          0          ERROR NUMBER TEMPORARY
247 S0001 DC        /0001      STMNT NO. INDR BIT MASK
248 SF802 DC        /F802      STMNT WORD COUNT MASK
249 S0020 DC        /0020      DEFINED BIT CONSTANT
250 *
251 *          CHECK FOR STMNT TERMINATOR
252 *
253 XY13  LD         1  0          NEXT STMNT WORD
254          BSC      L  XY7,7      PR IF NO TERMINATOR
255 XY11  BSI      L  PUT          PR TO MOVE 1 WD TO OUTPUT
256 ORCTN LD         I  IDSV2      STMNT ID WITHOUT WD COUNT
257          AND      L  SF802      ZERO WORD COUNT BITS
258          STO      XT4          STMNT ID WORD TEMPORARY
259          STX      2  XT3        INDEX REG 2 TEMPORARY
260          LD         XT3        NEXT STMNT ID WD ADDRESS
261          S         IDSV2      CURRENT STMNT ID WD ADDRESS
262          SLA      2          POSITION TO STORE STMNT CN
263          A         XT4          STMNT ID W/O WORD COUNT
264          STO      I  IDSV2      CORRECTED STMNT ID WORD
265          MDX      XY22        PR TO MOVE TO NEXT STMNT
266 *
267 *          CONSTANTS AND WORK AREA
268 *
269 XT3   DC          0          INDEX REG 2 TEMPORARY
270 XT4   DC          0          STMNT ID WORD TEMPORARY
271 ERID  DC        /A008      ERROR ID WORD
272 *
273 *          REPLACE STATEMENT WITH ERROR
274 *
275 XY7   LDX      L2  *-*        STMNT ID WORD ADDRESS
276 ATERR EQU      XY7          LABEL LAST INSTRUCTION
277 IDSV2 EQU      XY7 1        LABEL LAST INSTRUCTION ADD
278 *          *MODIFIABLE
279          LD         2  0          OUTPUT STMNT ID WORD
280          EOR      FIVE        REVERSE STMNT NO. INDR BIT
281          BSC      L  * 5,E      PR IF NOT A NUMBERED STMNT
282          LD         FIVE        STMNT NUMBERED INDR
283          A         ERID        SET BIT 15 ON AND INCR CNT
284          STO      2  0          REPLACE STMNT ID WORD
285          MDX      2  1          INCR STMNT POINTER
286          MDX      * 3          PR TO SET UP ERROR NUMBER
287          LD         L  ERID      ERROR ID WORD
288          STO      2  0          STMNT ID WORD
289          LD         L  ERNO      ERROR NUMBER
290          STO      2  1          STORE ON OUTPUT STRING
291          MDX      2  2          INCR STMNT POINTER
292 XY22  LDX      L1  *-*        NEXT STMNT ID WORD ADDRESS
293 NXTID EQU      XY22 1        LABEL FOR PRECEDING ADDRESS
294 *          *MODIFIABLE
295          BSC      L  XY1        PR TO SCAN NEXT STMNT
296 *
297 *          CHECK FOR AN ARITHMETIC STATEMENT
298 *
299 XY5   S         TARTH        ARITH STMNT CONSTANT

```

```

300      BSC  L  XY23,2  BR IF NOT ARITH STMNT
301      LDX  3  36      ERROR NUMBER 36
302      STX  L3 ERNC    SET UP ERROR NUMBER
303      LD   1  0      STMNT ID WORD
304      BSI          GETID  BR TO GET SYMBOL TABL ID WD
305      BSI  L  TESTV   BR TO TEST VARIABLE NAMES
306      LD   3  0      SYMBOL TABLE ID WORD
307      OR   L  S0020   SET DEFINED PIT ON
308      STO  3  0      RESTORE ID WORD
309      BSI  L  NAME    BR TO CHK NAME DIMENSIONIN
310      LD   L  SW1     TEST DIMENSIONED NAME SUPS
311      BSC  L  XY7,7   BR IF NOT SUBSCRIBED
312      *
313      *           SCAN ARITH EXPRESSION IF EQUAL SIGN
314      *
315  XY5A  LD   1  0      NEXT STMNT WORD
316      S          EQSGN  EQUAL SIGN CONSTANT
317      BSC  L  XY7,7   BR IF NOT EQUAL SIGN
318      BSI  L  MOVE1   BR TO MOVE ONE ELEMENT
319      BSI  L  ARITH   BR TO ARITH SCAN
320      MDX          XY13  BR TO CHK FOR SEMICOLON
321      *
322      *           CHECK FOR IF STATEMENT
323      *
324  XY23  S          TIF   IF STMNT CONSTANT
325      BSC  L  XY25,7  BR IF NOT IF STMNT
326      LDX  3  37      ERROR NUMBER 37
327      STX  L3 ERNC    SET UP ERROR NUMBER
328      LD   IFOP      IF OPERATOR
329      BSI          CHECK  BR TO CHK STRING OVERLAP
330      *
331      *           CHECK FOR LEFT PARENTHESIS
332      *
333      LD   1  0      NEXT STMNT WORD
334      S          ULP    LEFT PARENTHESIS CONSTANT
335      BSC  L  XY7,7   BR IF NOT LEFT PARENTHESIS
336      BSI  L  MOVE1   BR TO MOVE ONE ELEMENT
337      LDX  3  38      ERROR NUMBER 38
338      STX  L3 ERNC    SET UP ERROR WORD
339      BSI  L  ARITH   BR TO ARITH SCAN
340      LDX  3  37      ERROR NUMBER 37
341      STX  L3 ERNC    SET UP ERROR WORD
342      *
343      *           CHECK FOR RIGHT PARENTHESIS
344      *
345      LD   1  0      NEXT STMNT WORD
346      S          URP    RIGHT PARENTHESIS CONSTANT
347      BSC  L  XY7,7   BR IF NOT RIGHT PARENTHESI
348      BSI  L  MOVE1   BR TO MOVE ONE ELEMENT
349      *
350      *           CHECK FOR COMMA OPERATOR
351      *
352      LD   UCMA      COMMA OPERATOR CONSTANT
353      BSI          CHECK  BR TO CHECK STRING OVERLAF
354      BSI  L  LIST    BR TO CHK STMNT NO. LISTS
355      LD   L  CNT     STMNT NO. COUNT
356      S          FIVE   DECIMAL FIVE
357      BSC  L  XY7,7   BR IF NOT FIVE
358      MDX          XY13  BR TO CHK FOR SEMICOLON
359      *

```

```

360 *          CONSTANTS AND WORK AREA
361 *
362 TARTH DC      /00- /02   ARITHMETIC IF
363 ECGN  DC      /0E           OPERATOR
364 FIVE  DC      5           DECIMAL FIVE CONSTANT
365 TIF   DC      /0F- /00   IF ID
366 IFOP  DC      /14           IF OPERATOR
367 ULP   DC      /10           OPERATOR
368 URP   DC      /02           OPERATOR
369 UCMA  DC      /12           , OPERATOR
370 *****
371 *          THIS SUBROUTINE GETS THE ID WORD
372 *          FROM THE SYMBOL TABLE. THE ENTRY IS
373 *          A BSI INSTRUCTION TO THE LABEL GETID
374 *****
375 *
376 *          GET ID WORD
377 *
378 GETID DC      0           RETURN ADDRESS
379      BSC  L  XY7,-        BR IF NOT SYMBOL TBL PT
380      AND  SO1FF          EXTRACT TABLE DISPLACEMENT
381      SLA  2              EFFECTIVE MULTIPLY BY 4
382      STO  GET1C          TEMPORARY STORE
383      SRA  2              RESTORE - RIGHT JUSTIFY
384      S    GET1C          -3 * DISPLACEMENT
385      A    L  SOFST        START OF SYMPOLE TBL ADDR
386      STO  * 1            ADDR OF NEXT INSTRUCTION
387      LDY  L3 *-*         INDEX 3 SYMBOL TBL ADDR
388 GET1C EQU  *-1          MODIFIABLE
389      MDX  3 3            ADJUST ADDRESS
390      LD   3 0            SYMBOL TABLE ID WORD
391      BSC  I  GETID       BR TO EXIT GETID SUBROUTIN
392 *
393 *          CONSTANTS AND WORK AREA
394 *
395 SC1FF DC      /07FF        SYMBOL TBL POINTER MASK
396 *
397 *****
398 *          THIS SUBROUTINE PLACES THE ITEMS
399 *          ON THE NEW STRING. THE ENTRY IS A
400 *          BSI INSTRUCTION TO THE LABEL PUT.
401 *****
402 *          STORE ACCUMULATOR ON NEW STRING
403 *
404 PUT   DC      0           RETURN ADDRESS
405      STO  2 0            ONE ELEMENT TO NEW STRING
406      MDX  2 1            INCR NEW STRING POINTER
407      BSC  I  PUT         BR TO EXIT FROM PUT
408 *****
409 *          THIS SUBROUTINE CHECKS FOR POOM
410 *          BETWEEN THE TWO STRINGS. THE ENTRY
411 *          IS A BSI INSTRUCTION TO THE LABEL
412 *          CHECK.
413 *****
414 *
415 *          CALL PUT
416 *
417 CHECK DC      0           RETURN ADDRESS
418      BSI  PUT           BR TO MOVE ONE ELEMENT
419 *

```

```

420 *          CHK FOR XR1 GREATER THAN XR2
421 *
422     STX     1 CKTEM     INDEX REGISTER 1 TEMPORARY
423     LD      CKTEM     ADDR OF INPUT STRING
424     STX     2 CKTEM     INDEX REGISTER 2 TEMPORARY
425     S       CKTEM     ADDR OF OUTPUT STRING
426     BSC     L CK1,     BR IF STRING OVERLAP
427     BSC     I CHECK    BR TO EXIT
428 CK1 MDX     L ERROR,1  SET OVERLAP ERROR ON
429     BSC     L OVERF    BR TO RD AND GO TO NEXT PH
430 *
431 *          CONSTANTS AND WORK AREA
432 *
433 CKTEM DC      0          INDEX REG TEMPORARY STORE
434 *
435 *****
436 *          THIS SUBROUTINE OUTPUTS ONE ELEMENT
437 *          AND MOVES THE POINTER. THE ENTRY IS
438 *          A BSI INSTRUCTION TO THE LABEL MOVE1
439 *****
440 *          CALL PUT
441 *
442 MOVE1 DC      0          RETURN ADDRESS
443     LD      1 0          NEXT STMT WORD
444     BSI     PUT          BR TO MOVE ONE ELEMENT
445     MDX     1 1          INCR INPUT POINTER
446     BSC     I MOVE1     BR TO EXIT
447 *****
448 *          THIS SUBROUTINE OUTPUT UNDER CONTROL
449 *          OF XR3. THE ENTRY IS A BSI INSTRUC.
450 *          TO THE LABEL OUT.
451 *****
452 *          CALL MOVE1
453 *
454 OUT  DC      0          RETURN ADDRESS
455     BSI     MOVE1       BR TO MOVE ONE ELEMENT
456     MDX     3 -1        DECR MOVE COUNT
457     MDX     OUT 1       BR TO OUTPUT ONE MORE
458     BSC     I OUT       EXIT IF COUNT IS ZERO
459 *****
460 *          THIS SUBROUTINE HANDLES THE NAME
461 *          AND THE SUBSCRIPT IF THERE IS AND
462 *          SHOULD BE ONE. THE ENTRY IS A BSI
463 *          INSTRUCTION TO THE LABEL NAME.
464 *****
465 *
466 *          CALL PUT
467 *
468 NAME DC      0          RETURN ADDRESS
469     LD      1 0          NEXT STMT WORD
470     BSI     PUT          BR TO OUTPUT ONE ELEMENT
471     BSI     GETID       BR TO GET SYMBOL TBL TO WD
472     STO     SVID        TO WORD TEMPORARY
473     AND     S1800       EXTRACT DIMENSION RTTS
474     BSC     L NAME1,2   BR IF NOT DIMENSIONED NAME
475     MDX     1 1          INCR STMT POINTER
476     STX     0 SW7       SET DIMENSIONED NAME SW ON
477 NAME3 SLA     16        CLEAR ACCUMULATOR
478     STO     SW1         ZERO DIMENSION SUBSC INDR
479 NAME8 BSC     I NAME     BR TO EXIT

```



```

480 *
481 *           RESET DIMENSIONED NAME SWITCH
482 *
483 NAME1 SLA      16      CLEAR ACCUMULATOR
484         STO      SW7      RESET DIMENSIONED NAME SW
485         MDX      1 1      INCR STMT POINTER
486         LD       1 0      NEXT STMT WORD
487         S        NP3      3 DIMENSION OPERATOR CON
488         BSC     L NAME2,7  BR IF NOT 3 DIMENSION
489         LDX      3 15     SET UP OUTPUT SPACE IN XR3
490 NAME4 BSI      OUT      BR TO MOVE TO OUTPUT STRIN
491         MDX      NAME3    BR TO NORMALIZE SWITCH 1
492 NAME2 S        NP2      2 DIMENSION OPERATOR CON
493         BSC     L NAME5,7  BR IF NOT 2 DIMENSION
494         LDX      3 11     SET UP OUTPUT SPACE IN XR3
495         MDX      NAME4    BR TO OUTPUT
496 NAME5 S        NP1      1 DIMENSION OPERATOR CON
497         BSC     L NAME6,7  BR IF NOT 1 DIMENSION
498         LDX      3 7      SET UP OUTPUT SPACE IN XR3
499         MDX      NAME4    BR TO OUTPUT
500 NAME6 S        NP0      0 DIMENSION OPERATOR CON
501         BSC     L NAME7,7  BR IF NOT 0 DIMENSION
502         LDX      3 3      SET UP OUTPUT SPACE IN XR3
503         MDX      NAME4    BR TO OUTPUT
504 NAME7 STX      0 SW1     SET SWITCH 1 ON
505         MDX      NAME8    BR TO EXIT FROM NAME
506 *
507 *           CONSTANTS AND WORK AREA
508 *
509 SW1  DC        0        DIMENSIONED NAME SUBSC SW
510 SW7  DC        0        DIMENSIONED NAME SWITCH
511 S1800 DC      /1800     DIMENSION INDICATOR MASK
512 NP3  DC        /1E      3 OPERATOR CONSTANT
513 NP2  DC        /10- /1E  2 OPERATOR CONSTANT
514 NP1  DC        /1A- /10  1 OPERATOR CONSTANT
515 NP0  DC        /18- /1A  0 OPERATOR CONSTANT
516 SVID DC        *- *    SYMBOL TABLE ID WORD TEMP
517 *****
518 *           THIS SUBROUTINE CHECKS THE STATEMENT
519 *           NUMBER LIST OF IF AND GOTO STATEMENT
520 *           ENTRY IS A BSI INSTRUCTION TO THE
521 *           LABEL LIST
522 *****
523 LIST DC        0        RETURN ADDRESS
524         SLA      16      CLEAR ACCUMULATOR
525         STO      CNT     ZERO COUNT OF WORDS OUTPUT
526 *
527 *           CHK FOR NON-FORMAT STMT NUMBER
528 *
529 LIST3 LD       1 0      NEXT STMT WORD
530         BSI     L GETID   BR TO GET SYMBOL TABLE ID
531         AND     SFFDF    REMOVE REFERENCED NO. INCR
532         EOR     S0200    FLIP STMT NO. INCR BIT
533         BSC     L LIST2,7 BR IF NO STMT NUMBER
534         BSI     MOVE1    BR TO MOVE 1 WD TO OUTPUT
535         MDX     L CNT,1  INCR WORDS OUTPUT COUNT
536 *
537 *           CHECK FOR COMMA CHARACTER
538 *
539         LD       1 0      NEXT STMT WORD

```

```

540          S          LCMA          COMMA CONSTANT
541          BSC L      LIST4, -     PR IF COMMA
542          MDX          LIST5          BR TO EXIT
543 *
544 *          SET UP ERROR 43
545 *
546 LIST2 LDX      3 43          INDEX REG 3 43
547          STX L3 ERNO          SET UP ERROR WORD
548 *
549 *          RETURN
550 *
551 LIST5 BSC I      LIST          EXIT FROM LIST CHECK SUBR
552 LIST4 MDX L      CNT,1          INCR WORD OUTPUT COUNT
553          BSI L      MOVE1          PR TO OUTPUT ONE WORD
554          MDX          LIST3          PR TO GET SYMBOL TBL ID WD
555 *
556 *          CONSTANTS AND WORK AREA
557 *
558 CNT DC          0          NO. OF WDS OUTPUT COUNTER
559 LCMA DC          /12          COMMA CHARACTER CONSTANT
560 SFFDF DC          /FFDF          SYMBOL TABLE ID WORD MASK
561 S0200 DC          /0200          STMT NUMBER INDR MASK
562 *****
563 *          THIS SUBROUTINE CHECKS AND ALTERS
564 *          ALL ARITHMETIC EXPRESSIONS TO BE
565 *          SCANNED. THE ENTRY IS A BSI COMMAND
566 *          TO THE LABEL ARITH. INDEX REGISTER
567 *          1 POINTS AT THE RIGHT STRING AND
568 *          INDEX REGISTER 2 POINTS AT THE LEFT
569 *          STRING. THE ARITHMETIC EXPRESSION
570 *          IS MOVED FROM THE RIGHT STRING TO
571 *          THE LEFT STRING AS IT IS CHECKED.
572 *          A NEST CALL TABLE IS INSERTED IN THE
573 *          STRING AS NEEDED.
574 *****
575 *
576 *          INITIALIZE TO HANDLE EXPRESSIONS
577 *
578 ARITH DC          0          SUBR ENTRY POINT
579          SLA          16          CLEAR ACCUMULATOR
580          STO          PAR          CLEAR PARENTHESIS COUNTER
581          STO          SW4          CLEAR SIGN ALLOWABLE SWITC
582          STO          SW5          CLEAR PT INSIDE FUNCTION S
583          STO          CALLN          FUNCTION CALL INDICATOR
584          STO          ASFN          ARITH STMT FUNC INDICATOR
585          LDX      3 1          SET INDEX REG 3 1
586          STX      3 PARNC          INITIALIZE PARAMETER COUNT
587          LD L      NXTID
588          STO L      ATH14 1          ADDR OF NEXT STMT TO WORD
589 ATH3 LD      1 0          NEXT STMT WORD
590          BSC L      ATH1, Z          PR IF NOT OPERATOR
591          S          ARPN          LEFT PARENTHESIS CONSTANT
592          BSC L      ATH2,Z          PR IF NOT LEFT PARENTHESIS
593 *
594 *          PAR PAR 1
595          MDX L      PAR,1          INCR PARENTHESIS COUNT
596 ATH12 BSI L      MOVE1          PR TO MOVE ONE WD TO OUTPUT
597          SLA          16          CLEAR ACCUMULATOR
598          STO          SW4          NORMALIZE SWITCH /
599          MDX          ATH3          PR TO CHK NEXT STMT WORD

```

600	ATH2	LD		SW4	SIGN ALLOWABLE SWITCH
601		BSC	L	ATERR,Z	PR IF SIGN NOT ALLOWABLE
602		LD		1 0	NEXT STMT WORD
603		S	L	APSGN	PLUS SIGN CONSTANT
604		BSC	L	ATH4, -	PR IF PLUS SIGN FOUND
605		S	L	AMSGN	MINUS SIGN CONSTANT
606		BSC	L	ATERR,Z	PR IF NOT MINUS SIGN
607	*				
608	*			CALL PUT	U- OPERATOR
609	*				
610		LD	L	UMNS	U- OPERATOR
611		BSI	L	PUT	PR TO PUT ON OUTPUT STRING
612	ATH4	MDX		1 1	INCR STMT POINTER
613	ATH11	STX		0 SW4	SET SWITCH 4 ON
614		MDX		ATH3	PR TO CHK NEXT STMT WORD
615	ATH1	STO	L	SNAME	SAVE NAME IN SNAME
616		BSI	L	NAME	PR TO CHK NAME DIMENSIONIN
617		LD	L	SW1	DIMENSIONED NAME SUPSCR SW
618		BSC	L	ATH5,Z	PR IF NAME NOT SUBSCRIBED
619		LD	L	SW7	DIMENSIONED NAME SWITCH
620		BSC	L	ATH6, -	PR IF NAME NOT DIMENSIONED
621	*				
622	*			CHECK FOR	CONSTANT
623	*				
624		LD		SVID	SYMBOL TABLE ID WORD
625		BSC	L	ATH6, Z	PR TO CHECK FOR SEMICOLON
626		LD		1 0	NEXT STMT WORD
627		S		ARPEN	LEFT PARENTHESIS CONSTANT
628		BSC	L	ATH7, -	PR IF LEFT PARENTHESIS
629		LD		SVID	LOAD ID
630		AND		S0190	CHK IF ARITH STMT
631		BSC	L	ATH6, -	PR IF ARITHMETIC STMT
632		EOR		H0090	
633		BSC	L	ATH6, -	RETURN IF EXTERNAL
634		LD	L	SORF	TEST FOR SUBR OF FUNCTION
635		BSC	L	XY7,	BRANCH IF SUBROUTINE
636		LD	L	FNAME	LOAD PROG NAME SYM TBL PT
637		S		SNAME	COMPARE WITH FUNCTION PT
638		BSC	L	XY7,Z	PR TO ERR IF FNAME NE SNAM
639	ATH6	LD		1 0	NEXT STMT WORD
640		BSC	L	ATH8, -	PR IF SEMICOLON
641		S		APSGN	PLUS SIGN CONSTANT
642		BSC	L	ATH9, -	PR IF PLUS SIGN
643		S		AMSGN	MINUS SIGN CONSTANT
644		BSC	L	ATH9, -	PR IF MINUS SIGN
645		S		ASLF	SLASH CONSTANT
646		BSC	L	ATH9, -	PR IF SLASH
647		S		ASTER	ASTERISK CONSTANT
648		BSC	L	ATH10,Z	PR IF NOT ASTERISK
649		LD		1 1	NEXT STMT WORD
650		S		ASK	ASTERISK CONSTANT
651		BSC	L	ATH9,7	PR IF NOT ASTERISK
652	*				
653	*			CALL PUT	** OPERATOR
654	*				
655		MDX		1 2	MOVE POINTER BY 2
656		LD		AEXP	** OPERATOR
657		BSI	L	PUT	PR TO PUT ON OUTPUT STRING
658		MDX		ATH11	PR TO TAG SWITCH 4
659	ATH9	BSI	L	MOVE1	PR TO ONE WD TO OUTPUT

```

660          MDX          ATH11      PR TO TAG SWITCH 4
661  ATH8    LD          PAR          PARENTHESIS COUNT
662          BSC    L    ATERR,Z      PR IF NOT ZERO TO SFT ERRC
663          BSC    I    APITH        PR TO EXIT FROM APITH SUBR
664  *
665  *          CONSTANTS AND WORK AREA
666  *
667  PAR      DC          0           PARENTHESIS COUNT
668  SW4      DC          0           SWITCH 4 - SIGN ALLOWABLE
669  SW5      DC          0           POINTER INSIDE FUNC CALL S
670  ARPEN    DC          /10        LEFT PARENTHESIS CONSTANT
671  CALLN    DC          0           FUNCTION CALL INDICATOR
672  PARNC    DC          0           PARAMETER COUNT
673  ASFN     DC          0           ARITH STMT FUNC INDICATOR
674  S0190    DC          /0190      ARITH STMT MASK
675  H0090    DC          /0090      EXTERNAL STMT MASK
676  *
677  *          CALL CHECK C OPERATOR
678  *
679  ATH7     LD          ACALL        C OPERATOR
680          BSI    L    CHECK        CHECK FOR OVERLAP
681          STX    0    SWE          SET SWITCH 5 ON
682          MDX    L    CALLN,1      CALLN CALLN 1
683  *
684  *          CHK FOR LEGAL SUBPROGRAM NAME
685  *
686          LD          SNAME        IS IT A REFERENCE      *V1M3
687          S          L    FNAME     *TO THIS PROGRAM      *V1M3
688          BSC    L    ATERR,+ -    YES - ERROR 36        *V1M3
689          LD          SNAME        NAME
690          BSI    L    GETID        GET SYMBOL TABLE ID WORD
691          SLA          8           SUBPROGRAM NAME BIT TO SIG
692          BSC    L    ATH19,C      BR IF STMT FUNCTION NAME
693          BSC    L    ATH17, Z     PR IF SUPERPROGRAM NAME
694          LD          3 0          SYMBOL TABLE ID WORD
695          AND          SBEZF        LEGAL SUBPROGRAM NAME MASK
696          BSC    L    ATERR,Z      BR TO SET UP ERROR
697  ATH15    LD          3 0          SYMBOL TABLE ID WORD
698          OR          S0080        SUBPROGRAM INDICATOR BIT
699          STO          3 0          RESTORE SYMBOL TBL ID WORD
700  *
701  *          XR2 2 LESS THAN XR1
702  ATH19    LDX        I3    IDSV2   IS THIS THE NAME      *V1M3
703          MDX        3 1          *OF THE STATEMENT      *V1M3
704          STX        3 **+1       *JUST BEING          *V1M3
705          LD          L    *- *    *LOOKED              *V1M3
706          S          SNAME        *AT                  *V1M3
707          BSC    L    ATERR,+ -    YES-ERROR 42        *V1M3
708  ATH17    STX        2    ATEMP    INDEX REG 2 TEMPORARY
709          LD          ATEMP        XR2 CONTENTS-OUTPUT STRING
710          A          ATWC          INCR BY 2
711          STX        1    ATEMP    INDEX REG 1 TEMPORARY
712          S          ATEMP        ADDR OF INPUT STRING
713          BSC    L    CK1,-       BR IF OVERLAP ERROR
714  *
715  *          OPEN STRING 2 WORDS
716  *
717          LD          ATH14 1      NEXT STMT ID WORD ADDRESS
718          S          ATEMP        CURRENT STMT ADDRESS
719          STO          * 1         ADDR NEXT INSTRUCTION

```

```

720          LDX  L3 *--*      RANGE - NO. WDS TO MOVE
721 *
722          MDX  1 -2         *MODIFIABLE
723          STX  1 ATH1R      DECR CURRENT STMT ADDRESS
724 ATH1A LD    1 2          NEW CURRENT STMT ADDR
725          STO  1 0          NXT WD TO MOVE
726          MDX  1 1          OPEN STRING 2 WORDS
727          MDX  3 -1         INCR STMT MOVE POINTER
728          MDX          ATH1A  DECR NO. OF WORDS TO MOVE
729          LDX  L1 *--*      PR TO CONTINUE MOVE
730 ATH1B EQU  *--1         SET UP NEW STMT POINTER
731 *
732 *
733 *          PUT PAR,PARNO AND ASFN IN THE
734 *          NESTED CALL TABLE
735          LDX  I3 ATH14 1    ADDR OF NEXT STMT ID WORD
736          MDX  3 -2         DECR BY 2
737          STX  3 ATH14 1    ADDR NESTED CALL TABLE
738          LD    ASFN        ARITH STMT FUNC NAME
739          STO  3 0          CALL TABLE
740          LD    PAR         PARENTHESIS COUNT
741          SLA          8
742          OR    PARNO
743          STO  3 1
744          LD    SNAME
745          STO  ASFN
746          LDX  3 1
747          STX  3 PAR
748          STX  3 PARNO
749          BSC  L  ATH12     MOVE ANOTHER WORD
750 *
751 *          CONSTANTS AND WORK AREA
752 *
753 AFSGN DC    /04
754 AMSGN DC    /06-/04    --
755 UMNS  DC    /20        U- OPERATOR
756 SNAME DC    0          LAST NAME
757 ASLH  DC    /08-/06    /--
758 ASTER DC    /00-/08    *-/
759 ASK   DC    /00        *
760 AEXP  DC    /0A        ** OPERATOR
761 ACALL DC    /2E        C OPERATOR
762 ATEMP DC    0          INDEX REGISTERS TEMPORARY
763 ACOMA DC    /12        ,
764 ARPC  DC    /02-/12    -,
765 SBE7F DC    /BA6E     LEGAL SUBPROGRAM ID MASK
766 ATWO  DC    2          DECIMAL TWO CONSTANT
767 S0080 DC    /0080     SUBPROGRAM NAME INDR BIT
768 *
769 *          TEST SWITCH 5
770 *
771 ATH5  LD    SW5
772          BSC  L  ATEPR, -  ERROR IF NOT SET
773 *
774 *          CHK FOR COMMA  1
775 *
776 ATH10 LD    1 0
777          S    ACOMA
778          BSC  L  ATH13,Z  GO TO CHK FOR RT PAREN
779 *

```

```

780 *          TEST SWITCH 5
781 *
782          LD          SW5          PR NO POINTER INSIDE
783          BSC L      ATH8, -      FUNCTION CALL.
784          MDX L      PARNO,1      INCR PARAMETER COUNT BY 1
785          BSC L      ATH12         PR TO MOVE 1 ELEMENT
786 *
787 *          CHECK FOR RIGHT PARENTHESIS 1
788 *
789 ATH13 S          ARPO          RIGHT PARENTHESIS CONSTANT
790          BSC L      ATEMP,Z      PR IF NOT RIGHT PARENTHESI
791          LD          PAR          PARENTHESIS COUNT
792          BSC L      ATH8, -      BR IF ZERO
793          MDX L      PAR,-1       DECR PAR BY 1 IF NOT ZERO
794          MDX          ATH16       BR IF PAR IS NOW ZERO
795          LD          SW5          PR IF NO POINTER INSIDE
796          BSC L      ATH16, -     FUNCTION CALL.
797          BSI          CKASF       PR TO CHK ARITH STMT FUNC
798          MDX L      CALLN,-1     CALLN CALLN -1
799          MDX          ATH14       BR IF CALLN IS NOW ZERO
800          SLA          16         CLEAR ACCUMULATOR
801          STO          SW5        NORMALIZE SWITCH 5
802 *
803 *          GET PAR, PARNO, AND ASFN FROM THE
804 *          NESTED CALL TABLE.
805 *
806 ATH14 LDX L3 *-*          ADDR OF NESTED CALL TABLE
807 *          *MODIFIABLE
808          LD          3 0          FIRST TABLE ELEMENT
809          STO          ASFN        ARITH STMT FUNC NAME
810          LD          3 1          SECOND TABLE ELEMENT
811          SRT          8          RIGHT JUSTIFY PAR
812          STO L      PAR          PARENTHESIS COUNT
813          SLA          16         CLEAR ACCUMULATOR
814          SLT          8          SHIFT IN 8 FROM EXTENSION
815          STO L      PARNO        PARAMETER COUNT
816 *
817 *          CLOSE STRING 2 WORDS
818 *
819          STX          1 ATEMP     INDEX REG 1 TEMPORARY
820          LD          ATH14 1      ADDR OF NESTED CALL TABLE
821          S          ATEMP        CURRENT STMT POINTER
822          STO          * 1         ADDR OF NEXT INSTRUCTION
823          LDX L3 *-*          RANGE OF MOVE TO CLOSE
824 *          *MODIFIABLE
825          MDX          1 2         INCR CURRENT POINTER BY 2
826          MDX L      ATH14 1,2    INCR ADDR NESTED CALL TABL
827          STX          1 ATH18 1   NEW STMT POINTER
828          LDX I1 ATH14 1         XR1 EQUALS MOVE ADDRESS
829 ATH10 LD          1 -3          MOVE TO CLOSE
830          STO          1 -1        STRING TWO WORDS
831          MDX          1 -1        DECR MOVE ADDR
832          MDX          3 -1        DECR MOVE COUNTER
833          MDX          ATH10       PR TO CONTINUE MOVE
834 ATH18 LDX L1 *-*          NEW STMT PT. MODIFIABLE
835 ATH16 BSI L      MOVE1          PR TO MOVE 1 ELEMENT
836          BSC L      ATH6        PR TO CHK FOR SEMICOLON
837 *
838 *****
839 *          THIS ROUTINE CHECKS THE ASF TABLE TO

```

```

840 *           INSURE THAT CALLS TO ASF STATEMENTS
841 *           HAVE THE CORRECT NO OF PARAMETERS.
842 *           THE ENTRY IS A RSI INSTRUCTION TO
843 *           THE LABEL CKASF. THE ROUTINE RETURNS
844 *           ONLY IF THE MATCH IS CORRECT.
845 *****
846 *
847 CKASF JC      0           SUBR ENTRY POINT
848 CKAS  LDX    L3 1       INITIALIZE INDEX REG 3
849 *           *MODIFIABLE
850 CKS1  MDX    3 -2       SKIP NEXT IF XR3 NEG OR 0
851      MDX    * 2         BRANCH IF XR3 POSITIVE
852 CKS1A BSC    I  CKASF   EXIT FROM CKASF
853 CKS2  LD     L3 *-*     PRESTORED ASF TABLE ADDR
854 *           *MODIFIABLE
855      S      L  ASFN     COMPARE NAMES
856      BSC    L  CKS1,7   BR IF NOT SAME NAME
857 CKS3  LD     L3 *-*     PRESTORED ASF TABLE ADDR
858 *           *MODIFIABLE
859      S      L  PARNC    COMPARE PARAMETER COUNTS
860      BSC    L  ATERR,Z  BR TO ERROR IF NOT EQUAL
861      MDX    L  CKS1A   PR TO EXIT IF MATCH
862 *
863 *           CHECK FOR VALID VARIABLE IN
864 *           ARITHMETIC EXPRESSION
865 *
866 TESTV DC      0           ENTRY POINT
867      LD     L  SORF     SUBPROGRAM OR FUNC INDR
868      BSC    L  TESTW,  BR IF NOT FUNCTION
869      LD     L  1 0      NEXT STMT WORD
870      S      L  FNAME    FUNCTION NAME
871      BSC    I  TESTV, - EXIT IF NAME COMPARE
872      LD     L  SA7DE    LEGAL FUNCTION ID MASK
873      MDX    L  TESTX   BR TO MASK SYMBOL TBL ID W
874 TESTW LD     L  SA3DE   LEGAL SUBROUTINE ID MASK
875 TESTX AND    3 0       MASK SYMBOL TABLE ID WORD
876      BSC    L  XY7,7   BR TO ERROR IF ANY BITS ON
877      BSC    I  TESTV   EXIT IF LEGAL SUBR OR FUNC
878 *
879 SA3DE DC     /83DE     SUBROUTINE ID MASK
880 SA7DE DC     /A7DE     FUNCTION ID MASK
881 *
882 *           CHECK CALL STMT SYNTAX
883 *
884 XY25  S      L  CCALL   CALL ID CONSTANT
885      BSC    L  XY28,7  BR IF NOT CALL
886      LDX    3 39       SET XR3 39
887      STX    L3 ERNC    SET UP ERROR NUMBER 39
888      LD     L  1 0     IS THIS A CALL *V1M3
889      S      L  FNAME   *TO ITSELF *V1M3
890      BSC    L  ATERR,+ YES-ERROR 39 *V1M3
891      LD     L  1 0     NEXT STMT WORD
892      BSI    L  GETID   GET SYMBOL
893      AND    L  SUB1X   LEGAL SUBPROGRAM MASK
894      BSC    L  ATERR,Z PR TO ERROR IF NOT LEGAL
895      LD     L  3 0     SYMBOL TABLE ID WORD
896      OR     L  SUBFC   SET SUBPROG INDR BIT ON
897      STO    3 0       RESTORE SYMBOL TABLE ID WD
898      BSI    L  NAME    CALL NAME
899      LD     L  SW7     DIMENSIONED NAME SWITCH

```

```

900      BSC  L  ATERR, -  PR IF NOT DIMENSIONED NAME
901      LD   ROPR      C OPERATOR
902      BSI  L  CHECK   CHECK FOR OVERLAP
903      *
904      *           CH FOR LEFT PARENTHESIS 1
905      *
906      LD   1 0       NEXT STMT WORD
907      S    LPAR      LEFT PARENTHESIS CONSTANT
908      BSC  L  XY29,7  PR IF NOT LEFT PARENTHESIS
909      XY30 BSI  L  MOVF1  PR TO MOVE ONE ELEMENT
910      *
911      *           CHECK FOR SUBSCRIPTED ELEMENT
912      *
913      STX  1 XY30B 1  SAVE XR1
914      LD   1 1       NEXT STMT WORD
915      S    LX1       SUBSC CHARACTER LOW LIMIT
916      BSC  L  XY30A, 7 PR IF NOT SUBSCRIPT CHAR
917      S    LX2       SUBSC CHARACTER HIGH LIMIT
918      BSC  L  XY30A,-Z PR IF NOT SUBSCRIPT CHAR
919      A    LX3       RESTORE SUBSCRIPT VALUE
920      SLA  1        MPY RY 2
921      STO  * 1       ADDR NEXT INSTRUCTION
922      MDX  L1 *-*    INCR TO SKIP OVER SUBSC
923      *           *MODIFIABLE
924      MDX  1 1       ADJUST BY ONE MORE
925      *
926      *           CHK FOR COMMA OR RT PARENTHESIS
927      *
928      XY30A LD   1 1  NEXT CHAR BEYOND SUBSCRIPT
929      XY30B LDX  L1 *-* RESTORE STMT POINTER
930      *           *MODIFIABLE
931      S    LX4       COMMA CHARACTER CONSTANT
932      BSC  L  XY30C, - PR IF COMMA
933      S    LX5       RIGHT PARENTHESIS CONSTANT
934      BSC  L  XY30C,7 PR IF NOT RIGHT PARENTHESI
935      XY30C LDX  3 39 INDEX REG 3 39
936      STX  L3 ERNO   SET UP ERROR NUMBER 39
937      *
938      *           CHECK FOR LEGAL VARIABLE
939      *
940      LD   1 0       NEXT STMT WORD
941      BSI  L  GETID   GET SYMBOL TABLE ID WORD
942      AND  LX6       LEGAL VARIABLE ID MASK
943      BSC  L  XY30D,7 PR IF NOT LEGAL VARIABLE
944      *
945      *           CHECK FOR SUBPROGRAM NAME
946      *
947      LD   3 0       SYMBOL TABLE ID WORD
948      SLA  8        SUBPROGRAM NAME BIT TO SIG
949      BSC  L  XY30E, 7 PR IF SUBPROGRAM NAME
950      LD   3 0       SYMBOL TABLE ID WORD
951      OR   LX7       REFERENCED INDICATOR BIT 0
952      STO  3 0       RESTORE SYMBOL TABLE ID WD
953      *
954      *           CALL NAME
955      *
956      XY30F BSI  L  NAME  PR TO CHK NAME DIMENSIONIN
957      MDX  XY30E    PR TO CHK FOR COMMA
958      *
959      *           SET UP ERROR NUMBER 40

```



```

960 *
961 XY30D LDX 3 40 INDEX REG 3 40
962 STX L3 ERNO SET UP ERROR NUMBER 40
963 *
964 * CALL ARITH EXPRESSION SCAN
965 *
966 BSI L ARITH BR TO APITH SCAN SUBROUTIN
967 *
968 * CHECK FOR COMMA 1
969 *
970 XY30E LD 1 0 NEXT STMT WORD
971 S CMAS COMMA CHAR CONSTANT
972 BSC L XY30, - BR IF COMMA FOUND
973 S RPAR RIGHT PARENTHESIS 1
974 BSC L XY7,7 BR IF NOT RT PARENTHESIS
975 BSI L MOVE1 BR TO MOVE ONE ELEMENT
976 XY31 BSC L XY13 BR TO CHECK FOR SEMICOLON
977 XY29 LD 1 0 NEXT STMT WORD
978 BSC L XY7,2 BR TO ERR IF NOT SEMICOLON
979 LD RFAC FLOATING ACC INDICATOR
980 BSI L CHECK OUTPUT AND CHK FOR OVERLAP
981 MDX XY31 BR TO CHK FOR SEMICOLON
982 *
983 XY30G SLA 3 TEST IF EXTERNAL
984 BSC Z+ SKIP IF NOT *V1M3
985 MDX XY30F CONTINUE *V1M3
986 LD L SORF IS THIS A FUNCTION *V1M3
987 BSC L ATERR,+ NO - ERROR 39 *V1M3
988 LD 1 0 PARAMETER IN CALL *V1M3
989 S L FNAME FUNCTION NAME *V1M3
990 BSC L ATERR,7 ERROR IF NOT SAME *V1M3
991 LD 3 0 SET DEFINED *V1M3
992 OR LX7 *VARIABLE BIT *V1M3
993 STO 3 0 *IN ID WORD *V1M3
994 MDX XY30F CONTINUE
995 *
996 * CONSTANTS AND WORK AREA
997 *
998 LX1 DC /18 SUBSC CHAR LOW LIMIT CON
999 LX2 DC /1E-/18 SUBSC CHAR HIGH LIMIT CON
1000 LX3 DC 7 RESTORE SUBSC VALUE CON
1001 LX4 DC /12 COMMA CHAR CONSTANT
1002 LX5 DC /02-/12 RIGHT PARENTHESIS CONSTANT
1003 LX6 DC /834E LEGAL VARIABLE MASK
1004 LX7 DC /0020 REFERENCED STMT NO. INDR
1005 LX8 DC /8FFE LEGAL ARITH STMT FUNC MAS
1006 LX9 DC /0100 ARITH STMT FUNC INDR BIT
1007 CCALL DC /06-/0F CALL ID
1008 LPAR DC /10 LEFT PARENTHESIS CONSTANT
1009 CMAS DC /12 COMMA CHAR CONSTANT
1010 RCOP DC /2E CALL OPERATOR
1011 RPAR DC /02-/12 RIGHT PARENTHESIS CONSTANT
1012 RFAC DC /8000 FLOATING ACC INDR CONSTANT
1013 ASFID DC /1A-/06 ASF ID
1014 RASF DC /20 ASF OPERATOR
1015 SUBPO DC /0080 SUBPROGRAM INDR BIT
1016 SUB1X DC /8B6E LEGAL SUBPROGRAM MASK
1017 *
1018 * CHK FOR APITH STMT FUNCTION
1019 *

```

1020	XY28	S		ASFID	ARITH STMNT FUNC ID
1021		BSC	L	XY26,Z	PR IF NOT ARITH STMNT FUNC
1022		LDX	3	41	INDEX REG 3 41
1023		STX	L3	ERNO	SET UP ERROR NUMBER 43
1024	*				
1025	*				CHK FOR VALID ARITH STMNT FUNC NAME
1026	*				
1027		LD	1	0	NEXT STMNT WORD
1028		BSI	L	GETID	GET SYMBOL TABLE ID WORD
1029		AND		LX8	VALID ASF NAME MASK
1030		BSC	L	XY7,Z	BR IF NOT VALID TO SET ERR
1031		LD	3	0	SYMBOL TABLE ID WORD
1032		OR		LX9	SET STMNT FUNC NAME BIT ON
1033		STO	3	0	RESTORE SYMBOL TABLE ID WD
1034		BSI	L	MOVE1	PR TO MOVE ON ELEMENT
1035	*				
1036	*				OPEN STRING 2 WDS IF REQUIRED
1037	*				
1038		MDX	2	2	INCR OUTPUT STRING ADDRESS
1039		STX	2	XQ1	INDEX REG 2 TO TEMPORARY
1040		LD		XQ1	OUTPUT STRING ADDRESS
1041		STX	1	XQ2	INDEX REG 1 TO TEMPORARY
1042		S		XQ2	INPUT STRING ADDRESS
1043		BSC	L	CK1,-	BR IF OVERLAP ERROR
1044		LD		XQ1	CURRENT OUTPUT STRING PT
1045		S	L	SOFS	START OF STRING ADDRESS
1046		STO		* 1	ADDR NEXT INSTRUCTION
1047		LDX	L3	*-*	RANGE OF MOVE TO OPEN
1048	*				*MODIFIABLE
1049		MDX	3	-2	NO. OF WORDS TO MOVE
1050		LD	2	-3	NEXT WORD TO MOVE TO OPEN
1051		STO	2	-1	OPEN STRING 2 WORDS
1052		MDX	2	-1	DECR MOVE ADDRESS
1053		MDX	3	-1	DECR MOVE WORD COUNT
1054		MDX		*-5	PR TO CONTINUE MOVE
1055		LDX	I2	SOFS	NEW START OF STPING ADDRESS
1056		MDX	L	IDSV2,2	INCR STMNT WORD COUNT BY 2
1057	*				
1058	*				PUT NAME AND ZERO IN ARITH STMNT
1059	*				FUNCTION TABLE.
1060	*				
1061		LD	1	-1	ASF NAME
1062		STO	2	0	ASF FUNCTION TABLE
1063		SLA		16	CLEAR ACCUMULATOR
1064		STO	2	1	STORE ZERO IN ASF TABLE
1065		MDX	L	CKAS 1,2	INCR ASF TABLE PCINTER BY
1066		LDX	L2	*-*	RESTORE OUTPUT STRING PT
1067	XQ1	EQU		*-1	ADDR FOR LAST INSTRUCTION
1068	*				*MODIFIABLE
1069		LDX	L1	*-*	RESTORE INPUT STRING PT
1070	XQ2	EQU		*-1	ADDR FOR LAST INSTRUCTION
1071	*				*MODIFIABLE
1072	*				
1073	*				CALL CHECK ASF OPERATOR
1074		LD		RASF	ASF OPERATOR
1075		BSI	L	CHECK	OUTPUT AND CHK FOR OVERLAP
1076		LD	1	0	NEXT STMNT WORD
1077		S		LPAR	LEFT PARENTHESIS CONSTANT
1078		BSC	L	XY7,7	PR IF NOT LEFT PARENTHESIS
1079	XY32	BSI	L	MOVE1	PR TO OUPUT ONE ELEMENT

```

1080 *
1081 *           CHK FOR VALID DUMMY ARGUMENT
1082 *
1083         LD      1 0           NEXT STMT WORD
1084         BSI    L  GETID      PR TO GET SYMBOL TABLE ID
1085         BSC    L  XY7, Z     PR TO ERROR IF MINUS
1086         BSI    L  MOVE1      PR TO OUTPUT ONE ELEMENT
1087 XY33 MDX    L  *-*, 1      COUNT ASE ARGUMENTS
1088 *           *MODIFIABLE
1089         LD      1 0           NEXT STMT WORD
1090         S          CMAS      COMMA CHAR CONSTANT
1091         BSC    L  XY32, -    PR IF COMMA FOUND
1092         S          RPAR      RIGHT PARENTHESIS CONSTANT
1093         BSC    L  XY7, 7     PR TO ERR IF NOT RT PAREN
1094         BSI    L  MOVE1      PR TO OUTPUT ONE ELEMENT
1095         LD      1 0           NEXT STMT WORD
1096         S          L  EQSN     EQUAL SIGN CHAR CONSTANT
1097         BSC    L  XY7, 2     PR TO ERR IF NO EQUAL SIGN
1098         LDX    3 42         INDEX REG 3 42
1099         STX   L3 ERNO      SET UP ERROR NUMBER 42
1100         BSC    L  XY5A      PR TO CHK FOR EQUAL SIGN
1101 *
1102 *           CONSTANTS FOR DISK READ/WRITE SCAN
1103 *
1104 REDID DC      /12-/1A      READ ID
1105 WRTID DC      /11-/12      WRITE ID
1106 FNDID DC      /10-/11      FIND ID
1107 APOST DC      /42-/02      APOSTROPHE
1108 RPARN DC      /02         RIGHT PARENTHESIS
1109 CCLON DC      /32         COLON OPERATOR
1110 *
1111 *           SEEK READ/WRITE OPERATOR
1112 *
1113 XY26 S        REDID      READ ID CONSTANT
1114         BSC    Z          SKIP IF READ
1115         S        WRTID     WRITE ID CONSTANT
1116         BSC    Z          SKIP IF FIND
1117         S        FNDID     FIND ID CONSTANT
1118         BSC    L  XY27, 7  BRANCH UNLESS READ OR WRIT
1119 *
1120 *           TEST FOR SIOK READ/WRITE
1121 *
1122         LDX    3 73         ERROR NUMBER
1123         STX   L3 ERNO      SET ERROR NO JUST IN CASE
1124 MOVCH BSI    L  MOVE1      MOVE ONE CHAR
1125         LD      1 -1        GET CHARACTER JUST MOVED
1126 *
1127 *           CHECK FOR APOSTROPHE
1128 *
1129         BSC    L  XY22, -    BRANCH IF SEMICOLON
1130         S          RPARN     TEST FOR RIGHT PAREN
1131         BSC    L  MVRST, -   BRANCH IF NOT DISK P/W
1132         S          APOST     APOSTROPHE CHAR CONSTANT
1133         BSC    Z          SKIP ON APOSTROPHE
1134         MDX    L  MOVCH     PR TO MOVE ONE CHARACTER
1135 *
1136 *           GO SCAN FOR SECTOR ADDR
1137 *
1138         BSI    L  ARITH     GO TO SCAN ARITH EXPRESSIO
1139         LD      1 0           REQUIRE RIGHT PAREN

```

```

1140          S          RPAPN      AS TERMINATOR
1141          BSC  L      XY7,2      BRANCH TO ERROR IF NOT
1142          *
1143          *          PUT COLON OVER THE RIGHT PARENTHESIS
1144          *
1145          LD          COLON      COLON OPERATOR
1146          STO  1  0          PUT OVER RIGHT PARENTHESIS
1147          *
1148          *          MOVE REST OF STMT
1149          *
1150          MVRST BSI  L  MOVE1      MOVE REST OF STATEMENT
1151          LD  1  -1          STOP WHEN SEMICOLON
1152          BSC  L  MVRST,7        FOUND.
1153          BSC  L  CRCTN        GO CORRECT NORM
1154          *
1155          *          MOVE STATEMENT
1156          *
1157          XY27  LDX  L3  *-*      RANGE OF STMT MOVE
1158          *          *MODIFIABLE
1159          MDX  3  -1          NO. OF WORDS TO MOVE
1160          MDX  *  1          RR IF RANGE NOT ZERO
1161          MDX          XY27A      RR TO GO NEXT STMT
1162          BSI  L  MOVE1        GO TO MOVE ONE ELEMENT
1163          MDX  3  -1          DECR MOVE COUNTER
1164          MDX          XY27 3     RR TO MOVE NEXT WORD
1165          XY27A BSC  L  XY22      GO GET NEXT ID
1166          *
1167          *          END OF ASCAN PHASE
1168          BSS          OVERL-**+320*3  PHASE-12 PATCH AREA
1169          END          START

```