

```

1 * GENERAL AUTOMATION, INC. ALL RIGHTS RESERVED
2 *****
3 *
4 * PROGRAM NAME FORTRAN PH-04
5 *
6 * MODEL NUMBER 8F004
7 *
8 * PURPOSE FORTRAN PHASE=04
9 *
10 * PROGRAMMER DICK WALLMANN, MODS-MARK ELFIELD
11 *
12 ***** REVISION LIST *****
13 *
14 * RV DATE SCO BY REASON FOR CHANGE
15 * -----
16 *
17 * 01 11/16/70 RPH INITIAL RELEASE
18 *
19 *****
20 *****
21 * GA 18/30 FORTRAN COMPILER 05/01/70
22 *STATUS - VERSION 1, MODIFICATION 0
23 *FUNCTION/OPERATION-
24 * * PLACES COMMON STMT VARIABLE INTO THE SYMBOL
25 * TABLE, INCLUDES DIMENSION INFORMATION, IF
26 * PRESENT.
27 * * REMOVES COMMON STMTS FROM THE STRING
28 * * CHECKS FOR A SUBROUTINE OR FUNCTION STMT,
29 * * PLACES THE NAME AND DUMMY ARGUMENTS OF THE
30 * SUBROUTINE OR FUNCTION STATEMENT INTO THE
31 * SYMBOL TABLE, DELETES THE STMT FROM THE
32 * STMT STRING.
33 * * CHECKS REAL AND INTEGER STATEMENTS FOR THE
34 * WORD FUNCTION
35 * * PHASE 4 IS A TWO PASS PHASE, THE FIRST PASS
36 * PROCESSES COMMON STATEMENTS, THE SECOND
37 * PASS PROCESSES A SUBROUTINE OR FUNCTION
38 * STATEMENTS, INCLUDING A FUNCTION DESIGNATED
39 * IN A REAL OR INTEGER STATEMENT.
40 *ENTRY POINT
41 * * TOP IS THE LABEL ON THE ENTRY POINT ADDR,
42 * PHASE 4 IS SEQUENTIALLY LOADED FROM PHASE 3
43 * VIA A CALL TO ROLRX.
44 *INPUT-
45 * NONE IN THE USUAL SENSE, HOWEVER, THE STMT
46 * STRING SCANNED BY THE COMPILER MAY BE
47 * CONSIDERED INPUT.
48 *OUTPUT-
49 * NONE IN THE USUAL SENSE, HOWEVER, PHASE 4
50 * REMOVES STMTS FROM THE STMT STRING AND ADDS
51 * ENTRIES TO THE SYMBOL TABLE.
52 *EXTERNAL ROUTINES-N/A
53 *EXITS
54 * * NORMAL - THE NORMAL EXIT IS TO PHASE 5 VIA
55 * ROLRX IF NO OVERLAP ERROR HAS
56 * OCCURRED.
57 * * ERRORS - THE NORMAL FLOW OF THE COMPILER IS
58 * NOT INTERRUPTED BY SYNTAX ERRORS,
59 * AN OVERLAP ERROR TERMINATES THE

```

```

60 *          COMPILATION, CONTROL IS PASSED
61 *          FROM PHASE TO PHASE UNTIL PHASE 21
62 *          IS LOADED, AT THAT TIME A MESSAGE
63 *          STATING THE OVERLAP IS PRINTED.
64 *TABLES/WORK AREAS-
65 *   NON EXCEPT STATEMENT STRING AREA, SYMBOL
66 *   TABLE, AND FORTRAN COMMUNICATION AREA
67 *ATTRIBUTES-N/A
68 *   * THE COMPILATION AND SYNTAX ERRORS NUMBERING
69 *     7, 8, 10, 11, 12, 13, 14, 15, 74 ARE
70 *     DETECTED BY PHASE 4.
71 *NOTES-
72 *   THE SWITCHES USED IN PHASE 4 FOLLOW, IF NON-
73 *   ZERO, THE SWITCH IS TRANSFER=T. IF ZERO, THE
74 *   SWITCH IS NORMAL=N.
75 *     * NORM2-SYNTAX ERROR IN COMMON STMT
76 *       T=SYNTAX ERROR PRESENT
77 *     * ASW1-RIGHT PARENTHESIS ENCOUNTERED
78 *       N=NOT ENCOUNTERED
79 *     * ASW2-REAL STATEMENT ENCOUNTERED
80 *       N=NOT ENCOUNTERED
81 *     * ASW3-INTEGER STATEMENT ENCOUNTERED
82 *       N=NOT ENCOUNTERED
83 *     * SW5A-FIO STATEMENT PRESENT
84 *       N=NOT PRESENT
85 *     ABS      REF CORE
86 *     SYSTEM AND FORTRAN EQUATES
87 MEMRY EQU      FFFF CORE MAXIMUM CORE SIZE
88 PHSIZ EQU      4*320    MAXIMUM PHASE SIZE
89 OVERL EQU      MEMRY-PHSIZ    PHASES 2-29 START
90 FCOM EQU      OVERL-22    FORTRAN COMM. TABLE
91 PHNTB EQU      FCOM-56    PHASE TABLE
92 ROLRX EQU      PHNTB-50    INTERPHASE CALL
93 *     FORTRAN COMMUNICATION AREA
94 *     FORTRAN COMMUNICATION AREA
95 *     ORG      FCOM
96 SOFS BSS      1          START OF STRING
97 EOFS BSS      1          END OF STRING
98 SOFST BSS     1          START OF SYMBOL TABLE
99 SOFNS BSS     1          START OF NON-STMT NUMBERS
100 SOFXT BSS    1          START OF SUBSCRIPTED TEMPS
101 SOFGT BSS    1          START OF GENERATED TEMPS.
102 EOFST BSS    1          END OF SYMBOL TABLE
103 COMON BSS    1          NEXT AVAILABLE COMMON
104 CSIZE BSS    1          SIZE OF COMMON
105 ERROR BSS    1          OVERLAP ERROR
106 FNAME BSS    1          PROGRAM NAME
107          BSS    1          2ND WORD OF NAME
108 SORF BSS     1          SUBR (=) OR FUNCT (+)
109 CCWD BSS     1          CONTROL CARD WORD
110 *          BIT 15 TRANSFER TRACE
111 *          BIT 14 ARITHMETIC TRACE
112 *          BIT 13 EXTENDED PRECISION
113 *          BIT 12 LIST SYMBOL TABLE
114 *          BIT 11 LIST SUBPROGRAM NAMES
115 *          BIT 10 LIST SOURCE PROGRAM
116 *          BIT 9  ONE WORD INTEGERS
117 IOCS BSS     1          IOCS CONTROL CARD WORD
118 *          BIT 15 CARD
119 *          BIT 14 PAPER TAPE

```

```

120 *          BIT 13 TYPEWRITER
121 *          BIT 12 1443 PRINTER
122 *          BIT 11 MAGNETIC TAPE
123 *          BIT 10 KEYBOARD
124 *          BIT 9  1442 PUNCH
125 *          BIT 8  DISK
126 *          BIT 7  NOT USED
127 *          BIT 6  NOT USED
128 *          BIT 5  NOT USED
129 *          BIT 4  NOT USED
130 *          BIT 3  PLOITER
131 *          BIT 2  NOT USED
132 *          BIT 1  UNFORMATTED DISK
133 *          BIT 0  UNFORMATTED TAPE
134 DFCNT BSS      1          DEFINE FILE COUNT
135 LCOMN BSS      2          INSKEL COMMON USE
136 ICCER BSS      2          IOCS CONTROL CARD ERROR
137          BSS      2          SYSTEM LOADER USE
138 *          END OF FORTRAN COMMUNICATION
139 *          AREA
140          ORG      OVERL
141 TOP     LDX      I1 SOFS      INITIALIZE STRING I/P PT
142 *          OVERLAP ERROR ON
143          LD       ERROR      TEST FOR OVERLAP ERROR
144          BSC      L  START,Z  GET PHS 5 IF OVLP ERROR
145 *          CHECKS FOR END AND COMMON STMT
146 *          UPDATES THE I/P STRING PT (XR1)
147 *          TO MOVE TO THE NEXT STMT
148 TST     LD       1 0          LOAD STMT ID WORD
149          SRA      1
150          AND      IDTPE      GET ID TYPE
151          S        ENDC       IS IT END
152          BSC      L  START,+  GET PHS 5 IF END CARD
153          S        COMMC      IS IT COMMON
154          BSC      +-         SKIP IF NOT
155          MDX      FIX        CHECK FOR TYPE OF COMMON
156 *          MOVE TO NEXT STMT
157 MV      LD       1          LOAD STMT ID WORD
158          SRA      2          SHIFT NORM TO RIGHT
159          AND      IDNRM      ISOLATE NORM
160          STO      NXID+1
161 NXID    MDX      L1         MOVE POINTER TO NEXT STMT
162          MDX      TST        GO TEST STMT TYPE
163 FIX     SLA      16
164          STO      COMID      CLEAR COMID
165 *          INITIALIZE TO SCAN COMMON STMT
166          STX      1 COMSV    SAVE INPUT POINTER
167          MDX      1 1        MOVE POINTER
168          SLA      16
169          STO      NORM2      CLEAR NORM2
170 *          CHECKS FOR STMT TERMINATOR
171 *          CHECKS FOR SLASH
172 PTB    LD       1 0          IS CHARACTER
173          S        SEMIC      SEMICOLON
174          BSC      L  RMOVE,+  BRANCH IF YES
175          LD       NORM2      IS THERE SYNTAX ERROR IN
176          BSC      L  SETUP,E  COMMON STMT - BRANCH YES
177          LD       1 0          LOAD WORD
178          S        SLH        IS CHARACTER SLASH
179          BSC      L  QCOM,+  BRANCH IF YES

```

```

180 *                CHECKS THE NAME FOR VADIDTY
181 DD2  LD          1 0          LOAD WORD
182      BSC          -          SKIP IF NOT SPECIAL CHAR
183      MDX          ERAH        GO TO ERROR 7
184      STO          WD1        SAVE WORD
185      SLA          2
186      BSC          L ERAH,C+Z  BRANCH IF NUMERIC
187      LD           1 1          LOAD 2ND WORD
188      BSC          L MAD,+Z    BRANCH IF PART OF NAME
189      LD           ZERO        LOAD ZEROES
190      STO          WD2        STORE AS 2ND WORD IN NAME
191      MDX          ZORRO       CHECK FOR DUPLICATES
192 MAD   STO          WD2        STORE 2ND WORD
193      MDX          1 1          MOVE POINTER
194      LD           1 1          LOAD NEXT WORD
195      BSC          L ERAH,+Z    BRANCH NOT SPECIAL CHAR
196 *                SCANS THE SYM TBL FOR
197 *                DUPLICATION OF THE VAR NAME
198 ZORRO LDX        13 SOFNS     INITIALIZE SYMBOL TABLE P
199      MDX          JACK        START LOOP
200 LOOP  LD          3 1          DOES FIRST WORD IN NAME
201      S            WD1        MATCH TABLE WORD
202      BSC          Z          SKIP IF YES
203      MDX          EAT        GET NEXT TABLE ENTRY
204      LD           3 2          DOES SECOND WORD IN NAME
205      S            WD2        EQUAL TABLE 2ND WORD
206      BSC          L HELP,+--  BRANCH IF YES
207 EAT   MDX        3 -3         GET NEXT TABLE ENTRY
208 JACK  STX        3 SAVEQ      SAVE TABLE POINTER
209      LD           SAVEQ      HAS ENTIRE TABLE BEEN
210      S            EOFST     CHECKED FOR DUPLICATES
211      BSC          Z          SKIP IF YES
212      MDX          LOOP       CONTINUE LOOP
213      MDX          PLACE      PUT IN SYMBOL TABLE
214 *                CONSTANTS
215 COMMC DC          /1400      COMMON
216 ENDC  DC          /0800      END
217 IDTPE DC          /7C00      STMT ID TYPE MASK
218 IDNRM DC          /01FF     NORM MASK
219 SEMIC DC          /001E     SEMICOLON
220 NORM2 DC          0         SWITCH TWO
221 SW2   DC          1         SWITCH 2
222 COMSV DC          0         POINTER
223 WD1   DC          0         WORD 1 OF COMMON NAME
224 WD2   DC          0         WORD 2 OF COMMON NAME
225 SAVEQ DC          0         SYMBOL TABLE POINTER
226 ERRNO DC          0         ERROR NUMBER
227 ERR10 DC         10        ERROR NO, 10
228 ERR11 DC         11        ERROR NO, 11
229 ERR7  DC          7         ERROR NO, 7
230 ERR8  DC          8         ERROR NO, 8
231 ERRID DC          /A008     ERROR STMT ID WORD
232 GET   DC          0         ADDRESS OF NEXT STMT
233 SVV   DC          0         STRING POINTER
234 ZERO  DC          /8000     USEFUL CONSTANT
235 SLH   DC          /21       SLASH
236 COMID DC          0         TYPE OF COMMON
237 *                SETUP ERRORS
238 HELP  LD          ERR11      SET ERROR NO, 11
239      STO          ERRNO      GO TO CLOSE

```

240		MDX		MOVE		UP STRING
241	SETUP	LD		ERR10		SET ERROR NO, 10
242		STO		ERRNO		GO TO CLOSE
243		MDX		MOVE		UP STRING
244	GEEP	LD		ERR8		SET ERROR NO, 8
245		STO		ERRNO		GO TO CLOSE
246		MDX		MOVE		UP STRING
247	ERAH	LD		ERR7		SET ERROR NO, 7
248		STO		ERRNO		
249	*					REPLACES AN ERRONEOUS STNNT WIT
250	*					AN ERROR MESSAGE - CLOSSES UP
251	*					THE STRING
252	MOVE	LDX	I1	COMSV		RESET STRING POINTER
253		LD	1	0		LOAD STNNT ID WORD
254		SRA		2		RIGHT-JUSTIFY NORM
255		AND		IDNRM		CLEAR ALL BUT NORM
256		STO		NRMSV		ADD NORM TO STRING POINTE
257		A		COMSV		TO GET ADDRESS OF NEXT
258		STO		GET		STATEMENT
259		LDX	I2	GET		INITIALIZE INPUT LOOP PT
260		LD		ERRID		LOAD ERROR STNNT ID WORD
261		STO	1	0		PUT ON STRING
262		LD		ERRNO		LOAD ERROR NUMBER
263		STO	1	1		PUT ON STRING
264		MDX	1	1		MOVE POINTER
265		LD	L	EOFS		COMPUTE RANGE OF LOOP
266		S		COMSV		SUBTRACT POINTER THEN
267		S		NRMSV		SUBTRACT NORM TO GET
268		STO		RANGE		RANGE OF LOOP
269		LDX	I3	RANGE		INITIALIZE RANGE COUNTER
270		STX	1	SVV		SAVE STRING POINTER
271		MDX	3	1		
272	LOOP1	LD	2	0		MOVE WORD DOWN
273		STO	1	1		IN STRING
274		MDX	1	1		MOVE POINTERS
275		MDX	2	1		
276		MDX	3	-1		DECREMENT RANGE COUNTER
277		MDX		LOOP1		CONTINUE LOOP
278		STX	L1	EOFS		NEW END OF STRING
279		LDX	I1	SVV		RESTORE INPUT POINTER
280		MDX	1	1		MOVE STRING POINTER
281		BSC	L	TST		CHECK NEXT STNNT
282	*					PLACES A VARIABLE NAME IN THE
283	*					SYM TBL - SETS THE COMMON AND
284	*					TYPE INDICATORS IN THE SYM TBL
285	*					ID WORD - CHECKS FOR A SYM TBL
286	*					OVERLAP
287	PLACE	LD		WD2		PUT 2ND WORD INTO
288		STO	3	2		SYMBOL TABLE
289		LD		WD1		PUT 1ST WORD INTO
290		STO	3	1		SYMBOL TABLE
291	*					SET TYPE AND COMMON INDICATORS
292		LD		WD1		GET FIRST LETTER
293		AND		PIT1		OF NAME
294		S		RGTI		IS IT I OR GREATER
295		BSC		+Z		SKIP IF YES
296		MDX		REEL		GO SET REAL ID WORD
297		S		RGTN		IS IT N OR LESS
298		BSC	L	REEL,Z-		BRANCH IF NOT
299		LD		IDCI		LOAD INIEGER INDICATOR

300		MDX		VIP	GO PUT IN SYMBOL TABLE ID
301	REEL	LD		IDCR	LOAD REAL SYM TBL ID
302	VIP	OR		COMID	PUT IN COMMON TYPE BIT
303		STO	3	0	PUT IN SYMBOL TABLE
304		MDX	L	EOFST,-3	MODIFY SYMBOL
305		MDX	L	SOFXT,-3	TABLE ADDRESSES
306		MDX	L	SOFGT,-3	
307	*				SYMBOL TABLE OVERLAP
308		LD	L	EOFST	LOAD END OF SYMBOL TABLE
309		A		CONST	ADD TWO
310		S	L	EOFS	SUBTRACT END OF STRING
311		BSC	L	JIMY,+Z	BRANCH IF NEGATIVE-OVERLA
312		MDX	1	1	MOVE POINTER
313		LD	1	0	LOAD NEXT WORD
314		S		PARNL	IS CHAR LEFT PARENTHESIS
315		BSC	L	PRTE,+Z	BRANCH IF YES
316	*				CHECKS FOR A COMMA OPTR
317	DD3	LD	1	0	LOAD WORD
318		S		COMAC	IS CHARACTER COMMA
319		BSC		+Z	SKIP IF NO
320		MDX		DD1	GO GET NEXT WORD
321		LD	1	0	LOAD WORD
322		S		SLH	IS CHARACTER SLASH
323		BSC	L	OCOM,+Z	BRANCH IF YES
324		LD		SW2	SET SYNTAX ERROR IN
325		STO		NORM2	COMMON STMT SWITCH
326		BSC	L	PTB	GO CHECK CHARACTER
327	*				UPDATES THE I/P STRING PT (XR1)
328	*				TO MOVE TO THE NEXT STRING WORD
329	DD1	MDX	1	1	MOVE POINTER
330		BSC	L	DD2	GET NEXT WORD
331	*				CONSTANTS
332	NRMSV	DC		0	NORM
333	RANGE	DC		0	RANGE OF LOOP
334	CONST	DC		2	TWO
335	PIT	DC		/6000	NUMERIC ZONE BIT MASK
336	PIT1	DC		/7E00	ONE CHAR EBC MASK
337	RGTI	DC		/1200	I
338	RGTN	DC		/1800	N
339	IDCI	DC		/6020	INTEGER SYMBOL TABLE ID
340	IDCR	DC		/2020	REAL SYMBOL TABLE ID
341	PARNL	DC		/000D	LEFT PARENTHESIS
342	COMAC	DC		/002B	COMMA
343		BSS	E	0	
344	TEM	DC		0	EBC NUMBER TO
345		DC		0	BE CONVERTED
346	COT1	DC		/7E00	EBC NUMERIC MASK
347	COT2	DC		/7800	EBC NUMERIC BITS MASK
348	HELP1	DC		0	TEMPORARY STORAGE
349	NINT	DC		/4800	NINE
350	SUMM	DC		0	TEMPORARY TOTAL
351	CNT	DC		0	DIMENSION COUNTER
352	PARNR	DC		/001D	RIGHT PARENTHESIS
353	*				CONVERTS THE DIMENSION CONSTANT
354	*				TO BINARY
355	PRTE	SLA		16	
356		STO		CNT	CLEAR COUNTER
357		MDX	1	1	MOVE POINTER
358		LD	1	0	LOAD WORD
359		AND		PIT	IS IT NUMERIC

360	EOR		PIT			
361	BSC	L	GEEP,Z		BRANCH IF NOT	
362	*			CHANGE	CONSTANT TO BINARY NO,	
363	LDS		0		CLEAR STATUS INDICATORS	
364	SLT		32		CLEAR ACC AND EXTENSION	
365	STO		SUMM		ZERO TOTAL	
366	LD	1			LOAD FIRST WORD	
367	STD		TEM		STORE IN WORK AREA	
368	LD	1	1		LOAD 2ND WORD	
369	BSC	L	AOT,-		BRANCH ON OPERATOR	
370	SLA		1		GET RID OF INDICATOR BIT	
371	STO		TEM+1		STORE 2ND WORD	
372	LD	1	2		LOAD 3RD WORD	
373	MDX	1	1		MOVE POINTER	
374	BSC	L	GEEP,+		BRANCH ON NO OPERATOR	
375	AOT		LD	TEM	LOAD FIRST WORD	
376	AOT1		AND	COT1	GET FIRST CHAR FROM WORD	
377	SLA			2	SHIFT TO TEST IF NUMERIC	
378	BSC	L	PILL,C		BRANCH CARRY ON	
379	BSC	L	NEX,+		BRANCH END OF CONVERSION	
380	BSC	L	GEEP		BRANCH ERROR	
381	PILL	BSC	L	GEEP,-	BRANCH NOT NUMERIC	
382			AND	COT2	GET NUMBER BITS	
383			S	NINT	IS NO, GREATER THAN NINE	
384		BSC	L	GEEP,-Z	BRANCH IF YES	
385			A	NINT	RESTORE ORIGINAL VALUE	
386			SRA	11	RIGHT JUSTIFY NUMBER	
387			STO	HELP1	SAVE NUMBER	
388			LD	SUMM	LOAD PREVIOUS TOTAL	
389			SLA	1	MULTIPLY BY TWO	
390			STO	TEMPT	SAVE	
391		BSC	L	GEEP,+Z	BRANCH IF OVERFLOW	
392			SLA	2	MULTIPLY TOTAL BY 8	
393		BSC	L	GEEP,C+Z	BRANCH IF ERROR IN NUMBER	
394			A	TEMPT	GET PREVIOUS TOTAL X 10	
395			A	HELP1	ADD UNITS VALUE	
396		BSC	L	GEEP,0	BRANCH IF OVERFLOW	
397			STO	SUMM	STORE NEW TOTAL	
398			LDD	TEM	SHIFT NUMBER TO	
399			SLT	6	GET EBC NUMBER	
400			STD	TEM	IN POSITION	
401			MDX	AOT1	GO PROCESS NUMBER	
402	*			C	0	TEMPORARY STORAGE
403	THRE	DC		3		THREE
404	ZTEMP	DC		0		TEMPORARY STORAGE
405	FLOG	DC		0		PRINTER STORAGE
406	BOUND	DC		0		RANGE OF LOOP
407	*					PLACES A DIMENSION CONSTANT INT
408	*					THE SYMBOL TABLE
409	NEX	MDX	L	CNT,1		INCREMENT DIMENSION COUNT
410			LD	SUMM		LOAD DIMENSION VALUE
411			BSC	L	GEEP,+	BRANCH IF ZERO
412	*					PUT CONSTANT IN SYMBOL TABLE
413			LD	CNT		LOAD DIMENSION COUNT
414			S	L	SW2	WAS COUNT ONE
415			BSC		Z	SKIP IF YES
416			MDX		02	CHECK FOR COUNT OF TWO
417			LD		SUMM	LOAD DIMENSION VALUE
418			STO	3	-1	PUT IN ALL
419			STO	3	-2	THREE DIMENSION

420		STO	3	-3	WORDS
421		MDX		NEXP	GO GET NEXT NUMBER
422	02	S	L	SW2	WAS COUNT TWO
423		BSC		Z	SKIP IF YES
424		MDX		03	CHECK FOR COUNT OF THREE
425		LD	3	-1	GET VALUE FOR 2ND AND 3RD
426		M		SJMM	DIMENSION BY MULTIPLYING
427		SLT		16	FIRST VALUE BY SECOND
428		STO	3	-2	VALUE
429		STO	3	-3	STORE IN TABLE
430		MDX		NEXP	GO GET NEXT NUMBER
431	03	LD	3	-3	GET VALUE FOR 3RD DIM
432		M		SJMM	BY MULTIPLYING 2ND VALUE
433		SLT		16	BY 3RD VALUE
434		STO	3	-3	STORE IN TABLE
435	*				CHECKS FOR A COMMA, A RIGHT
436	*				PARENTHESIS, AND AN OVERLAP
437	*				ERROR - INDICATES THE
438	*				DIMENSIONING LEVEL IN THE SYM
439	*				TBL ID WORD
440	NEXP	MDX	1	1	MOVE STRING POINTER
441		LD	1	0	LOAD WORD
442		S		COMAC	IS CHARACTER COMMA
443		BSC	L	TCNT,+	BRANCH IF YES
444		A		COMAC	RESTORE ORIGINAL VALUE
445		S		PARNR	IS IT RIGHT PARENTHESIS
446		BSC	L	GEEP,Z	BRANCH IF NOT
447	*			PUT	DIMENSION COUNT IN SYM TBL
448		LD		CNT	LOAD DIMENSION COUNT
449		SLA		11	POSITION FOR SYMBOL TABLE
450		OR	3	0	COMBINE WITH ID WORD
451		STO	3	0	NEW SYMBOL TABLE ID WORD
452		MDX	L	EOFST,-3	MODIFY SYMBOL
453		MDX	L	SOFXT,-3	TABLE ADDRESSES
454		MDX	L	SOFGT,-3	
455		LD	L	EOFST	TEST FOR TABLE OVERLAP
456		A		TWO	ADD TWO
457		S	L	EOFST	SUBTRACT END OF STRING
458		BSC	L	JIMY,+Z	BRANCH IF OVERLAP
459		MDX	1	1	MOVE POINTER
460		BSC	L	DD3	GO CHECK NEXT WORD
461	*			CONSTANT	
462	TWO	DC		2	TWO
463	JIMY	MDX	L	ERROR,1	SET OVERLAP ERROR
464	*				CHECKS THAT DIMENSIONING
465	*				DOES NOT EXCEED THREE LEVELS
466	TCNT	LD	L	CNT	LOAD DIMENSION COUNT
467		S		THRE	IS IT THREE OR GREATER
468		BSC	L	PRTE+2,+Z	BRANCH IF NOT
469		BSC	L	GEEP	ERROR - BRANCH
470	*				REMOVES A STMT FROM THE STRING
471	*				CLOSES UP THE STRING
472	RMOVE	LDX	I1	COMSV	RESET INPUT POINTER
473		MDX	1	-1	
474		LD	1	1	LOAD STMT ID WORD
475		SRA		2	RIGHT-JUSTIFY NORM
476		AND	L	IDNRM	CLEAR ALL BUT NORM
477		STO		ZTEMP	
478		LD	L	EOFST	LOAD END OF STRING
479		S	L	COMSV	SUBTRACT ORIGINAL ID ADDR

480		S		ZTEMP	SUBTRACT NORM
481		STO		BOUND	GET RANGE OF LOOP
482		LDX	I3	BOUND	INITIALIZE RANGE COUNTER
483		MDX	3	1	
484		LD	L	COMSV	LOAD ORIGINAL ID ADDRESS
485		A		ZTEMP	GET NEXT ID ADDRESS
486		STO		ZTEMP	PUT IN ZTEMP
487		LDX	I2	ZTEMP	INITIALIZE INPUT LOOP PT
488		STX	1	FLOG	SAVE STRING INPUT POINTER
489	LPP	LD	2		MOVE WORD DOWN
490		STO	1	1	IN STRING
491		MDX	1	1	MOVE POINTERS
492		MDX	2	1	
493		MDX	3	-1	DECREMENT COUNTER
494		MDX		LPP	CONTINUE LOOP
495		STX	L1	EOFS	NEW END OF STRING ADDRESS
496		LDX	I1	FLOG	RESTORE INPUT POINTER
497		MDX	1	1	MOVE POINTER
498		BSC	L	TST	GO TO CHECK FOR END
499	*				CHECKS FOR A SLASH (/) FOLLOWIN
500	*				A SLASH (/) - IF FOUND INDICATE
501	*				A CORE LOAD COMMON VARIABLE
502	QCOM	LD	1	1	LOAD WORD
503		S		SLAS	IS CHARACTER SLASH
504		BSC	L	Q2,Z	BRANCH IF NOT
505		MDX	1	2	MOVE POINTER
506		SLA		16	
507	Q1	STO	L	COMID	CLEAR COMMON TYPE INDICAT
508		BSC	L	PTB	GO TO CHARACTER CHECK
509	Q2	LD	1	1	RELOAD THE STRING WORD
510		S		QN1	COMPARE TO INSKEL
511		BSC	L	SETUP,Z	BR IF NO COMPARISON
512		LD	1	2	COMPARE ANOTHER PART OF WD
513		S		QN2	WITH INSKEL
514		BSC	L	SETUP,Z	AGAIN, BR IF NO COMPARISON
515		LD	1	3	GET THE LAST PART AND MAKE
516		S		QN3	FINAL COMPARISON
517		BSC	L	SETUP,Z	BR OUT IF COMPARISON FAILS
518	*				CHECK FOR A SLASH
519		LD	1	4	GET THE FOLLOWING CHARACTE
520		S		SLAS	AND COMPARE IT TO A (/)
521		BSC	L	SETUP,Z	BR OUT IF NO COMPARISON
522		MDX	1	5	MOVE POINTER
523	*				CHECK FOR ONE WORD INTEGERS
524		LD	L	CCWD	GET CONTROL CARD WORD AND
525		SLA		9	CHECK FOR BIT 9 ON
526		BSC	L	EROWD,-	BR AND SET ERROR IF NOT
527	*				SET COMMON ID TO ONE
528		LD		QN4	GET THE CONSTANT
529		STO	L	COMID	PUT IT IN STORAGE
530		BSC	L	DD2	EXIT TO CHECK NAME
531	*				PUT OUT ERROR NUMBER 74
532	EROWD	LDX	1	74	SET AN ERROR COND AND
533		STX	L1	ERRNO	PUT IT IN STORAGE
534		BSC	L	MOVE	EXIT TO CLOSE UP STRING
535	*				CONSTANTS USED FOR INSKEL
536	QN1	DC		/92AC	CONSTANTS FOR
537	QN2	DC		/A485	DETERMINING
538	QN3	DC		/A600	INSKEL COMMON
539	QN4	DC		1	ID CONSTANT

```

540 SLAS DC /21 PACKED EBC SLASH
541 START LD L ERROR IS THERE OVERLAP ERROR
542 BSC L SOS,Z BRANCH IF YES
543 * INITIALIZE PHASE
544 LDX I1 S0FS INITIALIZE STRING POINTER
545 STARF LD 1 0 LD STMT ID
546 SRA 11
547 S FIOC TEST IF FIO CALL
548 BSC L STAR1,Z BRANCH IF NOT
549 STX SW5A SET FIO STMT PRESENT SW
550 LD 1 0 GET STMT NORM
551 SRA 2
552 AND IDNVQ
553 STO **1
554 MDX L1 *** MOVE TO NEXT STMT
555 MDX STARF CHECK NEXT STMT
556 * CHECKS THE FORTRAN COMMUNICATIO
557 * AREA (S0KF) FOR A SUBP I'DN
558 * IF NONE, CHECKS THE FIRST STMT
559 * TO SEE IF IT IS REAL OR INTEGER
560 STAR1 LD L SORF HAVE SUBROUTINE OR FUNC
561 BSC L D03,Z BRANCH IF YES
562 LD 1 0 LOAD STMT ID WORD
563 SRA 1
564 AND IDTPQ GET STMT ID TYPE
565 S REALC IS STMT REAL
566 BSC L TRY,+ BRANCH IF YES
567 S INTC IS STMT INTEGER
568 BSC L SOS,Z BRANCH IF NOT
569 STX L0 ASW3 SET INTEGER STMT SW
570 MDX TRY+2 CHECK FOR FUNCTION
571 * CHECKS FOR THE WORD *FUNCTION*
572 * IN A REAL OR INTEGER STMT
573 * IF FOUND, INDICATES A FUNCTION
574 * IN THE FORTRAN COMMUNICATION
575 * AREA (S0KF)
576 TRY STX L0 ASW2 SET REAL STMT SW
577 * ARE NEXT FOUR WORDS *FUNCTION*
578 LD 1 1 LOAD WORD
579 S FUNC1 ARE CHARACTERS *FU*
580 BSC L SOS,Z BRANCH IF NOT
581 LD 1 2 LOAD 2ND WORD
582 S FUNC2 ARE CHARACTERS *NC*
583 BSC L SOS,Z BRANCH IF NOT
584 LD 1 3 LOAD 3RD WORD
585 S FUNC3 ARE CHARACTERS *IO*
586 BSC L SOS,Z BRANCH IF NOT
587 LD 1 4 LOAD 4TH WORD
588 S FUNC4 IS CHARACTER *N*
589 BSC L SOS,Z BRANCH IF NOT
590 STX 1 SUBSV SAVE STRING POINTER
591 MDX 1 5 MOVE POINTER
592 * INDICATE FUNCTION IN
593 * COMMUNICATION AREA
594 MDX L SORF,1 SET FUNCTION IN SORF
595 MDX D02+1 CHECK FOR LEGAL NAME
596 * CHECKS FOR A SUBP OR FUNC STMT
597 D03 LD 1 0 LOAD STMT ID WORD
598 SRA 11 GET STMT ID TYPE
599 S SF1 IS IT FUNCTION

```

600	BSC	L	DO2,+	BRANCH IF YES
601	S		SF2	IS IT SUBROUTINE
602	BSC	L	SQS,Z	BRANCH IF NO
603	*			CHECKS FOR A VALID SUBP NAME
604	DO2	STX	1 SUBSV	SAVE INPUT POINTER
605		MDX	1 1	MOVE POINTER
606		LD	1 0	LOAD WORD
607		BSC	-	SKIP IF NUMBER OR LETTER
608		MDX	ER7	ERROR
609		STO	WRD1	SAVE WORD
610		LD	1 1	LOAD 2ND WORD
611		BSC	L MADS,+Z	BRANCH IF NAME TWO WORDS
612		LD	Z0	SET ZEROS FOR
613		STO	WRD2	SECOND WORD OF NAME
614		MDX	ZOR	POT NAME IN SYMBOL TABLE
615	MADS	STO	WRD2	STORE 2ND WORD
616		MDX	1 1	MOVE POINTER
617		LD	1 1	LOAD NEXT WORD
618		BSC	L ER7,+Z	ERROR IF NOT SPECIAL OPER
619	*			SCANS THE SYM TBL FOR
620	*			DUPLICATION OF THE SUBP NAME
621	ZOR	LDX	13 SOFNS	INITIALIZE SYMBOL TABLE P
622		MDX	HACK	START CHECK
623	LOP	LD	3 1	DOES FIRST WORD OF SYMBOL
624		S	WRD1	TABLE MATCH 1ST PART
625		BSC	Z	SKIP IF YES
626		MDX	EATQ	GET NEXT TABLE ENTRY
627		LD	3 2	DOES 2ND WORD OF SYMBOL
628		S	WRD2	TABLE MATCH 2ND PART
629		BSC	L CHK,+	BRANCH IF YES
630	EATQ	MDX	3 -3	GO TO NEXT TABLE ENTRY
631	HACK	STX	3 SAVE	STORE POINTER
632		LD	SAVE	HAS ENTIRE SYMBOL TABLE
633		S	L EQFST	BEEEN CHECKED FOR DUPLICAT
634		BSC	Z	SKIP IF YES
635		MDX	LOP	CONTINUE LOOP
636		MDX	PLACQ	PLACE NAME IN TABLE
637	*			CONSTANTS
638	SF1	DC	/0D	FUNCTION
639	SF2	DC	/04-/0D	SUBROUTINE
640	SW5A	DC	0	SWITCH 5A
641	FIOC	DC	/1B	FIO
642	REALC	DC	/2400	REAL
643	INTC	DC	/0400	INTEGER
644	FUNC1	DC	/8D22	FU
645	FUNC2	DC	/D0E3	NC
646	FUNC3	DC	/92B2	TI
647	FUNC4	DC	/D000	ON
648	WRD1	DC	**	TEMPORARY STORAGE FOR
649	WRD2	DC	**	SYMBOL TABLE ENTRY
650	IDNVQ	DC	/01FF	NORM MASK
651	SUBSV	DC	0	POINTER STORAGE
652	NVMSV	DC	**	NORM STORAGE
653	IDTPQ	DC	/7C00	STMT ID TYPE MASK
654	SAVE	DC	0	TEMPORARY STORAGE
655	ERNO	DC	**	ERROR NUMBER
656	ER7R	DC	7	ERROR NO, 7
657	ERR12	DC	12	ERROR NO, 12
658	ERR13	DC	13	ERROR NO, 13
659	ERR14	DC	14	ERROR NO, 14

```

660 ERID DC /A008 ERROR STMT ID WORD
661 ZO DC /8000 SECOND WORD OF 1 WORD NAM
662 * CHECKS THAT A PARAMETER NAME
663 * HAS NOT BEEN ENTERED INTO COMMO
664 CHK LD 3 0 LOAD SYMBOL TABLE ID WORD
665 SLA 2 IS IT IN COMMON
666 BSC +Z BRANCH IF NOT
667 MDX ER13 GO SET UP ERROR 13
668 * SET UP ERRORS
669 ER14 LD ERR14 SET UP
670 STO ERNO ERROR NO, 14
671 MDX CLOSE CLOSE UP STRING
672 ER7 LD ER7R SET UP
673 STO ERNO ERROR NO, 7
674 MDX CLOSE CLOSE UP STRING
675 ER12 LD ER12 SET UP
676 STO ERNO ERROR NO, 12
677 MDX CLOSE CLOSE UP STRING
678 ER13 LD ER13 SET UP
679 STO ERNO ERROR NO, 13
680 * CLOSES THE STRING AFTER
681 * REPLACING THE ERRONEOUS STMT
682 * WITH AN ERROR MESSAGE
683 CLOSE LDX I1 SUBSV LOAD STMT ID WORD ADDRESS
684 LD 1 0 LOAD STMT ID WORD
685 SRA 2
686 AND IDNVQ CLEAR ALL BUT NORM
687 STO NVMSV GET ADDRESS OF NEXT STMT
688 A SUBSV BY ADDING NORM TO ADDRESS
689 STO **1 OF PRESENT STMT
690 LDX L2 *** INITIALIZE INPUT LOOP PT
691 LD ERID LOAD ERROR STMT ID WORD
692 STO 1 0 PUT ON STRING
693 LD ERNO LOAD ERROR NUMBER
694 STO 1 1 PUT ON STRING
695 MDX 1 1 MOVE OUTPUT LOOP POINTER
696 LD L EOFs COMPUTE RANGE
697 S SUBSV OF MOVE LOOP
698 S NVMSV
699 STO **1 STORE RANGE IN
700 LDX L3 *** RANGE COUNTER
701 MDX 3 1
702 LOOPZ LD 2 0 MOVE WORD DOWN
703 STO 1 1 IN STMT STRING
704 MDX 1 1 MOVE POINTERS
705 MDX 2 1
706 MDX 3 -1 DECREMENT COUNTER
707 MDX LOOPZ CONTINUE LOOP
708 STX L1 EOFs NEW END OF STRING ADDRESS
709 * BRANCHES TO THE ROL ROUTINE TO
710 * LOAD THE NEXT PHASE
711 SOS BSI L ROLRX CALL PHASE ROLLER
712 DC 05 NEXT PHASE NUMBER
713 * CONSTANTS
714 H7E00 DC /7E00 STMT ID TYPE MASK
715 H1200 DC /1200 I
716 H1800 DC /1800 N
717 H4080 DC /4080 INTEGER SUBPROGRAM
718 H0080 DC /0080 REAL SUBPROGRAM
719 SEMI DC /001E SEMICOLON

```

720	RPAR	DC		/1D-/2B	RIGHT PARENTHESIS
721	LPAR	DC		/0D	LEFT PARENTHESIS
722	TWOS	DC		2	TWO
723	ASW2	DC		*-*	SWITCH 2A
724	ASW3	DC		*-*	SWITCH 3A
725	THREE	DC		3	THREE
726	SIGN1	DC		/8001	SIGN
727	*				PLACES THE SUBP NAME INTO THE
728	*				SYM TBL - PLACES THE ADDR OF TH
729	*				SYM TBL ENTRY INTO THE FORTRAN
730	*				COMMUNICATIONS AREA (FNAME)
731	PLACQ	LD		WRD2	LOAD SECOND WORD
732		STO	3	2	PUT IN SYMBOL TABLE
733		LD		WRD1	LOAD FIRST WORD
734		STO	3	1	PUT IN SYMBOL TABLE
735	*				PUT SYMBOL TABLE ADDRESS IN
736	*				COMMUNICATION AREA
737		LD	L	SOFST	FIND LOCATION OF PRESENT
738		S		SAVE	ENTRY RELATIVE TO START
739		SRT		16	OF SYMBOL TABLE
740		D		THREE	GET ENTRY POSITION
741		A		SIGN1	GIVE SIGN
742		STO	L	FNAME	SAVE
743		LD		ASW2	IS IT REAL
744		BSC	L	IGER,+*	BRANCH IF NOT
745		SLA		16	
746		STO		ASW2	CLEAR REAL STMT SW
747		MDX		REELQ	GO SET REAL SUBPROGRAM BI
748	IGER	LD		ASW3	ID IT INTEGER
749		BSC	L	PLAC1,+*	BRANCH IF NOT
750		SLA		16	
751		STO		ASW3	CLEAR INTEGER STMT SW
752		MDX		PLAC1+8	GO SET INTEGER SUBPRG BIT
753	*				SET THE SUBP AND TYPE INDRS IN
754	*				THE SYM TBL ID WORD OF A SUBP
755	*				SUBP NAME FOUND IN A SUBP, FJNC
756	*				REAL FUNCTION, OR INTEGER FUNC
757	*				STMT - CHECKS FOR A SYM TBL
758	*				OVERLAP
759	PLAC1	LD		WRD1	LOAD FIRST WORD
760		AND		H7E00	CLEAR ALL BUT FIRST WORD
761		S		H1200	IS IT LESS THAN I
762		BSC		+Z	SKIP IF YES
763		MDX		REELQ	GO SET REAL SUBP BITS
764		S		H1800	IS IT N OR LESS
765		BSC	L	REELQ,Z-	BRANCH IF NOT
766		LD		H4080	SET INTEGER SUBPRG BITS
767		MDX		VIPQ	SET INTEGER SUBP BITS
768	REELQ	LD		H0080	SET REAL SUBPROGRAM BITS
769	VIPQ	STO	3	0	PUT IN SYMBOL TABLE
770		MDX	3	-3	MOVE SYMBOL TABLE POINTER
771		STX	L3	EOFST	NEW END OF SYMBOL TABLE
772		MDX	L	SOFXT,-3	MOVE SYMBOL
773		MDX	L	SOFGT,-3	TABLE ADDRESSES
774	*				SYMBOL TABLE OVERLAP
775		LD	L	EOFST	LOAD END OF SYMBOL TABLE
776		A		TWOS	ADD TWO
777		S	L	EOFS	SUBTRACT END OF STRING
778		BSC	L	JIMMY,+Z	BRANCH IF OVERLAP ERROR
779	SLOP	MDX	1	1	MOVE POINTER

```

780 *          CHECKS FOR A LEFT PARENTHESIS OR
781 *          A STMT TERMINATOR FOLLOWING THE
782 *          SUBP NAME
783 DTB      LD      1 0          LOAD WORD
784          S          SEMI          IS CHARACTER SEMI-COLON
785          BSC      L  RMV,+ -    BRANCH IF YES
786          LD      ASW1          RIGHT PARENTHESIS FOUND
787          BSC      L  ER12,Z     BRANCH IF YES
788          LD      1 0          LOAD WORD
789          S          LPAR         IS CHARACTER LEFT PAREN
790          BSC      L  ER12,Z     BRANCH IF NOT
791 FLIP     MDX     1 1          MOVE POINTER
792 *          CHECKS FOR A VALID PARAM NAME
793 FLOP     LD      1 0          LOAD WORD
794          BSC      -          SKIP IF NOT SPECIAL CHAR
795          MDX     ER7          GO TO ERROR ROUTINE
796          STO     WDD1         SAVE WORD
797          LD      1 1          LOAD 2ND WORD
798          BSC      L  SADS,+Z    BRANCH IF PART OF NAME
799          LD      L  Z0          LOAD ZEROS AS 2ND WORD
800          STO     WDD2         STORE
801          MDX     ZARRO        GO PUT IN SYMBOL TABLE
802 SADS     STO     WDD2         STORE 2ND WORD
803          MDX     1 1          MOVE POINTER
804          LD      1 1          LOAD NEXT WORD
805          BSC      L  ER7,+Z    BRANCH NOT SPECIAL CHAR
806 *          SCANS THE SYM TBL FOR
807 *          DUPLICATION OF A VAR NAME
808 ZARRO    LDX     13 SOFNS     LOAD START OF CHECK ADDR
809          MDX     CAKE          START LOOP
810 LIPS     LD      3 1          DOES FIRST WORD OF NAME
811          S          WDD1       MATCH 1ST WORD OF TABLE
812          BSC      Z          SKIP IF YES
813          MDX     CLET         GET NEXT SYMBOL TBL ENTRY
814          LD      3 2          DOES 2ND WORD MATCH 2ND
815          S          WDD2       WORD OF SYMBOL TBL ENTRY
816          BSC      L  CHK,+ -    BRANCH IF YES
817 CLET     MDX     3 -3         MOVE PT TO NEXT TBL ENTRY
818 CAKE     STX     3 SIV        SAVE TABLE POINTER
819          LD      SIV          LOAD ENTRY ADDRESS
820          S          L  EOFST    SUBTRACT END OF TABLE
821          BSC      Z          SKIP IF FINISHED CHECK
822          MDX     LIPS         CONTINUE LOOP
823          MDX     PIECE        PUT NAME IN SYMBOL TABLE
824 *          CONSTANTS
825 WDD1     DC      0          TEMPORARY STORAGE FOR
826 WDD2     DC      0          SYMBOL TABLE ENTRY
827 SIV      DC      0          TEMPORARY STORAGE
828 IDCII    DC      /4400     INTEGER FUNCTION
829 IDCRR    DC      /0400     REAL FUNCTION
830 COMA     DC      /2B      COMMA
831 ASW1     DC      0          SWITCH 1A
832 ERR15    DC      15        ERROR NO, 15
833 *          PLACES THE PARAM NAME INTO THE
834 *          SYM TBL - SETS THE PARAM AND
835 *          TYPE INDICATORS IN THE SYM TBL
836 *          CHECKS FOR A SYM TBL OVERLAP
837 *          CHECKS FOR A COMMA OR RIGHT
838 *          PARENTHESIS
839 PIECE    LD      WDD2        LOAD 2ND WORD

```

840		STO	3 2	PUT IN SYMBOL TABLE
841		LD	WDD1	LOAD 1ST WORD
842		STO	3 1	PUT IN SYMBOL TABLE
843	*			INDICATE TYPE AND
844	*			FORMAL PARAMETER
845		LD	WDD1	LOAD FIRST WORD
846		AND	L H7E00	GET FIRST CHARACTER
847		S	L H1200	IS IT LESS THAN I
848		BSC	+Z	SKIP IF NO
849		MDX	KEEL	SET REAL INDICATORS
850		S	L H1800	IS IT GREATER THAN N
851		BSC	L KEEL,Z-	BRANCH IF YES
852		LD	IDCII	LOAD INTEGER FUNCTION ID
853		MDX	HIPS	GO PUT IN SYMBOL TABLE
854	KEEL	LD	IDCRR	LOAD REAL FUNCTION ID
855	HIPS	STO	3 0	PUT IN SYMBOL TABLE
856		MDX	3 -3	MOVE POINTER
857		STX	L3 EOFST	NEW END OF SYMBOL TABLE
858		MDX	L SOFXT,-3	MODIFY SYMBOL
859		MDX	L SOFGT,-3	TABLE ADDRESSES
860	*			SYMBOL TABLE OVERLAP
861		LD	L EOFST	LOAD END OF SYMBOL TABLE
862		A	L TWOS	ADD TWO
863		S	L EOFS	SUBTRACT END OF STRING
864		BSC	L JIMMY,+Z	BRANCH IF OVERLAP
865		MDX	1 1	MOVE STRING POINTER
866		LD	1 0	LOAD WORD
867		S	COMA	IS CHARACTER COMMA
868		BSC	L FLIP,+-	BRANCH IF YES
869		S	L RPAR	IS CHARACTER RIGHT PAREN
870		BSC	L ER12,Z	BRANCH IF NOT
871		STX	0 ASW1	SET RIGHT PARENTHESIS SW
872		MDX	SLOP	CHECK REST OF STMNT
873	*			REMOVES A STMNT FROM THE STRING
874	*			CLOSES UP THE STRING
875	RMV	LDX	I1 SUBSV	INITIALIZE INPUT LOOP PT
876		MDX	1 -1	
877		LD	1 1	LOAD STMNT ID WORD
878		SRA	2	
879		AND	L IDNVQ	GET STMNT NORM
880		STO	ZT+1	
881		LD	L EOFS	GET RANGE OF MOVE LOOP BY
882		S	L SUBSV	SUBTRACTING PRESENT STMNT
883		S	ZT+1	FROM END OF STRING - THEN
884		STO	**+1	SUBTRACTING STMNT SIZE
885		LDX	L3 **	LOAD RANGE COUNTER
886		LD	L SUBSV	GET ADDRESS OF NEXT STMNT
887		A	ZT+1	BY ADDING NORM TO PRESENT
888		STO	ZT+1	STMNT ADDRESS
889	ZT	LDX	L2 **	INITIALIZE INPUT LOOP PT
890		MDX	3 1	
891		LD	L SW5A	IS FID STMNT PRESENT
892		BSC	L LPPQ,+-	BRANCH IF NOT
893		LDX	I1 SOFS	LOAD START OF STRING ADDR
894		LD	ERR15	LOAD ERROR NO, 15
895		STO	1 1	PUT ON STRING
896		LD	L ERID	LOAD ERROR STMNT ID WORD
897		STO	1 0	PUT ON STRING
898		MDX	1 1	MOVE POINTER
899	*			REMOVES AN ERRONEOUS STMNT FROM

```

900 *          THE STRING
901 LPPQ LD      2 0      LOAD WORD
902      STO    1 1      PUT ON STRING
903      MDX    1 1      MOVE POINTER
904      MDX    2 1
905      MDX    3 -1     DECREMENT POINTERS
906      MDX    LPPQ     CONTINUE LOOP
907      STX    L1 EDFS  NEW END OF STRING
908      BSC    L SOS     GO TO END OF PHASE
909 JIMMY MDX    L ERROR,1 SET OVERLAP ERROR INDR
910      BSC    L SOS     GO TO END OF PHASE
911      BSS    OVERL-**+320*3 PHASE-04 PATCH AREA
912      END      TOP

```