

TABLE OF CONTENTS

((((((((

ABS

LOC 20000

1	000000		
4	000000		
5	020000	* PAGE	CONTENTS
6	020000	* 2	ENTRY POINTS
7	020000	* 4	EXTENDED MEMORY POINTERS
8	020000	* 5	PARSER MAIN PROGRAM AND PSETUP
9	020000	* 6	PARSIT - MAIN PARSE INNER LOOP
10	020000	* 8	P.PARS, C.PARS, E.PARS
11	020000	* 10	PARSTP (GET) AND BINOR2
12	020000	* 13	SCANNER
13	020000	* 16	SCAN: NUMBERS
14	020000	* 19	GCHAR, ALFA, DIGIT
15	020000	* 20	TABLE SEARCH
16	020000	* 23	SAVE, RESTORE REGISTERS
17	020000	* 24	NUMERIC VALUE
18	020000	* 26	LOGICAL PRIMARY
19	020000	* 28	NUMERIC EXPRESSION
20	020000	* 30	FACTOR, PRIMARY
21	020000	* 31	GET STANDARD FUNCTION PARAMETERS
22	020000	* 33	NARREF, GETCHA, GET), SUBSCR
23	020000	* 34	NUMERIC REPRESENTATION
24	020000	* 36	STRING EXPRESSION
25	020000	* 38	STRING REFERENCE
26	020000	* 40	REMARK AND STRING CONSTANT
27	020000	* 42	PARSE TIME COMMON ROUTINES
28	020000	* 45	ASSIGN, RENAME
29	020000	* 46	OPTION BASE, ON KEY#, ON ERROR
30	020000	* 48	DIMENSION AND TYPE STATEMENT
31	020000	* 51	TRACE VARIABLE
32	020000	* 52	DEF FN STATEMENT
33	020000	* 55	FUNCTION NAME SUBROUTINE
34	020000	* 56	FUNCTION CALLS FNA(X), FNA*(X)
35	020000	* 59	INPUT STATEMENT PARSE
36	020000	* 60	DATA STATEMENT PARSE
37	020000	* 61	READ, READ# PARSE
38	020000	* 62	PRINT, PRINT USING, PRINT# PARSE
39	020000	* 65	LET STATEMENT
40	020000	* 66	FUNCTION LET
41	020000	* 67	REF STRING, ARITH VRBL, GOTO, GOSUB
42	020000	* 68	IF STATEMENT
43	020000	* 71	FOR NEXT STATEMENTS
44	020000	* 72	ON GOTO/GOSUB STATEMENT
45	020000	* 73	SEQUENCE NUMBERS, GOTO, GOSUB
46	020000	* 75	AUTO LINE NUMBER AND RENUMBER
47	020000	* 78	EXTERNALS

Entry Points

««««««««

020000	ENT ALFA
50 020000	ENT AUTO.
51 020000	ENT AUTOX
52 020000	ENT SP5ALT
53 020000	ENT COM
54 020000	ENT DATA
55 020000	ENT DEF
56 020000	ENT DIGIT
57 020000	ENT DIN
58 020000	ENT DMNOCR
59 020000	ENT E.PARS
60 020000	ENT ERR88U
61 020000	ENT FORMAR
62 020000	ENT FORMSA
63 020000	ENT FNEND
64 020000	ENT FHLET
65 020000	ENT FOR
66 020000	ENT G\$N
67 020000	ENT G\$N+NN
68 020000	ENT G012N
69 020000	ENT G01H
70 020000	ENT G00R2N
71 020000	ENT G120R4
72 020000	ENT G10R2N
73 020000	ENT GCHAR
74 020000	ENT GCDIG
75 020000	ENT GET)
76 020000	ENT GET\$N?
77 020000	ENT GET1\$
78 020000	ENT GET1N
79 020000	ENT GET2N
80 020000	ENT GET4N
81 020000	ENT GETCM?
82 020000	ENT GETCMA
83 020000	ENT GETPA?
84 020000	ENT GETPAR
85 020000	ENT GETST-
86 020000	ENT GETSTP
87 020000	ENT GLN/LB
88 020000	ENT GOTOSU
89 020000	ENT IDENT
90 020000	ENT IF
91 020000	ENT ILET
92 020000	ENT INPUT
93 020000	ENT INTEGR
94 020000	ENT LET
95 020000	ENT NARRE+
96 020000	ENT NARREF
97 020000	ENT NEXT
98 020000	ENT NUMBER
99 020000	ENT NUMCON
100 020000	ENT NUMEX+
020000	ENT NUMREP

>>>>>>		Entry Points	<<<<<<<<
020000		ENT NUMVA+	
103 020000		ENT NUMVAL	
104 020000		ENT OFKEY#	
105 020000		ENT ONERRD	
106 020000		ENT ONKEY#	
107 020000		ENT ONTIME	
108 020000		ENT ON	
109 020000		ENT OPTION	
110 020000		ENT PARPRM	
111 020000		ENT PARSER	
112 020000		ENT PARSTP	
113 020000		ENT PRINT	
114 020000		ENT P.PARS	
115 020000		ENT PUSH1A	
116 020000		ENT PUSH45	
117 020000		ENT PSRTH	
118 020000		ENT RAGAIN	
119 020000		ENT READ	
120 020000		ENT REFNUM	
020000		ENT REM	
122 020000		ENT RENUM.	
123 020000		ENT RESTOR	
124 020000		ENT RSTREG	
125 020000		ENT SAVREG	
126 020000		ENT SCAN	
127 020000		ENT SCAN+	
128 020000		ENT SEGNO+	
129 020000		ENT SET240	
130 020000		ENT SMLINT	
131 020000		ENT ST240+	
132 020000		ENT ST240-	
133 020000		ENT STREX+	
134 020000		ENT STREXP	
135 020000		ENT STREF	
136 020000		ENT SYSPR3	
137 020000		ENT TRCVB	
138 020000		ENT TRYIN	
139 020000		ENT TSTALO	
020000		ENT TYPSTM	
141 020000		ENT UNQUOT	

PARSER MAIN PROGRAM AND PSETUP

<<<<<<<

```

020000      *
143 020000      *
144 020000      *  PARSE  PARSE  PARSE  PARSE  PARSE
145 020000      *  THIS IS THE MAIN PARSE CALLED BY EXEC
146 020000      *  IT WILL BE HAPPY WITH EITHER PROGRAM
147 020000      *  MODE STATEMENTS, CALCULATOR MODE STATEMENTS,
148 020000      *  OR CALCULATOR MODE EXPRESSIONS.
149 020000      *  NOTE: IF IT REALLY IS THIS SIMPLE, IT
150 020000      *  SHOULD PROBABLY USE JUMPS, NOT JSB'S
151 020000      *
152 020000      *  A NEW STATEMENT WILL BE PARSED AT THE CURRENT
153 020000      *  END OF THE PROGRAM (IN EXTENDED MEMORY).
154 020000      *  POINTER PTR2 IS USED WITH AUTODECREMENT TO
155 020000      *  INSERT THE TOKENS IN THE ORDER REQUIRED
155 020000      *
*****
156 020000 316 333 207 PARSE  JSB =PRIDL  TRY ROM IDLE
157 020003 175 261 022      LDMD R75,=HXTMEM  ADDRESS OF FREE SPCE
157 020006 200
157 020007 263 314 377      STMD R75,=PTR2  RESET PARSE POINTER
159 020012 110 251 236      LDM  R10,=INPBUF  RESET INPUT POINTER
159 020015 200
160 020016 263 063 210      STMD R10,=SAVR10  SAVE FOR ERROR MESSAGES
161 020021 316 236 043      JSB =GCHAR  GET FIRST CHAR TO R20
162 020024
162 020024      *
162 020024      *  FOLLOWING 2 LINES ARE TO EXIT WITHOUT
162 020024      *  ERRORS, WHEN EMPTY LINES ARE INPUT
164 020024      *
*****
164 020024
165 020024
166 020024 120 310 015      CMB  R20,=CR  EMPTY LINE?
167 020027 367 070      JZR  PSRTH  JIF YES
168 020031
169 020031 316 331 042      JSB =INTEGR  GET LINE# IF ANY
170 020034 230      BIN  RETURN TO BINARY
171 020035 172 261 114      LDMD R72,=ERLIN#  GET ERRL, ERRN, ERRON,-#
171 020040 200
171 020041 006 345      PUMD R72,+R6  ONTO STACK
173 020043 370 005      JEN  PROGRAM  JIF PROG OR EXPRESSION
174 020045 316 233 041      JSB =C.PARS  DO CALC MODE STMT
175 020050 360 034      JMP  RSTORE  RESTORE ERRL, ERRN?
176 020052 316 317 040 PROGRAM JSB =P.PARS  DO PROGRAM MODE
177 020055 117 310 300      CMB  R17,=300  ANY ERRORS
178 020060 372 024      JNC  RSTORE  JIF NO
179 020062 116 210      ICB  R16  SET CALC
180 020064 112 261 346      LDMD R12,=OLDTOS  RESET R12 TO BE SURE
180 020067 203
181 020070 145 261 341      LDMD R45,=STSIZE  RESET PTR2 TO
181 020073 203
182 020074 313 003 000      ADM  R45,=3,0,0  STSIZE + 3 TO MAKE
182 020077 000
182 020100 263 314 377      STMD R45,=PTR2  UP FOR LINE NUMBER

```

```

>>>>>          PARSER MAIN PROGRAM AND PSETUP          <<<<<<<<
020103 316 313 041      JSB =E.PARS          DO EXPRESSION PARS
135 020106 172 006 343 RSTORE  FOMD R72,-R6      RESTORE STACK
136 020111 117 220      TSB R17          ANY ERRORS NOW?
137 020113 364 004      JNG PSRTH          JIF YES, DONT RESTORE
138 020115 172 263 114      STMD R72,=ERLIN#      NOW ERRN WILL WORK
138 020120 200
189 020121 175 261 314 PSRTH  LDMD R75,=PTR2      SAVE POINTER
189 020124 377
190 020125 263 154 210      STMD R75,=SAVPT2
191 020130 236          RTN          DONE
  
```

PARSIT - MAIN PARSE INNER LOOP

<<<<<<<<

```

020131          *****
194 020131          *
195 020131          * PARSIT - MAIN PARSER INNER LOOP, CALLED
196 020131          * BY P.PARS, C.PARS, AND PARSTP
197 020131          *
198 020131          *****
199 020131          PARSED  DRP  114
200 020131 272 315 377          STBI R14,=PTR2-    FOR LINE CONTIN
201 020134 316 110 042 PARSIT  JSB  =SCAN          GET A BASIC STATEMENT
202 020137 114 310 004 PARSE!  CMB  R14,=4          IS IT A VARIABLE?
203 020142 372 004          JNC  IMPLT            JIF YES
204 020144 310 040          CMB  R14,=40          STRING ARRAY?
205 020146 766 010          JNZ  NOT123           JIF NO
206 020150          IMPLT  BSS  0
207 020150 114 250 221          LDB  R14,=221      IMPLIED LET TOKEN
208 020153 147 250 200          LDB  R47,=200      BASIC STMT
209 020156 760 002          JMP  BASIC?           DONT LOAD 14 WITH 43
210 020160 043 240          NOT123 LDB  R#,R43      KILL 370 AND 371!
211 020162          * We know the correct ROM is now selected
212 020162 136 261 100 BASIC? LDMD R36,=THENFL    NEED TO CHECK THEN?
213 020166 147          DRP  47          SAVE A DARP
214 020167 367 004          JZR  NOTHEN          JIF NOT IF-THEN
215 020171 310 300          CMB  R47,=300        LEGAL AFTER THEN?
216 020173 373 112          JCY  THENER          JIF NO
217 020175          NOTHEN DRP  147
218 020175 310 200          CMB  R47,=200        SEE IF SYSTEM TOKEN
219 020177 373 010          JCY  TYPEOK          JIF NO, DONT CHECK 16
220 020201 310 100          CMB  R47,=100        BASIC SYSTEM CALL?
221 020203 372 076          JNC  PRSERR          JIF NO, NOT BASIC
222 020205 116 220          TSB  R16            SYS MUST BE CALC MODE
223 020207 763 072          JEV  PRSERR          JIF NOT, ERROR
224 020211 136 261 006 TYPEOK LDMD R36,=R60+6    GET ROM PARSE ADDR
224 020214 140          TSB  R42            BUT IS IT A BINARY?
225 020215 142 220          JZR  PRSADR          JIF NO
226 020217 367 011          *****
227 020221          *
227 020221          * Change for Multiple Binaries
228 020221          *
229 020221          *****
230 020221 136 041 241          LDM  R36,R41      BINARY PRGM ADDRESS
231 020224 263 070 210          STMD R36,=BINTAB  MAKE IT ACTIVE
232 020227
233 020227 265 036 000          LDMD R36,X41,THIRT6 REPLACE WITH BINARY
234 020232 136 014 303 PRSADR ADM  R36,R14      ADD THE OFFSET
235 020235 303          ADM  R36,R14      *2
236 020236 036 245          LDMD R36,R36      GET PARSE ADDRESS
237 020240 306 000 000          JSB  X36,ZRO      JSB TO THE ROUTINE
238 020243 117 310 300          CMB  R17,=300    MAY BE 240, IF TRANS
239 020246 373 011          JCY  PRSRTN          JIF ERROR FOUND
240 020250 114 310 233          CMB  R14,=233    IN LINE COMMENT?

```

ARMA 03/14/81=====

ITEM LOC OBJECT CODE SRC=ORGFXX OBJ=QEXGFX 9/11/1981 11:11 AM PG 7

=====

```
>>>> PARSIT - MAIN PARSE INNER LOOP <<<<<<<<
241 020253 367 262 JZR PARSE! JIF YES
242 020255 310 100 CMB 14,=100 LINE CONTINUATION?
243 020257 367 250 JZR PARSE9 JIF YES, CONTINUE
244 020261 236 PRSRTN RTN DONE
245 020262 230 EOL BIN FOR REPEAT PARSSES
246 020263 114 310 015 CMB R14,=CR ARE WE DONE?
247 020266 367 004 JZR EOLOK JIF YES, OK
248 020270 316 377 377 JSB =ERROR CR NOT FOUND
249 020273 124 OCT 84D FALL THRU TO EOL
250 020274 136 250 016 EOLOK LDB R36,=16 PUSH LINE TERMINATOR
251 020277 272 315 377 STBI R36,=PTR2- TO OUTPUT
252 020302 236 RTN DONE
253 020303 316 377 377 PRSERR JSB =ERROR+ BAD BASIC STMT
254 020306 130 OCT 38D
255 020307 316 377 377 THENER JSB =ERROR+ ILLEGAL AFTER THEN
256 020312 126 OCT 36D
257 020313
258 020313 316 377 377 ERR90 JSB =ERROR+ ILLEGAL LINE#
259 020316 132 OCT 90D
260 020317
```

P.PARS, C.PARS, E.PARS

<<<<<<<<

```

020317      *
202 020317      *
263 020317      * P.PARS - PROGRAM MODE PARSE.
264 020317      * A LEADING INTEGER (POSSIBLE LINE#)
265 020317      * HAS BEEN SEEN
265 020317      *
265 020317      *
266 020317 155 040 241 P.PARS LDM R55,R40      Verify LN
267 020322 263 103 200      STMD R55,=AUTO#      For disc errors, AUTO
268 020325 273 315 377      STMI R55,=PTR2-      Push to output
269 020330 261 314 377      LDMD R55,=PTR2      Get pointer position
270 020333 263 341 203      STMD R55,=STSIZE      and save it
271 020336 213              DCM R55              skip STSIZE's place
272 020337 263 314 377      STMD R55,=PTR2      Update pointer
273 020342 112 263 346      STMD R12,=OLDTOS      Save R12 for errors
273 020345 203
274 020346 140 221          TSM R40              line zero?
275 020350 367 341          JZR ERR90            jif yes
276 020352 142 220          TSB R42              left nibble zero?
020354 375 335              JLN ERR90            jif no
278 020356 143 221          TSM R43              ln > 6 digits?
279 020360 366 331          JNZ ERR90            jif no
280 020362 120 310 015 LN#OK CMB R20,=CR        null line?
281 020365 367 061          JZR NULEOL           JIF YES (DONE)
282 020367      *
282 020367      *
283 020367      * CHECK IF THERE ARE AT LEAST 256
020367      * BYTES AVAILABLE FOR PARSING
284 020367      *
284 020367      *
285 020367      *
286 020367 145 261 022      LDMD R45,=NXTMEM      GET END OF CURR PROG
286 020372 200
287 020373 315 000 001      SBM R45,=0,1,0
287 020376 000
288 020377 321 025 200      CMMD R45,=LAVAIL      NXTMEM - LAVAIL > 256?
289 020402 372 164          JNC ERR19Z           JIF NO
290 020404 316 174 041      JSB =INDENT          INDENTATION
020407      *
292 020407      *===== CHANGE FOR LABELS =====
293 020407      *
294 020407 316 110 042      JSB =SCAN            GET A BASIC STATEMENT
295 020412 310 004          CMB R#,=4            IS IT A VARIABLE?
296 020414 373 024          JCY NOLABL           JIF NO
297 020416 120 310 072      CMB R20,=72          IS NEXT A COLON?
298 020421 366 017          JNZ NOLABL           JIF NO, IS VRBL
299 020423 250 257          LDB R20,=257          LOAD LABEL-RTIME-TOKEN
300 020425 272 315 377      STBI R20,=PTR2-      PUSH IT
301 020430 316 236 043      JSB =GCHAR          STEP OVER :
302 020433 316 107 051      JSB =PSH46           PUSH LEN + LABEL, SCAN
303 020436          DRP !14
304 020436 310 015          CMB R14,=CR          JUST THE LABEL?
305 020440 367 003          JZR LINEND           JIF YES

```

```

PARMA 03/14/81=====
ITEM LOC OBJECT CODE SRC=ORGFXN OBJ=GENGFX 9/11/1981 11:11 AM PG 9
=====
>>>>> P.PARS, C.PARS, E.PARS <<<<<<<
020442 316 137 040 NOLABL JSB =PARSE! PARSIT WITHOUT SCAN
307 020445 *
308 020445 *===== CHANGE ENDS =====
309 020445 *
310 020445 316 262 040 LINEND JSB =EOL DO END OF LINE
311 020450 230 NULEOL BIN FOR ADDITION
312 020451 155 261 314 LDMD R55,=PTR2 SAVE THIS POSITION
312 020454 377
313 020455 145 261 341 LDMD R45,=STSIZE STMT START POSITION
313 020460 203
314 020461 263 314 377 STMD R45,=PTR2 MOVE POINTER
315 020464 055 305 SBM R45,R55 STSIZE - OLD PTR2
316 020466 213 DCM R45
317 020467 146 220 TSB R46 - MORE THAN 256 BYTES?
318 020471 367 004 JZR LINSIZ JIF OK
319 020473 316 377 377 JSB =ERROR
320 020476 125 OCT 85D
321 020477 145 272 315 LINSIZ STB1 R45,=PTR2- insert into stmt
321 020502 377
322 020503 155 263 314 STMD R55,=PTR2 RESTORE PTR2
322 020506 377
323 020507 116 222 CLB R16 MUST BE IN IDLE!!!!
324 020511 316 377 377 JSB =LINEDR EDIT IN THE LINE
325 020514 *-----*
326 020514 * *
327 020514 * UPDATE AUTO LINE NUMBER IF APPLICABLE. NOTE *
328 020514 * THAT PROGRAM MODE PARSE IS THE ONLY TIME *
329 020514 * WHEN AUTO NUMBERING CAN BE ACTIVE *
330 020514 * IF THE LINE HAD ERRORS, JUST LEAVE AUTO# *
331 020514 * ALONE, IT CAN RESUME WHEN LINE CORRECT (?) *
332 020514 *
333 020514 *-----*
334 020514 130 251 236 SYSPRS LDM R30,=INPBUF FOR DCLIN#
334 020517 200
335 020520 117 220 TSB R17 ANY ERRORS?
336 020522 364 043 JNG P.PRTH JIF YES, DONT MESS
337 020524 165 261 106 LDMD R65,=AUTOI IN AUTO MODE?
337 020527 200
337 020530 367 035 JZR P.PRTH JIF NO
337 020532 231 BCD FOR BCD LINE NUMBERS
340 020533 323 103 200 ADMD R65,=AUTO# ADD CURRENT LINE#
341 020536 230 BIN FOR DECOMPILER
342 020537 374 001 JLZ PUTAUT JIF <= 999999
343 020541 223 CLM R65 ELSE MAKE IT ZERO
344 020542 *
345 020542 * DCLIN#: INPUT IN R65, OUTPUT ADDR IN R30
346 020542 *
347 020542 316 377 377 PUTAUT JSB =DCLIN# DECOMPILE LINE#
348 020545 126 251 236 LDMD R26,=INPBUF START FOR HLFLIN
348 020550 200
349 020551 130 026 305 SBM R30,R26 GET LENGTH OF LINE#
350 020554 036 243 STM R30,R36 MOVE FOR HLFLIN
351 020556 316 377 377 JSB =DECUR2 REMOVE CURSOR

```

```

=====
YARMA 03/14/81=====
ITEM   LOC   OBJECT CODE   SRC=GRGFX OBJ=CEXGFX   9/11/1981 11:11 AM PG 10
=====
>>>>>>      P.PARS, C.PARS, E.PARS                                     <<<<<<<<
020561 316 377 377      JSB =HLFLIN      SPIT OUT STRING
353 020564 316 377 377      JSB =CURS      SPIT OUT CURSOR
354 020567 236      P.PRTN RTN      PROGRAM PARSE DONE
355 020570
356 020570 316 377 377 ERR19Z JSB =ERROR+      MEMORY OVERFLOW
357 020573 023      OCT 19D
358 020574      *****
358 020574      *
359 020574      * INDENTATION: IF NO LEADING BLANKS ARE PRESENT THE
360 020574      * INDENTATION TOKEN IS SKIPPED
360 020574      *
360 020574      *****
361 020574      IND TOK EQU 367
362 020574
363 020574 130 010 241 INDENT LDM R30,R10      COPY R10
364 020577 213      DCM R30      BACK TO FIRST CHAR
365 020600 136 222      CLB R36      CLEAR COUNT
366 020602 135 030 342 CNTBLK POB R35,-R30      POP A CHAR
367 020605 310 040      CMB R35,=BLANK      IS IT A BLANK?
368 020607 366 004      JNZ INDEN2      JIF NO
369 020611 136 210      ICB R36      INCREMENT COUNT
370 020613 360 365      JMP CNTBLK      LOOP BACK
371 020615 136 310 002 INDEN2 CMB R36,=2      IS COUNT ZERO OR ONE?
372 020620 372 010      JNC INDEN3      JIF YES
373 020622 212      DCB R36      DECREMENT BY ONE
374 020623 137 250 367      LDB R37,=IND TOK      INDENTATION TOKEN
375 020626 136 273 315      STMI R36,=PTR2-      PUSH ON OUTPUT STK
376 020631 377
376 020632 236      INDEN3 RTN
377 020633
378 020633      *****
379 020633      *
380 020633      * C.PARS - CALCULATOR MODE PARSE
381 020633      *
382 020633      *****
383 020633 145 261 022 C.PARS LDMD R45,=NXTMEM      IS THERE ROOM?
383 020636 200
384 020637 263 341 203      STMD R45,=STSIZE      SAVE START LOCATION
385 020642 315 000 001      SBM R45,=0,1,0      NXTMEM - 256 > LAVAIL?
385 020645 000
386 020646 321 025 200      CMMD R45,=LAVAIL
387 020651 372 315      JNC ERR19Z      JIF NO
388 020653 152 223      CLM R52      CLEAR AUTO#, AUTINC
389 020655 263 103 200      STMD R52,=AUTO#      FOR ALL CALC MODE
390 020660 112 263 346      STMD R12,=OLDTOS      SAVE R12 START
390 020663 203
391 020664 116 220      TSB R16      STATE ACCEPTABLE?
392 020666 362 173      JOD ERR88      JIF NO
393 020670 210      ICB R16      MAKE R16 ODD
394 020671 316 134 040      JSB =PARSIT      TRY BASIC STMT PARSE
395 020674 117 220      TSB R17      SUCCESSFUL?
396 020676 365 113      JPS C.PDUN      JIF YES, DUN
397 020700 145 261 341      LDMD R45,=STSIZE      RESET POINTER 2

```

>>>> P.PARS, C.PARS, E.PARS

<<<<<<<

```

397 020703 203
398 020704 263 314 377      STMD R45,=PTR2
399 020707 112 261 346      LDMD R12,=OLDTOS      RESET R12
399 020712 203
400 020713
400 020713
401 020713
402 020713
403 020713
404 020713
405 020713
405 020713
405 020713
406 020713 112 263 346 E.PARS STMD R12,=OLDTOS      SAVE R12 START
406 020716 203
407 020717 145 263 341      STMD R45,=STSIZE      SET SIZE
407 020722 203
408 020723 117 317 077      ANM R17,=77          CLEAR ERROR BITS
409 020726 136 261 123      LDMD R36,=ERRORS      SAVE PARSE ERROR
409 020731 200
410 020732 006 345          PUMD R36,+R6          ON STACK
411 020734 142 261 114      LDMD R42,=ERLIN#      SAVE OLD ERLIN,NUM
411 020737 200
412 020740 345          PUMD R42,+R6          ON STACK
413 020741 136 261 063      LDMD R36,=SAVR10      PARSE POINTER
413 020744 210
414 020745 345          PUMD R36,+R6          ON STACK
415 020746 223          CLM R36          ZERO ERRORS
416 020747 263 123 200      STMD R36,=ERRORS      AND TRY FRESH
417 020752 110 251 236      LDM R10,=INPBUF      RESET INPUT BUFFER
417 020755 200
418 020756 316 105 042      JSB =SCAN+          DO INITIAL SCAN
419 020761 316 006 045      JSB =NUMVAL          TRY NUMERIC VALUE
420 020764 370 003          JEN EXPFND          JIF FOUND
421 020766 316 324 047      JSB =STREXP          TRY STRING EXPR
422 020771 134 006 343 EXPFND PUMD R34,-R6      RESTORE SAVR10
423 020774 172 343          PUMD R72,-R6      RESTORE ERLIN,NUM, ETC.
424 020776 136 343          PUMD R36,-R6      POP ERROR NUMBER
425 021000 371 015          JEZ ERRSTR          JIF EXPR NOT FOUND
426 021002 117 220          TSB R17          ANY PROBLEMS?
427 021004 364 011          JNG ERRSTR          JIF YES, RESTORE
428 021006 114 310 015      CMB R14,=CR          CARRIAGE RETURN?
429 021011 766 004          JNZ ERRSTR          JIF NO, REPORT ERROR
430 021013 316 262 040 C.PDUN JSB =EOL          GET END OF LINE
431 021016 236          RTN          DONE
432 021017 112 261 346 ERRSTR LDMD R12,=OLDTOS      RESET R12
432 021022 203
433 021023 145 261 341      LDMD R45,=STSIZE      RESET PTR2
433 021026 203
434 021027 263 314 377      STMD R45,=PTR2          TO START OF LINE
435 021032 136 263 123      STMD R36,=ERRORS      YES, RESTORE OLD ERROR
435 021035 200
436 021036 172 263 114      STMD R72,=ERLIN#      RESTORE OLD ERLIN,NUM

```

		P.PARS,	C.PARS,	E.PARS				
	021041	200						))))))
437	021042	134	263	063	STMD R34,=SAVR10	AND OLD SAVR10		
437	021045	210						
438	021046	251	000	300	LDM R34,=0,300	SET ERROR FLAG		
439	021051	262	030	377	STBD R34,=RSELEC	SELECT ROM 0		
440	021054	116	034	225	GRM R16,R34	IN R17		
441	021057	317	376	377	ANM R16,=376,377	KILL CALC MODE??		
442	021062				*IT WAS NOT REALLY A CALC MODE STATEMENT			
443	021062	236			RTN	DONE (WITH ERRORS)		
444	021063	316	377	377	ERR88 JSB =ERROR+	BAD STMT		
445	021066	130			OCT 380			
446	021067				*****PARSER ENDS HERE*****			
447	021067							
448	021067	116	222		ST240+ CLB R16	SET TO IDLE		
449	021071				ST240- BSS 0			
450	021071	136	006	344	SET240 PUBD R36,+6	SAVE REG		
451	021074	250	240		LDB R36,=240	OR THIS INTO		
452	021076	117	036	224	ORB R17,R36	XCOM (R17)		
3	021101	136	006	342	FOBD R36,-6	RESTORE 36		
4	021104	236			RTN	DONE		

```

=====
PARN 03/14/81=====
ITEM   LOC   OBJECT CODE  SRC=ORGFXX  OBJ=GE2GFX  9/11/1981 11:11 AM  PG 13
=====
>>>>  SCANNER  8-30-80                                     <<<<<<<<
454 021105          *****
456 021105          *
457 021105          * THE SCANNER SAVES R21-37 AND R50-57
458 021105          * IT USES REGISTERS 50-57
459 021105          * IF THE INPUT IS:          SCANNER DOES:
460 021105          * DIGIT                      FORMS INTEGER OR F.P. #
461 021105          * PERIOD                      "    F.P. NUMBER
462 021105          * QUOTE                      RETURNS 42, NO GCHAR
463 021105          * ANYTHING IN TABLES        RETURNS TOKEN
464 021105          * ALPHA NOT IN TABLES        RETURNS VARIABLE TYPE
465 021105          * OTHER NOT IN TABLS         RETURNS THE CHARACTER
466 021105          *-----*
467 021105          * SCANNER RETURNS:
468 021105          *          R40                FIRST CHAR SEARCHED
469 021105          *          R41,42             ROM # (R42=0)
470 021105          *                   OR...    BINARY PROG ADDR (R42#0)
471 021105          *                   OR...    SYSTEM "ROM" (R42=R43=0)
472 021105          *          R43                ROM TOKEN # (IF R14=370)
473 021105          *                   OR...    TYPE IF VARIABLE
474 021105          *          R44,45,46          PTR NAME IF VR5L, R46=LEN
475 021105          *                   OR...    INTEGER
476 021105          *                   OR...    SECONDARY ATTRIBS FOR FUNCS
477 021105          *          R47                CLASS (PRIMARY ATTRIB,)
478 021105          *
479 021105          * TOKEN RETURNS IN R14, R15=0 FOR ADDS MULTII
480 021105          * FLOATING POINT NUMBERS LEAVE RESULTS ON R12 STACK
481 021105          *
482 021105          *****
481 021105 316 236 043 SCAN+ JSB =GCHAR          FOR SOME FOLKS
482 021110 316 310 044 SCAN JSB =SAVREG          SAVE 21-37 AND 50-77
483 021113 370 006          JEN SCABRT          JIF NO ROOM
484 021115 316 132 042          JSB =SCAN++      NOW DO THE SCAN
485 021120 316 346 044          JSB =RSTREG      RESTORE 21-37, 50-77
486 021123 230          SCABRT BIN              RESTORE STATUS
487 021124 115 222          CLB R15             FOR ADM'S
488 021126 114          DRP 14                 FOR MOST PARSING
489 021127 036          ARP 36                  "
490 021130 235          CLE                      "
491 021131 236          RTN                      DONE
492 021132 117 220          SCAN++ TSB R17       ANY ERRORS?
493 021134 364 004          JNG SCAN+3          DONT SAVE R10 IF SO
494 021136 110 263 063          STMD 10,=SAVR10 SAVE LAST GOOD R10
494 021141 210
495 021142 140 223          SCAN+3 CLM R40        INITIALIZE FOR EVERYBODY
496 021144 262 030 377          STBD R40,=RSELEC ASSURE ROM 0
497 021147          *****
498 021147          * CHECK FOR NUMBERS OF THE FORM
499 021147          *   NNN.FF E +/- NN
500 021147          * OR   .FF E +/- NN
501 021147          *****
502 021147 316 000 043          JSB =NUMBER      IS IT A NUMBER?
503 021152 370 134          JEN SCRTN1         JIF YES (DONE)
504 021154          *****

```

```

CARMA 03/14/81=====
ITEM    LOC    OBJECT CODE  SRC=GRGFX OBJ=GENGFX  9/11/1981 11:11 AM  PG 14
=====
> >>>>>  SCANNER  8-30-80                                     <<<<<<<<

021154      *
506 021154      * NUMBER NOT FOUND, CALL TABLE SEARCH      *
507 021154      * WHICH WILL FIND ANY LEGAL WORD OR CHAR      *
508 021154      * EXCEPT VARIABLES                          *
509 021154      *
510 021154      *
511 021154 120 040 242 LBL_T? STB R20,R40          SAVE UNSHIFTED INPUT
512 021157
513 021157      * For quotes and CR, just push that char
514 021157      * for all other "special characters" just bomb
515 021157
516 021157 316 317 042      JSB =CK"CR          QUOTE OR CR?
517 021162 367 130      JZR QUOTCR          JIF YES
518 021164 316 327 043 GOSALT JSB =SALT          SEARCH ALL TABLES
519 021167 370 117      JEN SCRTN1          JIF FOUND, DONE
520 021171      *
521 021171      * =====MULTICHAR VARS AND LABELS=====
522 021171      *
523 021171 316 317 042 IDENT JSB =CK"CR          QUOTE OR CR?
524 021174 367 116      JZR QUOTCR          JIF YES
525 021176 371 111      JEZ BADSCH          JIF NO ALPHA
526 021200      *
526 021200      * *****
527 021200      * If the byte is a special character, and was
528 021200      * not found in the tables, it must be garbage.
529 021200      * This used to set R14 to R20, and caused a
530 021200      * problem with apostrophe looking like semicolon
531 021200      * CHANGED August 18, 1978
531 021200      *
531 021200      * *****
532 021200 110 044 243 GETVAR STM R10,R44          SAVE START OF NAME+1
533 021203 144 213      DCM R44          MOVE BACK ONE
534 021205 146 210      NMLOOP ICB R46          INCR LENGTH (R46 WAS 0!)
535 021207 310 040      CMB R46,=40          > 31?
536 021211 373 076      JCY BADSCH          JIF YES
537 021213 121 250 040      LDB R21,=40          SEE IF BLANK FOLLOWS
538 021216 010 330      CMBD R21,R10
539 021220 316 236 043      JSB =GCHAR          GET NEXT CHARACTER
540 021223 367 016      JZR VARTKN          IF YES, WE'VE GOT A VRBL
541 021225 316 256 043      JSB =ALFA          LETTER?
542 021230 370 353      JEN NMLOOP          ADD IT IF YES
543 021232 316 310 043      JSB =DIGIT          DIGIT?
544 021235 370 346      JEN NMLOOP          ADD IT IF YES
545 021237      DRP !20
546 021237 310 137      CMB R20,=137          UNDERLINE?
547 021241 367 342      JZR NMLOOP          JIF YES
548 021243 114 222      VARTKN CLB R14          For <letter> <blank>
549 021245 210          ICB R14          SET SIMPLE VARIABLE
550 021246 147 210      ICB R47          CLASS = 1 (R47 WAS 0!)
551 021250      *
552 021250      * ===== CHANGE ENDS =====
553 021250      *
554 021250 120 310 050      CMB R20,=OPEN          LEFT PAREN (ARRAY)?

```

```

PARMA 03/14/81=====
ITEM   LOC   OBJECT CODE   SRC=ORGFX  OBJ=GENGFX   9/11/1981 11:11 AM PG 15
=====
>>>>  SCANNER  8-30-80                                     <<<<<<<<
5 021253 367 024      JZR  ICB1      YES, TOKEN = 2
556 021255 310 044      CMB  R20,=DOLLAR  STRING VARIABLE?
557 021257 114          ORP  14
558 021260 366 024      JNZ  FNDVBL      NO, ARITH SIMPLE VRBL
559 021262          *      ICB  R14      STRING, INCREMENT TWICE
560 021262          *
561 021262          * CHANGE FOR STRING ARRAY
562 021262          *
563 021262 250 003      LDB  R14,=3      ASSUME SIMPLE STRING VAR
564 021264 316 236 043  JSB  =GCHAR      STEP OVER $
565 021267 120 310 050  CMB  R20,=OPEN  IS NEXT CHAR A (
566 021272 114          ORP  14
567 021273 366 011      JNZ  FNDVBL      JIF NOT
568 021275 250 040      LDB  R14,=40     IT IS A STRING ARRAY
569 021277 360 002      JMP  ICB1+      GOTO STEP OVER (
570 021301          *
571 021301          * CHANGE ENOS
572 021301          *
573 021301 114 210      ICB1  ICB  R14      AAY, INCREMENT ONCE
574 021303 316 236 043  ICB1+ JSB  =GCHAR      STEP OVER THE GARBAGE
575 021306          FNDVBL ORP  114
576 021306 043 242      STB  R14,R43      TYPE TO R43
577 021310 236          SCRTN1 RTN      ALL DONE
578 021311          *****
579 021311          *
579 021311          * Dont want GCHAR for QUOTES, dont need for CR
579 021311          *
580 021311          *****
581 021311 114 250 017  BADSCN LDB  R14,=17      ILLEGAL TOKEN
582 021314 014 242      QUOTCR STB  R#,R14      NO GCHAR AFTER "
583 021316 236          RTN      DONE
584 021317          *CLASS = 0 FOR CR AND QUOTES
585 021317
586 021317 316 256 043  CK"CR JSB  =ALFA      IS R20 A LETTER?
587 021322          ORP  120
588 021322 310 015      CMB  R20,=CR      CARRIAGE RETURN?
589 021324 367 002      JZR  FNDIT      JIF YES
590 021326 310 042      CMB  R20,=QUOTE  QUOTE?
591 021330 236          FNDIT  RTN
592 021331

```

>>>>> SCAN - NUMBERS <<<<<<<<

```

021331 *****
555 021331 *
596 021331 * INTEGER ROUTINE, ON RETURN R40 CONTAINS *
597 021331 * RIGHT JUSTIFIED INTEGER IF(!!) R36 = 15C *
598 021331 * - OTHERWISE IT CONTAINS INTEGER WITH AN *
599 021331 * EXPONENT OF (R36)-15C *
600 021331 * INTEGR IS INTENDED TO BE USED BY PARSER *
601 021331 * AND ANY OTHER PARTIES WANTING TO IGNORE *
602 021331 * DECIMAL POINTS AND EXPONENTS, SUCH AS *
603 021331 * SEQUENCE NUMBERS *
604 021331 *
605 021331 *****
606 021331 136 251 025 INTEGR LDM R36,=15C,0 SET UP EXPONENT
606 021334 000
607 021335 140 223 CLM R40 FOR NON-SCAN CALLS
608 021337 231 BCD FOR SHIFTS AND INCR
609 021340 122 222 CLB R22 INITIALIZE COUNT
610 021342 *R22 IS USED TO TELL IF ANY DIGITS SEEN
611 021342 316 310 043 MLOOP JSB =DIGIT WE HAVE A DIGIT?
612 021345 371 017 JEZ INTFND JIF NO
613 021347 122 210 ICB R22 COUNT UP
614 021351 147 220 TSB R47 HAVE 16 ALREADY?
615 021353 375 017 JLN M>16 JIF YES
616 021355 120 202 ERB R20 PUT DIGIT INTO E
617 021357 140 201 ELM R40 SHIFT INTO RESULT
618 021361 316 236 043 MGCHAR JSB =GCHAR GET NEXT CHAR
619 021364 360 354 JMP MLOOP GET MORE
620 021366 122 220 INTFND TSB R22 ANY FOUND?
621 021370 367 001 JZR INTRTH JIF NO
622 021372 234 ICE FLAG FOUND
623 021373 236 INTRTH RTN DONE
624 021374 136 211 M>16 ICM R36 STEP EXPONENT
625 021376 260 361 JMP MGCHAR GCHAR, LOOP
626 021400 *****
627 021400 *
628 021400 * NUMBER CREATES A FULL FLOATING POINT *
629 021400 * NUMBER IN R40, WITH EXPONENT IN R36 *
630 021400 * BY CALLING SHRONF, THIS NUMBER IS *
631 021400 * PACKED, NORMALIZED, AND ROUNDED *
632 021400 *
633 021400 *****
634 021400 100 222 NUMBER CLB R0 SET INTEGER FLAG
635 021402 316 331 042 JSB =INTEGR DO WE HAVE A NUMBER?
636 021405 120 310 056 CNB R20,=PERIOD A DECIMAL POINT?
637 021410 367 003 JZR DP1ST JIF JES
638 021412 370 037 JEN EXPON JIF DIGITS AND NO
639 021414 236 RTN NO DIGITS AND NO
640 021415 100 210 DP1ST ICB R0 KILL INTEGER FLAG
641 021417 *****
642 021417 *
643 021417 * DECIMAL POINT HAS BEEN SEEN, SHOVE *
644 021417 * REMAINING NUMBERS INTO R40, COUNTING *
645 021417 * DOWN THE EXPONENT. WE ARE SHIFTING THE *

```

```

>>>>>      SCAN - NUMBERS      <<<<<<<<
6 021417      * NUMBER LEFT SO THE EXPONENT MUST BE      *
647 021417      * COMPENSATED FOR POSITIONAL CHANGE      *
648 021417      * IF R40 FILLS UP, JUST THROW AWAY DIGITS *
649 021417      *                                          *
650 021417      * *****
651 021417 316 305 043 FLOOP JSB =GCDIG      THROW AWAY D.P.
652 021422 371 016 JEZ EXPON+      JIF NO MORE DIGITS
653 021424 122 210 ICB R22      ICR COUNT
654 021426 147 220 TSB R47      WE HAVE 16 NOW?
655 021430 375 365 JLN FLOOP      JIF YES
656 021432 136 213 DCM R36      COUNT DOWN EXP
657 021434 120 202 ERB R20      PUT DIGIT INTO E
658 021436 140 201 ELM R40      SHIFT INTO RESULT
659 021440 360 355 JMP FLOOP      LOOP IT
660 021442      * *****
661 021442      *
662 021442      * GET EXPONENT, IF ANY.      *
663 021442      * ONE PROBLEM WITH EXPONENTS IS THAT      *
664 021442      * THE CALCULATOR MODE STATEMENT:      *
665 021442      * 100E3 = 100000 (BOOLEAN EXPRESSION      *
666 021442      * WITH A VALUE OF TRUE)      *
667 021442      * IS INDISTINGUISHABLE FROM THE PROGRAM      *
668 021442      * MODE STATEMENT:      *
669 021442      * 100 E3=10000      *
670 021442      * THE SOLUTION IS TO DECIDE ARBITRARILY      *
671 021442      * IN FAVOR OF PROGRAM MODE. THE LOGIC TO      *
672 021442      * ACCOMPLISH THIS IS IN THE TEST FOR      *
673 021442      * EXPONENTS. AN EXPONENT IS AN "E"      *
674 021442      * FOLLOWED BY + OR - OR DIGIT. ANYTHING      *
675 021442      * ELSE TERMINATES THE SCAN.      *
676 021442      *
677 021442      * *****
678 021442 122 220 EXPON+ TSB R22      TEST DIGITS
679 021444 366 005 JNZ EXPON      JIF ANY
680 021446 114 250 056 LDB R14,=PERIOD      WE SKIPPED A DECIMAL
681 021451 234 ICE
682 021452 236 RTN      DONE
683 021453 316 256 043 EXPON JSB =ALFA      SET UPPER CASE
684 021456 021456 DRP 120
685 021456 310 105 CMB R20,=105      WE HAVE AN "E"?
686 021460 366 105 JNZ COMBINE      JIF NO
687 021462      * *****
688 021462      * SAVE STATE IN CASE E IS RESERVED WORD      *
689 021462      * *****
690 021462 130 010 241 LDM R30,R10      SAVE R10 POINTER
691 021465 126 020 240 LDB R26,R20      SAVE INPUT CHAR
692 021470 316 236 043 JSB =GCHAR      GET PAST E
693 021473 134 222 CLB R34      RESET SIGN FLAG
694 021475 122 223 CLM R22      CLEAR EXPONENT
695 021477 120 310 053 CMB 20,=PLUS      PLUS SEEN?
696 021502 367 023 JZR FNDE+      JIF + SEEN
697 021504 310 055 CMB 20,=MINUS      SEE -?
698 021506 367 015 JZR FNDE-      JIF YES
  
```

```

KARMA 03/14/81=====
ITEM   LOC   OBJECT CODE SRC=ORGFXX OBJ=GENGFX  9/11/1981 11:11 AM PG 18
=====
>>>>>      SCAN - NUMBERS                                     <<<<<<<<
021510 316 310 043      JSB =DIGIT      A DIGIT MAYBE?
700 021513 370 017      JEN FNDE      JIF VALID EXPONENT
701 021515      *****
702 021515      *THIS IS A CASE OF 234.5 ELSE (FALSE E) *
703 021515      *****
704 021515 110 030 241 FALSEE LDM R10,R30      RESTORE R10
705 021520 120 026 240      LDB R20,R26      RESTORE R20
706 021523 360 042      JMP COMBINE      NOW ITS AN INTEGER
707 021525 134 212      FNDE- DCB R34      SET SIGN FLAG
708 021527 316 305 043 FNDE+ JSB =GCDIG      GET PAST + OR -
709 021532      *If we don't see + or -, it isn't an exponent.
710 021532      *This allows SCAN not to flag errors.
711 021532 371 361      JEZ FALSEE      It ain't an EXP
712 021534 100 210      FNDE ICB R0      KILL INTEGER FLAG
713 021536 231      BCD      SHIFTS ARE BCD
714 021537 123 220      EXLOOP TSB R23      HAVE 3 DIGITS NOW?
715 021541 367 002      JZR SHFEXP      JIF NO
716 021543 250 001      LDB R23,=1      Set over(under)flow
717 021545      *Let SHRONF find over or underflow, not SCAN
718 021545 120 202      SHFEXP ERB R20      PUT DIGIT INTO E
719 021547 122 201      ELM R22      SHIFT DIGITS TO 22
720 021551 316 305 043      JSB =GCDIG      GET NEXT CHAR
721 021554 370 361      JEN EXLOOP      JIF DIGIT
722 021556 134 220      TSB R34      IS SIGN -?
723 021560 367 002      JZR ADMR36      JIF POSITIVE
724 021562 122 215      TCM R22      COMPLEMENT EXP
725 021564 136 022 303 ADMR36 ADM R36,R22      COMBINE EXPONENTS
726 021567 231      COMBINE BCD      FOR NO EXP CASE
727 021570 235      CLE      FLAG NUMBER
728 021571 234      ICE      FOUND
729 021572 100 220      TSB R0      POSSIBLE INTEGER?
730 021574 366 010      JNZ FIXFP#      JIF NO, . OR EXP
731 021576 143 221      TSM R43      ENOUGH ZEROES?
732 021600 366 004      JNZ FIXFP#      JIF NO
733 021602 142 220      TSB R42      ONLY 5 DIGITS?
734 021604 374 012      JLZ INTEG1      JIF YES, OK INTEGER
735 021606 132 222      FIXFP# CLB R32      SET SIGN TO POS
736 021610 316 377 377      JSB =SHRONF      SHIFT OUT 0'S, RND, PUSH
737 021613      *LEAVE THE ANSWER ON R12 STACK
738 021613 114 250 004      LDB 14,=4      FLOATING CONSTANT TOKEN
739 021616 360 010      JMP SETCL4      SET CLASS 4
740 021620 155 040 241 INTEG1 LDM R55,R40      MOVE UP
741 021623 044 243      STM R55,R44      SAVE IN R44,45,46
742 021625 114 250 032      LDB R14,=32      INTEGER TOKEN
743 021630 147 250 004 SETCL4 LDB R47,=4      CLASS = 4
744 021633 235      CLE      FLAG NUMBER
745 021634 234      ICE      FOUND
746 021635 236      RTN      DONE

```

```

>>>>      GCHAR, ALFA, DIGIT
748 021636 232      GCHAR      SAD      MUST RESPECT BIN/BCD
749 021637 230      BIN      NEED TO BE BINARY
750 021640 120 040 244 GCHI      LOBD 20,10      GET NEXT BYTE
751 021643 310 015      CMB 20,=CR      CARRIAGE RETURN?
752 021645 367 005      JZR GCHRTN      JIF YES
753 021647 340      POBD R20,+R10      INCREMENT R10
754 021650 310 040      CMB R20,=BLANK      SKIP BLANKS
755 021652 367 364      JZR GCHI      BY STARTING OVER
756 021654 237      GCHRTN      PAD      RESTORE STATUS
757 021655 236      RTN      OTHERWISE RETURN
758 021656
759 021656 230      ALFA      BIN      IF YOU CALL ALFA YOU WANT BIN
760 021657 235      CLE      PRESET "NOT FOUND"
761 021660 120 310 101      CMB 20,=101      >A?
762 021663 372 004      JNC NAM      NO
763 021665 310 133      CMB 20,=133      <I?
764 021667 372 012      JNC YEP      JIF YES
765 021671 310 141      NAM      CMB R#,=141      >SMALL A?
766 021673 372 007      JNC NOPE      NO
767 021675 310 173      CMB R#,=173      <SMALL Z+1?
768 021677 373 003      JCY NOPE      NOPE
769 021701 314 040      SBB R#,=40      CHANGE TO CAPITAL
770 021703 234      YEP      ICE      FOUND IT
771 021704 236      NOPE      RTN      DONE
772 021705
773 021705 316 236 043 GCDIG      JSB =GCHAR
774 021710 232      DIGIT      SAD
775 021711 230      BIN
776 021712 235      CLE
777 021713 120 310 060      CMB 20,=60      >"0"?
778 021716 372 005      JNC NO      NOPE
779 021720 310 072      CMB 20,=72      <"9"+1?
780 021722 373 001      JCY NO      NOPE
781 021724 234      ICE      FOUND
782 021725 237      NO      PAD      RETURN IN EITHER CASE
783 021726 236      RTN      DONE

```

>>>> TABLE SEARCH <<<<<<<<

```

021727 *****
785 021727 *
786 021727 * Entry: SAVR10 contains same R10 value
787 021727 * R10 points to the next character
788 021727 * R20 is first character of input
789 021727 *
790 021727 * SALT searches the BASIC statement ASCII table for
791 021727 * a legal BASIC statement type. If found, it returns
792 021727 * with E=0.
793 021727 *
794 021727 * SALT has been modified to search the ASCII tables
795 021727 * of up to 5 binary programs in addition to the ROM
796 021727 * and system tables.
796 021727 *
797 021727 *****
798 021727 THIRT2 DAD 32
799 021727 THIRT4 DAD 34
800 021727 THIRT6 DAD 36
801 021727
802 021727 110 030 243 SALT, STM R10,R30 SAVE INPUT R10
803 021732 134 251 073 LDM R34,=BINBAS BINARIES TABLE
803 021735 210
804 021736 136 250 005 LDB R36,=5 SET COUNTER
805 021741 122 034 341 BINLOP POMD R22,+R34 NEXT ADDRESS
806 021744 367 012 JZR NXTBIN JIF ENTRY EMPTY
807 021746 041 243 STM R22,R41 SET BINARY BASE
808 021750 265 034 000 LDMD R22,X41,THIRT4
809 021753 316 013 044 JSB =BPSALT SEARCH THIS BP'S TABLE
810 021756 370 023 JEN RTN JIF TOKEN FOUND
811 021760 136 212 NXTBIN DCB R36
812 021762 366 355 JNZ BINLOP JIF BP'S LEFT
813 021764 122 034 341 NXTROM POMD R22,+R34 GET NEXT TABLE POINTER
814 021767 041 243 STM R22,R41 SAVE ROM NUMBER
815 021771 311 377 001 CMM R22,=377,1 END OF LIST?
816 021774 367 006 JZR EXSALT JIF YES
817 021776 316 013 044 JSB =BPSALT SEARCH THIS ROM'S TBLE
818 022001 371 361 JEZ NXTROM JIF NOT FOUND
819 022003 236 RTN RTN
820 022004
821 022004 141 223 EXSALT CLM R41 SCRATCH BP/ROM NUMBER
822 022006 316 301 044 JSB =RS1020 RESET INPUT POINTERS
823 022011 235 CLE NOT FOUND
824 022012 236 RTN DONE
825 022013 114 223 BPSALT CLM R14 RESET TOKEN REGISTER
826 022015 123 220 TSB R23 A ROM?
827 022017 366 016 JNZ OUTERL JIF BINARY
828 022021 122 262 030 STBD R22,=RSELEC SELECT THAT ROM
828 022024 377
829 022025 261 004 140 LDMD R22,=R50+4 GET PARSE ADDRESS
830 022030 360 005 JMP OUTERL DO SEARCH
831 022032
832 022032 *****

```

<<<<<<<<

```

032 022032
033 022032
033 022032
033 022032
034 022032
035 022032 146 022 340 SKIP POBD R46,+R22 SKIP TO END OF
036 022035 365 373 JFS SKIP THIS ENTRY
037 022037 114 210 OUTERL ICB R14 BUMP ORDINAL
038 022041 316 301 044 JSB =RS1020 INITIALIZE INPUT
039 022044 147 020 240 INNERL LDB R47,R20 GET NEXT CHARACTER
040 022047 316 271 043 JSB =NAW SET UPPER CASE
041 022052 235 CLE UNDO WHAT NAW DID
042 022053
043 022053 * CMB R47,=141 IS IT LOWER CASE?
044 022053 * JNC SKBLNK JIF NO
045 022053 * CMB R47,=173 > SMALL Z?
046 022053 * JCY SKBLNK JIF YES
047 022053 * ANM R47,=337 SET UPPER CASE
048 022053
049 022053 146 022 340 SKBLNK POBD R46,+R22 GET NEXT CHAR FROM TABLE
050 022056 364 027 JNG LASTCH JIF LAST CHAR OF WORD
051 022060 310 040 CMB R46,=40 WAS IT A BLANK?
052 022062 366 011 JNZ NOTBLK JIF NO
053 022064 120 310 040 SKPBLK CMB R20,=BLANK BLANK IN INPUT ALSO?
054 022067 366 353 JNZ INNERL JIF NO
055 022071 010 340 POBD R20,+R10 GET NEXT CHARACTER
056 022073 360 367 JMP SKPBLK UNTIL # BLANK
057 022075 146 047 300 NOTBLK CMB R46,R47 IS THIS A MATCH?
058 022100 366 330 JNZ SKIP JIF NO
059 022102 * If no match, skip up to last character of this entry
060 022102 120 010 340 POBD R20,+R10 GET NEXT CHAR TO R20
061 022105 360 335 JMP INNERL DO INNER LOOP
062 022107
062 022107
063 022107 * The last character is special because it has a parity
064 022107 * bit stuck on. If R46 and R47 match, then the next inp
065 022107 * char will determine if table entry is found or not. *
066 022107 *
066 022107
067 022107
067 022107 LASTCH DRP 146
068 022107 310 377 CMB R46,=377 IS CHAR END OF TABLE?
069 022111 373 270 JCY RTN *** FOR MULTIPLE BINS ***
070 022113 204 LLB R46 CHOP OFF PARITY
071 022114 206 LRB R46 RESTORE WITH 0
072 022115 047 300 CMB R46,R47 COMPARE WITH INPUT
073 022117
074 022117 *
075 022117 * ===== CHANGE FOR MULTICHAR VARS =====
076 022117 366 070 JNZ TRLBLK JIF NO MATCH
077 022121 121 240 LDB R21,R47 SAVE R47 (=R20)
078 022123 316 310 043 JSB =DIGIT SEE IF DIGIT
079 022126 371 012 JEZ LSTAF? JIF NO

```

>>>>		TABLE SEARCH		<<<<<<<<	
022130	120 010 244	LDBD R20,R10		FETCH NEXT INPUT CHR	
381 022133	316 256 043	JSB =ALFA		SEE IF ALFA	
382 022136	370 277	JEN OUTERL		JIF YES, TRY IDENT	
383 022140	260 033	JMP HXTDG?		NEXT INPUT CHR DIGIT?	
384 022142	316 256 043 LSTAF?	JSB =ALFA		SEE IF PRESENT ALFA?	
385 022145	371 055	JEZ GOTVRB		JIF NO, TOKEN FOUND!	
386 022147		ORP 120			
387 022147	010 244	LDBD R20,R10		FETCH NEXT INPUT CHR	
388 022151	316 256 043	JSB =ALFA		SEE IF ALFA	
389 022154	371 017	JEZ HXTDG?		JIF NO	
390 022156	114 310 075	CMB R14,=75		LET FN - OFFSET (44)	
391 022161	367 041	JZR GOTVRB		JIF YES	
392 022163	310 143	CMB R14,=143		DEF FN - 44	
393 022165	367 035	JZR GOTVRB		JIF YES	
394 022167	310 175	CMB R14,=175		FN - 44	
395 022171	367 031	JZR GOTVRB		JIF YES	
396 022173	260 242	JMP OUTERL		NO FOUND	
397 022175	316 310 043 HXTDG?	JSB =DIGIT		DIGIT FOLLOWS?	
398 022200	370 235	JEN OUTERL		JIF YES	
399 022202	120 310 137	CMB R20,=137		UNDERLINE?	
900 022205	367 230	JZR OUTERL		JIF YES	
901 022207	360 013	JMP GOTVRB			
902 022211		*			
903 022211		+===== CHANGE ENDS =====			
904 022211		*			
905 022211	310 040	TRLBLK CMB R#,=BLANK		IF = 240 DONE	
906 022213	266 222	JNZ OUTERL		JIF NO GOOD	
907 022215	120 310 015	CMB R20,=CR		NEED TO DCR R10?	
908 022220	367 002	JZR GOTVRB		JIF NO (GCHAR DIDNT INR)	
909 022222	110 213	DCM R10		MAKE THAT GCHAR WORK	
910 022224		* -+--+--+MATCH HAS BEEN FOUND-+--+--+*			
911 022224		* ZRD SET, ROM# IN R41			
912 022224	235	GOTVRB CLE		FLAG FOUND	
913 022225	234	ICE		FLAG FOUND	
914 022226	122 041 241	LDM R22,R41		SYSTEM?	
915 022231	367 022	JZR SYSTOK		JIF YES	
916 022233		+NOTE: R15 = 0 FROM BEGINNING OF SALT			
917 022233	114 043 243	STM R14,R43		ROM'S TOKEN TO R43	
918 022236	250 370	LDB R14,=370		SET ROM TOKEN	
919 022240	123 220	TGB R23		BINARY PROG?	
920 022242	367 016	JZR ROMTOK		JIF ROM	
921 022244	114 210	ICB R14		BINARY PROG = 371	
922 022246	136 041 265	LDMD R36,X41 THIRT2		GET OFFSET FOR ROUTINE	
922 022251	032 000				
923 022253	260 011	JMP TOK.OK		CONTINUE	
924 022255	114 312 044 SYSTOK	ADB R14,=44		ADD SYSTEM OFFSET	
925 022260	043 243	STM R14,R43		SAVE 2ND COPY	
926 022262	136 261 002 ROMTOK	LDMD R36,=R50+2		ROM TABLE	
926 022265	140				
927 022266	043 303	TOK.OK ADM R#,R43		ADD OFFSET	
928 022270	303	ADM R#,R43		*2	
929 022271	036 245	LDMD R#,R36		GET START OF RUNTIME CODE	
930 022273	146 343	POMD R46,-R36		BACK UP TO ATTRIBS	

>>>>>		TABLE SEARCH		<<<<<<<
931	022275	316 236 043	USB =GCHAR	GET NEXT CHAR
932	022300	236	RTN	DONE
933	022301	120 040 240 RS1020	LDB R20,R40	RESTORE INPUT CHAR
934	022304	110 030 241	LDM R10,R30	RESET INPUT POINTER
935	022307	236	RTN	DONE

```

>>>>      SAVE, RESTORE REGISTERS                                     <<<<<<<<
022310      * THIS ROUTINE SAVES REGISTERS 21-37
938 022310      * REGISTER 70 NOT SAVED
939 022310 106 311 130 SAVREG CMM R5,=R6LIM1      ARE THERE 30 BYTES LEFT?
939 022313 205
940 022314 373 060      JCY ERR55      JIF OK
941 022316 102 006 343      POND R2,-R6      GET RETURN ADDR
942 022321 150 263 105      STMD R50,=ERSV50      SAVE R50
942 022324 202
943 022325 021 241      LDM 50,21      SAVE 21-31
944 022327 006 345      PUMD 50,+6
945 022331 030 241      LDM 50,30      SAVE 30-37
946 022333 006 345      PUMD 50,+6
947 022335 261 105 202      LDMD R50,=ERSV50      RESTORE R50
948 022340 160 345      PUMD R60,+R5      SAVE R60 TOO
949 022342 102 345      PUMD R2,+R6      RESTORE RETURN ADDR
950 022344 235      CLE      FLAG SAVE OK
951 022345 236      RTN      DONE
952 022346
953 022346 102 006 343 RSTREG      POND R2,-R6      GET RETURN ADDR
954 022351 160 343      POND R60,-R5      RESTORE R60
955 022353 150 263 105      STMD R50,=ERSV50      SAVE R50
955 022356 202
956 022357 343      POND 50,-6      RESTORE 30-37
957 022360 030 243      STM 50,30
958 022362 006 343      POND 50,-6      RESTORE 21,31
959 022364 021 243      STM 50,21      RESTORE 21,31
960 022366 102 006 345      PUMD 2,+6      TO STACK
961 022371 150 261 105      LDMD R50,=ERSV50      RESTORE R50
961 022374 202
962 022375 236      RTN
963 022376 316 164 045 ERR85      USB =ERR85X      EXPRESSION TO BIG
964 022401      *ERR85X DOES ERROR 85 AND CLE
965 022401 234      ICE      FLAG NOT FOUND
966 022402 236      RTN      DONE
967 022403      *WILL DO RETURN FROM CALLERS ROUTINE
968 022403      LETTOK EQU 142
969 022403      EQUALS EQU 65
970 022403      LBRACK EQU 336
971 022403      RBRACK EQU 337
972 022403      THEATO EQU 314
973 022403      JPSTOK EQU 33
974 022403      JMPTOK EQU 34
975 022403      FNSYM EQU 241
976 022403      ELSTOK EQU 327
  
```

```

>>>>>>      NUMERIC VALUE      <<<<<<<<
0000022403      *****
979022403      *
979022403      *RULES FOR PARSING EXPRESSIONS:
980022403      * 1. If you will accept either a numeric value
981022403      *    or a string, call NUMVAL first. Your are
982022403      *    guaranteed that:
983022403      *    A. If a NUMVAL found, E#0
984022403      *    B. If a NUMVAL not found, and some partial
985022403      *    numeric value found which is an obvious
986022403      *    error (e.g. SIN(X) AND nothing), an
987022403      *    error will be flagged and E=0.
988022403      *    C. If a NUMVAL is not found, and nothing
989022403      *    has been chewed up, the state will be
990022403      *    restored and E=0, no errors flagged.
991022403      *    R10,12,14,20,32-47 are restored.
992022403      *    You are then free to call STREXP.
993022403      * 2. When you call STREXP, no backup is done
994022403      *    on the assumption that you will not accept
995022403      *    the "not found" condition.
996022403      * + NUMVAL RESOURCES
997022403      * +    R15 TEMP
998022403      * <NUMERIC-VALUE> ::= <LOGICAL-EXP>
999022403      * <LOG-EXP>=<LOG-FACTOR> (<OR <LOG-FACTOR>)*
999022403      *
999022403      *****
1000022403 316 110 042 NUMVA+ JSB =SCAN      ,      ADDS SCAN TO NUMVAL
1001022406 316 044 045 NUMVAL JSB =LOGFAC      GET <LOG-FACTOR>
1002022411 371 150      JEZ PRIRTH      NONE FOUND
1003022413 146 311 002 LOGEX- CMM R46,=2,51      BINARY PREC 2?
1003022416 051
1004022417 266 142      JNZ PRIRTH      NO, DONE
1005022421 114 006 344      PUBD R14,+R6      SAVE TOKEN
1006022424 316 110 042      JSB =SCAN      SCAN OVER OR
1007022427 316 044 045      JSB =LOGFAC      GET 2ND <LOG-FAC>
1008022432 100 006 342      POBD R0,-R6      GET TOKEN
1009022435 272 315 377      STBI R0,=PTR2-      OR TO OUTPUT
1010022440 370 351      JEN LOGEX-      ITERATE
1011022442 360 034      JMP JERR81      Must find second one
1012022444
1013022444      *****
1013022444      *
1014022444      * <LOG-FACTOR>=<LOG-PRIM> (<AND <LOG-PRIM>)*
1014022444      *
1014022444      *****
1015022444 316 102 045 LOGFAC JSB =LOGPRIM      GET <LOG-PRIM>
1016022447 371 112      JEZ PRIRTH      NONE FOUND
1017022451 146 311 004 LOGFA- CMM R46,=4,51      BINARY PREC 4?
1017022454 051
1018022455 266 104      JNZ PRIRTH      NO, DONE
1019022457 114 006 344      PUBD R14,+R6      SAVE TOKEN
1020022462 316 110 042      JSB =SCAN      YES, SCAN OVER AND
1021022465 316 102 045      JSB =LOGPRIM      GET 2ND <LOG-PRIMARY>
1022022470 100 006 342      POBD R0,-R6      RESTORE TOKEN

```

```

=====
CARMA 03/14/81=====
ITEM   LOC   OBJECT CODE SRC=ORGFX OBJ=GEZGFX  9/11/1981 11:11 AM PG 26
=====
>>>>>      NUMERIC VALUE                                     <<<<<<<<
1023 022473 272 315 377      STBI R0,=PTR2-      TO OUTPUT
1024 022476 370 351      JEN LOGFA-      ITERATE
1025 022500 360 144      JERR81  JMP ERR91      Bad expression

```

```

>>>>> LOGICAL PRIMARY <<<<<<<
1025 022502 *****
1027 022502 *
1028 022502 * <LOGICAL-PRIM> = <NUMEXP>
1029 022502 * OR <NUMEXP><REL-OP><NUMEXP>
1030 022502 * OR <STRING-RELATION>
1031 022502 * <STRING-RELATION> ::= <STR-EXP><RELOP><STR-EXP>
1032 022502 * + LOGPRIM RESOURCES:
1033 022502 * + R36 TEMP
1034 022502 * This is another place where backup is required if
1035 022502 * this string expression is not followed by a relop
1035 022502 *
1035 022502 *****
1036 022502 106 311 130 LOGPRIM CMM R6,=R6LIM1 Enough room available?
1036 022505 205
1037 022506 373 054 JCY ERR55X JIF NO
1038 022510 110 006 345 PUMD R10,+R6 SAVE R10
1039 022513 155 261 314 LDMD R55,=PTR2 ALSO
1039 022516 377
1040 022517 345 PUMD R55,+R6 SAVE PTR2
1041 022520 114 344 PUBD R14,+R6 SAVE R14
1042 022522 120 344 PUBD R20,+R6 SAVE INPUT CHAR
1043 022524 140 345 PUMD R40,+R6 SAVE TOKEN REGISTER
1044 022526 316 324 047 JSB =STREXP GOOD STRING EXPRESSION?
1045 022531 371 037 JEZ NUMREL NONE FOUND, TRY NUMERIC
1046 022533 146 311 006 CMM R46,=6,53 BINARY STRING RELOP?
1046 022536 053
1047 022537 366 031 JNZ NUMREL RESTORE AND TRY NUM
1048 022541 114 006 344 PUBD R14,+R6 SAVE OPERATOR
1049 022544 316 321 047 JSB =STREXP+ GET SECOND EXPR
1050 022547 100 006 342 POBD R0,-R6 RESTORE R6
1051 022552 272 315 377 STBI R0,=PTR2- SEND IT OUT POSTFIX
1052 022555 371 013 JEZ NUMREL GUESSED WRONG, NUM
1053 022557 171 343 POND R71,-R6 REMOVE 7 BYTES FROM STACK
1054 022561 170 343 POND R70,-R6 " 8 "
1055 022563 236 PRIPTH RTN DONE
1056 022564 316 377 377 ERR55X JSB =ERROR EXPR TOO BIG
1057 022567 125 OCT 85D
1058 022570 235 CLE FLAG NOT FOUND
1059 022571 236 RTN
1060 022572 *****
1060 022572 *
1061 022572 * NOW DO NUMERIC RELATION
1062 022572 * NOTE THAT NUMEX IS CALLED, TO AVOID CONFUSION WITH
1063 022572 * STRING RELATION. (NUMVAL ALLOWS STRINGS)
1064 022572 * +NUMREL is a special purpose routine which backs
1065 022572 * up over characters clobbered by STREXP if necessary
1065 022572 *
1065 022572 *****
1066 022572 140 006 343 NUMREL POND R40,-R6 RESTORE TOKEN REGISTER
1067 022575 120 342 POBD R20,-R6 RESTORE INPUT CHARACTER
1068 022577 114 342 POBD R14,-R6 RESTORE R14
1069 022601 155 343 PUMD R55,-R6 RESTORE 2ND POINTER
1070 022603 263 314 377 STND R55,=PTR2
  
```

>>>>>		LOGICAL PRIMARY	<<<<<<<	
022606	110 343	POMD R10,-R6	RESTORE INPUT POINTER	
022610	316 344 061	JSB =ROMFX	GO SEL ROM	
022613	316 255 045	JSB =NUMEXP	GET FIRST COMPARAND	
022616	371 343	JEZ PRIRTH	RTH E=0 (BOMB LATER)	
022620	146 311 006	CMM R46,=6,53	BINARY REL OPERATOR?	
022623	053			
022624	766 335	JNZ PRIRTH	JIF NO OPS	
022626	114 312 010	ADB R14,=10	CHANGE TO NUMERIC OP	
022631	006 344	PUBD R14,+R6	SAVE THE OPERATOR	
022633	316 252 045	JSB =NUMEX+	SCAN, GET NEXT <NUM-EXP>	
022636	100 006 342	POBD R0,-R6	RESTORE R6	
022641	272 315 377	STBI R0,=PTR2-	PUSH TOKEN	
022644	370 315	JEN PRIRTH	JIF FOUND	
022646	316 377 377 ERR91	JSB =ERROR+	Bad expression	
022651	121	OCT 81D		
022652		* ALL RELATIONAL OPERATORS RETURN		
022652		* TRUE E = 1		
022652		* FALSE E = 0		

>>> NUMERIC EXPRESSION <<<<<<<

```

000 022652 *****
089 022652 *
090 022652 * <NUMERIC-EXPR> ::= <SIGN>?<TERMS>
091 022652 * Note: If <SIGN> found, <TERMS> is mandatory
092 022652 * If <SIGN> not found, <TERMS> is optional
093 022652 * This is to solve backup problem in the
094 022652 * case of PRINT -"HI". This MUST be an error.
095 022652 * + NUMEXP, NUMVAL RESOURCES: NONE (RESTORED)
096 022652 * + IN SUBROUTINES:
097 022652 * + R35 NUMEXP, R34 TERMS, R33 TERM, R32 FACTOR
097 022652 *
097 022652 *****
098 022652 316 110 042 NUMEX+ JSB =SCAN ADDS SCAN TO NUMEX
099 022655 152 032 241 NUMEXP LDM R52,R32 SAVE R32-R37
100 022650 006 345 PUMD R52,+R6 ON STACK
101 022662 135 222 CLB R35 PRESET "NO SIGN"
102 022664 146 311 007 CMM R46,=7,50 NOT?
102 022667 050
103 022670 367 024 JZR MONTRM JIF FOUND NOT
104 022672 114 310 055 CMB R14,=MINUS MONADIC MINUS?
105 022675 367 055 JZR MONAD- JIF YES
106 022677 *****
106 022677 *
107 022677 * <TERMS> ::= <TERM> (<+/-> <TERMS>)*
107 022677 *
107 022677 *****
108 022677 310 053 CMB R14,=PLUS MONADIC PLUS?
109 022701 367 047 JZR MONAD+ JIF YES
110 022703 316 360 045 TERMS JSB =TERM GET <TERM>
111 022706 371 034 JEZ NUMRST NOT FOUND, DONE
112 022710 146 311 007 TERMS- CMM R46,=7,51 BINARY OP PREC 7?
112 022713 051
113 022714 366 026 JNZ NUMRST JIF NO
114 022716 114 034 242 MONTRM STB R14,R34 SAVE OPERATOR
115 022721 316 110 042 JSB =SCAN Get next token
116 022724 316 360 045 JSB =TERM Get <TERM>
117 022727 134 220 TSB R34 MONADIC PLUS?
118 022731 367 003 JZR MONTR1 JIF YES
119 022733 272 315 377 STBI R34,=PTR2- Push token to output
120 022736 370 350 MONTR1 JEN TERMS- JIF found
121 022740 316 377 377 ERR31P JSB =ERROR BAD EXPRESSION
122 022743 121 OCT 81D FALL THRU TO NUMRST
123 022744 152 006 343 NUMRST FOMD R52,-R6 RESTORE R32-R37
124 022747 032 243 STM R52,R32 FOR RECURSION
125 022751 236 RET.NU RTN DONE
126 022752 250 365 MONAD+ LDB R#,=365 THEN 365+13=400 (0)
127 022754 312 013 MONAD- ADB R#,=13 CHANGE TO MONADIC -
128 022756 360 336 JMP MONTRM GET MONADIC TERM
129 022760 *****
129 022760 *
130 022760 * TERM = <FACTOR> (<MULTIPLIER> <FACTOR>)*
130 022760 *
130 022760 *****

```

KARMA 03/14/81=====

ITEM	LOC	OBJECT CODE	SRC=ORGFX	OBJ=GENGFX	9/11/1981 11:11 AM	PG 30
------	-----	-------------	-----------	------------	--------------------	-------

=====

		NUMERIC EXPRESSION			
1132	022760	316 014 046 TERM	JSB =FACTOR		
1133	022763	371 364	JEZ RET.NU		
1133	022765	146 311 012 TERM-	CMM R46,=12,51	BINARY OP PREC 12?	
1133	022770	051			
1134	022771	366 356	JNZ RET.NU	NO, RETURN	
1135	022773	114 033 242	STB R14,R33	SAVE OPERATOR	
1136	022776	316 110 042	JSB =SCAN	GET NEXT	
1137	023001	316 014 046	JSB =FACTOR	GET FACTOR	
1138	023004				
1139	023004	133 272 315	STBI R33,=PTR2-	PUSH TO OUTPUT	
1139	023007	377			
1140	023010	370 353	JEN TERM-	JIF found	
1141	023012	360 232	ERR31J JMP ERR81		

```

>>>> FACTOR <<<<<<<<
1143 023014 *****
1144 023014 *
1144 023014 * <FACTOR> ::= <PRIMARY> <<INVOLUTION> <FACTOR>)*
1144 023014 *
1145 023014 316 050 046 FACTOR JSB =PRIMARY
1146 023017 371 330 JEZ RET.NU
1147 023021 146 311 014 FACTO- CMM R46,=14,51 PWR OR SIMILAR OP?
1147 023024 051
1148 023025 766 322 JNZ RET.NU NO,RETURN
1149 023027 114 032 242 STB R14,R32 SAVE THE OP
1150 023032 316 110 042 JSB =SCAN GET NEXT
1151 023035 316 050 046 JSB =PRIMARY GET NEXT PRIMARY
1152 023040
1153 023040 132 272 315 STBI R32,=PTR2- ONTO OUTPUT
1153 023043 377
1154 023044 370 353 JEN FACTO- Iterate if found
1155 023046 360 342 JMP ERR91J Must find
1156 023050 *****
1157 023050 *
1158 023050 * <NUMERIC-PRIM> ::= <NUMERIC-VARIABLE>/<NUMERIC-REP>
1159 023050 * /<NUMERIC-FUNCTION-REF>/<OPEN> <NUMERIC-EXPR> <CLOSE>*
1160 023050 * + PRIMARY RESOURCES:
1161 023050 * + R36 TEMP
1162 023050 * <NUMERIC-FUNCTION-REF>::=<NUM-DEF-FUNC>
1162 023050 * ::= <NUM-SUPPLIED-FUNC>
1162 023050 *
1163 023050 *****
1163 023050 114 310 003 PRIMARY CMB R14,=3 NUMERIC SAV OR SNV?
1164 023053 372 047 JNC NUMVRB JIF YES (3-3 CY, 1/2-3 NC)
1165 023055 316 207 047 JSB =NUMREP NUMERIC REPRESENTATION?
1166 023060 370 027 JEN FNDPRI YES
1167 023062 114 310 241 CMB 14,=FNSYM FN?
1168 023065 367 023 JZR JFNCAL JIF YES
1169 023067 147 310 055 CMB R47,=55 <NUMERIC-SUPPLIED-FUNC>?
1170 023072 367 052 JZR FUHI JIF YES
1171 023074 114 310 050 CMB 14,=OPEN <NUM-EXP>?
1172 023077 766 010 JNZ FNDPRI NOPE, NONE FOUND,E=0
1173 023101 316 003 045 JSB =NUMVA+ SCAN, GET VALUE
1174 023104 371 304 JEZ ERR91J JIF bad open/close
1175 023106 316 050 047 JSB =GET) GET CLOSE
1176 023111 236 FNDPRI RTN
1177 023112
1178 023112 132 006 344 JFNCAL PUBD R32,+R6 SAVE OPERATOR
1179 023115 316 276 054 JSB =FNCALL GET THE CALL
1180 023120 132 006 342 POBD R32,-R6 RESTORE R32
1181 023123 236 RTN
1182 023124 006 344 NUMVRB PUBD R#,+R6 SAVE THAT TOKEN
1183 023126 316 076 051 JSB =PSH45 PUSH TOK, NAM
1184 023131 136 006 342 POBD R36,-R6 WAS IT SAV OR SNV?
1185 023134 362 353 JOD FNDPRI JIF SNV
1186 023136 316 123 047 JSB =SUBSCR GET SUBSCRIPTS
1187 023141

```

```
PARMA 03/14/81=====
ITEM    LOC    OBJECT CODE  SRC=ORGFX  OBJ=GEZGFX  9/11/1981 11:11 AM PG 32
=====
>>>>      FACTOR                                     <<<<<<<<
1 023141 136 272 315      STBI R36,=PTR2-      PUSH TO OUTPUT
1188 023144 377
1189 023145 236      RTN      DONE
```

GET STANDARD FUNCTION PARAMETER

))))))

191	023146	146 220	FUN1	TSB R46	ANY PARAMETERS?
192	023150	266 036	JNZ	GETPRM	JIF YES, GET THEM!
193	023152	114 272 315	STBI	R14,=PTR2-	PUSH INCOMING TOKEN
193	023155	377			
194	023156	310 370	CMB	R14,=370	ROM OR BINARY FUNCTION?
195	023160	372 021	JNC	NOTRBF	JIF NO
196	023162		*****		
196	023162		* ITS ROM OR THE BINARY PROGRAM (BPGM#)		
197	023162		*****		
197	023162		*****		
199	023162				
199	023162	142 220	TSB	R42	ROM OR BINARY?
200	023164	367 005	JZR	ITSROM	JIF ROM
201	023166	141 041 264	LDBD	R41,X41,BPGM#	
201	023171	007 000			
202	023173				
203	023173	141 272 315	ITSROM	STBI R41,=PTR2-	PUSH ROM# OR GARBAGE
203	023176	377			
204	023177	143 272 315	STBI	R43,=PTR2-	PUSH TOKEN WITHIN
204	023202	377			
205	023203	316 110 042	NOTRBF	JSB =SCAN	GET NEXT
206	023206	234		ICE	FLAG FOUND
207	023207	236		RTN	DONE WITH NO PARAMS
208	023210	230	GETPRM	BIN	FOR ADDS
209	023211	132 006 345	FUMD	R32,+R5	SAVE R32
210	023214	134 345	FUMD	R34,+R5	SAVE R34
211	023216	140 345	FUMD	R40,+R5	SAVE COPY OF R40
212	023220		* R40 contains ROM#, among other things		
213	023220	132 014 240	LDB	R32,R14	SAVE INCOMING TOKEN
214	023223		* R44 contains the doubled token		
215	023223	136 261 002	LDMD	R36,=R50+2	GET ROM RUNTIME
215	023226	140			
216	023227	142 220	TSB	R42	BINARY PROG?
217	023231	367 015	JZR	ROMTK.	JIF NO
218	023233		*****		
218	023233		* Change for Multiple Binaries		
219	023233		*****		
219	023233		*****		
220	023233				
221	023233	136 041 241	LDM	R36,R41	GET B.P. ADDRESS
222	023236	263 070 210	STMD	R36,=BINTAB	UPDATE BINTAB
223	023241	261 070 210	LDMD	R36,=BINTAB	GET TO RUNTIME (2 BYTES)
224	023244	036 265 032	LDMD	R36,X36 THIRT2	GET RUNTIME ADDR
224	023247	000			
225	023250	136 043 303	ROMTK.	ADM R36,R43	ADD OFFSET
226	023253	303	ADM	R36,R43	*2
227	023254	036 245	LDMD	R36,R36	GET ADDR OF ROUTINE
229	023256	213	DCM	R36	SKIP CLASS (ONLY 1 ICM)
229	023257	150 343	POMD	R50,-R36	GET ATTRIBUTES
230	023261	231	BCD		FOR SHIFTING BYTE
231	023262	205	LLM	R50	GET COUNT INTO E


```

>>>>> GET STANDARD FUNCTION PARAMETER
1280 023413 151 051 264          LDBD R51,X51,BPGM#          <<<<<<<<
1281 023416 007 000
1281 023420
1282 023420 151 272 315 ITSRM2  STBI R51,=PTR2-          ROM# OR GARBAGE (B.P.)
1282 023423 377
1283 023424 153 272 315          STBI R53,=PTR2-          ROM OR B.P. TOKEN
1283 023427 377
1284 023430 360 016          JMP GET)          DONE
1285 023432 150 006 343 POPFUN  POMB R50,-R6          RESTORE R50
1286 023435 140 343          POMB R40,-R6          RESTORE R40
1287 023437 134 343          POMB R34,-R6          RESTORE R34
1288 023441 132 343          POMB R32,-R6          RESTORE R32
1289 023443 235          ERR99  CLE          Clear found flag
1290 023444 316 377 377          JSB =ERROR+          Bad parameters
1291 023447 131          OCT 890
  
```

```

KARMA 03/14/81=====
ITEM   LOC   OBJECT CODE   SRC=ORGFX  OBJ=GENQFX   9/11/1981 11:11 AM  PG 36
=====
>>>>>  NARREF, GETMA, GET), SUBSCR                                     <<<<<<<<
023450 114 310 051 GET)      CMB 14,=CLOSE      END ) FOUND?
023453 367 041              JZR FNDCM?          JIF YES
023455
023455 316 377 377 ERR80     JSB =ERROR+        ) EXPECTED
023460 120                  OCT 80D
023461 316 110 042 NARRE+    JSB =SCAN          FOR SOME FOLKS
023464 235                  CLE                  Entry for FUN1 loop
023465 114 310 001 NARREF    CMB 14,=1
023470 366 351              JNZ ERR89           LOG ERROR AND CLEAR E
023472 210                  ICB 14             MAKE IT AN ARRAY
023473 316 076 051          JSB =PSH45
023476 236                  RTN
023477
023477 114 310 054 GETMA     CMB 14,=COMMA      Find ,?
023502 367 012              JZR FNDCM?          JIF YES
023504 316 377 377 ERR83X    JSB =ERROR+        ", " MISSING
023507 123                  OCT 83D
023510 235                  GETCM? CLE           RESET FLAG
023511 114 310 054          CMB 14,=COMMA      Find ,?
023514 366 004              JNZ NOCMA           JIF NO
023516 316 110 042 FNDCM?    JSB =SCAN          SCAN PAST IT
023521 234                  ICE
023522 236                  NOCMA RTN           RETURN WITH FOUND=E
023523
023523 *****
023523 *
023523 * + SUBSCR RETURN 1 OR 2 IN R36
023523 *
023523 *****
023523 SUBSCR JSB =NUMVAL      < WAS CHEWED UP
023526 371 313              JEZ ERR89           JIF not found
023530 136 250 013          LDB 36,=13         SET UP 1 SUBS
023533 114 310 054          CMB 14,=COMMA      TWO?
023536 366 310              JNZ GET)           Get close, RTN
023540 316 003 045 TUDSUB    JSB =NUMVA+        SCAN, GET EXPR
023543 371 110              JEZ ERR81           JIF not found
023545 136 210              ICB 36             TOKEN DOUBLE INDEX
023547 360 277              JMP GET)           Get close

```

```

>>>>>      NUMERIC REPRESENTATION      <<<<<<<<
328 023551      *
329 023551      * <NUMERIC-REP> = <SIGN> <SMALL-INTEGER>
330 023551      * <NUMERIC-CONST> = <SIGNIF> E <SIGN>? <INTEGER>
331 023551      * <SIGNIF> = <INTEGER> , / <INTEGER>? , <INTEGER>
332 023551      * + NUMREP RESOURCES: R36 TEMP
332 023551      *
333 023551 235      NUMCON CLE      FOR EXTERNAL CALLS
334 023552 114 310 004      CMB R14,=4      INTEGER?
335 023555 367 015      JZR FLTGP      JIF FLOATING CONST FOUND
336 023557 235      SMLINT CLE      FOR SMLINT CALLS
337 023560 114 310 032      CMB R14,=32      INTEGER?
338 023563 766 021      JNZ NUMRET      JIF NONE FOUND, E=0
339 023565 175 044 241      LDM R75,R44      GET COPY OF DIM-INDEX
340 023570 316 127 051      JSB =PUSH32      PUSH R14, R44, R45, R46
341 023573 236      RTN
342 023574 140 012 343 FLTGP      POMD R40,-12      THE NUMBER IS ON R12
343 023577 114 272 315      STBI R14,=PTR2-      PUSH OUT TOKEN
344 023602 377
344 023603 316 140 051      JSB =PUSH40      PUSH NUMBER AND SCAN
345 023606 236      NUMRET RTN      RETURN BYE BYE
346 023607
347 023607      *
348 023607      * NUMREP CHEWS UP LEADING + OR - AND PRODUCE
349 023607      * A SIGNED NUMERIC CONSTANT
350 023607 136 223      NUMREP CLM R36
351 023611 146 311 007      CMM R46,=7,51      SIGN?
351 023614 051
352 023615 367 006      JZR GETSGN      JIF YES, DEMAND NUMNUM
353 023617 316 151 047      JSB =NUMCON      GET OPTIONAL NUM
354 023622 370 035      JEN GETNMM      JIF FOUND
355 023624 236      RTN      OTHERWISE DONE
356 023625 114 310 053 GETSGN      CMB R14,=53      +?
357 023630 367 006      JZR SGN?1      JIF YES
358 023632 310 055      CMB R14,=55      -?
359 023634 766 007      JNZ SGNHUI      JIF NO
360 023636 137 210      ICB R37      INCR NEG COUNT
361 023640 316 110 042 SGN?1      JSB =SCAN      NEXT TOKEN
362 023643 760 360      JMP GETSGN      LOOP TO CHECK NEXT
363 023645 114 036 242 SGNHUI      STB R14,R36      SAVE TOKEN
364 023650 316 151 047      JSB =NUMCON      GO PROCESS NUMBER
365 023653      *CHEWED UP +/-'S WILL BE CHUCKED BY LOGPRI IF NEC.
366 023653 370 004      JEN GETNMM      JIF FOUND
367 023655 316 377 377 ERR31      JSB =ERROR+      MISSING PARAM
368 023660 133      OCT 910
369 023661 137 220      GETNMM TSB R37      NEG?
370 023663 763 033      JEY SGNEX      JIF NO
371 023665 136 310 032      CMB R36,=32      BCD INT
372 023670 766 013      JNZ SGNFLT      JIF NOT
373 023672 155 271 314      LDMI R55,=PTR2      POP VALUE
373 023675 377
374 023676 231      BCD
  
```

>>>>>				NUMERIC REPRESENTATION		<<<<<<<<	
1375	023677	215		TCM R55	MAKE IT NEG		
1376	023700	230		BIN			
1377	023701	273 314 377		STMI R55,=PTR2	PUSH IT BACK		
1378	023704	236		RTN	DONE		
1379	023705	150 271 314 3GNFLT		LDMI R50,=PTR2	POP VALUE		
1379	023710	377					
1380	023711	151 312 011		ADB R51,=9C	MAKE IT NEG		
1381	023714	150 273 314		STMI R50,=PTR2	PUSH IT BACK		
1381	023717	377					
1382	023720	236	3GNEX	RTN	DONE		

```

>>>>      STRING EXPRESSION      <<<<<<<<
*****
1384 023721      *
1385 023721      * <STRING-EXPR> ::= <SOURCE-STRING>
1386 023721      *      ::= <STRING-EXPR> <BINOP>
1387 023721      *      <SOURCE-STRING>
1388 023721      * + STREXP RESOURCES: R34,R36
1389 023721      * There are three possible return conditions for STREXP:
1390 023721      *   Successful return - E # 0
1391 023721      *   Error return - E = 0, ERRORS set
1392 023721      *   No string found - E = 0, input restored
1392 023721      *
*****
1393 023721 316 110 042 STREX+ JSB =SCAN      FOR SOME FOLKS
1394 023724 316 310 044 STREXP JSB =SAVREG    SAVE REGISTERS
1395 023727 371 002      JEZ DOSTR          JIF THERE WAS ROOM
1396 023731 235      CLE                    BAG IT, NO ROOM
1397 023732 236      RTN                    DONE
1398 023733 110 006 345 DOSTR PUMD R10,+R6    SAVE INPTR
1399 023736 120 344      PUBD R20,+R6        SAVE INPUT CHAR
1400 023740 316 016 050      JSB =SOURCE      GET 1ST <SOURCE-STRING>
1401 023743 370 007      JEN GET&           JIF FOUND
1402 023745 120 006 342      POBD R20,-R6     RESTORE INPUT CHAR
1403 023750 110 343      POMB R10,-R6        RESTORE INPTR
1404 023752 360 036      JMP STR,R+          FINISH UP
1405 023754      *!!!!!!IS THAT NECESSARY AND/OR SUFFICIENT??????
1406 023754      *SEE RULES FOR BACKUP IN FRONT OF NUMVAL
1407 023754 146 311 007 GET&  CMM R46,=7,53  BINARY STRING OP?
1407 023757 053
1408 023760 366 025      JNZ STR,RN          JIF NO, STILL OK
1409 023762 114 006 344      PUBD R14,+R6     SAVE OP
1410 023765 316 110 042      JSB =SCAN        SKIP OVER OP
1411 023770 316 016 050      JSB =SOURCE      GET NEXT SOURCE
1412 023773 136 006 342      POBD R36,-R6     RESTORE OP
1413 023776 272 315 377      STBI R36,=PTR2-  PUSH-TOKEN
1414 024001 370 351      JEN GET&           2ND OP MUST BE FOUND
1415 024003      *THIS IS ONLY REFERENCE TO THIS ERROR
1416 024003 316 377 377      JSB =ERROR        BAD STRING EXPRESSION
1417 024006 122      OCT 82D
1418 024007 175 006 343 STR,RN POMB R75,-R5    THROW AWAY R10,R20
1419 024012 316 346 044 STR,R+ JSB =RSTREG    RESTORE REGS
1420 024015 236      RTN                    DONE
1421 024016      *****
1421 024016      *
1422 024016      * <SOURCE-STRING> ::= <STRING-CONST>
1423 024016      *      ::= <STRING-FUNCTION>
1424 024016      *      ::= <STRING-REFERENCE>
1424 024016      *
*****
1425 024016 235      SOURCE CLE              PRESET NOT FOUND
1426 024017 114 310 042      CNB R14,=QUOTE  STRING CONST?
1427 024022 367 155      JZR STRCON          JIF YES
1428 024024 310 003      CNB R14,=3          STRING VRBL?
1429 024026 367 036      JZR STVRBL          JIF YES

```

	>>>>	STRING EXPRESSION		<<<<<<<<
024030		*		
431 024030		* CHANGE FOR STRING ARRAY		
432 024030		*		
433 024030 310 040		CMB R14,=40	STRING ARRAY	
434 024032 367 127		JZR STARAY	JIF YES	
435 024034		*		
436 024034		* CHANGE ENDS		
437 024034		*		
438 024034 147 310 056		CMB R47,=56	STRING FUNCTION?	
439 024037 366 004		JNZ USER#?	JIF NO	
440 024041 104 251 145		GTO FUN1	DO FUNCTION	
440 024044 046				
441 024045 114 310 241 USER#?		CMB R14,=FNSYM	USER STRING FUNCTION?	
442 024050 366 003		JNZ STXRTH	JIF NO, E=0	
443 024052 316 303 054		JSB =FN#CAL	TRY FN CALL, IF NUMERIC	
444 024055 236		STXRTH RTN	FN#CAL WILL RESTORE	

```

>>>>> STRING REFERENCE <<<<<<<<
1447 024056 * <STRING-VARIABLE>=<STR-VRBL-REP>
1448 024056 * OR <STR-VRBL-REP>[<NUMVAL>]
1449 024056 235 * OR <STR-VRBL-REP>[<NUMVAL>,<NUMVAL>]
1450 024057 114 310 003 STREF CLE PRESET FLAG
1451 024062 * CMB R14,=3 STRING VARIABLE?
1452 024062 * JNZ STXRTH JIF NO
1453 024062 * CHANGE FOR STRING ARRAY
1454 024062 *
1455 024062 766 057 JNZ STAREF JIF NO
1456 024064 *
1457 024064 * CHANGE ENDS
1458 024064 *
1459 024064 312 020 AOB R14,=20 CHANGE TO REFERENCE
1460 024066 316 076 051 STVRBL JSB =PSH45 PUSH NAME
1461 024071 235 STVRB+ CLE NOT FOUND
1462 024072 114 310 336 CMB 14,=LBRACK [<NUMVAL>?
1463 024075 766 040 JNZ NOSTR JIF NONE
1464 024077 316 003 045 JSB =NUMVA+ GET <NUM-EXP>
1465 024102 371 035 JEZ ERR32. JIF NONE FOUND
1466 024104 136 250 035 LDB 36,=35 SET SINGLE SUBS TOKEN
1467 024107 114 310 337 CMB 14,=RBRACK END OF LIST ?
1468 024112 367 020 JZR SUBDUN JIF DONE - R36 ALL SET
1469 024114 316 077 047 JSB =GETCMA NOW WE NEED ,<NUM-EXP>]
1470 024117 316 006 045 JSB =NUMVAL GET <NUM-EXP>
1471 024122 371 015 JEZ ERR32. JIF NONE FOUND
1472 024124 114 310 337 CMB 14,=RBRACK CHECK FOR ]
1473 024127 766 010 JNZ ERR32. JIF NO ]
1474 024131 136 250 036 LDB 36,=36 LOAD DOUBLE SUBS TOKEN
1475 024134 * ONE OR TWO PARAMETERS FOUND,
1476 024134 * PUSH APPROPRIATE SUBSTRING TOKEN
1477 024134 316 135 051 SUBDUN JSB =PU36SC PUSH 35 (1) OR 36 (2)
1478 024137 234 NOSTR ICE E:=1 OR 2
1479 024140 236 RTN DONE
1480 024141
1481 024141 ERR32. BSS 0
1482 024141 760 171 JMP ERR32Z RELAY JUMP
1483 024143
1484 024143 *
1485 024143 * CHANGE FOR STRING ARRAY
1486 024143 *
1487 024143 STAREF BSS 0 STRING ARRAY REFERENCE
1488 024143 114 310 040 CMB R14,=40 IS IT STRING ARRAY?
1489 024146 766 305 JNZ STXRTH JIF NOT
1490 024150 316 076 051 JSB =PSH45 PUSH NAME
1491 024153 316 123 047 JSB =SUBSCR GET SUBSCRIPT(S)
1492 024156 136 314 002 SBB R36,=2 MAKE IT ARRAY ADDRESS
1493 024161 760 006 JMP STARA+ GO DO MORE
1494 024163
1495 024163 STARAY BSS 0
1496 024163 316 076 051 JSB =PSH45 PUSH NAME
1497 024166 316 123 047 JSB =SUBSCR GET SUBSCRIPT(S)
1498 024171 136 272 315 STARA+ STBL R36,=PTR2- PUSH IT
  
```

KARMA 03/14/81=====

ITEM LOC OBJECT CODE SRC=ORGFX OBJ=GENGFX 9/11/1981 11:11 AM PG 42

>>>>> STRING REFERENCE

<<<<<<<

1 024174 377

149 024175 316 071 050

JSB =STVRB+

GO DO MORE

1500 024200 236

RTN

1501 024201

*

1502 024201

*

CHANGE ENDS

1503 024201

*

```

>>>> Remark and String Constant <<<<<<<
5 024201 *****
505 024201 *
506 024201 * The same routine is used for in-line remarks
507 024201 * and normal REM's. The appropriate token is
508 024201 * in R14. Any char but CR is allowed in remark.
509 024201 *
510 024201 * GUIDE TO ENTRY POINTS AND CALLS:
511 024201 * STRCON called by STREXP (SOURCE) and DATA
512 024201 * REMS- called by DATA
513 024201 * REM called by REM, IMAGE, and ! parses
514 024201 * UNQUOT is called by INPU$.
515 024201 * All others used on this page only
515 024201 *
516 024201 *****
517 024201 235 STRCON CLE Clear found flag
518 024202 114 310 042 CMB R14,=QUOTE A quoted string?
519 024205 766 147 JNZ NOCON# JIF no
520 024207 102 250 042 LDB R2,=QUOTE SET UP FOR QUOTED
521 024212 136 250 005 LDB R36,=5 STRING (TOKEN 5)
522 024215 760 043 JMP STRCN2 DO GENERAL STRING
523 024217 114 272 315 REM STBI R14,=PTR2- Push REM or IMAGE
523 024222 377
524 024223 102 250 015 LDB R2,=CR STOP ON CR
525 024226 020 300 CMB R2,R2 CR NOW?
526 024230 * JZR CRT014 Quit, make 14 CR
527 024230 367 002 JZR NODEC
528 024232 110 213 DCM R10 Compensate for GCHAR
529 024234 114 310 216 NODEC CMB R14,=216 IMAGE?
530 024237 367 130 JZR UNQUO- JIF YES
531 024241 *--PUT BACK LEADING BLANKS THAT GCHAR CHUCKED---
532 024241 120 DRP 20 SAVE LOOP TIME
533 024242 010 ARP 10 SAVE MOVE
534 024243 342 LEADBL POBD R#,-R# Check previous char
535 024244 310 040 CMB R#,-BLANK Is it a blank?
536 024246 767 373 JZR LEADBL JIF yes
537 024250 340 POBD R#,-R# Now go forward !
538 024251 340 POBD R#,-R# Check if there is
539 024252 310 040 CMB R#,-BLANK a leading blank
540 024254 *REM decompile will add a blank, so kill one
541 024254 367 001 JZR STRCN1 JIF yes, its gone
542 024256 342 POBD R#,-R# Undo the damage
543 024257 *If there is no leading blank, DCOMP will add one
544 024257
545 024257 136 250 006 STRCN1 LDB 36,=6 UNQUOTED STR
546 024262 235 STRCN2 CLE RESET FLAG
547 024263 *Is this CLE necessary?
548 024263
549 024263 272 315 377 STBI R#,-PTR2- Token (R36) to output
550 024266 155 261 314 LDMO R55,=PTR2 Get copy of this location
550 024271 377
551 024272 272 315 377 STBI R55,=PTR2- Reserve byte for length
552 024275 120 010 340 CHLOOP POBD R20,+R10 GET NEXT CHAR

```

		Remark and String Constant			
	>>>>				<<<<<<<<
554	024300 002 300	CMB 20,2		DONE?	
555	024302 367 011	JZR FNDCR-		YES	
556	024304 310 015	CMB =CR		LINE DONE?	
557	024306 367 011	JZR FNDCR		JIF yes, kill it	
558	024310				
559	024310 272 315 377	STBI R20,=PTR2-		STORE CHAR	
560	024313 360 360	JMP CHLOOP		GET MORE	
561	024315 310 042	FNDCR- CMB R#,=QUOTE		STOPPING ON QUOTE?	
562	024317 367 017	JZR ENDQUO		JIF YES	
563	024321 110 213	FNDCR DCM 10		BACK UP TO CR	
564	024323 102 310 042	CMB R2,=QUOTE		Looking for quote?	
565	024326 366 010	JNZ ENDQUO		JIF NO, ITS OK	
566	024330 110 263 063 ERR32S	STMD R10,=SAVR10		FOR ERROR MESSAGE	
567	024333 210				
568	024334 316 377 377 ERR32Z	JSB =ERROR+		BAD STRING EXPR	
569	024337 122	OCT 82D			
570	024340				
571	024340 165 055 241 ENDQUO	LDM R65,R55		GET COPY OF SIZE POS	
572	024343 325 314 377	SBMD R65,=PTR2		- CURRENT POSITION	
573	024346 213	DCM R65		SIZE IS ONE TOO BIG	
574	024347 316 020 051	JSB =UNQUO2		STORE STRING LENGTH	
575	024352 316 105 042 CRT014	JSB =SCAN+		GET NEXT TOKEN (E:=0)	
576	024355 234	ICE		SUCCESS FLAG!	
577	024356 236	NOCON\$ RTN		AND RETURN	
578	024357 120 310 015 REMS-	CMB R20,=CR		Null line?	
579	024362 367 366	JZR CRT014		JIF yes, set into R14	
580	024364 110 213	DCM R10		Back up input (BIN?)	
581	024366 102 250 054 UNQUOT	LDB R2,=COMMA		Stop on commas	
582	024371 316 257 050 UNQUO-	JSB =STRCH1		Do normal string	
583	024374				
584	024374				
585	024374				
586	024374				
587	024374				
588	024374				
589	024374				
590	024374				
591	024374				
592	024374				
593	024374 165 220	TSB R65		STR LEN = 0	
594	024376 367 020	UNQUOL JZR UNQUO2		JIF LEN 0	
595	024400 132 270 316	LOBI R32,=PTR2+		POP LAST	
596	024403 377				
597	024404 310 040	CMB R32,=BLANK		IS IT BLANK	
598	024406 366 004	JNZ UNQUO1		JIF NOT	
599	024410 165 212	DCB R65		DEC COUNT	
600	024412 360 362	JMP UNQUOL		AND LOOP	
601	024414 132 272 315 UNQUO1	STBI R32,=PTR2-		RESTORE LAST	
602	024417 377				
603	024420 175 261 314 UNQUO2	LDM R75,=PTR2		SAVE POSITION	
604	024423 377				
605	024424 155 263 314	STMD R55,=PTR2		SET TO \$ LENGTH	
606	024427 377				
607	024430 165 272 315	STBI R65,=PTR2-		STORE NEW COUNT	
608	024433 377				
609	024434 175 263 314	STMD R75,=PTR2		RESTORE	

PARMA 03/14/81=====

ITEM	LOC	OBJECT CODE	SRC=URGFXX	OBJ=GENGFX	9/11/1981	11:11 AM	PG 45
=====							
>>>>		Remark and String Constant				<<<<<<<	
150	024437	377					
1597	024440	236		RTN			
1598	024441						

PARSE TIME COMMON ROUTINES

CCCCCCCC

024441	114		PUSH1A	DRP	14	MOST USUAL CASE
601 024442	272	315 377	PUSH1#	STBI	R#,-PTR2-	PUSH OUT TOKEN
602 024445	316	110 042	SCANNN	JSB	=SCAN	GET NEXT
603 024450	234		ICE			SCAN DOES R14,R36,CLE
604 024451	236		RTN			DONE!!
605 024452	136		PU3637	DRP	36	PUSH 36,37+12
606 024453	360	001		JMP	PUSH##	LIKE OTHERS
607 024455	146		PUSH46	DRP	46	PUSH 46
608 024456	273	315 377	PUSH##	STMI	R#,-PTR2-	
609 024461	360	362		JMP	SCANNN	FINISH UP LIKE REST
610 024463	144	223	FNEND	CLM	R44	SEND FNEND, 3 ZEROES
611 024465	114	272 315	PUSH45	STBI	R14,-PTR2-	PUSH THE TOKEN
611 024470	377					
612 024471	155	044 241	PUSH44	LDM	R55,R44	PUSH R46, R45 & R44
613 024474	360	360		JMP	PUSH##	SCAN ON EXIT
614 024476						
615 024476			*			
616 024476			+=====CHANGE FOR MULTICHAR VARS =====			
617 024476			*			
618 024476	114	272 315	PSH45	STBI	R14,-PTR2-	PUSH THE TOKEN
619 024501	377					
620 024502	155	223	PSH44	CLM	R55	
621 024504	273	315 377		STMI	R55,-PTR2-	THREE BLANKS FOR ADDR
621 024507	146	272 315	PSH46	STBI	R46,-PTR2-	PUSH LENGTH
621 024512	377					
622 024513	121	044 340	PSHOUT	POBD	R21,+R44	GET ONE FROM BUFFER
623 024516	272	315 377		STBI	R21,-PTR2-	PUSH IT
624 024521	146	212		DCB	R46	
625 024523	366	366		JNZ	PSHOUT	JIF YES
626 024525	360	316		JMP	SCANNN	
627 024527						
628 024527			*			
629 024527			+===== CHANGE ENDS =====			
630 024527			*			
631 024527			*****			
631 024527			*			
632 024527			* PUSH32 IS FOR PUSHING INTEGERS (TOKEN 32)			
633 024527			* PUSH OUT R14, R44, R45 & R46, NOT R47			
634 024527			*			
634 024527			*			
634 024527			*****			
635 024527	114	047 242	PUSH32	STB	R14,R47	TO STORE MULTI
636 024532	144			DRP	44	PUSH 44,45,46,14
637 024533	360	321		JMP	PUSH##	FINISH UP
638 024535	136		PU363C	DRP	36	DO IT WITH 36
639 024536	360	302		JMP	PUSH1#	THAS ALL
640 024540	140		PUSH40	DRP	40	DO WITH 40
641 024541	360	313		JMP	PUSH##	PUND 40,+12
642 024543			*SECURE	JSB	=GETSTP	GET STRING EXPR
643 024543			*	JNZ	ERR83Y	JIF COMMA NOT FOUND
644 024543			*	LDB	R#,R#	RESTORE R14 FROM R36
645 024543	135	250 001	G#N	LDB	R35,=1	1 STRING, 1 NUMERIC
646 024546	316	256 051	GETPAS	JSB	=GETSTP	GET STRING EXPR

```

ARMA 03/14/81=====
ITEM   LOC   OBJECT CODE   SRC=ORGFXX OBJ=GEMGFX   9/11/1981 11:11 AM PG 47
=====
>>>>      PARSE TIME COMMON ROUTINES                                     <<<<<<<
649 024551 367 012      JZR GETPA-      JIF COMMA FOUND
649 024553 316 377 377 ERR33Y JSB =ERROR+      ", " MISSING
649 024556 123      OCT 83D
650 024557
651 024557 135 250 001 GET1N   LOB R35,=1      GET ONE PARAM
652 024562      *
652 024562      *
653 024562      *GETPAR GETS AS MANY PARAMS AS IT CAN,
654 024562      *RETURNS WITH COUNT IN 34 (DRP=34).
655 024562      *IF R35=0, ANY NUMBER OF PARAMS WILL BE OK
656 024562      *IF R35#0, NUMBER OF PARAMS MUST =R35
656 024562      *
656 024562      *
657 024562 114 036 242 GETPAR   STB R14,R36      SAVE INPUT TOKEN
658 024565 134 222      GETPA-   CLB 34      CLEAR COUNT
659 024567 316 003 045      JSB =NUMVA+      SCAN, GET NUM VAL
660 024572 371 016      JEZ P.DONE      JIF NOT FOUND
661 024574 134 210      GETCHT   ICB 34      COUNT UP
662 024576 114 310 054      CMB 14,=COMMA      Find separator?
663 024601 366 007      JNZ P.DONE      JIF NO
664 024603 316 003 045      JSB =NUMVA+      GET NEXT VALUE
665 024606 371 156      JEZ ERR91Y      THIS ONE MANDATORY
666 024610 360 362      JMP GETCHT      GET MORE
667 024612 136 272 315 P.DONE STBI R36,=PTR2-      PUSH TOKEN
667 024615 377
668 024616 135 220      TSB 35      REQUIRED NUMBER?
669 024620 367 004      JZR OKPAR      JIF NOT REQUIRED
670 024622 034 300      CMB 35,34      RIGHT NUMBER?
671 024624 366 127      JNZ ERR99Y      JIF UNEQUAL
672 024626 134      OKPAR   DRP 34      SET FOR CALLER
673 024627 236      RTN      DONE
674 024630 135 250 002 GET2N   LOB 35,=2      GET 2 NUMERIC
675 024633 360 325      JMP GETPAR      CALL GETPAR
676 024635 135 250 004 GET4N   LOB 35,=4      GET 4 NUMERIC
677 024640 360 320      JMP GETPAR      CALL GETPAR
678 024642 135 222      G$N+NN CLB R35      SET OPTIONAL PARAMETERS
679 024644 316 146 051      JSB =GETPA$      GET STRING, NUMERICS
680 024647 360 113      JMP G1.2++      NOW DID WE GET 1 OR 2?
681 024651 310 003      GXX3   CMB R#,=3      2-3NC,3-3CY
682 024653 373 100      JCYE89 JCY ERR89Y      JIF TOO MANY
683 024655 236      GETRTN   RTN      DONE
684 024656      *
684 024656      *
685 024656      *GETSTP IS FIRST PART OF MOST STRING "GETS"
686 024656      *REPORTS ERROR IF NO STRING EXPR FOUND, AND
687 024656      *RETURNS WITH DRP 14 FOR ALL AND "=COMMA" FOR SOME
687 024656      *
687 024656      *
688 024656 114 036 242 GETSTP   STB R14,R36      SAVE INPUT TOKEN
689 024661 316 321 047 GETST-   JSB =STREX+      GET STRING EXPR
690 024664 114 310 054      CMB R14,=COMMA      SATISFY MOST
691 024667 036      ARP 36      FOR SOME CALLERS
692 024670 370 363      JEN GETRTH      JIF STRING FOUND

```

```

PARMA 03/14/81=====
ITEM LOC OBJECT CODE SRC=ORGFX OBJ=GE%GFX 9/11/1981 11:11 AM PG 48
=====
>>>> PARSE TIME COMMON ROUTINES <<<<<<<<
1 024672 316 377 377 ERR32Y JSB =ERROR+ BAD STRING EXPR
1694 024675 122 OCT 82D
1695 024676
1696 024676 316 256 051 GET1$ JSB =GETSTP GET STRING EXP
1697 024701 136 272 315 G01NR STBI R36,=PTR2- SHOVE TOKEN
1697 024704 377
1698 024705 114 ORP 14
1699 024706 236 G1$RTN RTN DONE
1700 024707
1701 024707 +G012N and G01N get line #'s, not <numeric-values>
1702 024707 316 037 061 G012N JSB =SEQ#+ PUSH, SCAN, SEQ #?
1703 024712 371 365 JEZ G01NR JIF NO
1704 024714 316 127 051 JSB =PUSH32 PUSH#
1705 024717 114 310 054 CMB R14,=COMMA DEMAND COMMA
1706 024722 366 355 JNZ G01NR JIF NOT COMMA
1707 024724 036 240 LDB R14,R36 RESTORE TOKEN
1708 024726 316 037 061 G01N JSB =SEQ#+ PUSH, SCAN, GET SEQ #
1709 024731 371 346 JEZ G01NR JIF NONE?
1710 024733 316 127 051 JSB =PUSH32 PUSH #
1711 024736 360 341 JMP G01NR DONE
1712 024740
1713 024740
1714 024740 135 222 GETPA? CLB R35 SET OPTIONAL PARAMS
1715 024742 360 216 JMP GETPAR DO GETPAR
1716 024744
1717 024744 316 340 051 G00R2N JSB =GETPA? OPTIONAL PARAMS
1718 024747 371 304 JEZ GETRTH JIF NONE FOUND - OK
1719 024751 310 002 CMB R#, =2 Find 2?
1720 024753 367 300 JZR GETRTH JIF YES
1721 024755 316 377 377 ERR39Y JSB =ERROR+ BAD PARAMS
1722 024760 131 OCT 89D
1723 024761
1724 024761 316 340 051 G10R2N JSB =GETPA? OPTIONAL PARAMS
1725 024764 370 263 G1.2++ JEN GXX3 JIF FOUND
1726 024766 316 377 377 ERR31Y JSB =ERROR+ MISSING PARAM
1727 024771 133 OCT 91D
1728 024772
1729 024772 316 340 051 G120R4 JSB =GETPA? OPTIONAL PARAMS
1730 024775 310 004 CMB R#, =4 ARE THERE 4?
1731 024777 366 363 JNZ G1.2++ JIF NO, MUST BE 1 OR 2
1732 025001 236 RTN DONE IF 4
1733 025002 316 256 051 GET$N? JSB =GETSTP GET STRING EXPR
1734 025005 366 272 JNZ G01NR JIF NO COMMA
1735 025007 240 LDB R#,R# RESTORE R14
1736 025010 316 340 051 TRY1N JSB =GETPA? OPTIONAL PARAMS
1737 025013 310 002 CMB R#, =2 2-1 NC, 2-2 CY
1738 025015 360 234 JMP JCYE89 DO JCY ERR39Y

```

```

>>>>      ASSIGN, RENAME
1740 025017      TOTOK      EQU 244
1741 025017      TOTOK2    EQU 333
1742 025017      *ASSIGN    STB R14,R36      SAVE INPUT TOKEN
1743 025017      *          JSB =NUMYA+      GET NUMERIC
1744 025017      *          JEZ ERR91Y       JIF NOT FOUND
1745 025017      *          CMB R14,=TOTOK   ASSIGN - TO -?
1746 025017      *          JNZ ERR89Y       JIF NO
1747 025017      *          CMB R20,=STAR    ASSIGN TO * ?
1748 025017      *          JNZ RENASH       JIF NO
1749 025017      *CLOSEF    LDM R45,=52,1,6  UNQUOTED STRING
1750 025017      *          STMI R45,=PTR2-  PUSH IT
1751 025017      *          JSB =SCAN+       GCHAR, SCAN
1752 025017      *          JMP PUSHTO       PUSH "TO"
1753 025017      *JMP82Y    JMP ERR82Y       BAD STRING EXPR
1754 025017
1755 025017      *RENAME     JSB =GET1$      GET FROM NAME
1756 025017      *          DCM R12          MUST BE BINARY!
1757 025017      *          CMB R14,=TOTOK   Find "TO"?
1758 025017      *          JNZ ERR89Z       JIF NO
1759 025017      *RENASH     JSB =STREX+      SCAN, GET DEST
1760 025017      *JEZF82    JEZ JMP82Y       JIF NONE FOUND
1761 025017      *PUSHTO     LDB R37,=TOTOK   SHIFT SAVED TOKEN
1762 025017      *          STMI R36,=PTR2-  PUSH TO OUTPUT
1763 025017      *          TSB R16          CALC-MODE?
1764 025017      *          JEV RTHOK
1765 025017      *          GTO DMHDCR       JIF YES

```

```

>>> OPTION BASE AND ON KEY # <<<<<<<<
025017 316 041 051 OPTION JSB =PUSH1A PUSH OPTION TOKEN
768 025022 310 032 CMB R#,,=32 INTEGER FOUND?
769 025024 366 017 JNZ ERR51 JIF NO
770 025026 147 222 CLB R47
771 025030 145 221 TSM R45 CHECK IF 0 OR 1
772 025032 366 011 JNZ ERR51 JIF NO
773 025034 144 310 002 CMB R44,,=2 >1?
774 025037 373 004 JCY ERR51 JIF YES
775 025041 316 127 051 JSB =PUSH32 PUSH TOKEN AND #
776 025044 236 RTN
777 025045
778 025045 316 377 377 ERR51 JSB =ERROR+ OPTION BASE
779 025050 036 OCT 300
780 025051
780 025051
781 025051
781 025051
781 025051
*****
* ON KEY# *
*
*****
782 025051 114 034 242 KEYNAM STB R14,R34 SAVE ON KEY# TOKEN PART
783 025054 316 003 045 JSB =NUMVA+ SCAN & GET KEY#
784 025057 236 RTNOK RTN END ON-OFF KEY#
785 025060
786 025060 316 051 052 OFKEY# JSB =KEYNAM FIX UP NAME
787 025063 134 272 315 STBI R34,,=PTR2- PUSH OUT "OF KEY#"
787 025066 377
788 025067 236 RTN DONE
789 025070 316 230 051 ONTIME JSB =GET2N GET 2 NUMERICS
790 025073 360 032 JMP CKGOTO
791 025075 316 051 052 ONKEY# JSB =KEYNAM SET UP NAME
792 025100 371 176 JEZ ERR89Z JIF NO NUMVAL
793 025102 114 310 054 CMB R14,,=COMMA COMMA?
794 025105 235 CLE
795 025106 366 011 JNZ PUKTOK JIF NO, SEE IF GOTO
796 025110 316 261 051 JSB =GETST- GET 1ST STRING & COMMA
797 025113 235 CLE
798 025114 366 003 JNZ PUKTOK JIF FND, BUT NO COMMA
799 025116 316 261 051 JSB =GETST- GET 2ND STRING
800 025121 134 272 315 PUKTOK STBI R34,,=PTR2- PUSH OUT "ON KEY#"
801 025124 377
801 025125 370 330 JEN RTNOK JIF 3KEY DEFINITION
802 025127
803 025127 147 310 210 CKGOTO CMB R47,,=210 GOTO OR GOSUB?
804 025132 366 144 JNZ ERR89Z JIF NO
805 025134 316 317 060 TOGOTO JSB =GOTOSU GET GOSUB OR GOTO
806 025137 * JSB =DMNDCR DEMAND CARRIAGE RTN
807 025137 236 RTN DONE
808 025140
809 025140 316 041 051 ONERRO JSB =PUSH1A PUSH ON ERROR TOKEN
810 025143 360 362 JMP CKGOTO GET GOTO/GOSUB
811 025145

```

```

>>>> DIMENSION AND TYPE STATEMENTS <<<<<<<
181 025145 USTOK EQU 313
1814 025145 CRTOK EQU 354
1815 025145 CMATOK EQU 357
1816 025145 TABTOK EQU 315
1817 025145 STPKEY EQU 216 STOP KEY
1818 025145 BS EQU 225
1819 025145 CRKEY EQU 232
1820 025145 IN#TOK EQU 350
1821 025145 IN$TOK EQU 351
1822 025145 INPTOK EQU 137
1823 025145 REALTK EQU 231 REAL TOKEN
1824 025145 INTTOK EQU 177 INTEGER TOKEN
1825 025145 SHRTOK EQU 200 SHORT TOKEN
1826 025145
1827 025145
1827 025145
1828 025145 * <DIM-STMT>= DIM <DIM-LIST>
1828 025145 *
1828 025145 *
1828 025145 *
1829 025145 132 251 241 DIM LDM R32,=DIMITEM SET UP FOR JSB X32
1829 025150 052
1830 025151 360 004 JMP DIMCOM USE DIM/COM ROUTINE
1831 025153 132 251 213 COM LDM R32,=COMITEM SET UP FOR JSB X32
1831 025156 052
1832 025157 316 041 051 DIMCOM JSB =PUSH1A GET PAST COM/DIM TOKEN
1833 025162 032 306 000 DIMCM1 JSB X32,ZR0 JSB COMITEM OR DIMITEM
1833 025165 000
1834 025166 371 110 JEZ ERR89Z MUST FIND >=1!
1835 025170 316 110 047 DIMCOM? JSB =GETCM? GET POSSIBLE COMMA
1836 025173 370 365 JEN DIMCM1 JIF FOUND,ELSE DEMAND CR
1837 025175
1837 025175
1838 025175 * The following routine is used by anyone needing
1839 025175 * to demand CR or ! after the line.
1839 025175 *
1839 025175 *
1839 025175 *
1840 025175 114 310 233 DMNDCR CMB R14,=233 IN-LINE REM?
1841 025200 367 075 JZR DIMRTH JIF YES, OK
1842 025202 114 310 015 DMDCR- CMB R14,=CR LINE DONE?
1843 025205 367 070 JZR DIMRTH JIF YES
1844 025207 316 377 377 JSB =ERROR+ EXCESS CHARS
1845 025212 124 OCT 84D
1846 025213
1846 025213
1847 025213 * <DIM-LIST>= <DIM-ITEM>/<DIM-ITEM>,<DIM-LIST>
1848 025213 * <DIM-ITEM>=<NDIM-ITEM>/<SDIM-ITEM>
1849 025213 * <NDIM-ITEM>=<NUM-ARRAY><DIMENSIONS>
1850 025213 * <SDIM-ITEM>=<STRINGID> [ <DIMENSIONS> ]
1850 025213 *
1850 025213 *
1851 025213 114 310 177 COMITEM CMB 14,=INTTOK INTEGER?
1852 025216 367 010 JZR COMPU JIF SO
1853 025220 310 200 CMB 14,=SHRTOK SHORT?

```

```

KARMA 03/14/81=====
ITEM   LOC   OBJECT CODE SRC=ORGFXX OBJ=GENGFX  9/11/1981 11:11 AM PG 52
=====
>>>>>> DIMENSION AND TYPE STATEMENTS <<<<<<<<
1 025222 367 004 JZR COMPU JIF SO
1855 025224 310 231 CMB 14,=REALTK REAL?
1856 025226 766 003 JNZ COMSNV JIF NOT
1857 025230 316 041 051 COMPU JSB =PUSH1A
1858 025233 235 COMSNV CLE PRESET NOT FOUND
1859 025234 114 310 001 CMB 14,=1 SIMPLE N.V.?
1860 025237 367 047 JZR PU45CL JIF YES
1861 025241 DIMITEM BSS 0
1862 025241 235 SDIM CLE PRESET "NOT FOUND"
1863 025242 310 003 CMB R#, =3 <STRINGID>?
1864 025244 * JNZ TYPITH JIF NO
1865 025244 *
1866 025244 * CHANGE FOR STRING ARRAY
1867 025244 *
1868 025244 766 114 JNZ SAITH JIF NO
1869 025246 *
1870 025246 * CHANGE ENDS
1871 025246 *
1872 025246 316 076 051 JSB =PSH45 PUSH NAME TO OUTPUT
1873 025251 310 336 SDIM+ CMB R#, =LBRACK [ SYMBOL?
1874 025253 766 141 JNZ ERR91Z YES, OKOR
1875 025255 316 110 042 NOER92 JSB =SCAN GET NEXT TOKEN
1876 025260 316 001 053 JSB =SMLIN- GET INTEGER
1877 025263 371 013 JEZ ERR89Z JIF NONE
1878 025265 114 310 337 CMB R14, =RBRACK ] FOUND?
1879 025270 766 006 JNZ ERR89Z NOPE, ERROR
1880 025272 250 035 LDB R14, =35 1-DIM SUBSTRING
1881 025274 316 041 051 JSB =PUSH1A PUSH, SCAN, SET E
1882 025277 236 DIMRTN RTN DONE
1883 025300
1884 025300 316 377 377 ERR99Z JSB =ERROR+ INVALID PARAM
1885 025303 131 OCT 39D
1886 025304
1887 025304 310 001 NMVREL CMB R#, =1 SIMPLE NUMERIC?
1888 025306 766 004 JNZ TYPITH JIF NO
1889 025310 316 076 051 PU45CL JSB =PSH45 PUSH IT
1890 025313 236 RTN DONE
1891 025314
1892 025314 235 TYPITH CLE PRESET "NOT FOUND"
1893 025315 310 002 CMB R#, =2 ARRAY REFERENCE?
1894 025317 766 356 JNZ DIMRTN NO, MAYBE STRING
1895 025321 316 076 051 JSB =PSH45 PUSH NAME
1896 025324 316 001 053 SUBSCI JSB =SMLIN- GET FIRST DIMENSION
1897 025327 371 347 JEZ ERR89Z JIF NOT FOUND
1898 025331
1899 025331 * NOW GET DIMENSIONS (1 OR 2)
1900 025331 136 250 013 LDB R36, =13 PRESET FOR ONE DIM
1901 025334 316 110 047 JSB =GETCM? Find separator?
1902 025337 371 007 JEZ ONEDIM JIF NO
1903 025341 316 001 053 JSB =SMLIN- GET SECOND DIM
1904 025344 371 332 JEZ ERR89Z JIF NOT FOUND
1905 025346 136 210 ICB R36 CHANGE TO 2 DIM
1906 025350 136 272 315 ONEDIM STBI R36, =PTR2- PUSH THE DIM FLAG

```

DIMENSION AND TYPE STATEMENTS

<<<<<<<

```

907 025353 377
907 025354 316 050 047 JSB =GET) GET RIGHT PAREN
908 025357 236 RTN DONE
909 025360
910 025360 DMNDJP BSS 0
911 025360 760 213 JMP DMNDJR RELAY JUMP
912 025362
913 025362 *
914 025362 * CHANGES FOR STRING ARRAY
915 025362 *
916 025362 SAIMN BSS 0
917 025362 310 040 CMB R#,,=40 IS IT A STRING ARRAY
918 025364 766 326 JNZ TYPITM JIF NOT
919 025366 316 076 051 JSB =PSH45 PUSH NAME
920 025371 316 324 052 JSB =SUBSC! GET SUBSCRIPT(S)
921 025374 310 336 CMB R#,,=LBRACK [ SYMBOL?
922 025376 767 251 JZR SDIM+ JIF YES
923 025400 236 RTN
924 025401
925 025401 *
926 025401 * CHANGE ENDS
927 025401 316 157 047 SMLIN- JSB =SMLINT GET INTEGER
928 025404 231 BCD FOR COMPARE
929 025405 175 311 061 CNM R75,,=61,125,6 MORE THAN 64K-3?
929 025410 125 006
930 025412 373 061 JCY ERR89U JIF YES, BAD PARAM
931 025414 230 INDXOK BIN
932 025415 236 RTN
933 025416 316 377 377 ERR91Z JSB =ERROR+ MISSING PARAM
934 025421 133 OCT 91D
935 025422
936 025422 * <TYPE-STMT> ::= <TYPE-SPEC>
937 025422 * ::= <TYPE-SPEC> <NUM-VAR-LIST>
938 025422 * <TYPE-SPEC> ::= REAL / INTEGER / SHORT
939 025422 * <NUM-VAR-LIST> ::= <NUM-VAR-ELEM>
940 025422 * ::= <NUM-VAR-ELEM> <NUM-VAR-LIST>
941 025422 * <NUM-VAR-ELEM> ::= <SNV> / <FORMAL-ARRAY>
942 025422 316 041 051 TYPSTM JSB =PUSH1A PUSH THE TYPE
943 025425 316 304 052 JSB =NMVREL GET NUM-VAR-ELEM
944 025430 371 364 JEZ ERR91Z JIF DONE
945 025432 316 110 047 NMORE? JSB =GETCN? Find separator?
946 025435 371 321 JEZ DMNDJP JIF NO (DONE)
947 025437 316 304 052 JSB =NMVREL DEMAND ONE
948 025442 370 366 JEN NMORE? AFTER COMMA
949 025444 760 350 JMP ERR91Z

```

>>>>		TRACE VRBL					<<<<<<<<
952	025446	316	041	051	TRCVB	JSB =PUSH1A	PUSH TOKEN
953	025451	316	101	053		JSB =TRITEM	GET FIRST
954	025454	371	017			JEZ ERR89U	MUST BE ONE
955	025456	316	110	047	TRLOOP	JSB =GETCM?	COMMA
956	025461	371	005			JEZ TRCRTN	JIF NO
957	025463	316	101	053	PARPRM	JSB =TRITEM	GET NEXT
958	025466	760	366			JMP TRLOOP	AND LOOP
959	025470	236			TRCRTN	RTN	
960	025471	316	377	377	ERR88U	JSB =ERROR+	BAD STMT
961	025474	130				OCT 98D	
962	025475	316	377	377	ERR89U	JSB =ERROR+	INVALID PARAM
963	025500	131				OCT 99D	
964	025501	235			TRITEM	CLE	SET NOT FOUND
965	025502	114	310	001		CMB R14,=1	SNV
966	025505	367	050			JZR JPU45	JIF NO
967	025507	310	003			CMB R14,=3	STRING?
968	025511	367	044			JZR JPU45	JIF YES
969	025513	310	002			CMB R14,=2	ARRAY?
970	025515	766	005			JNZ CK\$AR	JIF NO
971	025517	251	363	001		LDM R14,=363,1	1-DIM, CORRECTN
972	025522	760	010			JMP ARDIMS	SEE IF COMMA
973	025524	114	310	040	CK\$AR	CMB R14,=40	STRING ARRAY?
974	025527	766	344			JNZ ERR89U	JIF NO
975	025531	251	273	005		LDM R14,=273,5	1-DIM, CORRECTN
976	025534	120	310	054	ARDIMS	CMB R20,=COMMA	
977	025537	766	006			JNZ ARRY1	JIF NOT COMMA
978	025541	114	015	302		ADB R14,R15	2-DIM
979	025544	316	236	043		JSB =GCHAR	NEXT CHAR
980	025547	120	310	051	ARRY1	CMB R20,=CLOSE	A() OR A(,) ELSE
981	025552	766	321			JNZ ERR89U	ERROR
982	025554	316	236	043		JSB =GCHAR	NEXT CHAR
983	025557	104	251	307	JPU45	GTO PU45CL	GO PUSH
984	025562	052					

```

>>>> DEF FN STATEMENT <<<<<<<
1984 025563 *****
1984 025563 *
1985 025563 * <DEF-STMT> ::= DEF FN<NUM-FUNC-HEAD'> = <NUM-VALUE> *
1986 025563 * ::= DEF FN<NUM-FUNC-HEAD'> *
1987 025563 * ::= DEF FN<STR-FUNC-HEAD'> = <STR-EXP> *
1988 025563 * ::= DEF FN<STR-FUNC-HEAD'> *
1989 025563 *NOTE: IN PUBLISHED SYNTAX, FN IS PART OF THE HEAD *
1989 025563 *
1989 025563 *****
1990 025563 116 220 DEF TSB R16 CALC?
1991 025565 362 302 JOD ERR88U ERROR IF YES
1992 025567 316 041 051 JSB =PUSH1A PUSH OUT DEF, SCAN
1993 025572 316 221 054 JSB =FNNAME GET NAME, TYPE
1994 025575 371 276 JEZ ERR99U JIF NOT FOUND
1995 025577 *****
1995 025577 *
1996 025577 * R31 NOW CONTAINS TYPE: *
1997 025577 * 0 (2) NUMERIC (WITH PARAMETERS) *
1998 025577 * 1 (3) STRING (WITH PARAMETERS) *
1998 025577 *
1998 025577 *****
1999 025577 155 261 314 LDMD R55,=PTR2 SAVE THIS POSITION
1999 025602 377
2000 025603 263 154 210 STMD R55,=SAVPT2
2001 025606 213 DCM R55
2002 025607 072 243 STM R55,R72 TO CONT IF NO PARS
2003 025611 126 223 CLM R26 AMOUNT OF RES. PAR.SPACE
2004 025613 272 315 377 STBI R26,=PTR2- SKIP PLACE FOR FNTYPE
2005 025616 106 263 066 STMD R6,=SAVER6 TO RESTORE IF ERRORS
2005 025621 210
2006 025622 131 310 002 CMB R31,=2 DO WE HAVE PARAMETERS?
2007 025625 373 004 JCY PRLOOP JIF YES
2008 025627 104 251 122 GTO GET= PUSH PAR#+FNTYPE, GET=
2008 025632 054
2009 025633 155 261 314 PRLOOP LDMD R55,=PTR2 MAKE POINTER LOCATION
2009 025636 377
2010 025637 325 022 200 SBMD R55,=NXTMEM REL TO START OF LINE
2011 025642 114 310 001 CMB R14,=1 NUMERIC PARAMETER?
2012 025645 366 017 JNZ STRPAR JIF NO
2013 025647 155 006 345 PUMD R55,+R6 SAVE IT
2014 025652 123 222 CLB R23 SET NUMERIC PARTYPE
2015 025654 344 PUBD R23,+R6 SAVE PARAMETER TYPE
2016 025655 316 107 051 JSB =PSH46 PUSH OUT LENGTH & NAME
2017 025660 126 313 014 ADM R26,=14,0 TYPE, REL.ADDR. & VALUE
2017 025663 000
2018 025664 360 101 JMP GETNOR MORE PARAMETERS?
2019 025666 310 003 STRPAR CMB R#,=3 STRING VARIABLE?
2020 025670 366 155 JNZ RSTR6 JIF NO
2021 025672 155 006 345 PUMD R55,+R6 SAVE REL (<0) PTR2
2022 025675 316 107 051 JSB =PSH46 PUSH OUT LENGTH & NAME
2023 025700 175 223 CLM R75
2024 025702 250 022 LDB R75,=22 DEFAULT STRING LENGTH
2025 025704 114 310 336 CMB R14,=LBRACK LENGTH GIVEN?

```

```

PARMA 03/14/81=====
ITEM   LOC   OBJECT CODE SRC=ORGFXN OBJ=GENGFX  9/11/1981 11:11 AM PG 56
=====
>>>> DEF FN STATEMENT                                     <<<<<<<<
2027 025707 366 032      JNZ  PUSHL$      JIF NO, DEFAULT
2027 025711 316 110 042   JSB  =SCAN      GET LENGTH
2028 025714      ORP  114
2029 025714 310 032      CMB  R14,=32      INTEGER?
2030 025716 366 127      JNZ  RSTR6      JIF NO
2031 025720 165 044 241   LDM  R65,R44      MOVE FOR BCDINT
2032 025723      *****
2032 025723      *
2033 025723      * Following check works for only one parameter
2034 025723      * CMM  R65,=310,20,0 MAX IS 231+24=255
2035 025723      * JCY  RSTR6      JIF TOO BIG
2035 025723      *
2035 025723      *****
2036 025723 316 377 377   JSB  =BCDINT      CONVERT BCD TO BIN
2037 025726 175 076 241   LDM  R75,R76      REPLACE DEFAULT LENGTH
2038 025731 177 222      CLB  R77      2 BYTES ONLY
2039 025733 120 310 135   CMB  R20,=RBRACK Find J?
2040 025736 366 107      JNZ  RSTR6      JIF NO
2041 025740 316 105 042   JSB  =SCAN+      SCAN PAST 32 AND J
2042 025743
2043 025743 175 211      PUSHL$ ICM  R75      INCLUDE ACTUAL
2044 025745 211      ICM  R75      IN TOTAL LENGTH
2045 025746 052 243      STM  R75,R52      COPY IT
2046 025750 155 006 343   FOMD R55,-R6      GET REL. ADDR.
2047 025753 152 345      FUMD R52,+R6      SAVE BOTH
2048 025755 123 250 200   LDB  R23,=200      SET STRING PARTYPE
2049 025760 344      PUBD R23,+R6      SAVE (STR) TYPE
2050 025761 126 313 011   ADM  R26,=11,0      TYPE,RA,TOTL,MAXL
2050 025764 000
2051 025765 075 303      ADM  R26,R75      ACTL & ROOM FOR STR ITSELF
2052 025767 130 210      GETMOR ICB  R30      INCREMENT PARAMETER COUNT
2053 025771 310 020      CMB  R30,=26      MORE THEN 16 PARAMS?
2054 025773 373 005      JCY  PARXCS      JIF YES
2055 025775 316 110 047   JSB  =GETCH?      GET COMMA, IF ANY
2056 026000 370 231      JEN  PRLOOP      JIF FOUND
2057 026002 316 050 047   PARXCS JSB  =GET)      GET CLOSE
2058 026005
2059 026005 130 023 242   STB  R30,R23      MAKE COPY OF PAR#
2060 026010 126 075 243   STM  R26,R75      PUT TOTLEN IN 3 BYTES
2061 026013 177 222      CLB  R77
2062 026015 155 261 314   LDMD R55,=PTR2      GET PTR1 LOCATION
2062 026020 377
2063 026021 305      SBM  R55,R75      ADVANCE BY TOTLEN
2064 026022 263 314 377   STMD R55,=PTR2      RESET IT
2065 026025 072 243      STM  R55,R72      SAVE THIS POSITION
2066 026027
2067 026027 154 006 342   PRTYPS POBD R54,-R6      GET TYPE
2068 026032 155 343      FOMD R55,-R6      GET REL. ADDR.
2069 026034 154 220      TSB  R54      NUM OR STR TYPE?
2070 026036 366 015      JNZ  STRTYP      JIF IF STRING
2071 026040 160 223      CLM  R60
2072 026042 273 316 377   STNI R60,=PTR2+      LEAVE ROOM FOR VALUE
2073 026045 360 040      JMP  PURA3T

```

>>>>		DEF FN STATEMENT		<<<<<<<	
2074	026047 106 261 066 RSTR6	LDMD R6,=SAVER6	RESTORE R6 STACK		
2075	026052 210				
2076	026053 260 135	JMP ERR89V	GO REPORT AN ERROR		
2077	026055	STRTYP ORP 154			
2078	026055 310 200	CMB R54,=200	STRING TYPE?		
2079	026057 266 131	JNZ ERR89V	JIF NO		
2080	026061 165 006 343	FOMD R65,=R6	GET TOTAL LENGTH		
2081	026064 075 243	STM R65,R75	SAVE IT		
2082	026066 323 314 377	ADMD R65,=PTR2	MAKE IT ABSOLUT		
2083	026071 263 314 377	STMD R65,=PTR2	SET POINTER		
2084	026074 166 241	LDM R66,R75	ACT LEN + 2		
2085	026076 213	DCM R66			
2086	026077 213	DCM R66			
2087	026100 273 316 377	STMI R66,=PTR2+	STORE THEM		
2088	026103 175 273 316	STMI R75,=PTR2+	STORE TOTAL LENGTH		
2089	026106 377				
2090	026107 155 273 316 FURA&T	STMI R55,=PTR2+	STORE RELATIVE ADDR.		
2091	026112 377				
2092	026113 154 272 316	STBI R54,=PTR2+	STORE (STR) TYPE		
2093	026116 377				
2094	026117 123 212	DECP# DCB R23	DECR. PARAMETER COUNT		
2095	026121 266 304	JNZ PRTYPS	JIF MORE		
2096	026123 155 261 154 GET=	LDMD R55,=SAVPT2	GET LOC FOR PAR#		
2097	026126 210				
2098	026127 263 314 377	STMD R55,=PTR2	SET POINTER		
2099	026132 130 317 377	ANM R30,=377,1	R30=COUNT, R31=0/1 (#/\$)		
2100	026135 001				
2101	026136 204	LLB R30	CLEAR LSB		
2102	026137 031 224	GRB R30,R31	PPPPPPPT		
2103	026141 272 315 377	STBI R30,=PTR2-	PAR# & FNTYPE (0/1)		
2104	026144 155 072 241	LDM R55,R72	FETCH SAVED PTR2		
2105	026147 263 314 377	STMD R55,=PTR2	RESTORE PTR2		
2106	026152 114 310 065	CMB R14,=EQUALS	IN LINE DEF?		
2107	026155 266 027	JNZ DEFRTN	JIF NO		
2108	026157 130 220	TSB R30	STRING OR NUMERIC?		
2109	026161 262 005	JOD RIGHT\$	JIF STRING		
2110	026163 316 003 045	JSB =NUMVA+	SCAN, GET NUM VAL		
2111	026166 260 003	JMP P.TAIL	GET TAIL		
2112	026170 316 321 047 RIGHT\$	JSB =STREX+	SCAN, GET STR EXPR		
2113	026173 371 015 P.TAIL	JEZ ERR89V	JIF EXPR NOT FOUND		
2114	026175 164 251 000	LDM R64,=0,0,0,352	PUSH STORE FN,		
2115	026200 000 000 352				
2116	026203 273 315 377	STMI R64,=PTR2-	ADDR SPACE, FEND		
2117	026206 316 175 052 DEFRTN	JSB =DMHDCR	DEMAND CR OR !		
2118	026211 236	RTN	TO OUTPUT - DONE		
2119	026212				
2120	026212 316 377 377 ERR89V	JSB =ERROR+			
2121	026215 131	OCT 890	INVALID PARAMETER		
2122	026216				

FUNCTION NAME SUBROUTINE

<<<<<<<<

```

026216 *****
0118 026216 *
0119 026216 *NOW GET THE (FN) NAME - POSSIBILITIES ARE
0120 026216 * TYPE      TOKEN      TYPE      TOKEN
0121 026216 * FNAB3=      1      FNAB3<      2
0122 026216 * FNAB3$=      3      FNAB3$(      3
0123 026216 *TOKEN IS 1,2, OR 3  FN IS NOT A KEY (1-8)
0124 026216 *FNNNAME PUSHES THE NAME (A3) WITH 3 ZEROES
0125 026216 * AND PUTS FOLLOWING MARKER IN R31
0126 026216 *      0 IF NUMERIC, NO PARAMS
0127 026216 *      1 IF STRING,  NO PARAMS
0128 026216 *      2 IF NUMERIC, WITH PARAMS
0129 026216 *      3 IF STRING,  WITH PARAMS
0129 026216 *
0130 026216 316 110 042 FNNAM+ JSB =SCAN      FOR FN CALLS
0131 026221 130 223 FNNAME CLM R30      CLEAR STRING FLAG + R31
0132 026223 135 014 240 LDB R35,14      SAVE TOKEN
0133 026226 112 263 344 STMD R12,=TOS      for RESTFN, if flot pt
0134 026231 203
0134 026232 316 102 051 JSB =PSH44      PUSH ADDR+LEN+NAME, SCAN
0135 026235 *      DRP !14
0136 026235 *      CMB R14,=4      Floating pt nr?
0137 026235 *      JNZ NOFLO      Jif no
0138 026235 *      POMO R50,-R12      Get that off stack!
0139 026235 155 223 NOFLO CLM R55      Save 3 bytes for
0140 026237 273 315 377 STMI R55,=PTR2- address?
0141 026242 235 CLE      Undo what SCAN did
0142 026243 135 310 040 CMB R35,=40      string func w/p?
0143 026246 366 004 JNZ NOSTRF      jif no
0144 026250 131 210 ICB R31      set string flag
0145 026252 360 015 JMP DEF.2
0146 026254 NOSTRF DRP !35
0147 026254 212 DCB R35      was it a 3NV?
0148 026255 367 010 JZR DEF.1      jif yes, numeric, no params
0149 026257 212 DCB R35      with params?
0150 026260 367 007 JZR DEF.2      jif yes, still numeric
0151 026262 212 DCB R35      string function?
0152 026263 366 003 JNZ FNNRTN      exit if no
0153 026265 131 210 ICB R31      SET STRING FLAG
0154 026267 234 DEF.1 ICE      SET FOUND FLAG
0155 026270 236 FNNRTN RTN      DONE
0156 026271 131 210 DEF.2 ICB R31      ADD 2 TO FLAG
0157 026273 210 ICB R31      FOR 1 PARAM
0158 026274 360 371 JMP DEF.1      GO PUSH FLAG
0159 026276

```

FUNCTION CALLS FNA(X), FNA\$(X)

<<<<<<<

```

2102 026276      *****
2161 026276      *
2162 026276      * Function calls must allow backup, since a string
2163 026276      * function call does not satisfy a NUMERIC-EXPR
2164 026276      * unless followed by a relational operator. That
2165 026276      * is the reason for saving so much on the stack.
2166 026276      * The other peculiarity of function calls is that
2167 026276      * the function type must first be determined, and
2168 026276      * all parameters must be outputted BEFORE the function
2169 026276      * name is pushed. The stuff down at EPLST is for
2170 026276      * functions with and without parameters, and pushes
2171 026276      * out the function name postfix, as well as parameter
2172 026276      * count and flags (200/000) for each parameter.
2173 026276      *-----
2174 026276      *Some additional comments 9-12-78:
2175 026276      *There is no such thing as an optional <STRING-EXP>
2176 026276      *but in the process of finding a <PRIMARY> it is
2177 026276      *reasonable to ask if we have here a string FN or
2178 026276      *a numeric FN. I'm not quite sure that we need
2179 026276      *backup here, but just in case it is included.
2180 026276      *Also note that FN$CAL is only called by STREXP
2181 026276      *which does a restore on R10, R20, and a RSTREG
2182 026276      *if not found.
2183 026276      * Internal Rep:
2184 026276      * <par1>...<parn> <FN-tok> <reladr> <len> <FNname>
2185 026276      * <#pars> <partyp1>...<partypn>
2186 026276      *
2187 026276      *****
2187 026276 132 250 026 FNCALL   LDB   R32,=26          FLAG NUMERIC
2188 026301 360 003      JMP   CALFF$          DO COMMON FN, FN$
2189 026303
2190 026303 132 250 027 FN$CAL   LDB   R32,=27          FLAG STRING
2191 026306 140 006 345 CALFF$   PUMD  R40,+R6          SAVE PARSE STATE
2192 026311 173 010 241      LDM   R73,R10          IN CASE WRONG FN TYPE
2193 026314 120 072 242      STB   R20,R72          SAVE INPUT CHAR
2194 026317 155 261 314      LDMD  R55,=PTR2         SAVE PTR2 POSITION
2194 026322 377
2195 026323 006 345      PUMD  R55,+R6          TO OVERWRITE NAME
2196 026325 316 110 042      JSB   =SCAN          SCAN
2197 026330 144 064 243      STM   R44,R64          COPY NAME ADR,LEN,ETC.
2198 026333 316 221 054      JSB   =FNNAME         PUSH NAME, SCAN
2199 026336 155 006 343      PUMD  R55,-R6          GET SAVED POSITION
2200 026341 263 314 377      STMD  R55,=PTR2         RESTORE PTR2
2201 026344 370 004      JEN   NAMEOK          JIF FOUND
2202 026346 104 251 157 JMPRES  GTO   RESTFN          ELSE GO RESTORE
2202 026351 055
2203 026352      *****
2203 026352      *
2204 026352      * If R31 is odd, string function found
2205 026352      * If R32 is odd, string function wanted
2206 026352      * so R32 XOR R31 is even if right type found
2206 026352      *

```

```

PARMA 03/14/81=====
ITEM   LOC   OBJECT CODE SRC=ORGFX OBJ=GENGFX  9/11/1981 11:11 AM PG 60
=====
>>>>> FUNCTION CALLS FNA(X), FNA$(X) <<<<<<<
1 026352 *****
2207 026352 *****
2208 026352 102 031 240 NAMEOK LDB R2,R31 MAKE COPY OF R31
2209 026355 032 226 XRB R2,R32 DO TYPES MATCH?
2210 026357 362 365 JGD JNPRES JIF NO, RESTORE
2211 026361 *****
2211 026361 *
2212 026361 * R60 = COPY OF R62
2213 026361 * R61 = FN-CALL TOKEN 26/27
2214 026361 * R62 = PARAMETER COUNT
2215 026361 * R63 = PARAMETER FLAG 000/200
2216 026361 * R64-67= LEN, NAME, ETC.
2216 026361 *
2216 026361 *****
2217 026361 161 240 LDB R61,R32 PUT 26 OR 27 IN R61
2218 026363 162 222 CLB R62 NO PARAMETERS YET
2219 026365 131 310 002 CMB R31,=2 2,3: PARAMS, 0,1: NONE
2220 026370 372 062 JNC EPLST JIF NO PARAMS
2221 026372 106 263 066 STND R6,=SAVER6 TO RESTORE IF ERRORS
2222 026375 210
2222 026376 163 222 PARLOP CLB R63 PRESET NUMERIC TYPE (0)
2223 026400 160 006 345 PUMD R60,+R5 SAVE R60
2224 026403 316 006 045 JSB =NUMVAL SCAN, GET NUMERIC
2225 026406 160 006 343 POMD R60,-R6 RESTORE R60
2226 026411 370 014 JEN GTMORE JIF PARAM FOUND
2227 026413 345 PUMD R60,+R6 STSAVE R60
2228 026414 316 324 047 JSB =STREXP GET STRING EXP
2229 026417 160 006 343 POMD R60,-R6 RESTORE R60
2230 026422 163 250 200 LDB R63,=200 CHANGE TO STRING (200)
2231 026425 371 121 JEZ ERR91X JIF NOT FOUND
2232 026427 162 210 GTMORE ICB R62 INCR PARAM COUNT
2233 026431 310 021 CMB R62,=21 > 16?
2234 026433 373 113 JCY ERR91X JIF YES
2235 026435 163 006 344 PUBD R63,+R5 SAVE TYPE FOR LATER
2236 026440 316 110 047 JSB =GETCN? COMMA?
2237 026443 370 331 JEN PARLOP JIF COMMA FOUND
2238 026445 316 050 047 NOPAR JSB =GET) GET CLOSE
2239 026450 370 002 JEN EPLST JIF FOUND
2240 026452 360 074 JMP ERR91X MISSING )
2241 026454 161 272 315 EPLST STBI R61,=PTR2- ADD 26 OR 27
2241 026457 377
2242 026460 155 223 CLM R55
2243 026462 273 315 377 STMI R55,=PTR2- 3 BYTES FOR ADDRESS
2244 026465 166 272 315 PULNGT STBI R66,=PTR2- PUSH LENGTH
2244 026470 377
2245 026471 121 064 340 FUNAME POBD R21,+R64 FETCH NEXT CHAR
2246 026474 272 315 377 STBI R21,=PTR2- PUSH IT OUT
2247 026477 166 212 DCB R66
2248 026501 366 366 JNZ FUNAME JIF ANY LEFT
2249 026503 162 272 315 STBI R62,=PTR2- PUSH PARAMETER COUNT
2249 026506 377
2250 026507 367 031 JZR NOPAR1 JIF ZERO PARAMS
2251 026511 155 223 CLM R55 FOR ADD MULTI

```

>>>>		FUNCTION CALLS FNA(X), FNA\$(X)		<<<<<<<<
252	026513 062 240	LDB	R55,R62	PARAM COUNT
253	026515 215	TCM	R55	MAKE IT NEGATIV
254	026516 323 314 377	ADMD	R55,=PTR2	ADVANCE POINTER
255	026521 263 314 377	STMD	R55,=PTR2	
256	026524 163 006 342 PUFLGS	POBD	R63,-R6	GET PARTYPE-FLAG
257	026527 272 316 377	STBI	R63,=PTR2+	PUSH IT OUT
258	026532 162 212	DCB	R62	
259	026534 766 366	JNZ	PUFLGS	JIF ANY LEFT
260	026536 155 263 314	STMD	R55,=PTR2	RESTORE POINTER PTR2
260	026541 377			
261	026542 235	NOPAR1	CLE	SET FOUND FLAG
262	026543 234		ICE	
263	026544 160 006 343	POMD	R60,-R6	CLEAN R6 (DONT KILL R40)
264	026547 236		RTN	DONE WITH FNCALLS
265	026550			
266	026550 106 261 066 ERR91X	LDMD	R6,=SAVER6	RESTORE STACK
266	026553 210			
267	026554 316 377 377	JSB	=ERROR	FN PARAM MISMATCH
267	026557 133	OCT	910	FALL THRU TO RESTORE
267	026560 235	RESTFN	CLE	FLAG NOT FOUND
270	026561 120 072 240	LDB	R20,R72	RESTORE NEXT CHAR
271	026564 173 010 243	STM	R73,R10	RESTORE R10
272	026567 140 006 343	POMD	R40,-R6	RESTORE R40
273	026572 112 261 344	LDMD	R12,=TOS	TRASH FP-NUMBER
273	026575 203			
274	026576 236		RTN	DONE
275	026577			

```

>>>>> INPUT STATEMENT PARSE <<<<<<<<
2 026577 *****
2277 026577 *
2278 026577 *
2279 026577 * <INPUT-STATEMENT> = INPUT <VARIABLE-LIST>
2280 026577 * <VARIABLE-LIST> = <VARIABLE>
2281 026577 * / <VARIABLE> , <VARIABLE-LIST>
2282 026577 * ANSI STANDARD BASIC REQUIRES THAT ALL INPUT BE
2283 026577 * CHECKED FOR ERRORS BEFORE STORING ANYTHING
2284 026577 * SO ITEMS ARE STACKED BY USING JSB =VRFND AFTER
2285 026577 * EACH ITEM, THEN A SERIES OF STORE STRING AND
2286 026577 * STORE NUMERIC TOKENS IS OUTPUT
2287 026577 *
2287 026577 *
2287 026577 *****
2288 026577 316 041 051 INPUT JSB =PUSH1A PUSH INPUT TOKEN
2289 026602 316 056 050 VLIST JSB =STRREF GET REF STRING VRBL IF ANY
2290 026605 371 011 JEZ NUMRF? JIF NOT FOUND
2291 026607 136 251 007 LDM R36,=7,351 INPUT STRING, STORE
2291 026612 351
2291 026613 273 315 377 STMI R36,=PTR2- TO OUTPUT
2293 026616 760 014 JMP VRFND
2294 026620 316 130 057 NUMRF? JSB =REFNUM GET NUMERIC REF IF ANY
2295 026623 371 130 JEZ JEZF09 OOPS, NO VARIABLES
2296 026625 136 251 010 LDM R36,=10,350 INPUT NUMERIC, STORE
2296 026630 350
2297 026631 273 315 377 STMI R36,=PTR2- TO OUTPUT
2297 026634 316 110 047 VRFND JSB =GETCM? MORE TO COME?
2297 026637 370 341 JEN VLIST LOOP IF SO
2300 026641 *To get past this point, E must be 0
2301 026641 234 ICE FLAG FOUND
2302 026642 236 RTN DONE
2303 026643

```

DATA Statement Parse

))))))

2300	026643	114	272	315	DATA	STBI R14,=PTR2-	PUSH DATA TOKEN
2305	026646	377					
2306	026647	110	006	345	DATLST	PUMD R10,+R6	SAVE CURRENT R10
2307	026652	120	344			PUBD R20,+R6	ALSO SAVE CHAR
2308	026654	316	110	042		JSB =SCAN	SCAN
2309	026657	316	207	047		JSB =NUMREP	SEE IF +,- NUMERIC
2310	026662	370	024			JEN DATCOM	JIF YES
2311	026664	316	201	050		JSB =STRCON	CIF QUOTED STRING
2312	026667	370	017			JEN DATCOM	JIF FOUND
2313	026671	114	310	015		CMB R14,=CR	
2314	026674	367	012			JZR DATCOM	JIF CR
2315	026676	120	006	342		POBD R20,-R6	RESTORE CHAR
2316	026701	110	343			POMD R10,-R6	RESTORE R10
2317	026703	316	357	050		JSB =REMS-	GET UNQUOTED STR
2318	026706	360	003			JMP DATCOM	
2319	026710	175	006	343	DATCOM	POMD R75,-R6	Trash old R10, R20
2320	026713	114	310	054	DATCOM	CMB R14,=COMMA	Find COMMA?
2321	026716	367	327			JZR DATLST	LOOP IF YES
2322	026720	316	202	052		JSB =DMCCR-	DEMAND CR OR !
2323	026723	236				RTN	DONE
2324	026724						
2325	026724				RSTL#	EQU 334	RESTORE LINE # TOKEN
2326	026724				RSTLB	EQU 335	RESTORE LABEL TOKEN
2327	026724	114	272	315	RESTOR	STBI R14,=PTR2-	PUSH TOKEN
2327	026727	377					
2328	026730	250	334			LDB R14,=RSTL#	
2329	026732	121	250	335		LDB R21,=RSTLB	TOKEN IF NO L# FOUND
2330	026735	316	344	060		JSB =GLN/LB	GET SEQ# OR LABEL
2331	026740	370	004			JEN RSTEX	JIF FOUND
2332	026742	177	270	316		LDBI R77,=PTR2+	TRASH TOKENR
2332	026745	377					
2333	026746				* RSTEX	JSB =SCAN	FETCH NEXT TOKEN
2334	026746	236				RTN	
2335	026747						

```

>>>>>      READ, READ# PARSE      <<<<<<<<
026747      RDNTK, EQU 346
026747      RDSTK, EQU 353
026747      RDA1P EQU 363
026747      READ3 EQU 364
026747 316 041 051 READ JSB =PUSH1A      push READ token
026752 316 372 055 REDLST JSB =REDITM      get next item
026755 371 122 JEZF89 JEZ ERR89X      jif yes, null
026757 316 110 047 JSB =GETCM?      Comma there?
026762 370 366 JEN REDLST      jif yes
026764 236 RTN      done if no comma
026765 316 034 056 RD#ITM JSB =FORMAR      find <formal-array>
026770 370 023 JEN REDRTN      jif found
026772 316 056 050 REDITM JSB =STRREF      String ref? (E:=0)
026775 122 250 353 LDB R22,=RDSTK,      Get string read token
027000 370 010 JEN PUSHRD      JIF string
027002 316 130 057 JSB =REFNUM      Get numeric
027005 371 006 JEZ REDRTN      JIF not found
027007 122 250 346 LDB R22,=RDNTK,      Get numeric read
027012
027012 272 315 377 PUSHRD STBI R#, =PTR2-      Push READ token
027015 236 REDRTN RTN      All done
027016 *READ# JSB =G1OR2N      1 OR 2 PARAMS
027016 * CMB 14,=SEMI      Semi separator?
027016 * JZR GOTSMI      JIF yes
027016 *DEMND2 CMB R34,=2      No semi, then demand 2
027016 * JNZ ERR89X      Wrong number of params
027016 * RTN      Position tape only
*****
*      Semi seen, get <read#-list>
*****
027016 316 041 051 GOTSMI JSB =PUSH1A      Push and SCAN
027021 316 365 055 RD#LST JSB =RD#ITM      Get <read#-item>
027024 371 053 JEZ ERR89X      JIF not found
027026 316 110 047 JSB =GETCM?      More to be found?
027031 370 366 JEN RD#LST      JIF yes, go get it!
027033 236 RTN      Done
027034 235 FORMAR CLE      <formal-array>
027035 114 310 002 CMB R14,=2      Array variable?
027040 366 016 JNZ FRMRTN      JIF no
027042 120 310 051 FORM$A CMB R20,=CLOSE      AC)?
027045 366 012 JNZ AWITH2      JIF no
027047 114 250 363 LDB R14,=RDA1P      READ# one dim
027052 316 236 043 PUSHR# JSB =GCHAR      GET PAST )
027055 316 076 051 JSB =PSH45      PUSH name, token
027060 236 FRMRTN RTN      E is now 1
027061 120 310 054 AWITH2 CMB R20,=COMMA      AC,)?
027064 366 372 JNZ FRMRTN      JIF no, done
027066 316 236 043 JSB =GCHAR      Get past comma
027071 114 250 364 LDB R14,=READ3      READ# two dims
027074 *Scanner sure to find comma, so throw away
027074 120 310 051 CMB R20,=CLOSE      Final )?

```

PARMA 03/14/81=====

ITEM	LOC	OBJECT CODE	SRC=ORGFX	OBJ=GEZGFX	3/11/1981	11:11 AM	PG 65
------	-----	-------------	-----------	------------	-----------	----------	-------

=====

>>>>>	READ, READ#	PARSE					<<<<<<<
027077	367	351	JZR PUSH#			Push if found	)

```

>>>>>> PRINT, PRINT USING, PRINT# PARSE
2 027101 316 377 377 ERR99X JSB =ERROR+ Duplicate?
2390 027104 131 OCT 29D
2391 027105 *PRINT# JSB =G1OR2N 1 OR 2 PARAMS
2392 027105 * CMB 14,=SEMI SEMI SEPARATOR
2393 027105 * JNZ DEMND2 If no semi, need 2
2394 027105 * STBD 14,=USING? SET USING TO FORCE
2395 027105 * END OF PRINT
2396 027105 * JSB =PUSH1A PUSH AND SCAN
2397 027105 * LDB R32,=377 Flag for PLIST
2398 027105 * JMP PLIST GET <print#-list>
2399 027105 *****
2399 027105 *
2400 027105 * PRINT STMT = PRINT <PRINTLIST>
2400 027105 *
2400 027105 *****
2401 027105 316 041 051 PRINT JSB =PUSH1A PUSH TOKEN TO OUTPUT
2402 027110 132 222 OLB R32 Flag plain PRINT
2403 027112 114 310 313 CMB R14,=USTOK PRINT USING?
2404 027115 366 042 JNZ PLIST JIF NO
2 027117 024 242 STB R14,R24 SAVE USING TOK
2406 027121 250 347 LDB R14,=ULNTOK USING line # token
2407 027123 121 250 345 LDB R21,=ULBTOK Option token
2408 027126 316 344 060 JSB =GLN/LB PUSH, GET L# OR LBL
2409 027131 370 014 JEN USEX1 JIF FOUND
2410 027133 272 316 377 STBI R#,=PTR2+ trash UL-tok & scan
2411 027136 316 324 047 JSB =STREXP Get in-line USING
2 027141 371 336 JEZ ERR99X JIF not there
2 027143
2414 027143 124 272 315 STBI R24,=PTR2- Push USING
2414 027146 377
2415 027147 132 210 USEX1 ICB R32 Flag PRINT USING
2416 027151 114 310 047 CMB R14,=SEMI Is there a PLIST?
2417 027154 366 114 JNZ EOPRT+ JIF no, done
2418 027156 316 041 051 JSB =PUSH1A Push and SCAN
2419 027161
2420 027161 ULNTOK EQU 347 USING LINE # TOKEN
2421 027161 ULBTOK EQU 345 USING LABEL TOKEN
2422 027161
2 027161 *****
2423 027161 *
2424 027161 * PRINT USING falls through to <print-list>
2425 027161 * <print-list> = <print-item>/
2426 027161 * /<print-item><print-sep><print-list>
2427 027161 *
2428 027161 * Register definitions:
2429 027161 * PRINT # R32 = -1
2430 027161 * PRINT USING R32 = 1
2431 027161 * PRINT R32 = 0
2432 027161 * String R33 = 0 (most common, so default)
2433 027161 * Numeric R33 = 1
2434 027161 * FORPAR flag R34
2435 027161 * Sep seen? R35
2 027161 *

```

```

CARNA 03/14/81=====
ITEM    LOC    OBJECT CODE SRC=ORGFXN OBJ=GEMGFX  9/11/1981 11:11 AM PG 67
=====
>>>>> PRINT, PRINT USING, PRINT# PARSE <<<<<<<<
2436 027161 135 222 *****
2437 027163 133 222 PLIST CLB R35 No separator seen
2438 027165 132 220 PLIST+ CLB R33 Preset numeric
2439 027167 367 016 TSB R32 PRINT#, PRINT USING?
2440 027171 365 021 JZR GETFUN JIF plain PRINT
2441 027173 JPS NOARRY JIF PRINT USING
2442 027173 316 034 056 *Fall through to PRINT# (allow <formal-array>)
2443 027176 371 014 JSB =FORMAR Get <formal-array>
2444 027200 114 310 047 JEZ NOARRY JIF not found
2445 027203 367 023 CMB R14,=SEMI PRINT#;?
2446 027205 360 036 JZR PRISEP JIF yes, make comma
2447 027207 147 310 045 GETFUN JMP PRCOM? Push and maybe SCAN
2448 027212 367 114 CMB R47,=45 Print function?
2449 027214 316 006 045 NOARRY JZR TABCAL JIF yes
2450 027217 371 004 JSB =NUMVAL <numeric-value>?
2451 027221 133 210 JEZ PSTRIH JIF no
2452 027223 360 003 ICB R33 Flag numeric
2453 027225 316 324 047 PSTRIH JMP PRISEP Now do separator
2454 027230 JSB =STREXP <string-exp>?
2455 027230 136 250 357 PRISEP *On return, if E = 0, element was <empty>
2456 027233 033 304 LDB 36,=CMATOK Put ",", token in R36
2457 027235 304 SBB R36,R33 Subtract 2
2458 027236 114 310 047 SBB R36,R33 if numeric
2459 027241 367 036 CMB 14,=SEMI <print-sep> = ","?
2460 027243 136 210 JZR DELIM- JIF yes
2461 027245 114 310 054 PRCOM? ICB 36 Change R36 to comma
2462 027250 367 035 CMB 14,=COMMA <print-sep> = ","?
2463 027252 371 006 JZR DELIM JIF yes
2464 027254 JEZ EOPRT JIF no item seen
2465 027254 136 272 315 STBI R36,=PTR2- Push comma token
2466 027257 377 JMP EOPRT+ No sep checking
2467 027262 132 220 EOPRT TSB R32 USING?
2468 027264 366 004 JNZ EOPRT+ FORCE EOL FOR USING
2469 027266 135 220 TSB R35 Any separators seen?
2470 027270 366 006 JNZ PLEND JIF yes, no EOL
2471 027272 136 250 354 EOPRT+ LDB R36,=CRTOK Terminate line
2472 027275 STBI R36,=PTR2- Push EOL token
2473 027275 272 315 377 RTN and return
2474 027300 236 PLEND PRINT# ??
2475 027301 132 220 DELIM- TSB R32
2476 027303 365 002 JPS DELIM JIF NO
2477 027305 136 210 ICB R36 MAKE COMMA TOKEN
2478 027307 370 007 DELIM JEN DELIM+ JIF item seen
2479 027311 130 251 000 LDM R30,=0,6 Empty unquoted
2480 027314 006 STMI R30,=PTR2- string to output
2481 027315 273 315 377 JSB =PU36SC FOUND ONE, PUSH IT
2482 027320 316 135 051 DELIM+ LDB R35,=1 Set delimiter seen
2483 027323 135 250 001 JMP PLIST+ Iterate <print-list>
2484 027330 TABCAL JSB =FUN1 Get function
2485 027330 316 146 046

```

```
ARMA 03/14/81=====
ITEM   LOC   OBJECT CODE SRC=ORGFXX OBJ=GENGFX  9/11/1981 11:11 AM PG 68
=====
>>>> PRINT, PRINT USING, PRINT# PARSE <<<<<<<
2 027333 235 CLE Force null string
2407 027334 360 272 JMP PRISEP Done
2488 027336 *
2489 027336 * IMAGE - (SEE REM)
2490 027336 *
```

```

CARMA 03/14/81=====
ITEM   LOC   OBJECT CODE SRC=ORGFXN OBJ=GENGFX  9/11/1981 11:11 AM PG 69
=====
>>>> LET STATEMENT 11-1-76 <<<<<<<
2492 027336 * <LET-STATEMENT>=
2493 027336 * LET? <STRING-REFERENCE>=<STRING-EXPRESSION>
2494 027336 * /LET <NUMERIC-VARIABLE> ( , <NUMERIC-VARIABLE>)*
2495 027336 * = <NUMERIC-VALUE>
2496 027336 114 043 240 ILET LDB R14,R43 RESTORE VARIABLE TYPE
2497 027341 360 003 JMP ILET++ SKIP PUSH
2498 027343 316 041 051 LET JSB =PUSH1A PUSH LET TO OUTPUT
2499 027346 316 056 050 ILET++ JSB =STRREF GET <STRING-REFERENCE>
2500 027351 134 250 007 LDB 34,=7 DEFAULT STORE STRING
2501 027354 370 054 JEN STRLET FOUND, CHECK FOR MULTIL
2502 027356 316 130 057 JSB =REFNUM TRY <NUMERIC-VARIABLE>
2503 027361 134 250 010 LDB 34,=10 DEFAULT STORE NUM
2504 027364 371 136 JEZ ERR92X JIF NOTHING FOUND
2505 027366 114 310 054 NUMLET CMB 14,=COMMA , <NUMVAR>?
2506 027371 366 023 JNZ RTHLF# NO, GET RIGHT
2507 027373 134 250 024 LDB 34,=24 MULTI STORE NUM
2508 027376 316 110 042 JSB =SCAN
2509 027401 316 130 057 JSB =REFNUM GET NEXT
2510 027404 371 116 JEZ ERR92X JIF NONE FOUND
2511 027406 316 366 056 JSB =NUMLET RECURS TO KEEP TRACK
2512 027411 *****
2512 027411 *
2513 027411 +NOW PUSH OUT 7 OR 10 AND RETURN TO PARSER, OR
2514 027411 * PUSH OUT 24 OR 25 AND RETURN TO NUMLET
2514 027411 *
2514 027411 *****
2515 027411 134 272 315 ENDSNV STBI 34,=PTR2- TO OUTPUT
2515 027414 377
2516 027415 236 RTN
2517 027416
2518 027416 114 310 065 RTHLF# CMB 14,=EQUALS
2519 027421 366 101 JNZ ERR92X JIF NOT FOUND
2520 027423 316 003 045 JSB =NUMVA+ SCAN, GET NUMERIC VALUE
2521 027426 370 361 JEN ENDSNV JIF NO FOUND
2522 027430 360 072 JMP ERR92X
2523 027432 114 310 054 STRLET CMB 14,=COMMA , <STRREF>?
2524 027435 366 053 JNZ RTHLF# NO, GET RIGHT HALF
2525 027437 134 250 025 LDB 34,=25 MULTI STORE STR
2526 027442 316 110 042 JSB =SCAN
2527 027445 316 056 050 JSB =STRREF GET <STRING-REF>
2528 027450 371 052 JEZ ERR92X JIF NONE FOUND
2529 027452 316 032 057 JSB =STRLET RECURS FOR MULTI
2530 027455 134 272 315 ENDSTR STBI R34,=PTR2- STORE STRING TO OUTPUT
2530 027460 377
2531 027461 236 RTN DONE

```

>>>>>		FUNCTION LET		<<<<<<<	
2534	027462 316 041 051 FHLET	JSB =PUSH1A		PUSH THE FHLET TOKEN	
		LDB R34,=10		DEFAULT TO NUMERIC	
2535	027470 114 310 001	CMB R14,=1		S.N.V.?	
2536	027473 367 006	JZR PUSHFN		JIF YES	
2537	027475 310 003	CMB R14,=3		STRING?	
2538	027477 366 023	JNZ ERR92X		JIF NO	
2539	027501 134 212	DCB R34		CHANGE FLAG TO 7	
2540	027503 316 076 051 PUSHFN	JSB =PSH45		PUSH TOKEN, NAME	
2541	027506 134 220	TSB R34		STRING OR NUMERIC FN?	
2542	027510 363 304	JEV RTHLF#		JIF NUMERIC	
2543	027512 114 310 065 RTHLF#	CMB 14,=EQUALS		=?	
2544	027515 366 005	JNZ ERR92X		JIF NO	
2545	027517	*TRY AND SALVAGE AN EXPRESSION			
2546	027517 316 321 047 FND=S	JSB =STREX+		SCAN, GET STRING EXP	
2547	027522 370 331	JEN END3TR		FOUND OK?	
2548	027524 316 377 377 ERR92X	JSB =ERROR+		SYNTAX	
2549	027527 134	OCT 92D			

```

>>>> REF STRING, ARITH VRBL; GOTO, GOSUB <<<<<<<
2550 027530 *****
2551 027530 *
2552 027530 * + NUMVAR RESOURCES:
2553 027530 * + R36 TEMP
2554 027530 * E IS ASSUMED TO BE 0 ON ENTRY
2554 027530 *
2554 027530 *****
2555 027530 114 310 003 REFNUM CMB R14,=3 (1 OR 2)-3 NO (>2)-3 CY
2556 027533 373 033 JCY NORTH JIF NOT THERE
2557 027535 312 020 ADB R14,=20 CHANGE TO REF
2558 027537 362 024 JOD PSSH45 JIF NOT ARRAY
2559 027541 316 076 051 JSB =PSH45 PUSH OUT THE VRBL
2560 027544 316 123 047 JSB =SUBSCR DO SUBSCRIPT
2561 027547 136 212 DCB R36 CHANGE TO REF
2562 027551 212 DCB R36
2563 027552
2564 027552 272 315 377 STBI R36,=PTR2- PUSH IT OUT
2565 027555 236 RTN DONE
2566 027556 134 310 003 NUMVAR CMB 34,=3 SNV OR SAV?
2567 027561 373 005 JCY NORTH JIF NO, DONE
2568 027563 363 004 JEV GETSAV JIF ARRAY VARIABLE
2569 027565 316 076 051 PSSH45 JSB =PSH45 PUSH 14,44,45
2570 027570 236 NORTH RTN DONE
2571 027571 316 124 046 GETSAV JSB =NUMVRB DO SAME AS IN PARSER
2572 027574 236 RTN DONE
2573 027575

```

IF STATEMENT 11-24-76

<<<<<<<

```

2575 027575 *****
2576 027575 *
2577 027575 * <IF STATEMENT>=IF <LOGICAL-EXPRESSION>
2578 027575 * THEN <THEN-PART> <ELSE <ELSE-PART>?
2579 027575 * <THEN-PART>=<SEQUENCE-NUMBER>/<LABEL-REF> <STMT>
2579 027575 * <ELSE-PART>=<SEQUENCE-NUMBER>/<LABEL-REF> <STMT>
2580 027575 *****
2581 027575 JMLB EQU 41
2582 027575 EJMLB EQU 43
2583 027575
2584 027575 316 003 045 IF JSB =NUMVA+ GET NUMERIC VALUE
2585 027600 371 322 JEZ ERR32X JIF NOT FOUND
2586 027602 114 310 314 CMB 14,=THENTO THEN?
2587 027605 766 315 JNZ ERR32X NO THEN
2588 027607 316 310 043 JSB =DIGIT IS NEXT A LINE#?
2589 027612 370 157 JEN THENSJ JIF YES
2590 027614 *****
2591 027614 *
2592 027614 *GET LINE# IF PRESENT, DONT SCAN BECAUSE OF POSSIBLE
2593 027614 *CALL TO PARST
2594 027614 *DO WE HAVE IF <RELEXP> THEN <SEQ-HUM>
2595 027614 * OR IS IT IF <RELEXP> THEN <STATEMENT>?
2596 027614 * PROCESS THEN <STATEMENT>
2597 027614 * STATEMENT MUST BE <LEGAL-AFTER-THEN>
2598 027614 *****
2598 027614 155 261 314 THENST LDMD R55,=PTR2 GET POINTER POSITION
2599 027617 377
2599 027620 263 154 210 STMD R55,=SAVPT2 SAVE IT
2600 027623 315 004 000 SBM R55,=4,0,0 SKIP TOKEN AND ADDR
2600 027626 000
2601 027627 263 100 200 STMD R55,=THENFL SET TO CALC REL JMP
2602 027632 250 033 LDB R55,=JPSTOK JUMP RELATIVE TOKEN
2603 027634 272 315 377 STBI R55,=PTR2- TO OUTPUT
2604 027637 223 CLM R55
2605 027640 273 315 377 STMI R55,=PTR2- 3 BYTES FOR REL JUMP
2606 027643 316 060 060 JSB =STM/LB FIND STMT OR LABEL
2607 027646 114 310 327 CMB 14,=ELSTOK ELSE?
2608 027651 766 056 JNZ ENDTN JIF NOT FOUND
2609 027653 *
2610 027653 *IF ELSE FOUND, NEED JMP TO END
2611 027653 *
2612 027653 165 261 100 STELSE LDMD R65,=THENFL SAVE OLD THENFL
2612 027656 200
2613 027657 155 261 314 LDMD R55,=PTR2 SAVE THIS POSITION
2613 027662 377
2614 027663 *
2615 027663 315 004 000 STMD R55,=SAVPT2 TO REPLACE TOKEN
2615 027666 000 SBM R55,=4,0,0 SKIP TOKEN AND ADDR
2616 027667 263 100 200 STMD R55,=THENFL SET TO CALC REL JMP

```

IF STATEMENT 11-24-76

<<<<<<<<

2	027672	250	034		LDB R55,=JMPTOK	JUMP REL TOKEN
2618	027674	272	315	377	STBI R55,=PTR2-	TO OUTPUT
2619	027677	223			CLM R55	
2620	027700	273	315	377	STMI R55,=PTR2-	3 BYTES FOR REL JMP
2621	027703	261	314	377	LDMD R55,=PTR2	
2622	027706	165	263	314	STMD R65,=PTR2	SET TO THENFL FOR ELSE
2622	027711	377				
2623	027712	316	360	057	JSB =FIXUPJ	FIX UP THEN'S JUMP
2624	027715	155	263	154	STMD R55,=SAVPT2	TO REPLACE TOKEN
2624	027720	210				
2625	027721	316	310	043	JSB =DIGIT	SCAN OVER ELSE
2626	027724	370	122		JEN ELSESN	JIF LINE#
2627	027726	316	060	060	JSB =STM/LB	FIND STMT OR LABEL
2628	027731	155	261	314	LDMD R55,=PTR2	COPY AND SAVE PTR2
2628	027734	377				
2629	027735	165	261	100	LDMD R65,=THENFL	GET POINTER
2629	027740	200				
2630	027741	263	314	377	STMD R65,=PTR2	SET EXT MEMORY PTR
2631	027744	316	360	057	JSB =FIXUPJ	FIX UP POINTERS
2632	027747	165	223		CLM R65	CLEAR THEN FLAG
2633	027751	263	100	200	STMD R65,=THENFL	
2634	027754	316	175	052	IFRTN	JSB =DMHDCR
2635	027757	236			RTN	DEMAND CR OR !
2636	027760					
2637	027760	165	055	305	FIXUPJ	SBM R65,R55
2638	027763	273	314	377	STMI R65,=PTR2	CALCULATE REL JUMP
2639	027766	155	263	314	STMD R55,=PTR2	INSERT IT IN PLACE
2640	027772	236				RESTORE PTR2
2641	027773				RTN	DONE
2641	027773				*****	
2642	027773				* <THEN-PART>=<SEQUENCE-NUMBER>	
2642	027773				*****	
2643	027773	114	250	030	THENSH	LDB 14,=30
2644	027776	316	022	061	JSB =SEQNO+	JTRUE# TOKEN
2645	030001	114	310	327	CMB 14,=ELSTOK	GET SEQUENCE NUMBER
2646	030004	366	346		JNZ IFRTN	ELSE?
2647	030006	316	310	043	JSB =DIGIT	JIF NO ELSE
2648	030011	371	010		JEZ SNELST	LINE#?
2649	030013	114	250	037	LDB R14,=37	JIF NO
2650	030016	316	022	061	JSB =SEQNO+	ELSJ# TOKEN
2651	030021	360	324		JMP CLTHNF	PUSH TOKEN AND L#
2652	030023	155	261	314	SNELST	Finish up
2652	030026	377			LDMD R55,=PTR2	SAVE THIS POSITION
2653	030027	263	154	210	STMD R55,=SAVPT2	IF WE REPLACE TOKEN
2654	030032	250	034		LDB R55,=JMPTOK	TOKEN 34
2655	030034	272	315	377	STBI R55,=PTR2-	PUSH IT TO OUTPUT
2656	030037	223			CLM R55	AND 3 ADDITL BYTES
2657	030040	273	315	377	STMI R55,=PTR2-	FOR DECOMPILER
2658	030043	316	060	060	JSB =STM/LB	GET STMTS OR LABEL
2659	030046	360	277		JMP CLTHNF	CLEAR THEN AND EXIT
2660	030050	114	250	037	ELSESN	ELSJ# TOKEN

```

PARMA 03/14/81=====
ITEM   LOC   OBJECT CODE   SRC=ORGFXN OBJ=GENGFX   9/11/1981 11:11 AM PG 74
=====
>>>> IF STATEMENT 11-24-76
266 030053 316 922 061 JSB =SEQNO+ DO THE JMP
2662 030056 360 251 JMP ENDTN FIX UP POINTER
2663 030060 *
2664 030060 *===== CHANGE FOR LABELS =====
2665 030060 *
2666 030060 110 006 345 STM/LB PUMD R10,+R5 SAVE INPUT POINTER
2667 030063 120 344 PUBD R20,+R5 SAVE CHARACTER
2668 030065 316 134 040 JSB =PARSIT PARSE THE LINE
2669 030070 117 310 300 CMB R17,=300 ANY ERRORS?
2670 030073 372 061 JNC STMEND JIF NO
2671 030075 147 310 300 CMB R47,=300 ILLEGAL AFTER THEN
2672 030100 373 054 JCY STMEND
2673 030102 117 317 077 ANM R17,=77 CLEAR ERROR BIT
2674 030105 120 222 CLB R20
2675 030107 262 123 200 STBD R20,=ERRORS CLEAR ERROR FLAG
2676 030112 006 342 POBD R20,-R5 RESTORE CHARACTER
2677 030114 110 343 POMD R10,-R5 RESTORE R10
2678 030116 155 261 154 LDMD R55,=SAVPT2 GET SAVED PTR2 POS
2679 030121 210
2680 030122 263 314 377 STMD R55,=PTR2 SET PTR2
2681 030125 134 270 317 LDBI R34,=PTR2+ FETCH PUSHED TOKEN
2682 030130 377
2683 030131 310 033 CMB R34,=JPSTOK ARE WE IN 'THEN'?
2684 030133 366 007 JNZ ELSPRT JIF NO
2685 030135 250 041 LDB R34,=JNPLB CHANGE TO JUMP LBL
2686 030137 272 315 377 STBI R34,=PTR2- PUSH IT
2687 030142 360 006 JMP FNDLBL GET THE LABEL
2688 030144 134 250 043 ELSPRT LDB R34,=EJMLB CHANGE TO ELSE LBL
2689 030147 272 315 377 STBI R34,=PTR2- PUSH NEW TOKEN
2690 030152 316 371 060 FNDLBL JSB =GETLBL
2691 030155 * JEZ ERR92Y JIF NONE
2692 030155 236 RTN
2693 030156 *GETLBL CLM R40
2694 030156 * JSB =LBL_T? IS LABEL A BASIC WORD?
2695 030156 ** JSB =IDENT FIND LABEL
2696 030156 * CLE CLEAR "FOUND" FLAG
2697 030156 * CMB R14,=1 RETURNS VAR TYPE
2698 030156 * JNZ ERR92Y JIF NOT FOUND
2699 030156 * CLE
2700 030156 ** JSB =PSH44 PUSH ADDR+LEN+LABEL
2701 030156 * ** SCAN & SET FLAG
2702 030156 * RTN
2703 030156 155 006 343 STMEND POMD R55,-R5 TRASH R10 & R20
2704 030161 236 RTN
2705 030162 *
2706 030162 *===== CHANGE ENOS =====
2707 030162 *
2708 030162 316 377 377 ERR92Y JSB =ERROR+ SYNTAX ERROR
2709 030165 134 OCT 920
2710 030166

```

```

>>>> FOR NEXT STATEMENTS 03-01-77 <<<<<<<<
2 2 030166 *
2713 030166 * SOURCE MCOMPS OUTPUT M2
2714 030166 *
2715 030166 STOTOK EQU 10 STORE TOKEN
2716 030166 STPTOK EQU 316 STEP TOKEN
2717 030166 *P EQU 120 ASCII "P"
2718 030166 *T EQU 124 ASCII "T"
2719 030166 *
2720 030166 *
2721 030166 316 041 051 FOR JSB =PUSH1A PUSH TOKEN, SCAN
2722 030171 310 001 CMB R#,,=1 SEE IF SNV
2723 030173 266 365 JNZ ERR92Y JIF NO
2724 030175 316 130 057 JSB =REFNUM CHANGE TO 21, PUSH
2725 030200 310 065 CMB R#,,=EQUALS SEE EQUAL SIGN?
2726 030202 266 356 JNZ ERR92Y JIF NO
2727 030204 114 250 010 LDB 14,,=STOTOK SO GETIN PUSHES STORE
2728 030207 316 157 051 JSB =GETIN SCAN, GET EXP, PUSH STORE
2729 030212 114 310 244 CMB R14,,=TOTOK TO?
2730 030215 266 343 JNZ ERR92Y JIF NOT FOUND
2731 030217 316 157 051 JSB =GETIN SCAN,GET EXP,PUSH TO
2732 030222 114 310 316 CMB R14,,=STPTOK STEP FOUND?
2733 030225 266 003 JNZ CALEOL JIF DEFAULT 1
2734 030227 316 157 051 JSB =GETIN SCAN,GET EXP, PUSH STEP
2735 030232 236 CALEOL RTN EXIT
2736 030233
2737 030233 114 036 242 NEXT STB R14,R36 SAVE NEXT TOKEN
2738 030236 316 110 042 JSB =SCAN
2739 030241 310 001 CMB R#,,=1 SEE IF SNV
2740 030243 266 315 JNZ ERR92Y JIF NO
2741 030245 316 130 057 JSB =REFNUM CHANGE TO 21, PUSH
2742 030250 136 272 315 STBI R36,,=PTR2- STORE NEXT TOKEN
2742 030253 577
2743 030254 236 RTN DONE
  
```

```

>>>> ON GOTO/GOSUB STATEMENT 01-12-77
2740 030255          GOTOK   EQU 132
2746 030255          GOSTOK  EQU 133
2747 030255          GOLBL   EQU 262
2748 030255          GOSLBL  EQU 270
2749 030255
2750 030255 114 036 242 ON      STB R14,R36      SAVE ON TOKEN
2751 030260 316 003 045        JSB =HUMVA+      GET EXPRESSION
2752 030263 136 272 315        STBI R36,=PTR2-   PUSH ON TOKEN
2752 030266 377
2753 030267 371 271      ERR92J JEZ ERR92Y        JIF EXPRESSION NOT FOUND
2754 030271 114 310 132      CMB 14,=GOTOK      ON ... GO?
2755 030274 367 004          JZR ONLOOP         JIF GOTO
2756 030276 310 133          CMB 14,=GOSTOK      ON ... GOSUB?
2757 030300 366 260          JNZ ERR92Y         JIF MISSING GO
2758 030302 024 242      ONLOOP STB R#,R24      SAVE GOTO/GOSUB TOKEN
2759 030304 024 240      ONLOP  LDB R#,R24      GET GOTO/GOSUB TOKEN
2760 030306 316 317 060      JSB =GOTOSU      GET LINE# AND PUSH
2761 030311 114 310 054      CMB R14,=COMMA    CHECK FOR COMMA
2762 030314 367 366          JZR ONLOP         JIF FOUND - LOOP
2763 030316 236          RTN                   DONE

```

```

=====
>>>> SEQUENCE NUMBERS, GOTO, GOSUB <<<<<<<
2763 030317 121 250 262 GOTOSU LDB R21,=262 GOTO-LABEL Token
2766 030322 114 220 TSB R14 132 OR 133?
2767 030324 363 003 JEY GOTOS+ JIF GOTO
2768 030326 121 250 270 LDB R21,=270 GOSUB-LABEL Token
2769 030331 116 310 001 GOTOS+ CMB R16,=1
2770 030334 367 060 JZR ERR88Z ERROR IF CALC
2771 030336 *****
2771 030336 *
2772 030336 * AUGUST 13,1981
2772 030336 *
2772 030336 *****
2773 030336 316 344 060 JSB =GLN/LB
2774 030341 371 217 JEZ ERR92Y JIF NEITHER L# NOR LB
2775 030343 236 RTN
2776 030344
2777 030344 110 006 345 GLN/LB PUMB R10,+R6 SAVE INPUT POINTER
2778 030347 120 344 PUBD R20,+R6 SAVE NEXT CHAR
2779 030351 316 022 061 JSB =SEQNO+ GET A LINE NUMBER
2780 030354 370 200 JEN STMFHD JIF FOUND
2781 030356 120 006 342 POBD R20,-R6 RESTORE
2782 030361 110 343 POMB R10,-R6 REGISTERS
2783 030363 114 021 240 LDB R14,R21 OPTION TOKEN
2784 030366 272 314 377 STBI R14,=PTR2 PUSH IT
2785 030371 140 223 GETLBL CLM R40
2786 030373 316 154 042 JSB =LBL_T? A VALID LABEL?
2787 030376 114 310 001 GLBCOM CMB R14,=1
2788 030401 366 067 JNZ NOLN CLEAR E ON EXIT
2789 030403 316 102 051 JSB =PSH44 PUSH LABEL
2790 030406 236 RTN
2791 030407
2792 030407 140 223 GETLB? CLM R40
2793 030411 316 171 042 JSB =IDENT A VARIABLE?
2794 030414 360 360 JMP GLBCOM
2795 030416
2796 030416
2797 030416 +GLN/LB PUMB R10,+R6 SAVE CURRENT R10
2798 030416 * PUBD R20,+R6 SAVE CHAR ALSO
2799 030416 * JSB =SEQNO+ Push, get line #
2800 030416 * JEZ LBLREF JIF NO LINE #
2801 030416 * POMB R55,-R6 TRASH R10,R20
2802 030416 * RTN
2803 030416 316 377 377 ERR88Z JSB =ERROR+
2804 030421 130 OCT 88D BAD STMT
2805 030422
2806 030422 114 272 315 SEQNO+ STBI R14,=PTR2- PUSH INCOMING TOKEN
2806 030425 377
2807 030426 316 042 061 SEQNO JSB =SEQ# SCAN GET VALID SEQ #
2808 030431 371 003 JEZ SEQRTH JIF NONE
2809 030433 316 071 051 JSB =PUSH44 PUSH IT
2810 030436 236 SEQRTH RTN
2811 030437
2812 030437 114 036 242 SEQ#+ STB R14,R36 SAVE INPUT R14
2813 030442 316 110 042 SEQ# JSB =SCAN NEXT TOKEN

```

```

>>>> SEQUENCE NUMBERS, GOTO, GOSUB <<<<<<<
2314 030445 234 ICE
2315 030446 114 310 032 CMB R14,=32 INTEGER?
2316 030451 766 017 JNZ NOLN JIF NO
2317 030453 146 220 TSB R46 5 DIGITS?
2318 030455 375 007 JLN ERR90X ERROR - TOO BIG
2319 030457 147 223 CLM R47 KILL CLASS FOR TSM
2320 030461 144 221 TSM R44
2321 030463 367 072 JZR ERR89T ZERO IS NOT ACCEPTABLE
2322 030465 236 RTN
2323 030466 316 377 377 ERR90X JSB =ERROR+ LINE # > 99999
2324 030471 132 OCT 900
2325 030472 235 NOLN CLE
2326 030473 236 RTN
2327 030474
2328 030474 *
2329 030474 +===== CHANGE FOR GOTO LABEL =====
2330 030474 *
2331 030474 *LBLREF POBD R20,-R6 RESTORE CHAR
2332 030474 * POND R10,-R6 RESTORE INPUT POINTER
2333 030474 * LDB R14,R21 GOTO-,REST-,USING-L.
2334 030474 * STBI R14,=PTR2 PUSH IT
2335 030474 * JSB =GETLBL GET + PUSH LABEL
2336 030474 * JEZ ERR920
2337 030474 * RTN
2338 030474 * SPC
2339 030474 *
2340 030474 +=====CHANGE ENDS=====
2341 030474 *

```

```

PARMA 03/14/81=====
ITEM   LOC   OBJECT CODE   SRC=ORGFXX OBJ=CEXGFX   9/11/1981 11:11 AM PG 79
=====
>>>>      AUTO LINE NUMBER AND RENUMBER      <<<<<<<<
2 3 030474      **** AUTO ATTRIBUTE TABLE ****
2844 030474 141      OCT 141      BASIC
2845 030475      *****
2846 030475      *
2847 030475      * LSS OF AUTOI IS CLEARED BY ALLOCATOR IF AN ERROR *
2848 030475      *
2849 030475      *****
2850 030475 316 116 061 AUTO. JSB =AUTOX
2851 030500 316 067 042 JSB =ST240+
2852 030503 165 062 241 LDM R55,R62 MOVE LINE# FOR PUTAUT
2853 030506 130 251 236 LDM R30,=INPBUF START FOR PUTAUT
2853 030511 200
2854 030512 316 142 041 JSB =PUTAUT PUSH LINE #
2855 030515 236 RTN DONE
2856 030516 316 377 377 AUTOX JSB =TO?I GET START# (44) & INC (54)
2857 030521 165 251 020 LDM R65,=100,0,0 SET DEFAULTS
2857 030524 000 000
2858 030526 062 243 STM R65,R62
2859 030530 371 014 JEZ AUTOS JIF NONE FOUND
2860 030532 145 243 STM R45,R62 AUTO #
2861 030534 367 015 JZR AUTERR
2862 030536 233 DCE
2863 030537 371 005 JEZ AUTOS JIF NO INC
2864 030541 155 065 243 STM R55,R65 PUT INC IN 65-67
2865 030544 367 005 JZR AUTERR?
2866 030546 162 263 103 AUTOS STMD R62,=AUTO# SET PARAMS (A# & AI)
2867 030552 236 RTN DONE
2868 030553 230 AUTERR BIN
2869 030554 106 213 DCM R5 TRASH 1 RETURN
2870 030556 213 DCM
2871 030557 316 377 377 ERR99T JSB =ERROR+ INVALID PARAMS
2872 030562 131 OCT 29D INVALID PARAMS
2873 030563 *****
2873 030563 *
2874 030563 * RENUMBER ATTRIBUTE TABLE *
2874 030563 *
2875 030563 *****
2876 030563 141 OCT 141 BASIC
2876 030564
2877 030564 316 116 061 RENUM. JSB =AUTOX GET START & INCR
2878 030567 230 BIN
2879 030570 162 263 103 STMD R62,=AUTO# SAVE AUTO# & AUTOI
2879 030573 200
2880 030574 316 067 042 JSB =ST240+ CLB R16,SET STSIZE,TOS
2881 030577 102 222 CLB R2 RESET OPTB FLAG
2882 030601 262 346 212 STBD R2,=OPTB
2883 030604 210 ICB R2 SET IT FOR ALLOCATOR
2884 030605 262 104 202 STBD R2,=DALLED
2885 030610 316 377 377 JSB =CUROPS GET CURR OPT BASE
2886 030613 316 377 377 JSB =ALLOC MAYBE SUBSET LATER
2887 030616 102 222 CLB R2 NOW IT'S ALLOCATED
2887 030620 262 104 202 STBD R2,=DALLED

```

```

>>>>>      AUTO LINE NUMBER AND RENUMBER      <<<<<<<
2890 030623 370 070      JEN FINREN      ALLOC ERROR,DONT REN!!!!
2890 030625 316 377 377      JSB =INIYAR      INITIALIZE VARIABLES
2891 030630 230      RAGAIN      BIN
2892 030631 165 261 006      LDMD R65,=FWCURR      GET PROGRAM ADDR
2892 030634 200
2893 030635 315 040 000      SBM R65,=40,0,0      ADD OFFSET OF P.GOG
2893 030640 000
2894 030641 263 310 377      STMD R65,=PTR1      SET POINTER
2895 030644 175 223      CLM R75      FIRST TIME AUTOI
2896 030646 360 006      JMP STOL      STORE FIRST LINE #
2897 030650
2898 030650 316 377 377 DONXT      JSB =SKPLN2      SKIP TO NEXT LINE
2899 030653      ORP 175
2900 030653 261 106 200      LDMD R75,=AUTOI      INCREMENT
2901 030656      STOL      BSS 0
2902 030656 145 271 311      LDMI R45,=PTR1-      CURRENT OLD LINE
2902 030661 377
2903 030662 231      BCD
2904 030663 311 231 231      CMN R45,=231,231,12      DONE ?
2904 030666 012
2905 030667 373 024      JCY FINREN      JIF YES
2906 030671 155 261 103      LDMD R55,=AUTO#      NEW LINE #
2906 030674 200
2907 030675 075 303      ADM R55,R75      NEXT NEW LINE #
2908 030677 311 231 231      CMN R55,=99C,99C,12      LINE # > 9999
2908 030702 012
2909 030703 365 016      JPS RENUMR      JIF YES
2910 030705 273 310 377      STMI R55,=PTR1      STORE IT
2911 030710 263 103 200      STMD R55,=AUTO#
2912 030713 360 333      JMP DONXT      AND LOOP
2913 030715
2914 030715 175 223      FINREN      CLM R75      KILL AUTO MODE
2915 030717 263 106 200      STMD R75,=AUTOI      AUTO active flag
2916 030722 236      RENRTN      RTN
2917 030723
2918 030723 223      RENUMR      CLM R#
2919 030724 211      ICM R#      (R75)
2920 030725 263 106 200      STMD R#,=AUTOI      RENUM WITH 1,1
2921 030730 263 103 200      STMD R#,=AUTO#
2922 030733 316 377 377      JSB =WARN
2923 030736 132      OCT 900      LINE #>10000
2924 030737 316 071 042      JSB =SET240
2925 030742 360 264      JMP RAGAIN
2926 030744
2927 030744 142 220      ROMFX      TSB R42      ROM?
2928 030746 366 004      JNZ GETNXP      JIF NO
2929 030750 141 262 030      STBD R41,=RSELEC      SELECT CORRECT ROM
2929 030753 377
2930 030754 236      GETNXP      RTN
2931 030755
2932 030755      RTNCOD      EQU 236
2933 030755
2934 030755      *

```

```

KARMA 03/14/81=====
ITEM   LOC   OBJECT CODE   SRC=ORGFXN OBJ=GENGFX   9/11/1981 11:11 AM PG 31
=====
>>>>>> AUTO LINE NUMBER AND RENUMBER <<<<<<<<
2935 030755 * SUBROUTINE TSTALO
2936 030755 *
2937 030755 * THIS ROUTINE RETURNS THE ALLOCATION
2938 030755 * FLAG IN "E".
2939 030755 *
2940 030755 * NOTE: MODE IS SET TO BCD
2941 030755 *
2942 030755 *
2943 030755 TSTALO BSS 0
2944 030755 155 261 006 LDMD R55,=FWCURR
2944 030760 200
2945 030761 315 006 000 SBM R55,=6,0,0 PT TO P.TYP
2945 030764 000
2946 030765 263 314 377 STMD R55,=PTR2
2947 030770 157 270 315 LDBI R57,=PTR2- FETCH P.TYP
2947 030773 377
2948 030774 317 040 ANM R57,=40 ISOLATE ALLOC BIT
2949 030776 231 BCD
2950 030777 200 ELB R57 SHIFT INTO E-REGISTER
2951 031000 236 RTN
2952 031001
2953 031001 EXTERNALS

```

>>>>>		EXTERNALS	<<<<<<<				
295	031001	EXT ALLOC					
2956	031001	EXT BCDINT					
2957	031001	EXT CUROPS					
2958	031001	EXT CURS					
2959	031001	EXT DECUR2					
2960	031001	EXT DCLIN#					
2961	031001	EXT DIMLIST					
2962	031001	EXT ERMSG					
2963	031001	EXT ERNUM.					
2964	031001	EXT ERROR					
2965	031001	EXT ERROR+					
2966	031001	EXT FIXBYT					
2967	031001	EXT FNDLIN					
2968	031001	EXT FNDRTN					
2969	031001	EXT FRMNA#					
2970	031001	EXT FRMNA#					
2971	031001	EXT FUNIA					
2972	031001	EXT FUNS					
2973	031001	EXT GRERAS					
2974	031001	EXT HLFLIN					
2975	031001	EXT INCHR					
2976	031001	EXT INIVAR					
2977	031001	EXT ITHEN					
2978	031001	EXT JEMPLN#					
2979	031001	EXT LINEDR					
2980	031001	EXT LINENO					
2981	031001	EXT LISTIP					
2982	031001	EXT LISTPR					
2983	031001	EXT LOGEXP					
2984	031001	EXT HXTPGM					
2985	031001	EXT OPTIM#					
2986	031001	EXT ON?I					
2987	031001	EXT ONEI					
2988	031001	EXT ONER					
2989	031001	EXT POSLP					
2990	031001	EXT PROVR1					
2991	031001	EXT PRINT					
2992	031001	EXT PSYM					
2993	031001	EXT PWO					
2994	031001	EXT RAMADD					
2995	031001	EXT REDROM					
2996	031001	EXT RUN.					
2997	031001	EXT SEMIC#					
2998	031001	EXT SHRONF					
2999	031001	EXT SKPLN					
3000	031001	EXT SKPLN2					
3001	031001	EXT SKPPAR					
3002	031001	EXT STOP.					
3003	031001	EXT SYSCOM					
3004	031001	EXT TABL3					
3005	031001	EXT TO?I					
3006	031001	EXT TTAB.A					
3007	031001	EXT WARN					

PARMA 03/14/81=====

ITEM	LOC	OBJECT CODE	SRC=ORGFX	OBJ=GEKGF	9/11/1981	11:11 AM	PG 83
------	-----	-------------	-----------	-----------	-----------	----------	-------

=====

>>>> EXTERNALS <<<<<<<<

3 3 031001 FIN

ADIR36	021564	LCL	1		
ALC	021656	ENT	7		
ALLOC	177777	EXT	1		
ARDIMS	025534	LCL	1		
ARRAY	023464	LCL	1		
ARRY1	025547	LCL	1		
AUTERR	030553	LCL	2		
AUTO#	100103	G DAD	8		
AUTO.	030475	ENT	1		
AUTOI	100106	G DAD	4		
AUTOS	030546	LCL	2		
AUTOX	030516	ENT	3		
AWITH2	027061	LCL	1		
BADSCN	021311	LCL	2		
BASIC?	020162	LCL	1		
BCDINT	177777	EXT	1		
BINBAS	104073	G DAD	1		
BINLOP	021741	LCL	1		
BINTAB	104079	G DAD	3		
BLANK	000040	G EQU	7		
B#	000007	G EQU	2		
BFILT	022013	ENT	3		
BS	000225	EQU	0	<--- NOT REFERENCED??	
C.PARS	020633	LCL	1		
C.PDUN	021013	LCL	1		
CALEOL	030232	LCL	1		
CALFF\$	026306	LCL	1		
CHLOOP	024275	LCL	1		
CK*CR	021317	LCL	2		
CK*AR	025524	LCL	1		
CKGOTO	025127	LCL	2		
CLOSE	000051	G EQU	4		
CLTHNF	027747	LCL	2		
CMATOK	000357	EQU	1		
CNTBLK	020602	LCL	1		
COM	025153	ENT	1		
COMBIN	021567	LCL	2		
COMITE	025213	LCL	1		
COMNA	000054	G EQU	15		
C*IPU	025230	LCL	2		
CUNV	025233	LCL	1		
CR	000015	G EQU	13		
CRKEY	000232	EQU	0	<--- NOT REFERENCED??	
CRT014	024352	LCL	1		
CRTOK	000354	EQU	1		
CUROPS	177777	EXT	1		
CURS	177777	EXT	1		
DALLED	101104	G DAD	2		
DATA	026643	ENT	1		
DATCM1	026713	LCL	1		
DATCOM	026710	LCL	3		
DATLST	026647	LCL	1		
DCLIN#	177777	EXT	1		
DECP#	026117	LCL	0	<--- NOT REFERENCED??	

SYMBOL	VALUE	TYPE	COUNT	Symbol Table
ECUR2	177777	EXT	1	
EF	025562	EXT	1	
EF.2	026267	LCL	2	
EF.2	026271	LCL	2	
EFRTN	026206	LCL	1	
ELIM	027397	LCL	2	
ELIM+	027320	LCL	1	
ELIM-	027301	LCL	1	
EGIT	021710	EXT	9	
EGIM	025145	EXT	1	
EGINCM1	025162	LCL	1	
EGINCOM	025157	LCL	1	
EGINITE	025241	LCL	1	
EGINLIS	177777	EXT	0	<--- NOT REFERENCED??
EGINRTN	025277	LCL	3	
EGINCOM?	025170	LCL	0	<--- NOT REFERENCED??
EGINDCR-	025202	LCL	1	
EGINDCR	025175	EXT	4	
EGINDCJP	025360	LCL	1	
EGOLLAR	000044	G EQU	1	
EGOT	030650	LCL	1	
EGOS	023733	LCL	1	
EGPIST	021415	LCL	1	
EGPARS	020713	EXT	2	
EGMPLB	000043	EQU	1	
ELSESH	030050	LCL	1	
ELSPRT	030144	LCL	1	
ELSTOK	000327	EQU	2	
EUO	024340	LCL	2	
ENDSNV	027411	LCL	1	
ENDSTR	027455	LCL	1	
ENDTHN	027731	LCL	2	
EOL	020262	LCL	2	
EOLOK	020274	LCL	1	
EOPRT	027262	LCL	1	
EOPRT+	027272	LCL	3	
EPLST	026454	LCL	2	
EQUALS	000065	EQU	4	
ERLIN#	100114	G DAD	4	
ERFG	177777	EXT	0	<--- NOT REFERENCED??
ERFIS	177777	EXT	0	<--- NOT REFERENCED??
ERNUM.	177777	EXT	0	<--- NOT REFERENCED??
ERR192	020570	LCL	2	
ERR51	025045	LCL	3	
ERR90	023455	LCL	0	<--- NOT REFERENCED??
ERR91	022646	LCL	2	
ERR91J	023012	LCL	2	
ERR91P	022740	LCL	0	<--- NOT REFERENCED??
ERR92.	024141	LCL	3	
ERR92S	024330	LCL	0	<--- NOT REFERENCED??
ERR92Y	024672	LCL	0	<--- NOT REFERENCED??
ERR92Z	024334	LCL	1	
ERR93X	023504	LCL	0	<--- NOT REFERENCED??
ERR93Y	024553	LCL	0	<--- NOT REFERENCED??

ERR55	022376	LCL	1		
ERR5X	022564	LCL	2		
ERR6	021067	LCL	1		
ERR88U	025471	ENT	2		
ERR88Z	030416	LCL	1		
ERR99	023443	LCL	2		
ERR99T	030557	LCL	1		
ERR99U	025475	LCL	5		
ERR99V	026212	LCL	3		
ERR99X	027101	LCL	3		
ERR99Y	024755	LCL	2		
ERR99Z	025300	LCL	7		
ERR90	020313	LCL	3		
ERR90X	030466	LCL	1		
ERR91	023655	LCL	1		
ERR91X	026550	LCL	3		
ERR91Y	024766	LCL	1		
ERR91Z	025416	LCL	3		
ERR92J	030267	LCL	0	<--- NOT REFERENCED??	
ERR92X	027524	LCL	9		
ERR92Y	030162	LCL	7		
ERR9R	177777	EXT	6		
ERROR+	177777	EXT	28		
ERRORS	100123	G DAD	4		
ERRSTR	021017	LCL	3		
ERSV50	101105	G DAD	4		
EXLOOP	021537	LCL	1		
EXPEND	020771	LCL	1		
EXPON	021453	LCL	2		
EXPON+	021442	LCL	1		
EXSALT	022004	LCL	1		
FACTO-	023021	LCL	1		
FACTOR	023014	LCL	2		
FALSEE	021515	LCL	1		
FINREN	030715	LCL	2		
FIXBYT	177777	EXT	0	<--- NOT REFERENCED??	
FINFP#	021606	LCL	2		
FIXUPJ	027760	LCL	2		
FLOOP	021417	LCL	2		
FTGPT	023574	LCL	1		
FLAL	026303	LCL	1		
FNCALL	026276	LCL	1		
FND=S	027517	LCL	0	<--- NOT REFERENCED??	
FNDCM?	023516	LCL	2		
FNDOR	024321	LCL	1		
FNDOR-	024315	LCL	1		
FNDE	021534	LCL	1		
FNDE+	021527	LCL	1		
FNDE-	021525	LCL	1		
FNDIT	021330	LCL	1		
FNDLBL	030152	LCL	1		
FNDLIN	177777	EXT	0	<--- NOT REFERENCED??	
FNDPRI	023111	LCL	3		
FNDRTN	177777	EXT	0	<--- NOT REFERENCED??	

SYMBOL	021306	LCL	2	
SYMBOL	024463	ENT	1	
SYMBOL	027462	ENT	1	
SYMBOL	026216	LCL	0	<--- NOT REFERENCED??
SYMBOL	026221	LCL	2	
SYMBOL	026270	LCL	1	
SYMBOL	000241	EQU	2	
SYMBOL	030155	ENT	1	
SYMBOL	027042	ENT	1	
SYMBOL	027034	ENT	3	
SYMBOL	177777	EXT	0	<--- NOT REFERENCED??
SYMBOL	177777	EXT	0	<--- NOT REFERENCED??
SYMBOL	027060	LCL	2	
SYMBOL	023146	LCL	3	
SYMBOL	177777	EXT	0	<--- NOT REFERENCED??
SYMBOL	177777	EXT	0	<--- NOT REFERENCED??
SYMBOL	100006	G DAD	2	
SYMBOL	024543	ENT	1	
SYMBOL	024642	ENT	1	
SYMBOL	024707	ENT	1	
SYMBOL	024726	ENT	1	
SYMBOL	024701	LCL	5	
SYMBOL	024744	ENT	1	
SYMBOL	024706	LCL	0	<--- NOT REFERENCED??
SYMBOL	024764	LCL	2	
SYMBOL	024772	ENT	1	
SYMBOL	024761	ENT	1	
SYMBOL	021705	ENT	4	
SYMBOL	021640	LCL	1	
SYMBOL	021636	ENT	15	
SYMBOL	021654	LCL	1	
SYMBOL	025002	ENT	1	
SYMBOL	023754	LCL	2	
SYMBOL	023450	ENT	9	
SYMBOL	024676	ENT	1	
SYMBOL	024557	ENT	4	
SYMBOL	024630	ENT	2	
SYMBOL	024635	ENT	1	
SYMBOL	026123	LCL	1	
SYMBOL	023510	ENT	10	
SYMBOL	023477	ENT	3	
SYMBOL	024574	LCL	1	
SYMBOL	027207	LCL	1	
SYMBOL	030407	LCL	0	<--- NOT REFERENCED??
SYMBOL	030371	LCL	1	
SYMBOL	025767	LCL	1	
SYMBOL	023661	LCL	2	
SYMBOL	030754	LCL	1	
SYMBOL	024546	LCL	1	
SYMBOL	024565	LCL	1	
SYMBOL	024740	ENT	5	
SYMBOL	024562	ENT	4	
SYMBOL	023210	LCL	1	
SYMBOL	024655	LCL	3	

```

=====
GETSAV 027571 LCL 1
GETSGN 023623 LCL 2
GETT- 024661 ENT 3
GETSTP 024656 ENT 4
GETVAR 021206 LCL 0 <--- NOT REFERENCED??
GLBCOM 030376 LCL 1
GLNPLB 030344 ENT 4
GLBL 000262 EQU 0 <--- NOT REFERENCED??
GOSALT 021164 LCL 0 <--- NOT REFERENCED??
GOSLBL 000279 EQU 0 <--- NOT REFERENCED??
GOSTOK 000133 EQU 1
GOTOK 000132 EQU 1
GOTOS+ 030331 LCL 1
GOTOSU 030317 ENT 3
GOTSMI 027916 LCL 0 <--- NOT REFERENCED??
GOTVRB 022224 LCL 6
GRERAS 177777 EXT 0 <--- NOT REFERENCED??
GTMORE 026427 LCL 1
GXX3 024651 LCL 1
HLFLIN 177777 EXT 1
H 021301 LCL 1
HLL+ 021303 LCL 1
IDENT 021171 ENT 2
IF 027575 ENT 1
IFRTH 027754 LCL 1
ILET 027336 ENT 1
ILET++ 027346 LCL 1
IMPLET 020150 LCL 1
IN#TOK 000350 EQU 0 <--- NOT REFERENCED??
IN$TOK 000351 EQU 0 <--- NOT REFERENCED??
INCHR 177777 EXT 0 <--- NOT REFERENCED??
INDEN2 020615 LCL 1
INDEN3 020632 LCL 1
INDENT 020574 LCL 1
INDTOK 000367 EQU 1
INDXOK 025414 LCL 0 <--- NOT REFERENCED??
INIVAR 177777 EXT 1
INNERL 022044 LCL 2
INPBUF 100236 G DAD 5
INPTOK 000137 EQU 0 <--- NOT REFERENCED??
INT 026577 ENT 1
INTEG! 021620 LCL 1
INTEGR 021331 ENT 3
INTFND 021366 LCL 1
INTRTH 021373 LCL 1
INTTOK 000177 EQU 1
ITHEN 177777 EXT 0 <--- NOT REFERENCED??
ITSRM2 023420 LCL 1
ITSROM 023173 LCL 1
JCYE89 024653 LCL 1
JERR81 022590 LCL 1
JEZF89 026755 LCL 1
JFNCAL 023112 LCL 1
JMLPB 000041 EQU 1

```

SYMBOL	VALUE	TYPE	COUNT	Symbol Table
AMPLN#	177777	EXT	0	<--- NOT REFERENCED??
ATES	026346	LCL	1	
ATOK	000034	EQU	2	
BPSTOK	000033	EQU	2	
CPU45	025557	LCL	2	
KEYHAM	025051	LCL	2	
LASTCH	022107	LCL	1	
LAVAIL	100025	G DAD	2	
LBL_T?	021154	LCL	1	
LBRAK	000336	EQU	4	
LEADBL	024243	LCL	1	
LET	027343	ENT	1	
LETTOK	000142	EQU	0	<--- NOT REFERENCED??
LINEDR	177777	EXT	1	
LINEHD	020445	LCL	1	
LINEHD	177777	EXT	0	<--- NOT REFERENCED??
LINSIZ	020477	LCL	1	
LISTIP	177777	EXT	0	<--- NOT REFERENCED??
LISTPR	177777	EXT	0	<--- NOT REFERENCED??
LN#OK	020362	LCL	0	<--- NOT REFERENCED??
LOGEXP	022413	LCL	1	
LOGFA-	022451	LCL	1	
LOGFAC	022444	LCL	2	
LOGPRI	022502	LCL	2	
LSTAF?	022142	LCL	1	
MD16	021374	LCL	1	
MDCHAR	021361	LCL	1	
MDUS	000055	G EQU	2	
MDLOOP	021342	LCL	1	
MONAD+	022752	LCL	1	
MONAD-	022754	LCL	1	
MONTR1	022736	LCL	1	
MONTRM	022716	LCL	2	
NAMEOK	026352	LCL	1	
NARRE+	023461	ENT	1	
NARREF	023465	ENT	1	
NAM	021671	LCL	2	
NEXT	030233	ENT	1	
NLOOP	021205	LCL	3	
NNE?	025432	LCL	1	
NMPARS	023364	LCL	1	
NMYREL	025304	LCL	2	
NO	021725	LCL	2	
NOARRY	027214	LCL	2	
NOOMA	023522	LCL	1	
NOCON#	024356	LCL	1	
NODEC	024234	LCL	1	
NOER92	025255	LCL	0	<--- NOT REFERENCED??
NOFLO	026235	LCL	0	<--- NOT REFERENCED??
NOLABL	020442	LCL	2	
NOLN	030472	LCL	2	
NOPAR	026445	LCL	0	<--- NOT REFERENCED??
NOPARI	026542	LCL	1	

ONE	021704	LCL	2		
OPEN	027579	LCL	2		
OSTR	024137	LCL	1		
OSTRF	026254	LCL	1		
OT123	020160	LCL	1		
OTBLK	022075	LCL	1		
OTHEN	020175	LCL	1		
OTRBF	023203	LCL	1		
ULEOL	020450	LCL	1		
UMBER	021400	ENT	2		
UMCON	023551	ENT	3		
UMEX+	022652	ENT	2		
UMEXP	022655	LCL	1		
UMLET	027366	LCL	1		
UMREL	022572	LCL	3		
UMREP	023607	ENT	3		
UMRET	023606	LCL	1		
UMRF?	026620	LCL	1		
UMRST	022744	LCL	2		
UMVA+	022403	ENT	11		
UMVAL	022406	ENT	7		
UMVAR	027556	LCL	0	<--- NOT REFERENCED??	
UMVRB	023124	LCL	2		
XTBIN	021760	LCL	1		
XTDG?	022175	LCL	2		
XTHEM	100022	G DAD	4		
XTPGM	177777	EXT	0	<--- NOT REFERENCED??	
XTROM	021764	LCL	1		
OFKEY#	025060	ENT	1		
OFTIM+	177777	EXT	0	<--- NOT REFERENCED??	
OKPAR	024626	LCL	1		
OLDTOS	101746	G DAD	6		
ON	030255	ENT	1		
ON?I	177777	EXT	0	<--- NOT REFERENCED??	
ONEDIM	025350	LCL	1		
ONEI	177777	EXT	0	<--- NOT REFERENCED??	
ONER	177777	EXT	0	<--- NOT REFERENCED??	
ONERRO	025140	ENT	1		
ONKEY#	025075	ENT	1		
ONOP	030302	LCL	1		
ONLOP	030304	LCL	1		
ONTIME	025079	ENT	1		
OPEN	000050	G EQU	4		
OPTB	105346	G DAD	1		
OPTION	025017	ENT	1		
OUTERL	022037	LCL	7		
P.DONE	024612	LCL	2		
P.PARS	020317	ENT	2		
P.PRTH	020567	LCL	2		
P.TAIL	026173	LCL	1		
PARLOP	026376	LCL	1		
PARPRM	025463	ENT	1		
PARSE!	020137	LCL	2		
PARSE0	020131	LCL	1		

```

PARMA 03/14/81=====
SYMBOL  VALUE      TYPE      COUNT      Symbol Table      9/11/1981 11:11 AM PG 91
=====
ARSR  020000      ENT        1
IT    020134      LCL        2
ACS   026002      LCL        1
PERIOD 000056      G EQU       2
PLEND 027300      LCL        1
PLIST 027161      LCL        1
PLIST+ 027163      LCL        1
PLUS  000053      G EQU       2
POPFUN 023432      LCL        2
POSLP  177777      EXT        0      <--- NOT REFERENCED??
PROCOM? 027245      LCL        1
PRDVR1 177777      EXT        0      <--- NOT REFERENCED??
PRIMAR 023050      LCL        2
PRINIT 177777      EXT        0      <--- NOT REFERENCED??
PRINT  027105      ENT        1
PRIRTN 022563      LCL        7
PRISEP 027230      LCL        3
PRLQOP 025633      LCL        2
PRMLQOP 023304      LCL        1
)OGRA 020052      LCL        1
PRDR  020232      LCL        1
PRSEER 020303      LCL        2
PRSIDL 103733      G DAD       1
PRSRTH 020261      LCL        1
PRTYPS 026027      LCL        1
PSH44  024502      LCL        2
PSH45  024476      LCL       13
PSH46  024507      LCL        3
OUT    024513      LCL        1
PSRTN  020121      ENT        3
PSSH45 027565      LCL        1
PSTRIN 027225      LCL        1
PSYM   177777      EXT        0      <--- NOT REFERENCED??
PTR1   177710      G DAD       2
PTR1-  177711      G DAD       1
PTR2   177714      G DAD      46
PTR2+  177716      G DAD       9
PTR2-  177715      G DAD      74
TR2-+  177717      G DAD       1
TIMES  023354      LCL        1
PU3637 024452      LCL        0      <--- NOT REFERENCED??
PU36SC 024535      LCL        2
PU45CL 025310      LCL        2
PUFLGS 026524      LCL        1
PUKTOK 025121      LCL        2
PULNGT 026465      LCL        0      <--- NOT REFERENCED??
PUNAME 026471      LCL        1
PURA&T 026107      LCL        1
PUSH## 024456      LCL        4
PUSH1# 024442      LCL        1
PUSH1A 024441      ENT       17
PUSH32 024527      LCL        4
PUSH40 024540      LCL        1
PU444  024471      LCL        1

```

SH45	024465	ENT	1	
PU 46	024455	LCL	0	<--- NOT REFERENCED??
PUSHFN	027503	LCL	1	
PUSHL\$	025743	LCL	1	
PUSHR#	027052	LCL	1	
PUSHRD	027012	LCL	1	
PUTAUT	020542	LCL	2	
PWO	177777	EXT	0	<--- NOT REFERENCED??
QUOTCR	021314	LCL	2	
QUOTE	000042	G EQU	6	
R60+2	060002	G DAD	2	
R60+4	060004	G DAD	1	
R60+6	060006	G DAD	1	
R6LIM1	102530	G DAD	2	
RAGAIN	030630	ENT	2	
RAMPDD	177777	EXT	0	<--- NOT REFERENCED??
RBRACK	000135	G EQU	1	
RBRACKT	000337	EQU	3	
RD#ITM	026765	LCL	1	
RD#LST	027021	LCL	1	
RD#P	000367	EQU	1	
RD#K.	000346	EQU	1	
RDSTK.	000353	EQU	1	
READ	026747	ENT	1	
READ3	000364	EQU	1	
REALTK	000231	EQU	1	
REDITM	026772	LCL	1	
REDLST	026752	LCL	1	
REDROM	177777	EXT	0	<--- NOT REFERENCED??
REDRTH	027015	LCL	2	
REFNUM	027530	ENT	7	
REM	024217	ENT	1	
REMS-	024357	LCL	1	
RENRTN	030722	LCL	0	<--- NOT REFERENCED??
RENUM.	030564	ENT	1	
RENUMR	030723	LCL	1	
RESTFN	026560	LCL	1	
RESTOR	026724	ENT	1	
RT.NU	022751	LCL	4	
RT#HT\$	026170	LCL	1	
ROMX	030744	LCL	1	
ROMTK.	023250	LCL	1	
ROMTOK	022262	LCL	1	
RS1020	022301	LCL	2	
RSELEC	177430	G DAD	4	
RSTEX	026746	LCL	1	
RSTL#	000334	EQU	1	
RSTLB	000335	EQU	1	
RSTORE	020106	LCL	2	
RSTR6	026047	LCL	3	
RSTREG	022346	ENT	3	
RTHLF#	027416	LCL	2	
RTHLF\$	027512	LCL	1	
RTN	022003	LCL	2	

```

=====
ARNA 03/14/81=====
SYMBOL  VALUE      TYPE      COUNT      Symbol Table      9/11/1981 11:11 AM PG 93
=====
COD      000236      EQU       0      <--- NOT REFERENCED??
CK      025057      LCL       1
      177777      EXT       0      <--- NOT REFERENCED??
SAITH    025362      LCL       1
SALT     021727      LCL       1
SAVER6   104066      G DAD     4
SAVPT2   104154      G DAD     7
SAVR10   104063      G DAD     5
SAVREG   022310      ENT       3
SCABRT   021123      LCL       1
SCAN     021110      ENT      28
SCAN+    021105      ENT       4
SCAN++   021132      LCL       1
SCAN+3   021142      LCL       1
SCANNN   024445      LCL       2
SCRTN1   021310      LCL       2
SDIM     025241      LCL       0      <--- NOT REFERENCED??
SDIM+    025251      LCL       1
SEMI     000047      G EQU     3
SEMIC*   177777      EXT       0      <--- NOT REFERENCED??
SEMI     030442      LCL       1
SEMI+    030437      LCL       2
SEGNO    030426      LCL       0      <--- NOT REFERENCED??
SEGNO+   030422      ENT       5
SEQRTH   030436      LCL       1
SET240   021071      ENT       2
SETCL4   021630      LCL       1
SON?1    023640      LCL       1
EX       023720      LCL       1
SGNFLT   023705      LCL       1
SGNNU1   023645      LCL       1
SHFEXP   021545      LCL       1
SHRONE   177777      EXT       1
SHRTOK   000200      EQU       1
SKBLNK   022053      LCL       0      <--- NOT REFERENCED??
SKIP     022032      LCL       2
SKPBLK   022064      LCL       1
SKPLN    177777      EXT       0      <--- NOT REFERENCED??
SKPLN2   177777      EXT       1
S. PAR   177777      EXT       0      <--- NOT REFERENCED??
SML-1-   025401      LCL       3
SMLINT   023557      ENT       2
SHELST   030023      LCL       1
SOURCE   024016      LCL       2
ST240+   021067      ENT       3
ST240-   021071      ENT       1
STAR+    024171      LCL       1
STARAY   024163      LCL       1
STAREF   024143      LCL       1
STELSE   027653      LCL       0      <--- NOT REFERENCED??
STM/LB   030060      LCL       3
STMFND   030156      LCL       3
STOL     030656      LCL       1
STOP,    177777      EXT       0      <--- NOT REFERENCED??

```

```

PARMA 03/14/81=====
SYMBOL  VALUE      TYPE      CCUNT      Symbol Table      9/11/1981 11:11 AM  PG 94
=====
TOTOK    000010      EQU       1
STAEY    000216      EQU       0      <--- NOT REFERENCED??
STPTOK   000316      EQU       1
STR.R+   024012      LCL       1
STR.RN   024007      LCL       1
STRANG   103715      G DAD     1
STRCN1   024257      LCL       2
STRCN2   024262      LCL       1
STRCON   024201      LCL       2
STREX+   023721      ENT       5
STREXP   023724      ENT       7
STRLET   027432      LCL       2
STRPAR   025665      LCL       1
STRREF   024056      ENT       5
STRTYP   026055      LCL       1
STSIZE   101741      G DAD     7
STVRB+   024071      LCL       1
STVRBL   024066      LCL       1
STXRTH   024055      LCL       2
SUBDUN   024134      LCL       1
SUBE     025324      LCL       1
SUBSCR   023523      LCL       4
SYSCOM   177777      EXT       0      <--- NOT REFERENCED??
SYSPRS   020514      ENT       1
SYSTOK   022255      LCL       1
TABCAL   027330      LCL       1
TABL3    177777      EXT       0      <--- NOT REFERENCED??
TABTOK   000315      EQU       0      <--- NOT REFERENCED??
TERM     022760      LCL       2
TERM-    022765      LCL       1
TERMS    022703      LCL       0      <--- NOT REFERENCED??
TERMS-   022710      LCL       1
THENER   020307      LCL       1
THENFL   100100      G DAD     6
THENSH   027773      LCL       1
THENST   027614      LCL       0      <--- NOT REFERENCED??
THENTO   000314      EQU       1
THIRT2   000032      DAD       2
THIRT4   000034      DAD       1
THIR6    000036      DAD       1
TO?1     177777      EXT       1
TOGOTO   025134      LCL       0      <--- NOT REFERENCED??
TOK.OK   022266      LCL       1
TOS      101744      G DAD     2
TOTOK    000244      EQU       1
TOTOK2   000333      EQU       0      <--- NOT REFERENCED??
TRCRTN   025479      LCL       1
TRCVB    025446      ENT       1
TRITEM   025501      LCL       2
TRLBLK   022211      LCL       1
TRLOOP   025456      LCL       1
TRYIN    025010      ENT       1
TSTALO   030755      ENT       1
TTAB.A   177777      EXT       0      <--- NOT REFERENCED??

```

TWOSUB	023548	LCL	0	<--- NOT REFERENCED??	
TOK	020211	LCL	1		
ITM	025314	LCL	2		
TYPSTM	025422	ENT	1		
ULBTOK	000345	EQU	1		
ULNTOK	000347	EQU	1		
UNQUO-	024371	LCL	1		
UNQUO1	024414	LCL	1		
UNQUO2	024420	LCL	2		
UNQUOL	024376	LCL	1		
UNQUOT	024366	ENT	1		
USER#?	024045	LCL	1		
USEX1	027147	LCL	1		
USTOK	000313	EQU	1		
VARTKN	021243	LCL	1		
VLIST	026602	LCL	1		
VRFND	026634	LCL	1		
WARN	177777	EXT	1		
WEP	021703	LCL	1		
ZRO	000000	G DAD	3		

=====	
YARMA 03/14/81=====	
HEADING	PAGE
=====	
Table of Contents	9/11/1981 PG 96
=====	
TABLE OF CONTENTS	1
by Points	2
USER MAIN PROGRAM AND PSETUP	4
PARSIT - MAIN PARSE INNER LOOP	6
P.PARS, C.PARS, E.PARS	8
SCANNER 8-30-80	13
SCAN - NUMBERS	16
GCHAR, ALFA, DIGIT	19
TABLE SEARCH	20
SAVE, RESTORE REGISTERS	24
NUMERIC VALUE	25
LOGICAL PRIMARY	27
NUMERIC EXPRESSION	29
FACTOR	31
GET STANDARD FUNCTION PARAMETER	33
HARREF, GETCHA, GET), SUBSCR	36
NUMERIC REPRESENTATION	37
STRING EXPRESSION	39
STRING REFERENCE	41
Remark and String Constant	43
USE TIME COMMON ROUTINES	46
ASSIGN, RENAME	49
OPTION BASE AND ON KEY #	50
DIMENSION AND TYPE STATEMENTS	51
TRACE VRBL	54
DEF FN STATEMENT	55
FUNCTION NAME SUBROUTINE	58
FUNCTION CALLS FNA(X), FNA\$(X)	59
)UT STATEMENT PARSE	62
DATA Statement Parse	63
READ, READ# PARSE	64
PRINT, PRINT USING, PRINT# PARSE	66
LET STATEMENT 11-1-76	69
FUNCTION LET	70
REF STRING, ARITH VRBL; GOTO, GOSUB	71
IF STATEMENT 11-24-76	72
FOR NEXT STATEMENTS 03-01-77	75
ON GOTO/GOSUB STATEMENT 01-12-77	76
SEQUENCE NUMBERS, GOTO, GOSUB	77
) LINE NUMBER AND RENUMBER	79
EXTERNALS	82

ORGFXX HAD 0 ERRORS 0 WARNINGS 614 LABELS LAST ERROR AT 0