

ENTRY POINTS

<<<<<<<

4	000000	ENT	BINCON
5	000000	ENT	CLEAN
6	000000	ENT	CONST-
7	000000	ENT	CONST
8	000000	ENT	DCLIN#
9	000000	ENT	DCSTR1
10	000000	ENT	DCSTRN
11	000000	ENT	DECOM
12	000000	ENT	DECOM2
13	000000	ENT	EADDR
14	000000	ENT	ELINE#
15	000000	ENT	ELNCAP
16	000000	ENT	EMOYDN
17	000000	ENT	EMOYUP
18	000000	ENT	EXTR1
19	000000	ENT	EXTRJM
20	000000	ENT	FETVAR
21	000000	ENT	FETVA+
22	000000	ENT	FNDLIN
23	000000	ENT	GETNXT
24	000000	ENT	MYLOP
25	000000	ENT	PUMRK
26	000000	ENT	PUMRKR
27	000000	ENT	PU=
28	000000	ENT	SKIP1
29	000000	ENT	SKPBYT
30	000000	ENT	SKPLN2
31	000000	ENT	SKPNAM
32	000000	ENT	SKPNM1
33	000000	ENT	SKPNM+
34	000000	ENT	SCNST
35	000000	ENT	SUBSCR
36	000000	ENT	UNSTAK
37	000000		
38	000000		

>>>> EXTERNALS <<<<<<)

40	000000	EXT	ASCHMI
41	000000	EXT	BOP
42	000000	EXT	CONBIN
43	000000	EXT	CVNUM
44	000000	EXT	ERROR
45	000000	EXT	ERROR+
46	000000	EXT	FBPGM
47	000000	EXT	FXRSET
48	000000	EXT	INTORL
49	000000	EXT	MOVDN
50	000000	EXT	ROMCL-
51	000000	EXT	ROMINI
52	000000	EXT	SKPANM
53	000000		
54	000000		

BLOCK DATA MOVE IN EXTENDED MEMORY

<<<<<<<<

57 000000

57 000000

58 000000

59 000000

60 000000

61 000000

62 000000

63 000000

64 000000

65 000000

66 000000

67 000000

68 000000

69 000000

69 000000

70 000000

70 000000

71 000000

72 000000

73 000000

74 000000

75 000000

76 000000

77 000000

77 000000

78 000000

79 000000

80 000000

81 000000

81 000000

82 000000

83 000000

84 000000

84 000000

85 000000

86 000000

87 000000

88 000000

89 000000

90 000000

91 000000

92 000000

92 000000

93 000000

94 000000

95 000000

96 000000

97 000000

98 000000

99 000000

100 000000

1 000000

1 000000

1 000000

1 000000

1 000000

1 000000

1 000000

1 000000

1 000000

1 000000

1 000000

1 000000

1 000000

1 000000

BLOCK MOVE DOWN IN EXTENDED MEMORY

IN:

PTR1=LWA OF SOURCE

PTR2=LWA OF SINK

R45=BYTE COUNT

OUT:

PTR1=FWA+1 OF SOURCE

PTR2=FWA+1 OF SINK

R45=0

EMOVDN	BSS	0	MOVE BLOCK DOWN
	PUMD	R50,+R6	SAVE R50
EMOVD8	BSS	0	
*****	SBM	R45,=10,0,0	MORE THAN 8 TO MOVE
	OCT	145,315	
EIGHT	OCT	10,0,0	
	JNG	EMOVD-	JIF NO
	LDMI	R50,=PTR1+	MOVE 8 BYTES
	STMI	R50,=PTR2+	
	JMP	EMOVD8	LOOP BACK
EMOVD-	BSS	0	
	ADMD	R45,=EIGHT	RESTORE COUNT
	JZR	ENDMOV	JIF NO MORE
EMOVDL	BSS	0	
	LDBI	R50,=PTR1+	MOVE 1 BYTE
	STBI	R50,=PTR2+	
	DCM	R45	DECR COUNT
	JNZ	EMOVDL	LOOP
ENDMOV	BSS	0	
	POMD	R50,-R6	RESTORE R50
	RTN		DONE

BLOCK MOVE UP IN EXTENDED MEMORY

IN:

PTR1=FWA+1 OF SOURCE

PTR2=FWA+1 OF SINK

R45=BYTE COUNT

OUT:

PTR1=LWA OF SOURCE

```

ITEM      LOC      OBJECT CODE      SRC=CCDCOM  OBJ=GEZDCO      9/11/1981  11:20 AM  PG  4

```

◀◀◀◀◀◀◀◀

PC	OPCODE	INSTR	COMMENT
102	000050		
103	000050		
103	000050		
103	000050		
104	000050		
105	000050		
106	000050	150 006 345	
107	000053		
108	000053	145 325 005	
108	000056	000	
109	000057	364 011	
110	000061	150 271 311	
110	000064	377	
111	000065	273 315 377	
112	000070	360 361	
113	000072		
114	000072	145 323 005	
114	000075	000	
115	000076	367 344	
116	000100		
117	000100	150 270 311	
117	000103	377	
118	000104	272 315 377	
119	000107	145 213	
120	000111	366 365	
121	000113	360 327	
122	000115		

>>> SKIP A LINE <<<<<<<<

```

124 000115
125 000115
125 000115
126 000115
127 000115
128 000115
129 000115
130 000115
131 000115
132 000115
132 000115
132 000115
133 000115
134 000115
135 000115
136 000115 175 006 345
137 000120 316 345 001
138 000123 175 006 343
    000126 236
140 000127

```

```

*****
*
*   SKIP A LINE
*
*   IN:
*   PTR1=POINTS TO STSIZE-1
*
*   OUT:
*   PTR1=POINTS TO NEXT LINE-1
*
*****

```

```

SKPHAM  BSS  0
SKPLN2  BSS  0
FUMD   R75,+R6      SAVE WORKING REG
JSB    =SKPHM1       skip the name
FOMD   R75,-R6      RESTORE REG
RTN

```

FETCH LINE#/ADDRESS

(((((((

10000127							
143000127							
143000127							
144000127							
145000127							
146000127							
147000127							
148000127							
149000127							
150000127							
151000127							
152000127							
153000127							
153000127							
153000127							
154000127							
155000127							
156000127							
157000127							
10000127	165	260	157				
158000132	210						
159000133	367	005					
160000135							
161000135	165	271	311				
161000140	377						
162000141	236						
163000142							
164000142	166	271	311				
164000145	377						
165000146	167	065	242				
166000151	222						
167000152	236						
168000153							
169000153							
170000153	165	271	311				
170000156	377						
171000157	164	260	157				
171000162	210						
10000163	366	006					
173000165	065	240					
174000167	167	242					
175000171	064	240					
176000173	236						
177000174							

FETCH A LINE NUMBER OR AN ADDRESS OR AN INTEGER

IN:

PTR1=POINTS TO LINE#-1 (OR ADDR - 1)

GEMINI=1: IF GEMINI PROGRAM

0: IF CAPRICORN PROGRAM

OUT:

PTR1=POINTS TO STSIZE-1 (OR AFTER ADDR)

R65=LINE NUMBER (OR ADDR)

EADDR BSS 0

ELINE# BSS 0

LDBD R65,=GEMINI GET GEMINI FLAG

JZR ELNCAP JIF CAPRICORN

ELNGEM BSS 0 FETCH GEM LINE#

LDMI R65,=PTR1- LOAD LINE#

RTN

ELNCAP BSS 0 FETCH CAP LINE#

LDMI R66,=PTR1- LOAD 2 BYTES

STB R67,R65 PUT IN ORDER

CLB R67 CLEAR HIGH BYTE

RTN

EBIN BSS 0

LDMI R65,=PTR1- LOAD A BIN CONST

LDBD R64,=GEMINI LOAD FLAG

JNZ EBINX JIF GEMINI

LDB R64,R65 ELSE SWAP R65 & R67

STB R67,R65

LDB R67,R64

EBINX RTN

>>>> FIND A LINE <<<<<<<<

9	000174					
180	000174					
180	000174					
181	000174					
182	000174					
183	000174					
184	000174					
185	000174					
186	000174					
187	000174					
188	000174					
189	000174					
189	000174					
182	000174					
190	000174					
191	000174					
192	000174	230				
193	000175	165 006 345				
194	000200	316 377 377				
195	000203					
196	000203	263 310 377				
197	000206					
198	000206	235				
199	000207					
200	000207					
201	000207					
202	000207	165 271 311				
202	000212	377				
203	000213	231				
204	000214	075 301				
205	000216	230				
206	000217	367 010				
207	000221	373 005				
208	000223	316 115 000				
209	000226	360 356				
210	000230	233				
211	000231					
212	000231					
213	000231	006 343				
214	000233	236				
215	000234					

FIND A LINE		
IN:		
R75=LINE# TO BE FOUND		
OUT:		
PTR1=POINTS TO STSIZE OF THE LINE IF FOUND		
E=0 IF FOUND		
E#0 IF NOT FOUND		
FNDLIN BSS 0		
BIN		FOR SUBTRACTION
PUMD R65,+R6		SAVE REG
JSB =BOP		PT TO BEG OF PRGM
DRP 165		
STMD R65,=PTR1		SET UP POINTER
FNDLOP BSS 0		
CLE		RESET FLAG
***** JSB =ELINE#		FETCH LINE#
LDMI R65,=PTR1-		FETCH LINE#
BCD		
CMM R65,R75		COMPARE
BIN		
JZR FNDRTN		JIF FOUND
JCY FNDEX		JIF PASSED
JSB =3KPLN2		SKIP THIS LINE
JMP FNDLOP		LOOP BACK
FNDEX DCE		SET NOT FOUND
FNDRTN BSS 0		DONE
ORP 165		
POMD R65,-R6		
RTN		

GET NEXT TOKEN

(((((((

```

200 000234
210 000234
218 000234
219 000234
220 000234
221 000234
222 000234
223 000234
224 000234
225 000234
226 000234
227 000234
227 000234
228 000234
229 000234
230 000234 136 261 002
230 000237 140
231 000240 156 223
232 000242 270 311 377
233 000245 262 232 200
234 000250 123 056 240
235 000253 314 371
236 000255 266 030
237 000257
238 000257 262 232 200
239 000262 122 270 311
239 000265 377
240 000266 316 377 377
241 000271 156 270 311
241 000274 377
242 000275 136 261 070
242 000300 210
243 000301 367 022
244 000303 036 265 032
244 000306 000
245 000307
246 000307 156 023 242
247 000312 205
248 000313 036 303
249 000315 056 245
250 000317 136 342
251 000321 317 077 000
252 000324 236
253 000325
254 000325
255 000325
256 000325
257 000325 316 377 377 ERRXX
258 000330 062
259 000331
260 000331

      GETNXT  BSS  0
                LDMD R36,=R50+2  RUN TIME TABLE ADDR
                CLM  R56
                LDBI R56,=PTR1-  GET TOKEN
                STBD R56,=BINFLG  ASSUME NOT BIN PRGM
                LDB  R23,R56      GET ONE MORE COPY
                SBB  R23,=371     BIN PGM
                JNZ  GNX2         JIF NOT
      GNX1      BSS  0
                STBD R#,=BINFLG  SET FLAG
                LDBI R22,=PTR1-  GET BPGM#
                JSB  =FBPGM       GO FIND BIN PRGM
                LDBI R56,=PTR1-  GET BIN TOKEN
                LDMD R36,=BINTAB  BIN PRGM ADDR
                JZR  ERRXX        JIF NOT LOADED
                LDMD R36,X36,THIRT2 RUN TAB BASE ADDR
      GNX2      BSS  0
                STB  R56,R23      SAVE TOKEN
                LLM  R56
                ADM  R56,R36      POINT TO ENTRY
                LDMD R56,R56      GET EXECUTION ADDR
                POBD R36,-R56     ATTRIBUTE
                ANM  R36,=77,0    ISOLATE CLASS
                RTN

      R50+2  EQU  60002
      THIRT2 EQU  32
      JSB  =ERROR+
      OCT  500

```

ARMA 03/14/81 =====
ITEM LOC OBJECT CODE SRC=CCDCOM OBJ=GEZDCO 3/11/1981 11:20 AM PG 9
=====

>>>> DECOMPILE 03-02-77

<<<<<<<

2 000331

DECOMPILE MAIN

CCCCCCCC

```

25 000331
265 000331
265 000331
266 000331
267 000331
269 000331
269 000331
270 000331
271 000331
272 000331
273 000331
273 000331
273 000331
274 000331
275 000331
276 000331 316 035 210
277 000334 136 250 012
278 000337 262 065 210
279 000342 222
280 000343 262 100 202
281 000346 262 264 203
282 000351 175 261 310
282 000354 377
283 000355 263 164 200
284 000360 316 127 000
285 000363 316 026 005
286 000366 175 270 311
286 000371 377
287 000372 261 025 200
288 000375 263 164 210
289 000400 112 263 344
289 000403 203
290 000404 316 024 001
291 000407 117 310 300
292 000412 372 370
293 000414
294 000414 175 261 164
294 000417 210
295 000420 263 025 200
296 000423 236
297 000424
298 000424
299 000424 230
300 000425 316 234 000
301 000430 316 377 377
302 000433 136 205
303 000435 036 265 047
303 000440 001
304 000441 306 000 000
305 000444 236
306 000445
307 000445 377 377
308 000447 205 001

DECOM BSS 0
JSB =DCIDLE JSB DECOM HOOK
LDB R36,=12 DECOM ROMFL
STBD R36,=ROMFL
CLB R36
STBD R36,=ONFLG ZERO "ON" FLAG
STBD R36,=COMFLG ZERO COM FLAG
LDMD R75,=PTR1 SAVE ADR OF BOL
STMD R75,=CURFUN (USED TO SAVE IN PCR)
JSB =ELINE# GET LINE #
JSB =DCLIN# DECOMPILE LINE NUMBER
LDBI R75,=PTR1- SKIP STSIZE
LDMD R75,=LAYAIL SAVE LAYAIL
STMD R75,=SAVLAY
STND R12,=TOS SAVE STACK PTR
DECOM1 JSB =DECOM2
CMB R17,=300 CHECK FOR ERROR
JNC DECOM1 DURING LISTING
RSTLAY BSS 0
LDMD R75,=SAVLAY RESTORE LAYAIL
STMD R75,=LAYAIL
RTN
DECOM2 BSS 0
BIN
JSB =GETNXT NEXT TOKEN
JSB =ROMCL- CIF ROM CLASS
LLM R36
LDMD R36,X36,DCTAB
JSB X36,ZRO
RTN
DCTAB DEF ROMINI -1 ROM CLASS > 56
DEF EOL 0 END OF LINE

```

>>>> DECOMPILE MAIN

<<<<<<<<

000451	044	002	DEF FETVAR	1	FETCH VARIABLE
310 000453	364	001	DEF BINCON	2	BIN INTEGER
311 000455	027	002	DEF STOVAR	3	STORE VARIABLE
312 000457	255	001	DEF CONST	4	NUM FLOAT OR STR CONST
313 000461	327	001	DEF SCHST	5	STRING CONST
314 000463	002	007	DEF UFNCAL	6	USER FUNCTION CALL
315 000465	175	003	DEF JMWL#	7	COND JMP LINE #
316 000467	025	003	DEF GOLINE	10	GOTO GOSUB
317 000471	166	003	DEF JMPREL	11	JMP REL
318 000473	176	007	DEF UFNDEF	12	USER FN DEF
319 000475	271	002	DEF FNEND	13	FNEND
320 000477	163	010	DEF EXTROM	14	EXT ROM
321 000501	337	002	DEF RESWD	15	OPTION BASE
322 000503	274	002	DEF FNRTN	16	USER FN RETURN
323 000505	256	002	DEF FHASGN	17	FN ASSIGN
324 000507	337	002	DEF RESWD	20	DATA
325 000511	306	002	DEF RESWD+	21	DIM/REAL
326 000513	306	002	DEF RESWD+	22	SHORT
000515	306	002	DEF RESWD+	23	INTEGER
328 000517	300	002	DEF RESWD-	24	COM
329 000521	252	003	DEF EJMP#	25	ELSE JMP#
330 000523	216	003	DEF EJMPR	26	ELSE JMP REL
331 000525	276	003	DEF ULIN#	27	USING LINE #
332 000527	373	002	DEF ON	30	ON
333 000531	161	006	DEF PU=	31	STORE
334 000533	065	006	DEF SUBSCR	32	SUBSCR
000535	261	006	DEF INPUTR	33	INPUT
336 000537	102	006	DEF DIMSUB	34	DIM SUBS
337 000541	224	006	DEF PRNEOP	35	PRINT EOL
338 000543	231	006	DEF PRINTS	36	PRINT STUFF
339 000545	024	003	DEF MSCRTH	37	DUMMY
340 000547	024	003	DEF MSCRTH	40	IMMED EXECUTE
341 000551	337	002	DEF RESWD	41	OTHER RESERVED WORDS
342 000553	010	003	DEF MISC	42	MISC OUTPUT
343 000555	146	006	DEF MSTORE	43	MULTI STORE
344 000557	024	003	DEF MSCRTH	44	MISC IGNORE
345 000561	362	006	DEF PRTFUN	45	PRINT FUNCTIONS
000563	024	003	DEF MSCRTH	46	DUMMY
346 000565	024	003	DEF MSCRTH	47	DUMMY
348 000567	334	003	DEF UNOP	50	NUM UNARY OP
349 000571	346	003	DEF BINOP	51	NUM BINARY OP
350 000573	334	003	DEF UNOP\$	52	STR UNARY OP
351 000575	346	003	DEF BINOP\$	53	STR BINARY OP
352 000577	372	002	DEF RWEX	54	DUMMY
353 000601	266	006	DEF SYSFUN	55	NUM FUNC
354 000603	266	006	DEF SYSFN\$	56	STR FUNC

END OF LINE PROCESSING 03-01-77

))))))

```

355 000605
357 000605
357 000605
358 000605
359 000605
360 000605
361 000605
362 000605
363 000605
364 000605
365 000605
366 000605
366 000605
366 000605
367 000605
368 000605
369 000605 316 111 005
370 000610 174 006 343
    000613 030 342
371 000615 310 040
373 000617 367 001
374 000621 344
375 000622
376 000622 236
377 000623
378 000623
378 000623
379 000623
380 000623
381 000623
382 000623
383 000623
384 000623
385 000623
386 000623
387 000623
387 000623
387 000623
387 000623
389 000623
390 000623 170 271 311
390 000626 377
391 000627 100 260 157
391 000632 210
392 000633 766 017
393 000635
394 000635 231
395 000636 100 250 070
396 000641 101 201
397 000643
398 000643 231
399 000644 101 201
400 000646 201

      *****
      *
      *   END OF LINE
      *
      *   IN:
      *       R12=EXPRESSION STACK POINTER
      *       R30=OUTPUT STACK POINTER
      *
      *   OUT:
      *       R30=END OF OUTPUT
      *       DECOMPILED LINE IN OUTPUT STACK
      *
      *****

EOL      BSS  0
        JSB  =UNSTAK
        POMD R74,-R6          TRASH 2 RETURNS
        POBD R74,-R30         POP LAST CHAR
        CMB R74,=40          BLANK?
        JZR EOLEX            JIF BLANK
        FUBD R74,+R30
EOLEX    BSS  0
        RTN                  RETURN TO CALLING ROUTINE

      *****
      *
      *   FETCH A REAL CONSTANT
      *
      *   IN:
      *       PTR1=POINTS TO THE REAL CONSTANT
      *       GEMINI=GEMINI FLAG
      *
      *   OUT:
      *       PTR1=POINTS TO AFTER THE REAL CONSTANT
      *       R70=THE REAL CONSTANT
      *
      *****

EREAL    BSS  0              FETCH A REAL NUMBER
        LDNI R70,=PTR1-
        LDBD R0,=GEMINI      CHECK FLAG
        JNZ ERERTN          JIF GEMINI
        SWAP8 BSS  0         SWAP 8 BYTES
        BCD
        LDB R0,=70          POINT TO 1ST REG
        ELM R*              INITIALIZATION
        SWAP8: BSS  0        MAIN LOOP
        BCD
        ELM R*
        ELM R*
    
```

ARMA 03/14/81=====

ITEM LOC OBJECT CODE SRC=CCDCOM OBJ=GENDCO 9/11/1981 11:20 AM PG 13

=====
>>>> END OF LINE PROCESSING 03-01-77

<<<<<<<<

1	000647	230		BIN	
402	000650	100	210	ICB	R0 INC POINTER
403	000652	377	367	JRN	SWAPS:
404	000654	236		ERERTN	RTH
405	000655				

```

>>>>  CONSTANTS                                03-01-77                                <<<<<<
407 000655
408 000655
409 000655
410 000655
411 000655
412 000655
413 000655
414 000655
415 000655
416 000655
416 000655
416 000655
417 000655
418 000655
419 000655 316 223 001
420 000660 100 250 070
421 000663
422 000663
423 000663
424 000663
425 000663
426 000663
427 000663
428 000663
429 000663
430 000663
431 000663
432 000663
433 000663
434 000663
435 000663
435 000663
435 000663
436 000663
437 000663
438 000663 316 271 001
439 000666
440 000666 101 345
441 000670 236
442 000671
443 000671
443 000671
444 000671
445 000671
446 000671
447 000671
448 000671
449 000671
450 000671
451 000671
452 000671

*****
*
*   DECOMPILE A REAL CONSTANT
*
*   IN:
*       PTR1=POINTS TO THE REAL CONSTANT
*
*   OUT:
*       PTR1=POINTS TO AFTER THE REAL CONSTANT
*       TOKEN & REAL CONSTANT ON EXPRESSION STACK
*
*****

CONST      BSS  0
JSB  =EREAL      GET VALUE
CONST1  LDB R0,=79

*****
*
*   PUSH MARKER, TOKEN, & PARAMETER
*
*   IN:
*       LAVAIL=POINTS TO TOP OF OPERATOR STACK
*       R23=TOKEN
*       R*=PARAMETER
*
*   OUT:
*       LAVAIL=UPDATED
*       A MARK PUSHED ON OPERATOR STACK
*       A MISSING TOKEN ON EXPRESSION STACK
*       INPUT TOKEN ON EXPRESSION STACK
*       PARAMETER IS PUSHED ON EXPRESSION STACK
*
*****

CONST1-  BSS  0
JSB  =PUMRKR      PUSH A MARKER & TOKEN
ARP  !12
PUMD R*,+R12      PUSH PARAMETER
RTN

*****
*
*   PUSH MARKER & TOKEN
*
*   IN:
*       LAVAIL=STACK POINTER OF OPERATOR STACK
*       R23=TOKEN
*
*   OUT:
*       LAVAIL=UPDATED
*       A MARK PUSHED ON OPERATOR STACK

```

```

KARMA 03/14/81=====
ITEM   LOC   OBJECT CODE SRC=CCDCOM OBJ=GEWDCO  9/11/1981 11:20 AM  PG 15
=====
>>>  CONSTANTS                                03-01-77                                <<<<<<<<
3 000671 *                                     A MISSING TOKEN ON EXPRESSION STACK *
454 000671 *                                     INPUT TOKEN ON EXPRESSION STACK *
454 000671 *                                     *
454 000671 *                                     *
455 000671 *                                     *
456 000671 *                                     *
457 000671 316 277 001 PUMRKR BSS 0 *
458 000674 JSB =PUMRK PUSH MARK *
459 000674 122 345 ARP !12 *
460 000676 236 PUMC R22,+R12 PUSH GARBAGE & TOKEN *
461 000677 RTN *
462 000677 *
463 000677 145 261 025 PUMRK BSS 0 *
463 000702 200 LDMD R45,=LAVAIL LOAD OPRTR STK PTR *
464 000703 263 314 377 STMD R45,=PTR2 SET UP POINTER *
465 000706 313 003 000 ADM R45,=3,0,0 INC STK PTR *
465 000711 000 *
465 000712 263 025 200 STMD R45,=LAVAIL UPDATE SP *
465 000715 250 377 LDB R45,=377 MISSING TOKEN MARK *
465 000717 146 012 241 LDM R46,R12 ADDR OF MISSING TOK *
469 000722 145 273 316 STMI R45,=PTR2+ PUSH ON STACK *
469 000725 377 *
470 000726 236 RTN *
471 000727 *
472 000727 *
472 000727 *
3 000727 * DECOMPILE A STRING CONSTANT *
474 000727 * *
475 000727 * IN: *
476 000727 * PTR1=POINTS TO STRING CONSTANT *
477 000727 * *
478 000727 * OUT: *
479 000727 * PTR1=POINTS TO AFTER STRING CONSTANT *
480 000727 * A MARK ON OPERATOR STACK *
481 000727 * MISSING TOKEN ON EXPRESSION STACK *
482 000727 * INPUT TOKEN ON EXPRESSION STACK *
483 000727 * LENGTH OF STRING ON EXPRESSION STACK *
483 000727 * ADDR OF STRING CONST ON EXPRESSION STACK *
484 000727 *
484 000727 *
485 000727 *
486 000727 *
487 000727 175 223 SCHST BSS 0 *
488 000731 270 311 377 CLM R75 *
489 000734 316 047 002 LDBI R75,=PTR1- GET LENGTH *
490 000737 236 JSB =FETVA+ PUSH ON EXPRESSION STACK *
491 000740 RTN *
492 000740 *
492 000740 *
493 000740 * SKIP A FIXED # OF BYTES *
494 000740 * *
495 000740 * IN: *
; 000740 * R75=# OF BYTES TO SKIP *

```

```

>>>          CONSTANTS                      03-01-77                                <<<<<<<
4    000740      *
498  000740      *           OUT:
499  000740      *           PTR1 IS MOVED TO SKIP THE BYTES
499  000740      *
*****
500  000740
501  000740  241      OCT   241
502  000741      SKIP!   BSS   0
503  000741  175 270 311     LDBI R75,=PTR1-   SKIP TOKEN 6
503  000744  377
504  000745
505  000745  230      SKFNM1 BIN
506  000746      SKFNM+  BSS   0
507  000746  175 223      CLM   R75             reset accumulator
508  000750  270 311 377     LDBI R75,=PTR1-   get byte count
509  000753      *****BIN
510  000753      SKPBYT  BSS   0
511  000753  175 215      TCM   R75             NEGATE
512  000755  323 310 377     ADMD R75,=PTR1     GO PASS BYTES
513  000760  263 310 377     STMD R75,=PTR1     UPDATE POINTER
514  000763  236      RTN
515  000764
516  000764
517  000764      *****
517  000764      *
518  000764      *           DECOMPILE AN INTEGER CONSTANT
519  000764      *
520  000764      *           IN:
521  000764      *           PTR1=POINTS TO THE INTEGER CONSTANT
522  000764      *
523  000764      *           OUT:
524  000764      *           PTR1=POINTS TO NEXT TOKEN
524  000764      *
*****
525  000764      BINCON  BSS   0
526  000764  123 310 367     CMB   R23,=367       INDENTATION?
527  000767  367 017         JZR   DCINDT        JIF YES
528  000771  316 153 000     JSB   =EBIH         GET BCD CONST
529  000774  316 377 377     JSB   =INTORL       GO CONVERT
530  000777  230      BIN
531  001000  100 250 060     LDB   R0,=60
532  001003  123 250 004     LDB   R23,=4
533  001006  360 253         JMP   CONS1-        REAL TOKEN
534  001010                                     NOW IT LOOKS LIKE REAL
535  001010      *****
535  001010      *
536  001010      *           DECOMPILE INDENTATION
537  001010      *
538  001010      *           IN:
539  001010      *           PTR1=POINTS TO INDENTATION COUNT
540  001010      *
541  001010      *           OUT:
542  001010      *           PTR1=POINTS TO NEXT TOKEN

```

```

>>>>  CONSTANTS                                03-01-77                                <<<<<<<<
2 001010                                     *
542 001010                                     *
543 001010                                     *
544 001010                                     *
545 001010 157 250 040          DCINDT  BSS  0          LDB  R57,=BLANK      INIT FOR INDENTATION
546 001013 136 270 311          LDBI R36,=PTR1-        POP COUNT
546 001016 377
547 001017 157 030 344  OUTBLK  PUBD  R57,+R30        OUTPUT A BLANK
548 001022 136 212          DCB  R36          DECREMENT COUNT
549 001024 366 371          JNZ  OUTBLK          LOOP BACK
550 001026 236          DCINDN  RTN
551 001027

```

```

553 001027          *
554 001027          *   FETVAR
555 001027          *
556 001027          STOVAR   BSS   0
557 001027 123 310 257      CMB   R23,=257      LABEL?
558 001032 367 022          JZR   DCLABL        JIF YES
559 001034 310 040          CMB   R23,=40
560 001036 366 002          JNZ   STOVA1
561 001040 250 023          LDB   R23,=23
562 001042          STOVA1   BSS   0
563 001042          ORP      123
564 001042 314 020          ORB   R23,=20      LOOK LIKE VALUE
565 001044          STOVA+   BSS   0      DECOM LIKE VALUE
566 001044 316 110 002      FETVAR   JSB   =GETNM
567 001047 316 271 001      FETVA+   JSB   =PUMKR      PUSH MARKER & TOKEN
568 001052 316 133 002      JSB     =PUM      PUSH NAME
569 001055 236          FVRTN   RTN
570 001056
571 001056          DCLABL   BSS   0
572 001056 316 125 002      JSB   =GTNMG+      GET LABEL
573 001061 367 011          JZR   DCLRTN      JIF ZERO LEN
574 001063 316 075 002      JSB   =PUSTR      PUSH OUT LABEL
575 001066 250 072          LDB   R#,=72      PUSH ":"
576 001070 344          PUBD   R#,+R#
577 001071 250 040          LDB   R#,,=BLANK    PUSH ONE BLANK
578 001073 344          PUBD   R#,+R#
579 001074 236          DCLRTN   RTN
580 001075
581 001075          PUSTR     BSS   0
582 001075 170 270 311      LDBI   R70,=PTR1-    GET ONE CHAR
583 001100 377
584 001101 030 344          PUBD   R70,+R30      PUSH OUT
585 001103 175 212          DCB    R75          DEC COUNT
586 001105 366 366          JNZ   PUSTR          GO PUSH MORE
587 001107 236          RTN
588 001110
589 001110          *
590 001110          *   GET A VARIABLE NAME
591 001110          *   IN:
592 001110          *   PTR1=POINTS TO VAR NAME
593 001110          *
594 001110          *   OUT:
595 001110          *   PTR1=POINTS TO VAR NAME
596 001110          *   R75=LENGTH OF NAME
597 001110          *
598 001110          *
599 001110          *
600 001110          *
601 001110          *
602 001110          *
603 001110          *
604 001110          *
605 001110          *
606 001110          *
607 001110          *
608 001110          *
609 001110          *
610 001110          *
611 001110          *
612 001110          *
613 001110          *
614 001110          *
615 001110          *
616 001110          *
617 001110          *
618 001110          *
619 001110          *
620 001110          *
621 001110          *
622 001110          *
623 001110          *
624 001110          *
625 001110          *
626 001110          *
627 001110          *
628 001110          *
629 001110          *
630 001110          *
631 001110          *
632 001110          *
633 001110          *
634 001110          *
635 001110          *
636 001110          *
637 001110          *
638 001110          *
639 001110          *
640 001110          *
641 001110          *
642 001110          *
643 001110          *
644 001110          *
645 001110          *
646 001110          *
647 001110          *
648 001110          *
649 001110          *
650 001110          *
651 001110          *
652 001110          *
653 001110          *
654 001110          *
655 001110          *
656 001110          *
657 001110          *
658 001110          *
659 001110          *
660 001110          *
661 001110          *
662 001110          *
663 001110          *
664 001110          *
665 001110          *
666 001110          *
667 001110          *
668 001110          *
669 001110          *
670 001110          *
671 001110          *
672 001110          *
673 001110          *
674 001110          *
675 001110          *
676 001110          *
677 001110          *
678 001110          *
679 001110          *
680 001110          *
681 001110          *
682 001110          *
683 001110          *
684 001110          *
685 001110          *
686 001110          *
687 001110          *
688 001110          *
689 001110          *
690 001110          *
691 001110          *
692 001110          *
693 001110          *
694 001110          *
695 001110          *
696 001110          *
697 001110          *
698 001110          *
699 001110          *
700 001110          *
701 001110          *
702 001110          *
703 001110          *
704 001110          *
705 001110          *
706 001110          *
707 001110          *
708 001110          *
709 001110          *
710 001110          *
711 001110          *
712 001110          *
713 001110          *
714 001110          *
715 001110          *
716 001110          *
717 001110          *
718 001110          *
719 001110          *
720 001110          *
721 001110          *
722 001110          *
723 001110          *
724 001110          *
725 001110          *
726 001110          *
727 001110          *
728 001110          *
729 001110          *
730 001110          *
731 001110          *
732 001110          *
733 001110          *
734 001110          *
735 001110          *
736 001110          *
737 001110          *
738 001110          *
739 001110          *
740 001110          *
741 001110          *
742 001110          *
743 001110          *
744 001110          *
745 001110          *
746 001110          *
747 001110          *
748 001110          *
749 001110          *
750 001110          *
751 001110          *
752 001110          *
753 001110          *
754 001110          *
755 001110          *
756 001110          *
757 001110          *
758 001110          *
759 001110          *
760 001110          *
761 001110          *
762 001110          *
763 001110          *
764 001110          *
765 001110          *
766 001110          *
767 001110          *
768 001110          *
769 001110          *
770 001110          *
771 001110          *
772 001110          *
773 001110          *
774 001110          *
775 001110          *
776 001110          *
777 001110          *
778 001110          *
779 001110          *
780 001110          *
781 001110          *
782 001110          *
783 001110          *
784 001110          *
785 001110          *
786 001110          *
787 001110          *
788 001110          *
789 001110          *
790 001110          *
791 001110          *
792 001110          *
793 001110          *
794 001110          *
795 001110          *
796 001110          *
797 001110          *
798 001110          *
799 001110          *
800 001110          *
801 001110          *
802 001110          *
803 001110          *
804 001110          *
805 001110          *
806 001110          *
807 001110          *
808 001110          *
809 001110          *
810 001110          *
811 001110          *
812 001110          *
813 001110          *
814 001110          *
815 001110          *
816 001110          *
817 001110          *
818 001110          *
819 001110          *
820 001110          *
821 001110          *
822 001110          *
823 001110          *
824 001110          *
825 001110          *
826 001110          *
827 001110          *
828 001110          *
829 001110
```

```

=====
> >>>      FETCH VARIABLE      03-01-77      <<<<<<<<
  1 001116 223      CLM  R75
602 001117 250 002      LDB  R75,=2      SET LENGTH=2
603 001121 236      RTN
604 001122      GETNMG  BSS  0
605 001122      ORP  175
606 001122 271 311 377      LDMI R75,=PTR1-      SKIP ADDR
607 001125 175 223      GTNMG+ CLM  R75
608 001127 270 311 377      LDBI R75,=PTR1-      GET LENGTH
609 001132 236      RTN
610 001133
611 001133      *****
611 001133      *
612 001133      *      IN:
613 001133      *      PTR1=POINTS TO NAME
614 001133      *      R75=LENGTH OF NAME
615 001133      *
616 001133      *      OUT:
617 001133      *      PTR1=POINTS TO AFTER NAME
618 001133      *      ADDR OF NAME ON EXPRESSION STACK
618 001133      *
619 001133      *****
620 001133      PUNM  BSS  0
621 001133 166 075 241      ARP  112
622 001136 012 345      LDM  R66,R75      GET LENGTH
623 001140 165 261 310      PUMD R66,+R12      PUSH LHTH ON EXP STK
        ) 001143 377      LDMD R65,=PTR1      GET ADDR
624 001144 345      PUMD R65,+R12      PUSH ADDR ON EXP STK
625 001145 316 353 001      JSB  =SKP9YT      SKIP TO PASS NAME
626 001150 236      RTN

```

```

>>>> BASIC RES WORDS&FN END 03-09-77
628 001151 *
629 001151 * RESWD
630 001151 *
631 001151 120 223 MYWRD CLM R20
632 001153 023 240 LDB R20,R23
633 001155 213 DCM R20 BIAS FOR ROM
634 001156 126 261 004 LDMD R26,=R50+4 ASCII TAB ADDR
634 001161 140
635 001162 177 260 232 LDBD R77,=BINFLG BIN PROG
635 001165 200
636 001166 366 013 JNZ MYNBIN JIF NOT
637 001170 126 261 070 LDMD R26,=BINTAB
637 001173 210
638 001174 THIRT4 DAD 34
639 001174 126 026 265 LDMD R26,X26,THIRT4 ASCII TABLE
639 001177 034 000
640 001201 360 012 JMP MYW1
641 001203 177 260 000 MYNBIN LDBD R77,=R50K
641 001206 140
642 001207 366 004 JNZ MYW1 JIF NOT ROM 0
643 001211 120 315 044 SBM R20,=44,0 BIAS FOR SYSTEM
643 001214 000
644 001215 MYW1 BSS 0
645 001215 230 BATS BIN
646 001216 120 036 243 STM R20,R36 SAVE BIASED TOKEN
647 001221 136 213 POSLP DCM R36
648 001223 364 012 JNG MYLOP JIF WE ARE THERE
649 001225 132 026 340 POSLP1 POBD R32,+R26 BASIC TAB BYTE
650 001230 365 373 JPS POSLP1 LOOP IF NO PARITY
651 001232 317 177 000 ANM R32,=177,0 TRASH PARITY
652 001235 360 362 JMP POSLP JIF YES
653 001237 132 026 340 MYLOP POBD R32,+R26 POP ASCII
654 001242 024 242 STB R32,R24 SAVE IT
655 001244 317 177 000 ANM R32,=177,0 TRASH PARITY
656 001247 030 344 PUBD R32,+R30 AND PUSH IT
657 001251 124 220 TSB R24
658 001253 365 362 JPS MYLOP JIF MORE
659 001255 236 RTN
660 001256
661 001256 316 151 002 FHASGN JSB =MYWRD
662 001261 123 270 311 LDBI R23,=PTR1- 1 OR 3 TOKEN
662 001264 377
663 001265 316 044 002 JSB =FETVAR
664 001270 236 RTN
665 001271
666 001271 316 151 002 FNEHD JSB =MYWRD
667 001274 FNRTN BSS 0
668 001274 316 127 000 JSB =EADDR SKIP RETURN ADDR
669 001277 236 RTN
670 001300
671 001300 123 262 264 RESWD- STBD R23,=COMFLG SET COM FLAG
671 001303 203
672 001304 360 031 JMP RESWD

```

```

=====
KARMA 03/14/81=====
ITEM      LOC      OBJECT CODE  SRC=CCDCOM OBJ=GEZDCO  9/11/1981 11:20 AM  PG 21
=====
>>>>      BASIC RES WORDS&FN END 03-09-77                                     <<<<<<<<
3 001306 120 260 264 RESWD+  LDBD R20,=COMFLG
673 001311 203
674 001312 367 023          JZR RESWD          JIF NOT COMMON
675 001314 112 321 344      CMMD R12,=TOS      ANYTHING ON STAK
675 001317 203
676 001320 367 015          JZR RESWD          JIF NO
677 001322 123 006 344      PUBD R23,+R6      SAVE TOKEN
678 001325 316 111 005      JSB =UNSTAK
679 001330 123 250 054      LDB R23,=COMMA
680 001333 030 344          PUBD R23,+R30
681 001335 006 342          POBD R23,-R6      RESTORE TOKEN
682 001337 123 310 233 RESWD CMB R23,=233    BANG?
683 001342 766 006          JNZ RESWD1        JIF NO
684 001344 316 111 005      JSB =UNSTAK      ELSE UNSTAK
685 001347 123 250 233      LDB R23,=233    RELOAD BANG
686 001352 316 151 002 RESWD1 JSB =MVWRD
687 001355 132 250 040      LDB R32,=40
688 001360 030 344          PUBD R32,+R30
689 001362 136 222          RWRTN+          CLB R36
690 001364 262 100 202 RWRTN STBD R#,=0HFLG
691 001367 316 111 005      JSB =UNSTAK
692 001372 236          RWEX          RTN
693 001373 316 151 002 ON      JSB =MVWRD
694 001376 132 250 040      LDB R32,=40
695 001401 030 344          PUBD R32,+R30
696 001403 136 023 240      LDB R36,R23
697 001406 360 354          JMP RWRTN
698 001410 123 310 100 MISC    CMB R23,=100    @?
699 001413 766 004          JNZ MISCPU        JIF NO
700 001415 012 344          PUBD R23,+R12      PUSH TO STAK
701 001417 360 341          JMP RWRTN+        GO UNSTAK
702 001421 123 030 344 MISCPU PUBD R23,+R30
703 001424 236          MSCRTN          RTN

```

```

>>>> GOTO & GOSUB 03-01-77 <<<<<<<
700 001425 *
706 001425 * GOLINE
707 001425 *
708 001425 137 260 100 GOLINE LDBD R37,=ONFLG
709 001430 202
709 001431 367 006 JZR GOL1 JIF NOT ON
710 001433 364 016 JNG DDCOMA JIF NOT FIRST
711 001435 214 TCB R37
712 001436 262 100 202 STBD R37,=ONFLG
713 001441 316 151 002 GOL1 JSB =MYMRD
714 001444 132 250 040 LDB R32,=40 BLANK
715 001447 030 344 PUBD R32,+R30
716 001451 360 006 JMP DOLINE
717 001453 137 030 342 DDCOMA FDBD R37,-R30
718 001456 250 054 LDB R37,=COMMA
719 001460 344 PUBD R37,+30
720 001461 DOLINE BSS 0
721 001461 123 310 262 CMB R23,=262 GOTO/GOSUB LABEL?
722 001464 373 016 JCY DOLABL JIF YES
723 001466 316 116 003 JSB =DCLIN+ DCOM LINE#
724 001471 136 270 313 DCLIN+ LDBI R36,=PTR1-+ NEXT TOKEN
724 001474 377
725 001475 310 016 CMB R36,=16
726 001477 366 002 JNZ GRTH
727 001501 130 213 DCM R30 TRASH BLANK
728 001503 236 GRTH RTN
729 001504
730 001504 DOLABL BSS 0
731 001504 316 262 003 JSB =DCJLAB PUSH OUT LABEL
732 001507 132 250 040 LDB R32,=40 PUSH OUT BLANK
733 001512 030 344 PUBD R32,+R30
734 001514 360 353 JMP DCLIN+
735 001516
736 001516 DCLIN+ BSS 0
737 001516 316 127 000 JSB =ELINE# GET LINE #
738 001521 165 221 TSM R65 ALLOCATED ?
739 001523 365 011 JPS DCLIN1 JIF YES
740 001525 323 006 200 ADMD R65,=FWCURR MAKE IT ABS ADDR
741 001530 263 314 377 STMD R65,=PTR2 SET POINTER
742 001533 271 315 377 LDMI R65,=PTR2- GET LINE #
743 001536 DCLIN1 BSS 0
744 001536 316 026 005 JSB =DCLIN# DECOMP LINE #
745 001541 236 RTN
746 001542

```

```

>>>>>>  RELATIVE JUMPS      03-01-77      <<<<<<<<
001542      *
749 001542      *  JMPREL
750 001542      *
751 001542      RSTL#      EQU 334      RESTORE LINE # TOKEN
752 001542
753 001542 175 251 111 DCIF      LDM  R75,=111,106,040      "IF "
753 001545 106 040
754 001547 030 345      PUMD R75,+R30
755 001551 316 111 005      JSB  =UNSTAK
756 001554 173 251 124      LDM  R73,=124,110,105,116,040      "THEN "
756 001557 110 105 116
756 001562 040
757 001563 030 345      JMPEX+  PUMD R#,+R30
758 001565 236      JMPRTN  RTN
759 001566
760 001566      JMPREL  BSS  0
761 001566 316 142 003      JSB  =DCIF      OUTPUT IF PART
762 001571 316 127 000      JSB  =EADDR      SKIP REL ADDR
763 001574 236      RTN
764 001575
765 001575      JMPL#    BSS  0
766 001575 123 006 344      PUBD R23,+R6      SAVE R23
767 001600 316 142 003      JSB  =DCIF      OUTPUT IF PART
768 001603 123 006 342      POBD R23,-R6
769 001606 310 041      CMB  R23,=41      JUMP LABEL?
770 001610 367 050      JZR  DCJLAB      JIF YES
771 001612
772 001612 316 116 003      JMPL#+  BSS  0
773 001615 236      JSB  =DCLIN+      DCOMPILE LINE #
774 001616      RTN
775 001616      EJMPR    BSS  0
776 001616 316 127 000      JSB  =EADDR      SKIP REL ADDR
777 001621 175 270 313      LDBI R75,=PTR1-+      NEXT TOK TO 75
777 001624 377
778 001625 310 037      CMB  R75,=37      ELSE JMP # ???
779 001627 367 020      JZR  JMPEXX      YES, THEN NO ELSE
780 001631 310 043      CMB  R75,=43      ELSE JMP LABEL?
781 001633 367 014      JZR  JMPEXX      JIF YES
782 001635      PUELSE  BSS  0
783 001635 316 111 005      JSB  =UNSTAK      OUTPUT THEN PART
784 001640 173 251 105      LDM  R73,="ELSE "
784 001643 114 123 105
784 001646 040
785 001647 030 345      PUMD R73,+R30      OUTPUT "ELSE"
786 001651      JMPEXX  BSS  0
787 001651 236      RTN
788 001652
789 001652      EJMP#    BSS  0
790 001652 316 235 003      JSB  =PUELSE      OUT THEN PART & "ELSE"
791 001655 123 310 043      CMB  R23,=43      ELSE JUMP LINE #
792 001660 366 330      JNZ  JMPL#+      JIF YES
793 001662      DCJLAB  BSS  0
794 001662 316 110 002      JSB  =GETNM      GET LABEL LEN/ADDR

```

```

>>>>      RELATIVE JUMPS      03-01-77      <<<<<<<<
796 001665 367 006      JZR DCJRTN      JIF ZERO LEN
796 001667 316 075 002  JSB  =FUSTR      PUSH OUT LABEL
797 001672 250 040      LDB  R#,,=40      R75
798 001674 344      PUBD R#,,+R#      PUSH A BLANK
799 001675 236      DCJRTN RTN
800 001676
801 001676
802 001676 123 310 334 ULIN#  CMB R23,=RSTL#      RESTORE LINE # ?
803 001701 367 015      JZR ULIN#-
804 001703 310 335      CMB  R23,=335      RESTORE LABEL?
805 001705 367 023      JZR  ULABE-      JIF YES
806 001707 310 345      CMB  R23,=345      USING LABEL?
807 001711 367 011      JZR  ULABEL      JIF YES
808 001713 250 313      LDB  R23,=USTOK
809 001715 316 352 002  JSB  =RESWD1      DECOMPILE USING
810 001720      ULIN#-  BSS  0
811 001720 316 116 003  JSB  =DCLIN+      DECOMPILE LINE #
812 001723 236      RTN
813 001724      USTOK  EQU 313
814 001724
815 001724
816 001724      ULABEL  BSS  0
817 001724 123 250 313  LDB  R23,=USTOK      SET "USING"
818 001727 316 352 002  JSB  =RESWD1      OUTPUT KEYWORD
819 001732      ULABE-  BSS  0
820 001732 360 326      JMP  DCJLAB      OUTPUT LABEL
821 001734
822 001734

```

```

>>>> OPERATORS 03-01-77 <<<<<<<<
001734 *LST OPERATORS
325 001734 *
326 001734 * OPER
327 001734 *
328 001734 * INPUT
329 001734 *
330 001734 * R23 TOKEN
331 001734 * R56 POINTER TO LWA ATTRIBUTES
332 001734 *
333 001734 UNOP$ BSS 0
334 001734 155 056 342 UNOP POBD R55,-R56 OPER PREC
335 001737 * LDB R32=40
336 001737 * PUBD R32,+R30 PUSH A BLANK
337 001737 316 205 004 JSB =STAKB+
338 001742 316 363 003 JSB =PRETS1
339 001745 236 RTN
340 001746 BINOP$ BSS 0
341 001746 155 056 342 BINOP POBD R55,-R56 OPER PREC
342 001751 054 242 STB R55,R54 SAVE PREC
343 001753 * LDB R32,=40
344 001753 * PUBD R32,+R30
345 001753 316 245 004 JSB =STAKB
346 001756 316 363 003 PRETST JSB =PRETS1
347 001761 * LDB R32,=40
348 001761 * PUBD R32,+R30
349 001761 260 015 JMP PRETS2
350 001763 154 210 PRETS1 ICB R54 PREC + 1 FOR >= TEST
351 001765 134 056 243 STM R34,R56 SAVE LOC
352 001770 012 241 LDM R34,R12
353 001772 136 213 DCM R36
354 001774 316 021 004 JSB =OPTST
355 001777 236 RTN
356 002000
357 002000
358 002000 155 310 006 PRETS2 CMB R55,=6
359 002003 367 002 JZR PRTS2-
360 002005 154 212 DCB R54 RESTORE PREC
361 002007 134 002 241 PRTS2- LDM R34,R2 RESTORE ORIG R36
362 002012 316 373 004 JSB =POSIT
363 002015 316 021 004 JSB =OPTST
364 002020 PRETS3 BSS 0
365 002020 236 RTN
366 002021
367 002021 OPTST BSS 0
368 002021 175 261 025 LDMD R75,=LAVAIL GET PTR TO TOS
368 002024 200
369 002025 263 314 377 STMD R75,=PTR2 SETUP PTR
370 002030 OPTLOP BSS 0
371 002030 165 261 314 LDMD R65,=PTR2 GET CUR PTR
371 002033 377
372 002034 321 164 210 CMMD R65,=SAVLAV END OF STK?
373 002037 367 031 JZR OPTRTH EXIT IF YES
4 002041 271 315 377 LDNI R65,=PTR2- NEXT PREC/LOC

```

```

ARMA 03/14/81=====
ITEM   LOC   OBJECT CODE SRC=CCDCOM OBJ=GENDCO  9/11/1981 11:20 AM PG 26
=====
>>>>> OPERATORS 03-01-77 <<<<<<<
3 002044 166 036 301 CMM R66,R36
376 002047 372 357 JNC OPTLOP JIF STAK <= LOW
377 002051 367 355 JZR OPTLOP
378 002053 034 301 CMM R66,R34
379 002055 373 351 JCY OPTLOP JIF STAK > HI
880 002057 165 054 300 CMB R65,R54 STAK PREC < CURR PREC
881 002062 373 344 JCY OPTLOP JIF NO
882 002064 316 073 004 JSB =EXPER
883 002067 316 136 004 JSB =CLEAN
884 002072 OPTRTH BSS 0
885 002072 236 RTN
886 002073 132 223 EXPER CLM R32
887 002075 211 ICM R32
888 002076 316 337 004 JSB =EXPAND
889 002101 136 211 ICM R36
890 002103 170 250 050 LDB R70,=OPEN
891 002106 036 344 PUBD R70,+R36
892 002110 136 006 345 PUMD R36,+R6 SAVE R36 FOR CLEAN
893 002113 034 241 LDM R36,R34
894 002115 132 223 CLM R32
895 002117 211 ICM R32
896 002120 316 337 004 JSB =EXPAND
897 002123 136 211 ICM R36
898 002125 170 250 051 LDB R70,=CLOSE
899 002130 036 344 PUBD R70,+R36
900 002132 136 006 343 POMD R36,-R6 RESTORE R36
901 002135 236 RTN
902 002136 CLEAN BSS 0
903 002136 165 261 025 LDMD R65,=LAVAIL LOAD OPRTR SP
903 002141 200
904 002142 263 314 377 STMD R65,=PTR2 SET UP POINTER
905 002145 260 030 JMP CLNCMP ANYTHING ON STAK ?
906 002147 CLNLOP BSS 0
907 002147 ORP 165
908 002147 271 315 377 LDMD R65,=PTR2- NEXT PREC/LOC
909 002152 166 036 301 CMM R66,R36 STAK - LOW
910 002155 372 011 JNC CLNEX JIF STAK LO OR EQ
911 002157 367 007 JZR CLNEX
912 002161 034 301 CMM R66,R34
913 002163 373 003 JCY CLNEX JIF STAK HI
914 002165 165 250 100 LDB R65,=100 HIGHEST PREC + SOME
915 002170 CLNEX BSS 0
916 002170 165 273 314 STMI R65,=PTR2 STORE BACK
916 002173 377
917 002174 261 314 377 LDMD R65,=PTR2 GET POINTER
918 002177 CLNCMP BSS 0
919 002177 ORP 165
920 002177 321 164 210 CMM R65,=SAVLAV BOTTOM OF STACK?
921 002202 266 343 JNZ CLNLOP JIF NOT
922 002204 236 CLHRTN RTN
923 002205 316 373 004 STAKB+ JSB =POSIT NEXT MARKER
924 002210 132 223 CLM R32
925 002212 211 ICM R32

```

```

CARMA 03/14/81=====
ITEM   LOC   OBJECT CODE SRC=CCDCOM OBJ=GEZDCO  9/11/1981 11:20 AM PG 27
=====
>>>> OPERATORS 03-01-77 <<<<<<<<
927 002213 316 337 004 JSB =EXPAND
928 002216 136 211 ICM R36
928 002220 056 243 STM R36,R56 SAVE LOCATION
929 002222 175 261 025 LDMD R75,=LAVAIL PUSH A POINTER
929 002225 200
930 002226 263 314 377 STMD R75,=PTR2
931 002231 211 ICM R75 UPDATE STACK POINTER
932 002232 211 ICM R75
933 002233 211 ICM R75
934 002234 263 025 200 STMD R75,=LAVAIL
935 002237 155 273 314 STMI R55,=PTR2 PUSH PREC & ADDR
935 002242 377
936 002243 760 007 JMP STAKB1
937 002245 316 373 004 STAKB JSB =POSIT
938 002250 155 272 314 STBI R55,=PTR2 STORE PREC
938 002253 377
939 002254 136 006 345 STAKB1 PUMD R36,+R5 SAVE R36
940 002257 130 345 PUMD R30,+R5 SAVE R30
941 002261 316 151 002 JSB =MYWRD
942 002264 102 250 040 LDB R2,=BLANK APPEND A BLANK AT THE END
943 002267 030 246 STBD R2,R30
944 002271 130 032 243 STM R30,R32 NEW R30
945 002274 006 343 POMD R30,-R6 RESTORE ORIG R30
946 002276 136 343 POMD R36,-R6 RESTORE R36
947 002300 132 030 305 SBM R32,R30 LEN OF OPER
948 002303 213 DCM R32
949 002304 367 011 JZR NOEXPR JIF OPER 1 BYTE
950 002306 211 ICM R32 ADD ONE BLANK
951 002307 006 345 PUMD R32,+R5 SAVE LEN
952 002311 316 337 004 JSB =EXPAND MAKE ROOM
953 002314 132 006 343 POMD R32,-R6 LEN AGAIN
954 002317 136 002 243 NOEXPR STM R36,R2 SAVE R36
955 002322 134 030 241 LDM R34,R30
956 002325 177 034 340 DOITI1 POBD R77,+R34 NEXT BYTE OF OPER
957 002330 036 344 PUBD R77,+R36 TO STAK
958 002332 132 213 DCM R32
959 002334 365 367 JFS DOITI1 LOOP IF MORE
960 002336 236 RTN
961 002337
962 002337
963 002337 *****
964 002337 *KYWDOP BSS 0 *
965 002337 * PUMD R32,+R5 SAVE LENGTH *
966 002337 * JSB =EXPAND MAKE ROOM *
967 002337 * POMD R32,-R6 RESTORE LENGTH *
968 002337 * DCM R32 DEC LENGTH *
969 002337 * LDB R77,=BLANK GET A BLANK *
970 002337 * PUBD R77,+R36 PUSH OUT *
971 002337 * JMP NOEXPR PUSH OPERATOR *
972 002337 *****

```

>>>> SOME USEFUL UTILITIES 03-01-77 <<<<<<<<

9 002337 *
975 002337 * EXPAND
976 002337 *
977 002337 123 006 344 EXPAND FUBD R23,+R6 SAVE TOKEN
978 002342 124 345 FUMD R24,+R6 SAVE R24
979 002344 122 012 241 LDM R22,R12
980 002347 036 305 SBM R22,R36 # BYTES TO MOVE
981 002351 124 012 241 LDM R24,R12 LWA + 1 SOURCE
982 002354 126 241 LDM R26,R12
983 002356 032 303 ADM R26,R32 LWA + 1 SINK
984 002360 112 303 ADM R12,R32 UPDATE R12
985 002362 316 377 377 JSB =MOVDN
986 002365 124 006 343 POMD R24,-R6 RESTORE R24
987 002370 123 342 POBD R23,-R6 RESTORE TOKEN
988 002372 236 RTN

989 002373 *****
989 002373 *
990 002373 * LOCATE THE LAST MISSING TOKEN *
991 002373 * *
991 002373 * IN: *
993 002373 * PTR2=POINTS TO TOP OF OPERATOR STACK *
994 002373 * *
995 002373 * OUT: *
996 002373 * PTR2=POINTS TO THE LAST MISSING TOKEN *
997 002373 * R36=ADDR OF THE MISSING TOKEN *
997 002373 * *****

998 002373
999 002373 POSIT BSS 0
1000 002373 145 261 025 LDMD R45,=LAVAIL GET SP
1000 002376 200
1001 002377 263 314 377 STMD R45,=PTR2 SET UP FOR SEARCH
1002 002402 PLOOP BSS 0
1003 002402 136 271 315 LDMI R36,=PTR2- GET LOC
1003 002405 377
1004 002406 145 270 315 LDBI R45,=PTR2- GET PREC
1004 002411 377
1005 002412 365 366 JPS PLOOP JIF NOT MARKER
1006 002414 236 RTN

1007 002415 *****
1008 002415 *
1008 002415 * POSIT2 locates last missing token then unmark it. *
1009 002415 * *
1009 002415 *****

1010 002415
1011 002415 POSIT2 BSS 0
1012 002415 316 373 004 JSB =POSIT
1013 002420 ORP 145
1014 002420 250 100 LDB R45,=100 UNMARK THE MISSING TOKEN
1015 002422 272 314 377 STBI R45,=PTR2
1016 002425 236 RTN
1017 002426

>>>> LINE NUMBERS 03-01-77 <<<<<<<<

```

002426
1020 002426
1020 002426
1021 002426
1022 002426
1023 002426
1024 002426
1025 002426
1026 002426
1027 002426
1028 002426
1028 002426
1028 002426
1029 002426
1030 002426
1031 002426
1032 002426 231
1033 002427 157 250 006
1034 002432 165 221
1035 002434 366 005
1036 002436 157 250 001
1037 002441 360 016
1038 002443
1039 002443 167 207
1040 002445
1041 002445 167 220
1042 002447 377 010
1043 002451 164 205
1044 002453 157 212
1045 002455 360 366
1046 002457
1047 002457 164 205
1048 002461
1049 002461 167 317 017
1050 002464
1051 002464 310 012
1052 002466 366 001
1053 002470 212
1054 002471 167 312 060
1055 002474 030 344
1056 002476 157 212
1057 002500 366 355
1058 002502
1059 002502 157 250 040
1060 002505 030 344
1061 002507 230
1062 002510 236
1063 002511

```

```

*****
*
*
* DCLIN#
*
* INPUT:
* LINE # IN R65
*
* OUTPUT:
* ASCII LINE # ON R30 STACK
*
*****

```

```

DCLIN# BSS 0
BCD
LDB R57,=6 EXP
TSM R65 ZERO LINE #
JNZ SLOP0 JIF NO
LDB R57,=1
JMP LINLP1

```

```

SLOP0 LRM R67 INITIALIZE
SLOP ESS 0
TSB R67 NORMALIZED?
JRN LINLP1 JIF YES
LLM R64 SHIFT UP
DCB R57 DEC EXP
JMP SLOP

```

```

SDON BSS 0
LLM R64 GET NEXT DIGIT
LINLP1 BSS 0
ANM R67,=17 CONVERT TO ASCII
* TEST FOR 12 OCTAL AND MAKE IT 9 BCD
CMB R67,=12 END STOP MISSING LINE?
JNZ LINLP2 JIF NO
DCB R67 ELSE MAKE IT 11
ADB R57,=60 MAKE ASCII
PUBD R67,+R30 PUSH IT OUT
DCB R57 DEC EXP
JNZ SDON JIF MORE

```

```

BLK BSS 0
LDB R57,=BLANK ONE BLANK AFTER #
PUBD R57,+R30
BIN
RTN

```

```

>>>> PROCESS OPER STAK 03-01-77 <<<<<<<
000 002511
066 002511
066 002511
067 002511
068 002511
069 002511
069 002511
069 002511
070 002511 177 222
071 002513 262 030 377
072 002516 316 014 006
073 002521 112 321 344
073 002524 203
074 002525 367 137
075 002527 136 250 017
076 002532 012 344
077 002534 112 261 344
077 002537 203
079 002540 123 340
080 002542 310 016
080 002544 766 003
081 002546 123 012 340 UNSLOP
082 002551 123 310 017 UNSL1
083 002554 367 077
084 002556 310 273
085 002560 372 025
086 002562 316 305 005
087 002565 176 251 050
087 002570 051
088 002571 123 220
089 002573 176
090 002574 362 005
091 002576 175 251 050
091 002601 054 051
092 002603 030 345
093 002605 360 337
094 002607
095 002607 310 004
096 002611 765 005
097 002613 316 305 005
098 002616 360 326
099 002620 366 005
100 002622 316 364 005
101 002625 360 317
102 002627 123 310 007 HTNCON
103 002632 365 005
104 002634 316 374 005
105 002637 360 305
106 002641
107 002641 123 310 016 HTBANG
108 002644 366 002
109 002646 250 054
110 002650 123 030 344 PUSHIT

```

```

*
*
* UNSTAK
*
*
*****
UNSTAK CLB R77
STBD R77,=RSELEC ROM 0
JSB =CLNOPS
CMMO R12,=TOS
JZR UNSRT1 JIF STACK EMPTY
LDB R36,=17
PUBD R36,+R12
LDMD R12,=TOS
POBD R23,+R12 SCRAP FIRST MARKER
CMB R23,=16
JNZ UNSL1
POBD R23,+R12 NEXT TOKEN
CMB R23,=17
JZR UNSRTH
CMB R23,=273 A*( ),A*( ),AK ),AK( ), ?
JNC NOT364 JIF NO
JSB =DCVAR SPIT VARIABLE NAME
LDM R76,=50,51 SPIT ( )
TSB R23 1-D ?
ORP 76
JOD AA+ JIF YES
LDM R75,=50,54,51 SPIT (, ) FOR 2-D
AA+ PUND R5,+R30
JMP UNSLOP
NOT364 BSS 0
CMB R#,=4
JPS HTYAR JIF NOT VARIABLE
JSB =DCVAR
JMP UNSLOP
HTYAR JNZ HTNCON JIF NOT NUM CONST
JSB =DCNCON
JMP UNSLOP
HTNCON CMB R23,=7
JPS HTSCON JIF NOT STR CONST
JSB =DCSCON
JMP UNSLOP
HTSCON BSS 0
HTBANG CMB R23,=16
JNZ PUSHIT JIF NOT SEP
LDB R23,=CONMA
PUBD R23,+R30

```

```

>>>> PROCESS OPER STAK 03-01-77 <<<<<<<<
002653 760 271 JMP UNSLOP LOOP IF MORE
112 002655 112 261 344 UNSRTN LDM R12,=TOS RESET R12
112 002660 203
113 002661 134 250 040 LDB R34,=40
114 002664 030 344 PUBD R34,+R30 PUSH A BLANK
115 002666 236 UNSRT1 RTN
116 002667 *
117 002667 * DCVAR
118 002667 *
119 002667
120 002667 DCVARC BSS 0
121 002667 ORP 173
122 002667 012 341 POMD R73,+R12 GET LENGTH & NAME
123 002671 175 263 314 STMD R75,=PTR2 SET ADDR
123 002674 377
124 002675 176 271 315 LDHI R76,=PTR2- GET REVERSED NAME
124 002700 377
125 002701 030 345 PUMD R76,+R30 PUSH OUT
125 002703 360 011 JMP DCVAR1
125 002705
128 002705 DCVAR BSS 0
129 002705 173 260 157 LDBD R73,=GEMINI CAPRICORN PRGM ?
129 002710 210
130 002711 367 354 JZR DCVARC JIF YES
131 002713 316 335 005 JSB =DCSTRN SPIT NAME
132 002716 DCVAR1 BSS 0
133 002716 123 310 003 CMB R23,=3 STRING VAR?
134 002721 372 011 JNC STORTN JIF NO
135 002723 310 340 CMB R23,=340 NUMERICAL ARRAY ?
136 002725 373 005 JCY STORTN JIF YES
137 002727 122 250 044 LDB R22,=DOLLAR SPIT $
138 002732 030 344 PUBD R22,+R30
139 002734 236 STORTN RTN
140 002735
141 002735 DCSTRN BSS 0
142 002735 173 012 341 POMD R73,+R12 POP LENGTH & LOC
143 002740 DCSTR1 BSS 0
144 002740 175 263 314 STMD R75,=PTR2 SET POINTER
144 002743 377
145 002744 176 073 241 LDM R76,R73 ZERO LENGTH?
146 002747 367 363 JZR STORTN JIF YES
147 002751 DCSLOP BSS 0
148 002751 175 270 315 LDBI R75,=PTR2- FETCH A CHAR
148 002754 377
149 002755 030 344 PUBD R75,+R30 PUSH OUT
150 002757 176 213 DCM R76 DEC COUNT
151 002761 366 366 JNZ DCSLOP LOOP BACK
152 002763 236 DCSRTN RTN
153 002764
154 002764 *
155 002764 * DCNCON
156 002764 *
157 002764 140 012 341 DCNCON POMD R40,+R12 NUM CONST

```

```

=====
> >>> PROCESS OPER STAK 03-01-77 <<<<<<<
1158 002767 316 377 377 JSB =CVNUM
1159 002772 230 BIN
1160 002773 236 RTN
1161 002774
1162 002774 *
1163 002774 * DCSCON
1164 002774 *
1165 002774
1166 002774 DCSCON BSS 0
1167 002774 123 310 005 CMB R23,=5 QUOTED STRING?
1168 002777 366 334 JNZ DCSTRN JIF NOT
1169 003001 250 042 LDB R23,=QUOTE SPIT QUOTE
1170 003003 030 344 PUBD R23,+R30
1171 003005 316 335 005 JSB =DCSTRN SPIT STRING
1172 003010 123 030 344 PUBD R23,+R30 SPIT ANOTHER QUOTE
1173 003013 236 RTN
1174 003014
1175 003014 230 CLNOPS BIN
1176 003015 175 261 025 LDMD R75,=LAYAIL LOAD OPRTR SP
1177 003020 200
1178 003021 263 314 377 STMD R75,=PTR2 SET POINTER
1179 003024 316 014 001 JSB =RSTLAY RESTORE LAYAIL
1180 003027 DRP 175
1181 003027 321 314 377 CMMD R75,=PTR2 ANYTHING ON OPRTR STK?
1182 003032 367 030 JZR CLEXIT JIF NO
1183 003034 160 250 054 LDB R60,=COMMA LOAD COMMA
1184 003037 UNMRK BSS 0
1185 003037 165 271 315 LDMD R65,=PTR2- FETCH PREC/LOC
1186 003042 377
1187 003043 220 TSB R65 IS IT A MARK?
1188 003044 365 003 JPS UNMRK1 JIF NO
1189 003046 160 066 246 STBD R60,R66 PUSH A COMMA
1190 003051 UNMRK1 BSS 0
1191 003051 175 321 314 CMMD R75,=PTR2 END OF OPRTR STACK
1192 003054 377
1193 003055 366 360 JNZ UNMRK LOOP IF NOT
1194 003057 160 250 016 LDB R60,=16 SET 1ST MISSING TOKEN
1195 003062 066 246 STBD R60,R66
1196 003064 236 CLEXIT RTN
1197 003065

```

>>>>		MISC TOKENS		03-21-77		<<<<<<<<	
5	003065						
197	003065			*	MISC		
198	003065			*			
199	003065	176	251	050	SUBSCR	LDM R76,="()"	
199	003070	051					
200	003071	123	220			TSB R23	
201	003073	362	015			JDD CAL1	
202	003075	316	124	006	CAL2	JSB =TWO DIM	
203	003100	360	013			JMP TSTST0	
204	003102	176	251	133	DIMSUB	LDM R76,="[]"	
204	003105	135					
205	003106	123	220			TSB R23	
206	003110	363	363			JEV CAL2	JIF TWO DIM
207	003112	316	134	006	CAL1	JSB =ONEDIM	
208	003115	134	012	241	TSTST0	LDM R34,R12	
209	003120	316	136	004		JSB =CLEAN	
210	003123	236				RTN	
211	003124	316	015	005	TWO DIM	JSB =POSIT2	
212	003127	132	250	054		LDB R32,=COMMA	
213	003132	036	344			PUBD R32,+R36	
214	003134	316	015	005	ONEDIM	JSB =POSIT2	
215	003137	176	036	344		PUBD R76,+R36	
216	003142	177	012	344		PUBD R77,+R12	
217	003145	236				RTN	
218	003146	177	250	054	MSTOR	LDB R77,=COMMA	
219	003151	124	270	313		LDBI R24,=PTR1-+	GET NEXT TOKEN
220	003154	377					
220	003155	023	300			CMB R24,R23	A MULTI STORE TOO?
221	003157	367	012			JZR MSTOR-	JIF YES
222	003161	177	250	075	PU=	LDB R77,=75	"="
223	003164	316	173	006		JSB =MSTOR-	
224	003167	316	111	005		JSB =UNSTAK	
225	003172	236				RTN	
226	003173				MSTOR-	BSS 0	
227	003173	165	261	164		LDMD R65,=SAVLAY	GET ADDR OF MARKERS
227	003176	210					
228	003177	313	003	000		ADM R65,=3,0,0	SKIP 1ST MARKER
229	003202	000					
229	003203	263	314	377		STMD R65,=PTR2	SETUP POINTER
230	003206				FMARK	BSS 0	
231	003206					DRP 165	
232	003206	271	316	377		LDMD R65,=PTR2+	POP NEXT ENTRY
233	003211	220				TSB R65	IS IT A MARKER?
234	003212	765	372			JPS FMARK	JIF NO
235	003214	222				CLB R65	RESET MARK
236	003215	273	317	377		STMI R65,=PTR2-+	
237	003220	177	066	246		STBD R77,R66	PUSH DATA
238	003223	236				RTN	
239	003224						
240	003224	134	250	040	FRNEOP	LDB R34,=40	BLANK
241	003227	760	022			JMP ITSCOM	
242	003231	123	006	344	PRINTS	PUBD R23,+R6	
243	003234	316	111	005		JSB =UNSTAK	

```
>>>> MISC TOKENS 03-21-77 <<<<<<<<
244 003237 230 BIN
245 003240 134 250 054 LDB R34,=COMMA
246 003243 123 006 342 POBD R23,-R6
247 003246 363 003 JEV ITSCOM
248 003250 134 250 073 LDB R34,=73 ASCII ;
249 003253 130 213 ITSCOM DCM R30 TRASH WHATEVER
250 003255 134 030 344 PUBD R34,+R30
251 003260 236 RTN
252 003261 136 270 311 INPUTR LDBI R36,=PTR1- SKIP STORE TOKEN
252 003264 377
253 003265 236 RTN
```

>>>> SYSTEM FUNCTIONS 03-01-77 <<<<<<<<

1255	003266					*		
1256	003266					*	SYSFUN	
1257	003266					*		
1258	003266					*	INPUT	
1259	003266					*		
1260	003266					*	R23 TOKEN	
1261	003266					*	R56 POINTER TO LWA ATTRIBUTES	
1262	003266					*		
1263	003266					SYSFN#	BSS 0	
1264	003266	132	250	054		SYSFUN	LDB R32,=COMMA	LOAD UP
1265	003271	177	056	342			POBD R77,-R56	PARAM COUNT
1266	003274	231					BCD	
1267	003275	206					LRB R77	ISOLATE PARAM COUNT
1268	003276	230					BIN	
1269	003277	212					DCB R77	NO COMMAS IF ZERO OR ONE
1270	003300	265	014				JPS SYFN1	JIF ONE OR MORE
1271	003302	316	277	001			JSB =PUMRK	
1272	003305	112	211				ICM R12	
1273	003307	036	243				STM R12,R36	
1274	003311	211					ICM R12	
1275	003312	316	254	004			JSB =STAKB1	GO MOVE FUNC NAME
1276	003315	236					RTN	AND EXIT
1277	003316					SYFN1	BSS 0	
1278	003316	177	212				DCB R77	
1279	003320	264	010				JNG SYFN2	JIF DONE
1280	003322	316	015	005			JSB =POSIT2	FIND MARKER
1281	003325	132	036	344			PUBD R32,+R36	PUSH COMMA
1282	003330	360	364				JMP SYFN1	
1283	003332	316	373	004		SYFN2	JSB =POSIT	
1284	003335	134	012	241			LDM R34,R12	
1285	003340	316	136	004			JSB =CLEAN	
1286	003343	155	250	020			LDB R55,=20	PRECEDENCE OF SYS FUNC
1287	003346	316	205	004			JSB =STAKB+	GO MOVE FUN NAME
1288	003351	134	012	241			LDM R34,R12	
1289	003354	136	213				DCM R36	
1290	003356	316	073	004			JSB =EXPER	GO DO PARENS
1291	003361	236					RTN	
1292	003362	316	266	006		PRTFUN	JSB =SYSFUN	DO COMMON PART
1293	003365	136	271	311			LDMI R36,=PTR1-	NEXT TWO BYTES
1294	003370	377						
1295	003371	311	000	006			CMR R36,=0,6	NULL STR ?
1296	003374	367	003				JZR PRTFEX	JIF YES
1297	003376	273	312	377			STMI R36,=PTR1+	ELSE RESTORE IT
1298	003401	236				PRTFEX	RTN	
1299	003402					F	EQU 106	
	003402					N	EQU 116	

```

=====
ARMA 03/14/81=====
ITEM LOC OBJECT CODE SRC=CCDCOM OBJ=GEZDCO 9/11/1981 11:20 AM PG 36
=====
>>> USER FUNCTION CALLS 03-01-77 <<<<<<
1301 003402 *
1302 003402 * UFNCAL
1303 003402 *
1304 003402 316 110 002 UFNCAL JSB =GETNM NAME
1305 003405 166 075 241 LDM R66,R75 2-BYTE LENGTH OF NAME
1306 003410 006 345 PUMD R66,+R6 SAVE IT
1307 003412 165 261 310 LDMD R65,=PTR1 GET ADDR
1307 003415 377
1308 003416 345 PUMD R65,+R6 SAVE ADDR
1309 003417 316 353 001 JSB =3KPSYT SKIP THE NAME
1310 003422 177 270 311 LDBI R77,=PTR1- GET PAR COUNT
1310 003425 377
1311 003426 367 061 JZR FN0 JIF NO PARAMS
1312 003430 PLOP BSS 0
1313 003430 136 270 311 LDBI R36,=PTR1- THROW AWAY PAR TYPE
1313 003433 377
1314 003434 177 212 DCB R77 DEC PAR COUNT
1315 003436 367 012 JZR PLOP1 JIF LAST PAR
1316 003440 316 015 005 JSB =POSIT2
1317 003443 134 250 054 LDB R34,=CONMA
1318 003446 036 344 PUBD R34,+R36
1319 003450 360 356 JMP PLOP LOOP IF MORE
1320 003452 PLOP1 BSS 0
1321 003452 316 373 004 JSB =POSIT LOCATE LAST MISSING TOKEN
1322 003455 173 006 343 POMD R73,-R6 RESTORE FN NAME
1323 003460 316 130 007 JSB =FNNAM PACK FN NAME
1324 003463 316 337 004 JSB =EXPAND GO EXPAND IT
1325 003466 134 012 241 LDM R34,R12 CLEAN UP FIRST
1326 003471 316 136 004 JSB =CLEAN BEFORE R36 GET CHANGED
1327 003474 136 211 ICM R36 POINT TO INSERT LOC
1328 003476 170 036 345 PUMD R70,+R36 INSERT FUNC NAME
1329 003501 250 050 LDB R70,=OPEN INSERT A "<"
1330 003503 344 PUBD R70,+R36
1331 003504 250 051 LDB R70,=CLOSE PUT A ">"
1332 003506 012 344 PUBD R70,+R12
1333 003510 236 RTN
1334 003511 FN0 BSS 0
1335 003511 316 277 001 JSB =PUMRK PUSH MARKER
1336 003514 112 211 ICM R12
1337 003516 173 006 343 POMD R73,-R6 RESTORE FN NAME
1338 003521 316 130 007 JSB =FNNAM
1339 003524 170 012 345 PUMD R70,+R12 PUSH FN NAME
1340 003527 236 RTN
1341 003530 *
1342 003530 * FNNAM
1343 003530 *
1344 003530 FNNAM BSS 0
1345 003530 166 251 106 LDM R66,="FN"
1345 003533 116
1346 003534 070 243 STM R66,R70 PACK IT
1347 003536 123 314 024 SBB R23,=24 LOOK LIKE VAR NAME
1348 003541 072 242 STB R23,R72 PACK IT
1349 003543 132 223 CLM R32 SET EXPAND SIZE = 9

```

```

=====
PARMA 03/14/81=====
ITEM   LOC   OBJECT CODE   SRC=CCDCOM OBJ=GEZDCO   9/11/1981 11:20 AM PG 37
=====
>>>>      USER FUNCTION CALLS      03-01-77      <<<<<<<<
0 003545 250 011      LDB R32,=9D
1351 003547 236      RTN
1352 003550 040 075 040 BEQB      ASC 3, =
1353 003553 104 105 106 DEF      ASC 6,DEF FH
1353 003556 040 106 116

```

```

PARMA 03/14/81:=====
ITEM    LOG    OBJECT CODE  SRC=CCDCOM OBJ=GENDCO  9/11/1981 11:20 AM  PG 38
=====
>>>>  USER FUNCTION DEF  03-01-77  <<<<<<<<
1355 003561      *
1356 003561      *  UFNDEF
1357 003561      *
1358 003561
1359 003561      UFNDFC  BSS  0
1360 003561      ORP  176
1361 003561      ARP  130
1362 003561 271 311 377      LDMI R76,=PTR1-  GET REVERSED NMAE
1363 003564 345      PUMD R76,+R30  PUSH OUT
1364 003565 271 311 377      LDMI R76,=PTR1-  SKIP JUMP ADDR
1365 003570 123 270 311      LDBI R23,=PTR1-  GET PAR COUNT
1365 003573 377
1366 003574 260 053      JMP  SKPOK2
1367 003576
1368 003576      UFNDEF  BSS  0
1369 003576 172 261 153      LDMD R72,=DEF  GET "DEF FN"
1369 003601 007
1370 003602 030 345      PUMD R72,+R30  PUSH OUT
1371 003604 176 260 157      LDBD R76,=GEMINI  IS IT CAPRICORN?
1371 003607 210
1372 003610 367 347      JZR  UFNDFC  JIF YES
1373 003612      *****
1374 003612 175 271 311      LDMI R75,=PTR1-  MAY NEED TO SKIP ADDR
1374 003615 377
1375 003616      *****
1376 003616 270 311 377      LDBI R75,=PTR1-  GET CHAR COUNT
1377 003621      NMLoop  BSS  0  PUSH OUT FUNC NAME
1378 003621 177 270 311      LDBI R77,=PTR1-  GET ONE CHAR
1378 003624 377
1379 003625 030 344      PUBD R77,+R30  PUSH OUT
1380 003627 175 212      DCB  R75  DEC COUNT
1381 003631 366 366      JNZ  NMLoop
1382 003633 271 311 377      LDMI R75,=PTR1-  SKIP JUMP ADDR
1383 003636 123 270 311      LDBI R23,=PTR1-  GET PAR CNT/STR FLG
1383 003641 377
1384 003642 170 023 240      LDB  R70,R23  COPY IT
1385 003645 206      LRB  R70  SHIFT OUT STR FLG
1386 003646 316 377 377      JSB  =SKPANM  SKIP PAR NAMES
1387 003651      SKPOK2  BSS  0
1388 003651 123 220      TSB  R23
1389 003653 263 005      JEV  NOSTR  JIF NOT STRING
1390 003655 177 250 044      LDB  R77,=DOLLAR  PUSH OUT A "$"
1391 003660 030 344      PUBD R77,+R30
1392 003662      NOSTR  BSS  0
1393 003662 134 023 240      LDB  R34,R23
1394 003665 206      LRB  R34  PAR COUNT
1395 003666 367 033      JZR  NOPARS  JIF ZERO COUNT
1396 003670 135 250 050      LDB  R35,=OFEN  PUSH OUT A "<"
1397 003673 030 344      PUBD R35,+R30
1398 003675      PARLOP  BSS  0  PAR LOOP
1399 003675 316 277 001      JSB  =PUMRK  PUSH A MARKER
1400 003700 112 211      ICM  R12  HOLD THE PLACE
1401 003702 316 352 007      JSB  =PUPAR  PUSH A PARAMETER

```

```

ARMA 03/14/81=====
ITEM   LOC   OBJECT CODE SRC=CCDCOM OBJ=GENDCO  9/11/1981 11:20 AM PG 39
=====
>>>>  USER FUNCTION DEF      03-01-77      <<<<<<<<
1  2 003705 134 212          DCB R34          DEC COUNT
1403 003707 266 364          JNZ PARLOP        JIF NOT ZERO
1404 003711 135 250 051      LDB R35,=CLOSE
1405 003714 012 344          PUBD R35,+R12
1406 003716 316 111 005      JSB =UNSTAK
1407 003721 130 213          DCM R30          THROW AWAY LAST CHAR
1408 003723 175 261 150 NOPARS LDMD R75,=BEQB  GET " = "
1408 003726 907
1409 003727 136 270 313      LDBI R36,=PTR1++ GET NEXT TOKEN
1409 003732 377
1410 003733 310 016          CMB R36,=16      EOL?
1411 003735 367 004          JZR NOEQU        JIF YES
1412 003737 310 233          CMB R36,=233      "!" ?
1413 003741 266 003          JNZ PUEQU        JIF NO
1414 003743 177              NOEQU      ORP 77      MULTI LINE DEF FN
1415 003744 260 001          JMP PUEQ1
1416 003746 175              PUEQU      ORP 75      SINGLE LINE DEF FN
1417 003747 030 345          PUEQ1      PUMD R#,+R30
1418 003751 236              RTN
1419 003752
1420 003752              PUPAR      BSS 0          PUSH A PAR
1421 003752 170 250 001      LDB R70,=1      PRESUME SIMPLE VAR
1422 003755 165 251 010      LDM R65,=10,0,0 PRESUME LENGTH IS 8
1422 003760 000 000
1423 003762 137 270 313      LDBI R37,=PTR1++ GET PAR TYPE
1423 003765 377
1424 003766 265 003          JPS NSTR1        JIF NOT STRING
1425 003770 170 250 003      LDB R70,=3      IT'S STRING
1426 003773              NSTR1      BSS 0
1427 003773 170 012 344      PUBD R70,+R12  PUSH OUT
1428 003776 171 260 157      LDBD R71,=GEMINI IS IT GEMINI?
1428 004001 210
1429 004002 367 033          JZR CAPAR        JIF NOT
1430 004004 137 270 311      LDBI R37,=PTR1- SKIP PAR TYPE
1430 004007 377
1431 004010 175 271 311      LDMI R75,=PTR1- GET REL ADDR
1431 004013 377
1432 004014 323 164 200      ADMD R75,=CURFUN GET ABS ADDR
1432 004017 263 314 377      STMD R75,=PTR2  SETUP POINTER
1434 004022 176 223          CLM R76
1435 004024 270 315 377      LDBI R76,=PTR2- GET LEN OF NAME
1436 004027 345              PUMD R76,+R12
1437 004030 175 261 314      LDMD R75,=PTR2  GET ADDR
1437 004033 377
1438 004034 345              PUMD R75,+R12  PUSH ADDR
1439 004035 260 030          JMP CKSTR        GOTO MORE CHECK
1440 004037              CAPAR      BSS 0
1441 004037 136 271 311      LDMI R36,=PTR1- GET VAR NAME
1441 004042 377
1442 004043 316 377 377      JSB =ASCNM1      UNPACK IT
1443 004046 136 273 310      STMI R36,=PTR1  PUT IT BACK
1443 004051 377
1444 004052 173 223          CLM R73

```

>>>		USER FUNCTION DEF	03-01-77	<<<<<<<<
445	004054	211	ICM R73	LENGTH=2
446	004055	211	ICM R73	
447	004056	175 261 310	LDMD R75,=PTR1	GET ADDR
447	004061	377		
448	004062	211	ICM R75	
449	004063	211	ICM R75	BACK TO BEGIN OF NAME
450	004064	173 012 345	PUMD R73,+R12	PUSH OUT
451	004067		BSS 0	
452	004067	170 310 003	CMB R70,=3	IS IT A STRING?
453	004072	366 060	JNZ PUPARE	JIF NOT
454	004074	316 127 000	JSB =EADDR	GET TOTAL LEN
455	004077	174 271 311	LDMI R74,=PTR1-	SKIP MAX, ACT LEN
455	004102	377		
456	004103	137 260 157	LDBD R37,=GEMINI	IS IT GEMINI
456	004106	210		
457	004107	367 005	JZR CKSTR1	JIF NO
458	004111	176 065 243	STM R76,R65	USE MAX LEN
459	004114	167 222	CLB R67	
459	004116	175 251 135 CKSTR1	LDM R75,=135,133,4	GET JC & REAL TOK
460	004121	133 004		
461	004123	136 065 241	LDM R36,R65	GET LOW ORDER 2 BYTE
462	004126	311 022 000	CMM R36,=22,0	DEFAULT LENGTH?
463	004131	367 021	JZR PUPARE	JIF YES
464	004133	134 006 345	PUMD R34,+R6	SAVE R34
465	004136	316 377 377	JSB =CONBIN	CONVERT TO REAL
466	004141	230	BIN	
467	004142	176 012 345	PUMD R76,+R12	PUSH I4
468	004145	140 345	PUMD R40,+R12	PUSH REAL CONST
469	004147	175 344	PUBD R75,+R12	PUSHJ
470	004151	134 006 343	POMD R34,-R6	RESTORE R34
471	004154		BSS 0	
472	004154	175 065 241	LDM R75,R65	GET LENGTH OF PAR
473	004157	316 353 001	JSB =3KPBVT	SKIP THE PAR
474	004162	236	RTN	

```

>>>>>      EXTERNAL ROMS      <<<<<<<<
1  004163      *
1477 004163      * EXTR0M
1478 004163      *
1479 004163 136 251 024 EXTR0M LDM R36,=DECOM2
1479 004166 001
1480 004167 230      EXTR0M BIN
1481 004170 176 270 311      LDBI R76,=PTR1-      GET ROM #
1481 004173 377
1482 004174 310 360      CMB R76,=360      IS IT PLOTTER ROM#
1483 004176 366 011      JNZ EXTR0      JIF NOT
1484 004200 102 260 157      LDBD R2,=GEMINI      IS IT CAPRICORN ?
1484 004203 210
1485 004204 366 003      JNZ EXTR0      JIF NOT
1486 004206 176 250 001      LDB R76,=1      USE BUILT IN GRAPHICS
1487 004211      EXTR0      BSS 0
1488 004211 176 262 030      STBD R76,=RSELEC      ROM SELECT
1488 004214 377
1489 004215 320 000 140      CMBD R76,=R60K      IS ROM THERE
1  004220 366 012      JNZ ERR0M
1490 004222 036 306 000      JSB X36,ZR0      DO IT
1491 004225 000
1492 004226 177 222      EXTR1      CLB R77
1493 004230 262 030 377      STBD R77,=RSELEC      SELECT ROM 0
1494 004233 236      RTN
1495 004234 316 377 377 ERR0M      JSB =ERROR
1496 004237 025      OCT 21D      ROM MISSING
1  004240 360 364      JMP EXTR1
  
```

>>> BASIC TOKEN DECOMPILE 2-2-77

))))))

```

499 004242      *
500 004242      *
501 004242      *   BATS - BASIC TOKEN DECOMPILE
502 004242      *
503 004242      *   ROUTINE TO SEARCH THE PARSING TABLE
504 004242      *   AND BASTAB TO CONVERT TOKENS > 100
505 004242      *   OCTAL TO BASIC VERBS,
506 004242      *
507 004242      *
508 004242      *   THE ROUTINE IS ENTERED WITH THE
509 004242      *   FOLLOWING REGISTERS SET.
510 004242      *
511 004242      *   R20 - BIASED TOKEN
512 004242      *   R26 - BASE ADDRESS OF BASIC TAB
513 004242      *   R30 - OUTPUT BUFFER POINTER
514 004242      *
515 004242      *
516 004242      *   ENT POSLP
517 004242      *   ENT DCSLOP
518 004242      *   UNIL FOR FIN
519 004242      *   FIN
    
```

```

=====
AA+      002603      LCL      1
4- NM1    177777      EXT      1
5        001215      LCL      0      <--- NOT REFERENCED??
BEQB      003550      LCL      1
BINCON     000764      ENT      2
BINFLG     100232      G DAD      3
BINOP      001746      LCL      1
BINOP*     001746      LCL      1
BINTAB     104079      G DAD      2
BLANK      000040      G EQU      4
BLK        002502      LCL      0      <--- NOT REFERENCED??
BOP        177777      EXT      1
CAL1       003112      LCL      1
CAL2       003075      LCL      1
CAPAR      004037      LCL      1
CKSTR      004067      LCL      1
CKSTR1     004116      LCL      1
CLEAN      002136      ENT      5
CLEXIT     003064      LCL      1
CLNCMP     002177      LCL      1
CL EX      002179      LCL      3
CL OP      002147      LCL      1
CLNOPS     003014      LCL      1
CLRRTN     002204      LCL      0      <--- NOT REFERENCED??
CLOSE      000051      G EQU      3
COMFLG     101664      G DAD      3
COMMA      000054      G EQU      9
CONBIN     177777      EXT      1
C PI-     000663      ENT      2
CONST      000655      ENT      2
CONST1     000660      LCL      0      <--- NOT REFERENCED??
CURFUN     100164      G DAD      2
CVNUM      177777      EXT      1
DCIDLE     104035      G DAD      1
DCIF       001542      LCL      2
DCINDN     001026      LCL      0      <--- NOT REFERENCED??
DCINDT     001010      LCL      1
DCJLAB     001662      LCL      3
DCJRTN     001675      LCL      1
DC'ABL     001056      LCL      1
DC N#      002426      ENT      3
DCLIN+     001516      LCL      3
DCLINI     001536      LCL      1
DCLRTH     001074      LCL      1
DCNCON     002764      LCL      1
DCSCON     002774      LCL      1
DCSLOP     002751      ENT      2
DCSRTN     002763      LCL      0      <--- NOT REFERENCED??
DCSTR1     002740      ENT      1
DCSTRN     002735      ENT      4
DCTAB      000447      LCL      1
DCVAR      002705      LCL      2
DCVAR1     002716      LCL      1
DCVARC     002667      LCL      1
=====

```

ECOM	000331	ENT	1		
ECOM1	000404	LCL	1		
ECOM2	000424	ENT	3		
EF	003553	LCL	1		
IMSUB	003102	LCL	1		
OCOMA	001453	LCL	1		
OITI	002325	LCL	1		
OLABL	001504	LCL	1		
OLIN+	001471	LCL	1		
OLINE	001461	LCL	1		
OLLAR	000044	G EQU	2		
ADDR	000127	ENT	5		
BIN	000153	LCL	1		
BINX	000173	LCL	1		
BIGHT	000005	LCL	3		
BUMF#	001652	LCL	1		
BUMPR	001616	LCL	1		
ELINE#	000127	ENT	3		
ELNCAP	000142	ENT	2		
ELNGEM	000135	LCL	0	<--- NOT REFERENCED??	
EMOVD-	000023	LCL	1		
EMOVD8	000003	LCL	1		
EMOVDL	000031	LCL	1		
EMOVDN	000000	ENT	1		
EMOYU-	000072	LCL	1		
EMOYUS	000053	LCL	1		
EMOYUL	000100	LCL	1		
EMOYUP	000050	ENT	1		
ENDMOV	000044	LCL	3		
EOL	000605	LCL	1		
EOLX	000622	LCL	1		
EREAL	000623	LCL	1		
ERERTN	000654	LCL	1		
ERROM	004234	LCL	1		
ERROR	177777	EXT	1		
ERROR+	177777	EXT	1		
ERRXX	000325	LCL	1		
EXPAND	002337	LCL	5		
EXPER	002073	LCL	2		
EXTR0	004211	LCL	2		
EX1	004226	ENT	2		
EXTRJM	004167	ENT	1		
EXTROM	004163	LCL	1		
F	000106	EQU	0	<--- NOT REFERENCED??	
FPGM	177777	EXT	1		
FETVA+	001047	ENT	2		
FETVAR	001044	ENT	3		
FMARK	003206	LCL	1		
FNO	003511	LCL	1		
FNASGN	001256	LCL	1		
FNDX	000230	LCL	1		
FNDLIN	000174	ENT	1		
FNDLOP	000206	LCL	1		
FNDRTN	000231	LCL	1		

PNEND	001271	LCL	1		
PAM	003530	LCL	2		
PN	001274	LCL	1		
PVRTN	001055	LCL	0	<--- NOT REFERENCED??	
PMCURR	100006	G DAD	1		
PRRSET	177777	EXT	0	<--- NOT REFERENCED??	
SEHINI	104157	G DAD	9		
GETHM	001110	LCL	3		
GETHMC	001116	LCL	0	<--- NOT REFERENCED??	
GETHMG	001122	LCL	1		
GETNXT	000234	ENT	2		
GNX1	000257	LCL	0	<--- NOT REFERENCED??	
GNX2	000307	LCL	1		
GOL1	001441	LCL	1		
GOLINE	001425	LCL	1		
GRTN	001503	LCL	1		
GTNMG+	001125	LCL	1		
INPUTR	003261	LCL	1		
INTORL	177777	EXT	1		
ITSCOM	003253	LCL	2		
MPREX+	001563	LCL	0	<--- NOT REFERENCED??	
MPXX	001651	LCL	2		
MPPL#	001575	LCL	1		
MPPL#+	001612	LCL	1		
MPREL	001566	LCL	1		
MPRTH	001565	LCL	0	<--- NOT REFERENCED??	
LAVAIL	100025	G DAD	10		
LINLP1	002461	LCL	2		
L } LP2	002471	LCL	1		
MISC	001410	LCL	1		
MISCPU	001421	LCL	1		
MOVDN	177777	EXT	1		
MSCRTH	001424	LCL	5		
MSTOR	003146	LCL	1		
MSTOR-	003173	LCL	2		
MVLOP	001237	ENT	3		
MYNBIN	001203	LCL	1		
MYW1	001215	LCL	2		
MYWRD	001151	LCL	6		
N	000116	EQU	0	<--- NOT REFERENCED??	
N OP	003621	LCL	1		
NOEQU	003743	LCL	1		
NOEXPR	002317	LCL	1		
NOPARS	003723	LCL	1		
NOSTR	003662	LCL	1		
NOT364	002607	LCL	1		
NSTR1	003773	LCL	1		
NTSANG	002641	LCL	0	<--- NOT REFERENCED??	
NTNCON	002627	LCL	1		
NTSCON	002641	LCL	1		
NTVAR	002620	LCL	1		
ON	001373	LCL	1		
ONEDIM	003134	LCL	1		
ONFLG	101100	G DAD	4		

OPEN	000050	G EQU	3	
OPLOP	002030	LCL	4	
OPRTN	002072	LCL	1	
OPTST	002021	LCL	2	
OUTBLK	001017	LCL	1	
PARLOP	003675	LCL	1	
PLGOP	002402	LCL	1	
PLOP	003430	LCL	1	
PLOP1	003452	LCL	1	
POSIT	002373	LCL	6	
POSIT2	002415	LCL	4	
POSLP	001221	ENT	2	
POSLP1	001225	LCL	1	
PRETS1	001763	LCL	2	
PRETS2	002000	LCL	1	
PRETS3	002020	LCL	0	<--- NOT REFERENCED??
PRETST	001756	LCL	0	<--- NOT REFERENCED??
PRINTS	003231	LCL	1	
PRNEOP	003224	LCL	1	
PRTFEX	003401	LCL	1	
P3FUN	003362	LCL	1	
PR2-	002007	LCL	1	
PTR1	177710	G DAD	8	
PTR1+	177712	G DAD	3	
PTR1-	177711	G DAD	36	
PTR1-+	177713	G DAD	5	
PTR2	177714	G DAD	20	
PTR2+	177716	G DAD	4	
PTR2-	177715	G DAD	11	
PTR2-+	177717	G DAD	1	
PU=	003161	ENT	2	
PUElse	001635	LCL	1	
PUEQ1	003747	LCL	1	
PUEQU	003746	LCL	1	
PUMRK	000677	ENT	5	
PUMRKR	000671	ENT	3	
PUM	001133	LCL	1	
PUPAR	003752	LCL	1	
PUPARE	004154	LCL	2	
PUSHIT	002650	LCL	1	
PLR	001075	LCL	3	
QUOTE	000042	G EQU	1	
R60+2	060002	EQU	1	
R60+4	060004	G DAD	1	
R60K	060000	G DAD	2	
RESWD	001337	LCL	6	
RESWD+	001306	LCL	3	
RESWD-	001300	LCL	1	
RESWD1	001352	LCL	3	
ROMCL-	177777	EXT	1	
ROMFL	104065	G DAD	1	
ROMINI	177777	EXT	1	
RSELEC	177430	G DAD	3	
RSTL#	000334	EQU	1	

SYMBOL	VALUE	TYPE	COUNT	Symbol Table	9/11/1981 11:20 AM PG 47
BSTLAY	000414	LCL	1		
DEFX	001372	LCL	1		
DEFN	001364	LCL	1		
BWRTN+	001362	LCL	1		
BAYLAY	104164	G CAD	5		
BCHST	000727	ENT	2		
BDOH	002457	LCL	1		
BKIP!	000741	ENT	1		
BKPANM	177777	EXT	1		
BKPBYT	000753	ENT	4		
BKPLN2	000115	ENT	2		
BKPNAM	000115	ENT	1		
BKPNM+	000746	ENT	1		
BKPNM1	000745	ENT	2		
BKPOK2	003651	LCL	1		
BLOP	002445	LCL	1		
BLOP0	002443	LCL	1		
BTAKE	002245	LCL	1		
BTAKE+	002205	LCL	2		
BTAKE1	002254	LCL	2		
BWRTN	002734	LCL	3		
BWA+	001044	LCL	0	<--- NOT REFERENCED??	
BTOVA1	001042	LCL	1		
BTOVAR	001027	LCL	1		
BUBSOR	003065	ENT	2		
BWAP8	000635	LCL	0	<--- NOT REFERENCED??	
BWAP8:	000643	LCL	1		
BYFN1	003316	LCL	2		
BYFN2	003332	LCL	1		
BYFN\$	003265	LCL	1		
BYSFUN	003265	LCL	2		
BHIRT2	000032	EQU	1		
BHIRT4	000034	CAD	1		
BOS	101744	G CAD	5		
BSTSTO	003115	LCL	1		
BMODIM	003124	LCL	1		
BFNAL	003402	LCL	1		
BFNDEF	003576	LCL	1		
BFNDFC	003561	LCL	1		
BABE-	001732	LCL	1		
BABE	001724	LCL	1		
BULIN#	001676	LCL	1		
BULIN#-	001720	LCL	1		
BUNMRK	003037	LCL	1		
BUNMRK1	003051	LCL	1		
BUNOP	001734	LCL	1		
BUNOP\$	001734	LCL	1		
BUNSL1	002551	LCL	1		
BUNSLOP	002546	LCL	5		
BUNSR1	002665	LCL	1		
BUNSRN	002655	LCL	1		
BUNSTAK	002511	ENT	10		
BUSTOK	000313	EQU	2		
BZRO	000000	G CAD	2		

KARMA 03/14/81=====		
HEADING	Table of Contents	PAGE
=====		
ENTRY POINTS		1
EXTERNALS		2
STACK DATA MOVE IN EXTENDED MEMORY		3
SKIP A LINE		5
FETCH LINE# ADDRESS		6
FIND A LINE		7
GET NEXT TOKEN		8
DECOMPILE 03-02-77		9
DECOMPILE MAIN		10
END OF LINE PROCESSING 03-01-77		12
CONSTANTS 03-01-77		14
FETCH VARIABLE 03-01-77		18
BASIC RES WORDS&FN END 03-09-77		20
GOTO & GOSUB 03-01-77		22
RELATIVE JUMPS 03-01-77		23
OPERATORS 03-01-77		25
SOME USEFUL UTILITIES 03-01-77		28
LINE NUMBERS 03-01-77		29
PROCESS OPER STAK 03-01-77		30
MISC TOKENS 03-21-77		33
SYSTEM FUNCTIONS 03-01-77		35
JS. FUNCTION CALLS 03-01-77		36
USER FUNCTION DEF 03-01-77		38
EXTERNAL ROMS		41
BASIC TOKEN DECOMPILE 2-2-77		42

CCDCOM HAD 0 ERRORS 0 WARNINGS 270 LABELS LAST ERROR AT 0