

IDENTIFICATION

PRODUCT CODE:	DEC-S8-LBASA-A-LA
PRODUCT NAME:	LISTING OF OS/8 BASIC
DATE CREATED:	OCTOBER 2, 1972

COPYRIGHT © 1972
DIGITAL EQUIPMENT CORPORATION

```
1      /OS/8 BASIC EDITOR
2      /
3      /DEC-S8-LBASA-A-LA
4      /
5      /COPYRIGHT,1972
6      /
7      /DIGITAL EQUIPMENT CORPORATION
8      /MAYNARD,MASSACHUSETTS 01754
9      /
10     /AUGUST 19, 1972
11     /
12     /HANK MAURER AND LEN ELEKMAN
13     /
14     /
15     /
16     /
17     0100      VERSION=100
```

```

18      3700      HCSIZE=3700      / SIZE OF HCOMP.SV
19      7001      HCBEGN=7001      / START OF HCOMP
20      7604      INFO=7604      / INFORMATION AREA (FIELD 1)
21      3200      OSKRES=3200      / SWAP AREA FOR CSA
22      4000      DSKBUF=4000      / FILE BUFFER
23      4400      HANDLR=4400      / INPUT OUTPUT HANDLER ADDRESS
24      5000      TXTAREA=5000      / START OF TEXT AREA
25      0003      *3
26      00003 0000 SWAPT1, 0
27      00004 0000 SWAPT2, 0
28      00005 0000 SWAPT3, 0
29      00006 0000 SWAPT4, 0
30      00007 0100 VERS, VERSION      /VERSION NUMBER
31      00010 7603 X10, INFO=1
32      00011 3266 X11, NAMLIST=1
33      00012 0000 X12, 0
34      00013 0000 X13, 0
35      00014 0000 X14, 0
36      00015 0000 X15, 0
37      00016 0000 X16, 0
38      00017 0000 X17, 0
39      0020      *20
40      00020 0000 RDTMP, 0      /USED BY INPUT ROUTINE
41      00021 0000 RDPTR, 0
42      00022 0000 SIZE, 0      /USED BY LINE EDITOR STUFF
43      00023 0000 TEMP, 0
44      00024 0000 TEMP2, 0
45      00025 0000 TOWARD, 0
46      00026 0000 PTR, 0
47      00027 0000 NDIGS, 0
48      00030 0000 NCHARS, 0
49      00031 0000 COFLAG, 0      /=0 IF CTRL/O
50      00032 0000 CHNFLAG, 0      /=1 IF BACK FROM RUN, 0 IF OLD
51      00033 0000 RUNFLAG, 0      /=1 IF RUN, 0 IF SAVE
52      00034 0000 OLDFLAG, 0      /=1 IF INPUT COMING FROM FILE
53      00035 0000 LINENO, 0;0      /HOLDS MOST RECENT LINE NUM
54      00036 0000
55      00037 0000 EOFLIN, 0;0      /LAST LINE NUMBER
56      00040 0000
57      00041 0000 NAME, 0;0;0;0      /NAME BUFFER
58      00042 0000
59      00043 0000
60      00044 0000
61      00045 0000 FNAME, 0;0;0;0      /CURRENT FILE NAME
62      00046 0000
63      00047 0000
64      00050 0000
65      00051 0000 DEVHAN, 0      /ADDRESS OF DEVICE HANDLER
66      00052 0000 DEVNUM, 0      /CURRENT DEVICE NUMBER
67      00053 0000 SWPNUM, 0      /SWAPPER FLAG (FOR TC)
68      00054 7607 07607, 7607
69      00055 7700 07707, 7707
70      00056 0200 0200, 200
71      00057 0201 0201, 201
72      00060 7761 07761, 7761

```

73	00061	7764	07764,	7764
74	00062	0177	0177,	177
75	00063	0232	0232,	232
76	00064	7201	07201,	7201
77	00065	7706	07706,	7706
78	00066	0032	032,	32
79	00067	0072	072,	72
80	00070	7200	07200,	7200
81	00071	7600	07600,	7600
82	00072	0017	017,	17
83	00073	0260	0260,	260
84	00074	0237	0237,	237
85	00075	0013	013,	13
86	00076	0215	0215,	215
87	00077	0212	0212,	212
88	00100	0037	037,	37
89	00101	7741	07741,	7741
90	00102	7405	07405,	7405
91	00103	7701	07701,	7701
92	00104	7772	07772,	7772
93	00105	6171	06171,	6171
94	00106	6211	06211,	6211
95	00107	7770	07770,	7770
96	00110	6201	06201,	6201
97	00111	0010	010,	10
98	00112	7774	07774,	7774
99	00113	7766	07766,	7766
100	00114	7634	07634,	7634
101	00115	0137	0137,	137
102	00116	7746	07746,	7746
103	00117	0036	036,	36
104	00120	0077	077,	77
105	00121	7745	07745,	7745
106	00122	0012	012,	12
107	00123	7771	07771,	7771
108	00124	7400	07400,	7400
109	00125	7760	07760,	7760
110	00126	0400	0400,	400
111	00127	0014	014,	14

			PAGE	
112	00200	0000	GETLIN, 2	/GET A LINE FROM ITY.
113	00201	3027	DCA	NDIGS /CLEAR LINE NUMBER.
114	00201	3027	DCA	NCHARS /CLEAR TEXT COUNT.
115	00202	3030		
116	00203	7200	IGNORE, CLA	
117	00204	4577	JMS I	(GETCH
118	00205	5024	DCA	TEMP2 /SAVE CHAR
119	00206	7146	CLL CMA	RTL /CHECK FOR CONTROL C
120	00207	1024	TAD	TEMP2
121	00210	7450	SNA	
122	00211	5777	JMP	BYEbye /ITS "C" EXIT TO OS8
123	00212	1115	TAD	07766 /CHECK FOR CARRIAGE RETURN
124	00213	7450	SNA	
125	00214	5334	JMP	CARRET /JUMP IF 015 - CARRET.
126	00215	1107	TAD	07770 /CHECK FOR "U
127	00216	7450	SNA	
128	00217	5234	JMP	ALT /TREAT "U" AS ALTMODE
129	00220	1104	TAD	07772 /CHECK FOR ALTMODE
130	00221	7450	SNA	
131	00222	5234	JMP	ALT /JUMP IF 033 - ALTMODE.
132	00223	1114	TAD	07634 /CHECK FOR RUBOUT
133	00224	7450	SNA	
134	00225	5315	JMP	ARROW /TREAT LIKE BACK ARROW
135	00226	7001	IAC	/CHECK FOR ALTMODE
136	00227	7450	SNA	
137	00230	5234	JMP	ALT /JUMP IF 176 - ALTMODE.
138	00231	7001	IAC	
139	00232	7440	SZA	
140	00233	5237	JMP	.+4 /PRINT IF 175 - ALTMODE.
141	00234	4576	ALT, JMS I	[TYPE
142	00235	1116	MSGALT	
143	00236	5201	JMP	GETLIN+1
144	00237	1117	TAD	036 /CHECK FOR BACK ARROW
145	00240	7450	SNA	
146	00241	5315	JMP	ARROW /JUMP IF 137 - BACKARROW.
147	00242	7100	CLL	
148	00243	1120	TAD	077
149	00244	7420	SNL	
150	00245	5203	JMP	IGNORE /JUMP IF NOT PRINTABLE.
151	00246	7001	IAC	
152	00247	3023	DCA	TEMP /SAVE IT(SP=01,"=77,NO 00).
153	00250	1034	TAD	OLDFLAG /INPUT FROM FILE ?
154	00251	7640	SZA CLA	
155	00252	5255	JMP	.+3 /YES, DON'T ECHO
156	00253	1024	TAD	TEMP2
157	00254	4575	JMS I	[TTYOUT /PRINT ON ITY
158	00255	1030	TAD	NCHARS
159	00256	7640	SZA CLA	
160	00257	5264	JMP	ISTEXT /NOT LINE NUMBER.
161	00260	1023	TAD	TEMP
162	00261	1121	TAD	07745 /SEE IF ITS A DIGIT
163	00262	4574	JMS I	[LNDIG /PUT DIGIT INTO LINE NUM
164	00263	5203	JMP	IGNORE /GET NEXT CHAR
165	00264	1030	ISTEXT, TAD	NCHARS /PUT CHAR IN THE LINE.
166	00265	1102	TAD	07425

167	00266	7710	SPA	CLA	
168	00267	5274	JMP	.	+5
169	00270	4573	JMS	I	[CRLF
170	00271	4576	JMS	I	[TYPE /250 CHARS IS OK TO FIT 1 MORE.
171	00272	0571	MSG100		
172	00273	5201	JMP		GETLIN+1/IF AN EVEN NUMBER
173	00274	1030	TAD		NCHARS /OF CHARS SO FAR
174	00275	7110	CLL	RAR	
175	00276	1172	TAD		[LINE+2
176	00277	3026	DCA		PTR
177	00300	7430	SZL		
178	00301	5307	JMP		RIGHTY
179	00302	1023	TAD		TEMP /THEN STORE AS THE LEFT
180	00303	7106	CLL	RTL	/HALF OF THE WORD.
181	00304	7006	RTL		
182	00305	7006	RTL		
183	00306	5312	JMP	.	+4
184	00307	1426	RIGHTY, TAD	I	PTR /ELSE AS THE RIGHT.
185	00310	0055	AND		07700
186	00311	1023	TAD		TEMP
187	00312	3426	DCA	I	PTR
188	00313	2030	ISZ		NCHARS
189	00314	5203	JMP		IGNORE
190	00315	1034	ARROW, TAD		OLDFLAG /INPUT FROM FILE ?
191	00316	7640	SZA	CLA	
192	00317	5322	JMP	.	+3
193	00320	1115	TAD		0137 /NO, PRINT BACK ARROW
194	00321	4575	JMS	I	[TTYOUT
195	00322	7240	CLA	CMA	
196	00323	1030	TAD		NCHARS /IF THERE IS A TEXT CHAR TO ERASE
197	00324	7510	SPA		
198	00325	5330	JMP	.	+3
199	00326	3030	DCA		NCHARS /THEN ERASE IT.
200	00327	5203	JMP		IGNORE
201	00330	1027	TAD		NDIGS /OTHERWISE, IF THERE IS A LINE#
202	00331	7500	SMA		/CHARACTER TO ERASE THEN ERASE
203	00332	3027	DCA		NDIGS /THAT.
204	00333	5203	JMP		IGNORE /OTHERWISE, NEVER MIND.
205	00334	1034	CARRET, TAD		OLDFLAG /INPUT FROM FILE ?
206	00335	7650	SNA	CLA	
207	00336	4573	JMS	I	[CRLF /NO, PRINT CR-LF
208	00337	1030	TAD		NCHARS
209	00340	7440	SZA		
210	00341	5345	JMP	.	+4 /NOT AN EMPTY LINE
211	00342	1027	TAD		NDIGS /ANY CHARS AT ALL ?
212	00343	7650	SNA	CLA	
213	00344	5203	JMP		IGNORE /NO, IGNORE EMPTY LINES
214	00345	7110	CLL	RAR	
215	00346	1172	TAD		[LINE+2
216	00347	3026	DCA		PTR
217	00350	7430	SZL		
218	00351	1426	TAD	I	PTR /STORE 00 (CAR. RET.) LIKE ALL
219	00352	0055	AND		07700 /THE OTHER CHARACTERS.
220	00353	3426	DCA	I	PTR
221	00354	4571	JMS	I	[NORM /NORMALIZE LINE NUMBER

222	00355	5600		JMP I	GETLIN
223	00356	0000	CRLF,	4	/PRINT CR-LF
224	00357	1076		TAD	0215
225	00360	4575		JMS I	[TTYOUT
226	00361	1077		TAD	0212
227	00362	4575		JMS I	[TTYOUT
228	00363	5756		JMP I	CRLF

229	00377	1070			
230	00400	0400		PAGE	
231	00400	0000	LNDIG,	0	/GET DIGIT INTO LINE NUM
232	00401	7100		CLL	
233	00402	1122		TAD	012
234	00403	7430		SZL	
235	00404	5210		JMP	.*4
236	00405	7200		CLA	/NOT A DIGIT
237	00406	2200		ISZ	LNDIG /RETURN +1
238	00407	5600		JMP I	LNDIG
239	00410	3023		DCA	TEMP /SAVE DIGIT
240	00411	2027		ISZ	NDIGS
241	00412	1027		TAD	NDIGS
242	00413	1123		TAD	07771
243	00414	7500		SMA	
244	00415	5600		JMP I	LNDIG /IGNORE MORE THAN 6 DIGITS.
245	00416	1221		TAD	.*3
246	00417	3220		DCA	.*1 /PUT IN THE NTH DIGIT IN THE
247	00420	7402		HLT	/4 BIT BCD LINE NUMBER:
248	00421	5230		JMP	.*7
249	00422	5263		JMP	DIG1
250	00423	5253		JMP	DIG2
251	00424	5250		JMP	DIG3
252	00425	5242		JMP	DIG4
253	00426	5232		JMP	DIG5
254	00427	1036	DIG6,	TAD	LINENO+1/WHERE N=6
255	00430	0125		AND	07760
256	00431	5240		JMP	DIG56
257	00432	1023	DIG5,	TAD	TEMP /WHERE N=5
258	00433	7106		CLL	RTL
259	00434	7006		RTL	
260	00435	3023		DCA	TEMP
261	00436	1036		TAD	LINENO+1
262	00437	0124		AND	07400
263	00440	1023	DIG56,	TAD	TEMP
264	00441	5246		JMP	DIG456
265	00442	1023	DIG4,	TAD	TEMP /WHERE N=4
266	00443	7110		CLL	RAR
267	00444	7012		RTR	
268	00445	7012		RTR	
269	00446	3036	DIG456,	DCA	LINENO+1
270	00447	5600		JMP I	LNDIG
271	00450	1035	DIG3,	TAD	LINENO /WHERE N=3
272	00451	0125		AND	07760
273	00452	5261		JMP	DIG23
274	00453	1023	DIG2,	TAD	TEMP /WHERE N=2
275	00454	7106		CLL	RTL
276	00455	7006		RTL	
277	00456	3023		DCA	TEMP
278	00457	1035		TAD	LINENO
279	00460	0124		AND	07400
280	00461	1023	DIG23,	TAD	TEMP
281	00462	5267		JMP	DIG123
282	00463	1023	DIG1,	TAD	TEMP /WHERE N=1
283	00464	7110		CLL	RAR

```

284 00465 7012      RTE
285 00465 7012      RTE
286 00467 3035      DIG123, DCA      LINEND
287 00470 5600      JMP I      LNDIG
288 00471 0000      TYPE, 0      /TYPE A MESSAGE
289 00472 3347      DCA      CRSWIT /SAVE CARRIAGE RETURN SWITCH
290 00473 1671      TAD I      TYPE /GET ADDR OF MESSAGE
291 00474 2271      ISZ      TYPE
292 00475 3333      DCA      PASS
293 00476 4570      TLOOP, JMS I  ICTRL0 /CHECK FOR CTRL/D
294 00477 5321      JMP      TCRLF /YES, STOP PRINTING
295 00500 1733      TAD I      PASS /GET HIGH CHAR
296 00501 7112      CLL RTR    /SHIFT RIGHT
297 00502 7012      RTR
298 00503 7012      RTR
299 00504 0120      AND      077    /SIX BITS
300 00505 7450      SNA
301 00506 5321      JMP      TCRLF /END OF MESSAGE
302 00507 1074      TAD      0237 /CONVERT TO ASCII
303 00510 4325      JMS      TTYOUT /PRINT CHAR
304 00511 1733      TAD I      PASS /GET LOWER CHAR
305 00512 2333      ISZ      PASS
306 00513 0120      AND      077
307 00514 7450      SNA
308 00515 5321      JMP      TCRLF /END OF LINE
309 00516 1074      TAD      0237 /CONVERT TO ASCII
310 00517 4325      JMS      TTYOUT /PRINT
311 00520 5276      JMP      TLOOP
312 00521 1347      TCRLF, TAD  CRSWIT /RETURN THE CARRIAGE ?
313 00522 7650      SNA CLA
314 00523 4573      JMS I  ICTRL /YES
315 00524 5671      JMP I  TYPE  /DONE
316 00525 0000      TTYOUT, 0    /PRINT ONE CHAR
317 00526 6041      TSF
318 00527 5326      JMP      .-1
319 00530 6046      TLS
320 00531 7200      CLA
321 00532 5725      JMP I  TTYOUT
322 00533 0000      PASS, 0      /SKIP OVER LINE
323 00534 2023      ISZ      TEMP
324 00535 7410      SKP
325 00536 4363      JMS      FINCR
326 00537 1423      TAD I      TEMP
327 00540 0120      AND      077
328 00541 7640      SZA CLA
329 00542 5334      JMP      PASS+1
330 00543 2023      ISZ      TEMP
331 00544 5733      JMP I  PASS
332 00545 4363      JMS      FINCR
333 00546 5733      JMP I  PASS
334 00547 0000      CRSWIT,
335 00550 1023      FUECR, 0      /DECR. POINTER AND FIELD
336 00551 7640      TAD      TEMP
337 00552 5337      SZA CLA
338 00552 5337      JMP      .+1

```

339	00553	6214	RDF	
340	00554	1105	TAD	06171
341	00555	3356	DCA	.+1
342	00556	7402	HLT	
343	00557	7240	CLA	CNA
344	00560	1023	TAD	TEMP
345	00561	3023	DCA	TEMP
346	00562	5747	JMP I	FDECR
347	00563	0000	FINCR,	0 /INCR. CURRENT DATA FIELD
348	00564	6214	RDF	
349	00565	1106	TAD	06211
350	00566	3367	DCA	.+1
351	00567	7402	HLT	
352	00570	5763	JMP I	FINCR
353	00571	5552	MSGT00,	5552/5746/0165/6060/0155/6057/5000
354	00572	5746		
355	00573	0165		
356	00574	6060		
357	00575	0155		
358	00576	6057		
359	00577	5000		

			PAGE
360	0600	0600	
361	00600	6201	CMDDONE, CDF
362	00601	4573	JMS I [CRLF /TYPE READY MESSAGE
363	00602	4576	JMS I [TYPE
364	00603	1123	MSGRDY
365	00604	6201	MAINLUP, CDF
366	00605	4567	JMS I [GETLIN /GET AN EDITED LINE.
367	00606	7650	SNA CLA
368	00607	5214	JMP NOCOMD /NOT A COMMAND
369	00610	1034	TAD OLDFLAG /IN OLD MODE ?
370	00611	7650	SNA CLA
371	00612	5566	JMP I [COMMAND/NO, MUST BE A COMMAND
372	00613	5204	JMP MAINLUP /OTHERWISE IGNORE
373	00614	1026	NOCOMD, TAD PTR /OR A LINE WITH A LINE
374	00615	7040	CMA /NUMBER ON IT.
375	00616	1165	TAD [LINE
376	00617	3022	DCA SIZE /SET UP SIZE OF LINE.
377	00620	1572	TAD I [LINE+2 /IS LINE EMPTY ??
378	00621	7650	SNA CLA
379	00622	3022	DCA SIZE /POSSIBLY ZERO.
380	00623	1035	TAD LINENO /IS IT > LAST LINE ?
381	00624	7141	CIA CLL
382	00625	1037	TAD EOFLIN
383	00626	7640	SZA CLA
384	00627	5233	JMP ,+4 /HI PART NOT =, FORGET LOW
385	00630	1036	TAD LINENO+1
386	00631	7141	CIA CLL
387	00632	1040	TAD EOFLIN+1 /COMPARE LOW PARTS
388	00633	7630	SZL CLA
389	00634	5245	JMP NOTLAST /NOT > LAST
390	00635	4564	JMS I [GETEOF /GET EOF
391	00636	1023	TAD TEMP /MAKE IT LOOK LIKE
392	00637	3026	DCA PTR /A CALL TO FINDLN
393	00640	1035	TAD LINENO /SAVE NEW LAST LINE
394	00641	3037	DCA EOFLIN
395	00642	1036	TAD LINENO+1
396	00643	3040	DCA EOFLIN+1
397	00644	7410	SKP
398	00645	4563	NOTLAST, JMS I [FINDLN /GENERAL CASE - SEARCH
399	00646	1023	INSERT, TAD TEMP /THERE ARE (TEMP-PTR) WORDS IN
400	00647	7140	CLL CMA /THE OLD LINE WHICH ARE TO BE
401	00650	1026	TAD PTR /REPLACED BY (=SIZE) WORDS IN
402	00651	7200	CLA /NEW LINE.
403	00652	6214	RDF
404	00653	7430	SZL
405	00654	1107	TAD 07770
406	00655	1110	TAD 06201
407	00656	3346	DCA PTRFLD /GET FIELD OF START OF OLD LINE
408	00657	1026	TAD PTR
409	00660	7141	CLL CIA
410	00661	1023	TAD TEMP
411	00662	1022	TAD SIZE /WHICH WAY ?
412	00663	7450	SNA
413	00664	5331	JMP MOVE /SAME SIZE, MOVE IN NEW LINE
414	00665	7510	SPA

415	00666	5777	JMP	EXPAND	/MAKE MORE ROOM FOR NEW LINE
416	00667	7041	CIA		
417	00670	1023	TAD	TEMP	/SHRINK THE FILE
418	00671	3025	DCA	TOWARD	/MOVE FILE DOWN TO HERE
419	00672	6214	RDF		
420	00673	1110	TAD	06201	
421	00674	3304	DCA	TMPFLD	/GET FIELD OF READ POINTER
422	00675	1025	TAD	TOWARD	
423	00676	7140	CLL	CMA	
424	00677	1023	TAD	TEMP	
425	00700	7620	SNL	CLA	
426	00701	1107	TAD	07770	
427	00702	1304	TAD	TMPFLD	
428	00703	3306	DCA	TWDFLD	/GET FIELD OF WRITE POINTER
429	00704	7402	TMPFLD, HLT		
430	00705	1423	TAD I	TEMP	
431	00706	7402	TWDFLD, HLT		
432	00707	3425	DCA I	TOWARD	/MOVE DOWN
433	00710	1425	TAD I	TOWARD	
434	00711	1103	TAD	07701	/END OF FILE ???
435	00712	7650	SNA	CLA	
436	00713	5327	JMP	LWREOF	/YES, PUT NEW LINE IN AT END
437	00714	2023	ISZ	TEMP	/INCREMENT POINTERS
438	00715	5321	JMP	.+4	
439	00716	1304	TAD	TMPFLD	/AND FIELDS IF NECESSARY
440	00717	1111	TAD	010	
441	00720	3304	DCA	TMPFLD	
442	00721	2025	ISZ	TOWARD	
443	00722	5304	JMP	TMPFLD	
444	00723	1306	TAD	TWDFLD	
445	00724	1111	TAD	010	
446	00725	3306	DCA	TWDFLD	
447	00726	5304	JMP	TMPFLD	/KEEP SHRINKING
448	00727	1025	LWREOF, TAD	TOWARD	/SET NEW EOF
449	00730	4562	JMS I	[SETEOF	
450	00731	1022	MOVE, TAD	SIZE	
451	00732	7650	SNA	CLA	
452	00733	5204	JMP	MAINLUP	/IT WAS A DELETE
453	00734	6201	CDF 00		
454	00735	1035	TAD	LINEND	/PUT IN LINE NUMBER
455	00736	3565	DCA I	[LINE	
456	00737	1036	TAD	LINEND+1	
457	00740	3561	DCA I	[LINE+1	
458	00741	1165	MOVENTR, TAD	[LINE	
459	00742	3023	DCA	TEMP	
460	00743	6201	MOVLUP, CDF		/MOVE IN NEW LINE
461	00744	1423	TAD I	TEMP	
462	00745	2023	ISZ	TEMP	
463	00746	7402	PTRFLD, HLT		
464	00747	3426	DCA I	PTR	
465	00750	2026	ISZ	PTR	/INCREMENT POINTERS
466	00751	5355	JMP	.+4	
467	00752	1346	TAD	PTRFLD	/WATCH OUT FOR FIELDS
468	00753	1111	TAD	010	/(W.C. OR E.M. ?)
469	00754	3346	DCA	PTRFLD	

/OS/8 BASIC EDITOR

PAL8-V7 10/24/72 PAGE 5-2

470	00755	2022
471	00756	5343
472	00757	5204

ISZ	SIZE
JMP	MOVLUP
JMP	MAINLUP

473	00777	1000			
474		1000			
475	01000	7041	EXPAND, CIA	PAGE	
476	01001	3025			/EXTRA ROOM NEEDED
477	01002	1423	DCA	TOWARD	
478	01003	3024	TAD I	TEMP	/SAVE THIS PLACE
479	01004	1100	DCA	TEMP2	
480	01005	3423	TAD	037	/NOW MARK THIS PLACE
481	01006	4564	DCA I	TEMP	
482	01007	6214	JMS I	[GETEOF	/GET EOF
483	01010	1110	RDF		
484	01011	3237	TAD	06201	
485	01012	7100	DCA	TMP2FLD	/GET FIELD OF END OF FILE
486	01013	1023	CLL		
487	01014	1025	TAD	TEMP	/MOVE FILE UP
488	01015	3025	TAD	TOWARD	/TO
489	01016	7430	DCA	TOWARD	/HERE
490	01017	4560	SZL		
491	01020	3425	JMS I	[FINCR	/MIGHT BE ACROSS A FIELD
492	01021	1425	DCA I	TOWARD	
493	01022	7640	TAD I	TOWARD	/IS THERE ENOUGH CORE ?
494	01023	5273	SZA CLA		
495	01024	7240	JMP	COREOV	/NO, GIVE ERROR
496	01025	3425	CLA CMA		
497	01026	1425	DCA I	TOWARD	
498	01027	7001	TAD I	TOWARD	
499	01030	7640	IAC		
500	01031	5273	SZA CLA		
501	01032	6214	JMP	COREOV	/NO, GIVE ERROR
502	01033	1110	RDF		
503	01034	3241	TAD	06201	
504	01035	1025	DCA	TWD2FLD	/SAVE NEW EOF FIELD
505	01036	4562	TAD	TOWARD	/SAVE NEW EOF
506	01037	7402	JMS I	[SETEOF	
507	01040	1423	TMP2FLD,HLT		
508	01041	7402	TAD I	TEMP	
509	01042	3425	TWD2FLD,HLT		
510	01043	1425	DCA I	TOWARD	/MOVE UP ONE WORD
511	01044	1101	TAD I	TOWARD	
512	01045	7650	TAD	07741	/IS THE MARK ?
513	01046	5270	SNA CLA		
514	01047	7340	JMP	LASTWD	/YES, PUT IN LAST WORD
515	01050	1025	CLA CLL	CMA	
516	01051	3025	TAD	TOWARD	/BACK UP POINTERS
517	01052	7430	DCA	TOWARD	
518	01053	5257	SZL		
519	01054	1241	JMP	.+4	
520	01055	1107	TAD	TWD2FLD	/AND FIELDS (MAYBE)
521	01056	3241	TAD	07770	
522	01057	7340	DCA	TWD2FLD	
523	01060	1025	CLA CLL	CMA	
524	01061	3025	TAD	TEMP	
525	01062	7430	DCA	TEMP	
526	01063	5237	SZL		
527	01064	1237	JMP	TMP2FLD	
			TAD	TMP2FLD	

```

528 01065 1107      TAD      07770
529 01066 3237      DCA      TMP2FLD
530 01067 5237      JMP      TMP2FLD
531 01070 1024      LASTWD, TAD      TEMP2      /PUT IN SAVED WORD
532 01071 3425      DCA I      TOWARD
533 01072 5557      JMP I      [MOVE      /GO MOVE IN NEW LINE
534 01073 4576      COREOV, JMS I      [TYPE      /FILE TOO BIG
535 01074 1105      MSGBIG
536 01075 5556      JMP I      [MAINLUP      1064 V3
537 01076 7201      BYEBYE, CLA IAC
538 01077 0053      AND      SWPNUM      /IS OS8 RES IN PLACE ?
539 01100 7640      SZA CLA      /YES IF EVEN NUMBER OF SWAPS
540 01101 4555      BYE,      JMS I      [SWAP      /PUT BACK OS8
541 01102 6041      TSF      /WAIT FOR TTY SO OS8 DOESN'T
542 01103 5302      JMP      .-1      /TRAMPLE ON MY LINE FEED
543 01104 5777      JMP      7605      /EXIT TO OS8
544 01105 4442      MSGBIG, 4442;5710;6501;4752;6516;5250;5760;6346;4500
545 01106 5710
546 01107 6501
547 01110 4752
548 01111 6516
549 01112 5250
550 01113 5760
551 01114 6346
552 01115 4500
553 01116 0145      MSGALT, 0145;4655;4665;4645;0
554 01117 4655
555 01120 4665
556 01121 4645
557 01122 0000
558 01123 6346      MSGRDY, 6346;4245;7200
559 01124 4245
560 01125 7200
561 01126 7051      MSGWHAT,7051;4265;4000
562 01127 4265
563 01130 4000
564 01131 1154      SCRATCH,TAD      [TXTAREA/SCRATCH FILE
565 01132 4562      JMS I      [SETEOF
566 01133 3037      DCA      EOFLIN      /ZERO LAST LINE NUM
567 01134 3040      DCA      EOFLIN+1
568 01135 5553      JMP I      [CMDDONE
569 01136 0000      NORM,      0      /LINE NUMBER NORMALIZER
570 01137 1027      TAD      NDIGS
571 01140 7440      SZA
572 01141 5344      JMP      .+3      /IF THERE ARE NO DIGITS IN THE
573 01142 7240      CLA CMA      /LINE NUMBER THEN
574 01143 5736      JMP I      NORM      /RETURN -1.
575 01144 1104      TAD      07772
576 01145 7700      SMA CLA
577 01146 5736      JMP I      NORM      /IF THE LINENO HAS BEEN FILLED
578 01147 1112      TAD      07774      /OUT TO 6 DIGITS(LEADING 0'S)
579 01150 3023      DCA      TEMP      /THEN RETURN.
580 01151 1035      TAD      LINENO      /OTHERWISE, SHIFT RIGHT 1 DIGIT
581 01152 7110      CLL RAR
582 01153 3035      DCA      LINENO      /AND CHECK AGAIN.

```

583	01154	1036
584	01155	7010
585	01156	3036
586	01157	2023
587	01160	5351
588	01161	2027
589	01162	5337

TAD	LINENO+1
RAR	
DCA	LINENO+1
ISZ	TEMP
JMP	.-7
ISZ	NDIGS
JMP	NORM+1

```

590 01177 7605
591      1200      PAGE
592 01200 1220 COMMAND,TAD COMTBL /COMMAND LIST POINTER
593 01201 3023      DCA      TEMP
594 01202 2023 COMLUP, ISZ      TEMP /GET 2 CHAR COMMAND
595 01203 1423      TAD I     TEMP
596 01204 2023      ISZ      TEMP
597 01205 7450      SNA
598 01206 5215      JMP      WHAT /END OF LIST
599 01207 1572      TAD I     (LINE+2 /IS THIS IT ?
600 01210 7640      SZA CLA
601 01211 5202      JMP      COMLUP /NO, LOOK AGAIN
602 01212 1423      TAD I     TEMP /GET COMMAND ADDR
603 01213 3023      DCA      TEMP /AND GO TO IT
604 01214 5423      JMP I     TEMP
605 01215 4576 WHAT, JMS I     (TYPE /TYPE WHAT?
606 01216 1126      MSGWHAT
607 01217 5556      JMP I     (MAINLUP
608 01220 1220 COMTBL,
609 01221 2226      -5552
610 01222 1400      LIST
611 01223 1723      -6055
612 01224 2723      OLD
613 01225 1336      -6442
614 01226 2223      SAVE
615 01227 1412      -6366
616 01230 2200      RUN
617 01231 1334      -6444
618 01232 1131      SCRATCH
619 01233 3406      -4372
620 01234 1101      BYE
621 01235 2032      -5746
622 01236 1763      NEW
623 01237 2036      -5742
624 01240 1754      RENAME
625 01241 0000      0
626 01242 0145 WSSAVE, 0145;6454;3343;4264;5244;1770;6400
627 01243 6454
628 01244 3343
629 01245 4264
630 01246 5244
631 01247 1770
632 01250 6400
633 01251 0000 HEADING,0
634 01252 4573 JMS I     (CRLF /LATER
635 01253 1152 TAD      (FNAME /SET UP FOR CONVERSION
636 01254 3023 DCA      TEMP /POINTER TO FILE NAME
637 01255 1265 TAD      XTITLE /WHERE IT GOES
638 01256 3026 DCA      PTR
639 01257 4270 JMS      CONV /OUTPUT FIRST TWO CHARS
640 01260 4270 JMS      CONV /NEXT TWO
641 01261 4270 JMS      CONV /THIRD TWO
642 01262 2026 ISZ      PTR /SKIP FOR EXT
643 01263 4270 JMS      CONV /OUTPUT EXTENSION
644 01264 4576 JMS I     (TYPE /TYPE HEADING

```

```

645 01265 1312 XTITLE, TITLE
646 01266 4573 JMS I [CRLF /FOLLOWED BY A CRLF
647 01267 5651 JMP I HEADING
648 01270 0000 CONV, 0 /CONVERT TO SIX BIT ASCII
649 01271 1423 TAD I TEMP /GET NEXT WORD
650 01272 0120 AND 077 /CHECK FOR 0
651 01273 7450 SNA /SUBSTITUTE BLANKS
652 01274 1377 TAD (40
653 01275 1101 TAD 07741 /SUBTRACT 37
654 01276 0120 AND 077 /SIX BITS
655 01277 3426 DCA I PTR
656 01300 1423 TAD I TEMP /DO UPPER CHAR
657 01301 0055 AND 07700
658 01302 7450 SNA
659 01303 7130 CLL CML RAR
660 01304 1376 TAD (-3700 /SAME WAY
661 01305 1426 TAD I PTR /COMBINE THEM
662 01306 3426 DCA I PTR
663 01307 2023 ISZ TEMP
664 01310 2026 ISZ PTR
665 01311 5670 JMP I CONV
666 01312 0000 TITLE, 0101010110 /FOR THE PROG NAME
667 01313 0000
668 01314 0000
669 01315 0101
670 01316 0000
671 01317 0101 010110101 /SOME BLANKS
672 01320 0101
673 01321 2217 VERSION*700+2117 /VERSION NUMBER X.Y
674 01322 2101 VERSION*10*700+2101
675 01323 0101 010110101 /MORE BLANKS
676 01324 0101
677 01325 0000 DATE, 010101163010 /DATE
678 01326 0000
679 01327 0000
680 01330 1630
681 01331 0000
682 01332 0000 GETNC, 0 /GET A CHAR FOR A FILE NAME
683 01333 1030 TAD NCHARS /GET CHAR POINTER
684 01334 2030 ISZ NCHARS /BUMP IT
685 01335 7110 CLL RAR /DIVIDE BY 2
686 01336 1172 TAD [LINE+2 /ADD BASE
687 01337 3023 DCA TEMP /GIVES ADDR OF WORD
688 01340 1423 TAD I TEMP /GET 2 CHARS
689 01341 7430 SZL /ODD EVEN BIT IS IN LINK
690 01342 5346 JMP TESTCH /GO CHECK THE CHAR
691 01343 7012 RTR
692 01344 7012 RTR
693 01345 7012 RTR
694 01346 0120 TESTCH, AND 077 /ONLY 6 BITS
695 01347 7450 SNA
696 01350 5732 JMP I GETNC /END OF NAME
697 01351 1060 TAD 07761 /WAS IT . ??
698 01352 7440 SZA
699 01353 5361 JMP NOEXT /NO

```

700	01354	7240	CLA CMA	/SET SIZE TO -1
701	01355	3022	DCA	SIZE
702	01356	1151	TAD	(NAME+3 /SET POINTER TO FXT WORD
703	01357	3024	DCA	TEMP2
704	01360	5775	JMP	NAMLUP /START ON EVEN CHAR
705	01361	1061	TAD	07764 /IS IT : ??
706	01362	7440	SZA	
707	01363	5366	JMP	.+3
708	01364	7201	CLA IAC	/YES, RETURN NONZERO
709	01365	5732	JMP I	GETNC
710	01366	1067	TAD	072 /RESTORE THE CHAR TO
711	01367	0120	AND	077 /STRIPPED SIX BIT
712	01370	2332	ISZ	GETNC
713	01371	5732	JMP I	GETNC

NOEXT,

714	01375	1736	
715	01376	4100	
716	01377	0040	
717		1400	PAGE
718	01400	1550	LIST, TAD I [LINE+3 /LISTNH ?
719	01401	1547	TAD I [LINE+4
720	01402	1377	TAD [-4436 /PSEUDO TEST
721	01403	7650	SNA CLA
722	01404	5211	JMP LISTNH /NO HEADING
723	01405	4546	JMS I [HEADING/GIVE HEADING
724	01406	1031	TAD COFLAG /WAS CTRL/O TYPED ?
725	01407	7650	SNA CLA
726	01410	5553	JMP I [CMDDONE/YES, ABORT LISTING
727	01411	3030	LISTNH, DCA NCHARS /SET POINTER
728	01412	3027	DCA NDIGS /AND DIGIT COUNTER
729	01413	4545	JMS I [GETNC /SKIP UNTIL BLANK
730	01414	5225	JMP NUMDUN /DONE
731	01415	1376	TAD [-40
732	01416	7640	SZA CLA
733	01417	5213	JMP .-4 /NO BLANK YET
734	01420	4545	JMS I [GETNC /GET A CHAR
735	01421	5225	JMP NUMDUN /END OF NUMBER
736	01422	1065	TAD 07706 /SUBTRACT 72
737	01423	4574	JMS I [LNDIG /GO SEE IF ITS A DIGIT
738	01424	5220	JMP .-4 /IT WAS, CONTINUE
739	01425	1154	NUMDUN, TAD [TXTAREA
740	01426	3026	DCA PTR /SET UP POINTER
741	01427	4571	JMS I [NORM /NORMALIZE THE NUM
742	01430	7700	SMA CLA /ANY NUMBER ?
743	01431	4563	JMS I [FINDLN /YES, LOCATE IT
744	01432	6214	RDF /GET THE FIELD
745	01433	1110	TAD 06201
746	01434	3267	DCA PTR2FLD /SAVE IT
747	01435	3023	DCA TEMP
748	01436	4570	LSTLUP, JMS I [CTRL /CHECK FOR CTRL/O
749	01437	5553	JMP I [CMDDONE/YES, EXIT
750	01440	4244	JMS GETFIL /GET CHARACTER TO LIST
751	01441	5553	JMP I [CMDDONE
752	01442	4575	JMS I [TTYOUT /PRINT IT
753	01443	5236	JMP LSTLUP /LOOP
754	01444	0000	GETFIL, 0 /GET CHARACTER FROM FILE
755	01445	1023	TAD TEMP
756	01446	2023	ISZ TEMP
757	01447	1252	TAD .+3
758	01450	3251	DCA .+1
759	01451	7402	HLT
760	01452	5653	JMP I .+1 /SEQUENCE OF OPERATIONS
761	01453	1467	PTR2FLD /GET FIRST WORD
762	01454	1506	FRSTDIG /FIRST DIGIT OF LINE NUMBER
763	01455	1507	DIGIT /SECOND DIGIT
764	01456	1507	DIGIT /THIRD DIGIT
765	01457	1467	PTR2FLD /GET NEXT WORD OF LINE NUMBER
766	01460	1507	DIGIT /FOURTH DIGIT
767	01461	1507	DIGIT /FIFTH DIGIT
768	01462	1505	LASTDIG /SIXTH AND LAST DIGIT

769	01463	1467	PTR2FLD	/GET WORD OF TEXT
770	01464	1526	LEFTXT	/LEFT CHARACTER
771	01465	1533	RITETXT	/RIGHT CHARACTER
772	01466	1550	LINFTXT	/LINE FEED CHARACTER
773	01467	7402	PTR2FLD,HLT	/CHECK FOR EOF
774	01470	1426	TAD I PTR	
775	01471	6201	CDF	
776	01472	1103	TAD	07701
777	01473	7450	SNA	
778	01474	5644	JMP I	GETFIL /YES, RETURN UNSKIPPED
779	01475	1120	TAD	077
780	01476	3024	DCA	TEMP2 /NO, SAVE WORD
781	01477	2026	ISZ	PTR /BUMP POINTER
782	01500	5245	JMP	GETFIL+1
783	01501	1267	TAD	PTR2FLD
784	01502	1111	TAD	010
785	01503	3267	DCA	PTR2FLD
786	01504	5245	JMP	GETFIL+1
787	01505	7201	LASTDIG,CLA	IAC /FORCE LAST DIGIT (EVEN IF 0)
788	01506	3027	FRSTDIG,DCA	NDIGS /ZERO DIGIT COUNT
789	01507	1024	DIGIT, TAD	TEMP2
790	01510	7006	RTL	
791	01511	7006	RTL	
792	01512	3024	DCA	TEMP2 /SHIFT LEFT ONE DIGIT
793	01513	1024	TAD	TEMP2
794	01514	7004	RAL	
795	01515	0072	AND	017 /GET DIGIT
796	01516	7440	SZA	
797	01517	5323	JMP	NZDIGIT /ITS NOT ZERO
798	01520	1027	TAD	NDIGS /IS IT A LEADING ZERO ?
799	01521	7650	SNA CLA	
800	01522	5245	JMP	GETFIL+1 /YES, DON'T PRINT IT
801	01523	2027	NZDIGIT,ISZ	NDIGS /NON ZERO OR NON LEADING ZERO
802	01524	1073	TAD	0260 /SO PRINT IT
803	01525	5342	JMP	GFRET
804	01526	1024	LEFTXT,TAD	TEMP2 /GET LEFT CHAR
805	01527	7012	RTR	
806	01530	7012	RTR	
807	01531	7012	RTR	
808	01532	5336	JMP	.+4
809	01533	1111	RITETXT,TAD	010 /SETUP FOR LEFT CHAR NEXT
810	01534	3023	DCA	TEMP
811	01535	1024	TAD	TEMP2
812	01536	0120	AND	077 /SIXBITIZE AC
813	01537	7450	SNA	
814	01540	5344	JMP	ZEROTXT /0 IS END OF THE LINE
815	01541	1074	TAD	0237 /MAKE IT ASCII
816	01542	2244	GFRET, ISZ	GETFIL
817	01543	5644	JMP I	GETFIL
818	01544	1075	ZEROTXT,TAD	013 /SETUP FOR LF NEXT
819	01545	3023	DCA	TEMP
820	01546	1076	TAD	0215 /RETURN CR
821	01547	5342	JMP	GFRET
822	01550	3023	LINFTXT,DCA	TEMP /CLEAR SEQUENCER AND RETURN LF
823	01551	1077	TAD	0212

824	01552	5342		JMP	GFRET	
825	01553	0000	CTRLO,	0		/CHECK FOR ^O AND ^C
826	01554	6031		KSF		/CHECK FOR ^O
827	01555	5366		JMP	CTRLOX	/NO KEYBOARD
828	01556	6036		KRB		
829	01557	0062		AND	0177	
830	01560	1060		TAD	07761	
831	01561	7450		SNA		
832	01562	5370		JMP	CTRLOX+2/SET FLAG OFF	
833	01563	1127		TAD	014	/^C ?
834	01564	7650		SNA CLA		
835	01565	5775		JMP	BYEBYE	/YES, END
836	01566	2353	CTRLOX,	ISZ	CTRLO	/FIX RETURN
837	01567	7201		CLA IAC		/SET FLAG #1 IF NO CTRL/O
838	01570	3031		DCA	COFLAG	
839	01571	5753		JMP I	CTRLO	

840	01575	1076			
841	01576	7740			
842	01577	3342			
843		1600			
844	01600	0000	GETFN,	PAGE 0	/GET A FILE NAME (ALSO FETCH ITS HANDLER)
845	01601	3323		DCA	SAVFLAG /#1 FOR SAVE, 0 FOR OLD OR NEW
846	01602	1032		TAD	CHNFLAG /RETURNING FROM RUN ?
847	01603	7640		SZA CLA	
848	01604	5215		JMP	NOFUSR /YES, DON'T FETCH USR
849	01605	4555		JMS I	[SWAP /GET OS8 RESIDENT
850	01606	1323		TAD	SAVFLAG /IS IT OLD OR NEW ??
851	01607	7650		SNA CLA	
852	01610	7001		IAC	/YES, DON'T SWAP 10000-11777
853	01611	3777		DCA	7746 /DO IF SAVE, SO ALTER JSW
854	01612	6212		CIF	10 /GET THE USR
855	01613	4455		JMS I	07700
856	01614	0010		10	
857	01615	3030	NOFUSR,	DCA	NCHARS /RESET CHAR POINTER
858	01616	4776	BSKIP,	JMS	GETNC /GET A CHAR
859	01617	5275		JMP	ASKNAM /ASK FOR FILE NAME
860	01620	1260		TAD	M40 /BLANK ?
861	01621	7640		SZA CLA	
862	01622	5216		JMP	BSKIP /NO, LOOP
863	01623	4324	NOSKIP,	JMS	GETNAM /GET A NAME
864	01624	7650		SNA CLA	
865	01625	5236		JMP	USEDISK /NO DEVICE SPECIFIED, USE DSK:
866	01626	1041		TAD	NAME /PUT IN THE DEVICE NAME
867	01627	3247		DCA	DEV /AS SPECIFIED
868	01630	1042		TAD	NAME+1
869	01631	3250		DCA	DEV+1
870	01632	4324		JMS	GETNAM /FETCH THE FILE NAME
871	01633	7640		SZA CLA	
872	01634	5775		JMP	IOERR /BAD SYNTAX IN FILE DESCRIPTOR
873	01635	5242		JMP	GETHAN /GO FETCH THE HANDLER
874	01636	1374	USEDISK,	TAD	(0423 /SET DEVICE NAME TO DSK:
875	01637	3247		DCA	DEV
876	01640	1373		TAD	(1300
877	01641	3250		DCA	DEV+1
878	01642	1144	GETHAN,	TAD	[HANDLR+1
879	01643	3251		DCA	DEV+2 /ALSO THE HANDLER ORIGIN
880	01644	6212		CIF	10
881	01645	4456		JMS I	0200 /CALL THE USR
882	01646	0001		1	/FETCH HANDLER BY NAME
883	01647	0000	DEV,	01010	
884	01650	0000			
885	01651	0000			
886	01652	5775		JMP	IOERR /BAD DEVICE
887	01653	1250		TAD	DEV+1 /SAVE THE DEVICE NUMBER
888	01654	3052		DCA	DEVNUM
889	01655	1251		TAD	DEV+2 /AND THE HANDLER ENTRY POINT
890	01656	3051		DCA	DEVHAN
891	01657	1323	MOVEFN,	TAD	SAVFLAG /WAS IT A SAVE ?
892	01660	7740	M40,	SMA SZA CLA	
893	01661	5600		JMP I	GETFN /YES, JUST RETURN
894	01662	1041		TAD	NAME /NEW OR OLD, ANY NAME GIVEN ?

```

895 01663 7450      SNA
896 01664 5600      JMP I   GETFN   /NO, PROBABLY JUST A DEVICE
897 01665 3045      DCA     FNAME   /YES, SAVE IT
898 01666 1042      TAD     NAME+1
899 01667 3046      DCA     FNAME+1
900 01670 1043      TAD     NAME+2
901 01671 3047      DCA     FNAME+2
902 01672 1044      TAD     NAME+3
903 01673 3050      DCA     FNAME+3
904 01674 5600      JMP I   GETFN
905 01675 1323      ASKNAM, TAD     SAVFLAG /WAS THIS A SAVE ?
906 01676 7750      SPA SNA CLA
907 01677 5313      JMP     ASKNM   /NO, GO ASK FOR A NAME
908 01700 1045      TAD     FNAME   /IT WAS A SAVE, ANY OLD NAME TO USE ?
909 01701 7450      SNA
910 01702 5313      JMP     ASKNM   /NO, GO ASK FOR ONE
911 01703 3041      DCA     NAME    /YES, MOVE INTO NAME
912 01704 1046      TAD     FNAME+1
913 01705 3042      DCA     NAME+1
914 01706 1047      TAD     FNAME+2
915 01707 3043      DCA     NAME+2
916 01710 1050      TAD     FNAME+3
917 01711 3044      DCA     NAME+3
918 01712 5600      JMP I   GETFN
919 01713 7201      ASKNM, CLA IAC   /ASK FOR FILE NAME
920 01714 4576      JMS I   [TYPE
921 01715 2167      ASKFN
922 01716 4567      JMS I   [GETLIN /GET THE INPUT LINE
923 01717 7650      SNA CLA
924 01720 5543      JMP I   [WHAT   /???
925 01721 3030      DCA     NCHARS  /SET CHAR POINTER
926 01722 5223      JMP     NOSKIP  /GO GET THE NAME
927 01723 0000      SAVFLAG, 0
928 01724 0000      GETNAM, 0
929 01725 3041      DCA     NAME     /GET A FILE OR DEVICE NAME
930 01726 3042      DCA     NAME+1   /ZERO THE NAME BUFFER
931 01727 3043      DCA     NAME+2
932 01730 1057      TAD     0201     /USE DEFAULT EXT .BA
933 01731 3044      DCA     NAME+3
934 01732 1142      TAD     [NAME    /SETUP POINTER
935 01733 3024      DCA     TEMP2
936 01734 1112      TAD     07774   /SETUP SIZE (MAX 4 WORDS)
937 01735 3022      DCA     SIZE
938 01736 4545      NAMLUP, JMS I   [GETNC /GET A CHAR
939 01737 5724      JMP I   GETNAM
940 01740 7106      CLL RTL
941 01741 7006      RTL
942 01742 7006      RTL
943 01743 3424      DCA I   TEMP2   /SAVE IT
944 01744 4545      JMS I   [GETNC
945 01745 5724      JMP I   GETNAM
946 01746 1424      TAD I   TEMP2   /COMBINE THE 2
947 01747 3424      DCA I   TEMP2
948 01750 2024      ISZ     TEMP2
949 01751 2022      ISZ     SIZE    /ANY MORE ?

```

950	01752	5336		JMP	NAMLUP	
951	01753	5724		JMP I	GETNAM	
952	01754	7130	RENAME,	CLL CML	RAW	/SAVE USR AREA
953	01755	4200		JMS	GETFN	/GET FILE NAME
954	01756	6212		CIF	10	
955	01757	4456		JMS I	0200	/REMOVE USR
956	01760	0011		11		/AND RESTORE 10000-11777
957	01761	4555		JMS I	[SWAP	/SWAP OS8 RESIDENT
958	01762	5553		JMP I	[CMDDONE	
959	01763	1154	NEW,	TAD	[TXTAREA	/SCRATCH
960	01764	4562		JMS I	[SETEOF	
961	01765	3037		DCA	EOFLIN	/ZERO LAST LINE NUM
962	01766	3040		DCA	EOFLIN+1	
963	01767	4200		JMS	GETFN	/GET THE FILE NAME
964	01770	4555		JMS I	[SWAP	/REMOVE OS8
965	01771	5553		JMP I	[CMDDONE	

966	01773	1300		
967	01774	0423		
968	01775	2715		
969	01776	1332		
970	01777	7746		
971		2000	PAGE	
972	02000	0000	PUTFIL, 0	/WRITE THE FILE
973	02001	1154	TAD	[TXTAREA
974	02002	3026	DCA	PTR /GET POINTER TO TEXT
975	02003	1110	TAD	06201 /GET FIELD OF TEXT
976	02004	3541	DCA I	[PTR2FLD
977	02005	3023	DCA	TEMP /ZERO LINE SEQUENCER
978	02006	1140	TAD	[DSKBUF /GET ADDR OF DISK BUFFER
979	02007	3003	DCA	SWAPT1 /BUFFER POINTER
980	02010	1071	TAD	07600
981	02011	3004	DCA	SWAPT2 /DOUBLE WORD COUNTER
982	02012	1310	TAD	JMPINS /SET 3 WAY SWITCH
983	02013	3233	DCA	PUTJMP
984	02014	4537	PFLOOP, JMS I	[GETFIL /GET A CHAR FROM TEXT AREA
985	02015	5220	JMP	PFCTLZ /END OF FILE
986	02016	4231	JMS	PUTCH /OUTPUT IT
987	02017	5214	JMP	PFLOOP /DO NEXT CHAR
988	02020	1063	PFCTLZ, TAD	0232 /PUT CTRL-Z
989	02021	4231	JMS	PUTCH
990	02022	1064	TAD	07201 /BUFFER DUMP COUNT
991	02023	3230	DCA	PFTEMP
992	02024	4231	JMS	PUTCH /FILL WITH ZEROES
993	02025	2230	ISZ	PFTEMP
994	02026	5224	JMP	. =2
995	02027	5600	JMP I	PUTFIL /DONE
996	02030	0000	PFTEMP, 0	
997	02031	0000	PUTCH, 0	/PUT A CHAR ONTO THE OS8 FILE
998	02032	3006	DCA	SWAPT4 /SAVE THE CHAR
999	02033	7402	PUTJMP, HLT	/JUMP TO CORRECT PLACE
1000	02034	5304	JMP	PH1 /FIRST CHAR
1001	02035	5301	JMP	PH2 /SECOND CHAR
1002	02036	1310	PH3, TAD	JMPINS /RESTORE SWITCH
1003	02037	3233	DCA	PUTJMP
1004	02040	1006	TAD	SWAPT4 /GET THE CHAR
1005	02041	0072	AND	017 /LOW FOUR BITS
1006	02042	7110	CLL RAR	
1007	02043	7012	RTR	/INTO THE HIGH PART OF WORD TWO
1008	02044	7012	RTR	
1009	02045	1403	TAD I	SWAPT1 /COMBINE WITH CHAR 2
1010	02046	3403	DCA I	SWAPT1
1011	02047	1006	TAD	SWAPT4 /GET THE HIGH FOUR BITS
1012	02050	7006	RTL	
1013	02051	7006	RTL	/INTO THE HIGH PART OF WORD ONE
1014	02052	0124	AND	07400
1015	02053	1405	TAD I	SWAPT3 /COMBINE WITH WORD ONE
1016	02054	3405	DCA I	SWAPT3
1017	02055	2003	ISZ	SWAPT1 /BUMP POINTER
1018	02056	2004	ISZ	SWAPT2 /BUMP DOUBLE WORD COUNT
1019	02057	5631	JMP I	PUTCH /RETURN
1020	02060	4555	JMS I	[SWAP /SWAP IN OS8

```

1021 02061 4451 JMS I DEVHAN /WRITE THIS BUFFER
1022 02062 4200 4200
1023 02063 4000 DSKBUF
1024 02064 0000 WRBLOK, 0
1025 02065 5777 JMP OUERR
1026 02066 2311 ISZ OUSIZE /ANY ROOM LEFT ?
1027 02067 7410 SKP
1028 02070 5777 JMP OUERR /NO, ERROR
1029 02071 2264 ISZ WRBLOK /BUMP BLOCK NUMBER
1030 02072 2776 ISZ OULEN /BUMP ACTUAL SIZE
1031 02073 4555 JMS I ISWAP /SWAP BACK
1032 02074 1140 TAD DSKBUF /SET UP BUFFER POINTER
1033 02075 3003 DCA SWAPT1
1034 02076 1071 TAD 07600
1035 02077 3004 DCA SWAPT2 /SET UP COUNT
1036 02100 5631 JMP I PUTCH
1037 02101 1003 PH2, TAD SWAPT1 /SAVE POINTER TO FIRST
1038 02102 3005 DCA SWAPT3
1039 02103 2003 ISZ SWAPT1 /BUMP POINTER
1040 02104 1006 PH1, TAD SWAPT4 /GET CHAR
1041 02105 3403 DCA I SWAPT1 /INTO BUFFER
1042 02106 2233 ISZ PUTJMP /BUMP SWITCH
1043 02107 5631 JMP I PUTCH
1044 02110 5234 JMPINS, JMP PUTJMP+1
1045 02111 0000 OUSIZE, 0
1046 02112 0000 SWAP, 0 /SWAP OS8 RESIDENT
1047 02113 2053 ISZ SWPNUM /FLIP BYEBYE SWITCH
1048 02114 7000 NOP
1049 02115 4327 JMS SWAP2 /WITH OS8RES THROUGH OS8RES+577
1050 02116 3200 OS8RES
1051 02117 6201 CDF
1052 02120 4327 JMS SWAP2
1053 02121 3400 OS8RES+200
1054 02122 6211 CDF 10
1055 02123 4327 JMS SWAP2 for 8K patch to 5712
1056 02124 3600 OS8RES+400
1057 02125 6221 CDF 20
1058 02126 5712 JMP I SWAP
1059 02127 0000 SWAP2, 0
1060 02130 1071 TAD 07600
1061 02131 3005 DCA SWAPT3
1062 02132 1727 TAD I SWAP2
1063 02133 2327 ISZ SWAP2
1064 02134 3006 DCA SWAPT4
1065 02135 1727 TAD I SWAP2
1066 02136 2327 ISZ SWAP2
1067 02137 3347 DCA SWPFLD
1068 02140 5345 JMP +5
1069 02141 6201 SWPLUP, CDF 00
1070 02142 1003 TAD SWAPT1
1071 02143 3406 DCA I SWAPT4
1072 02144 2006 ISZ SWAPT4
1073 02145 1406 TAD I SWAPT4
1074 02146 3004 DCA SWAPT2
1075 02147 7402 SWPFLD, HLT

```

✓ 3 Same

1076 02150 1405
1077 02151 3003
1078 02152 1004
1079 02153 3405
1080 02154 2005
1081 02155 5341
1082 02156 6201
1083 02157 1003
1084 02160 3406
1085 02161 5727
1086 02162 4342
1087 02163 4501
1088 02164 4752
1089 02165 5546
1090 02166 0000
1091 02167 4752
1092 02170 5546
1093 02171 0157
1094 02172 4256
1095 02173 4616
1096 02174 1600

TAD I SWAPT3 c(3600)
DCA SWAPT1
TAD SWAPT2 c(3600)
DCA I SWAPT3
ISZ SWAPT3
JMP SWPLUR
CDF 00
TAD SWAPT1
DCA I SWAPT4
JMP I SWAP2

BAOFIL, 4342/4501/4752/5546/0

ASKFN, 4752/5546/0157/4256/4616/1600

1097	02176	2266	
1098	02177	2720	
1099		2200	
1100	02200	1550	RUN, PAGE
1101	02201	1547	TAD I [LINE+3 /RUNNH ?
1102	02202	1377	TAD I [LINE+4
1103	02203	7640	TAD (-3057
1104	02204	4546	SZA CLA
1105	02205	1172	JMS I [HEADING/GIVE A HEADING
1106	02206	3023	TAD [LINE+2 /SET UP FAKE LINE
1107	02207	1136	DCA TEMP
1108	02210	3026	TAD [WSSAVE
1109	02211	1123	DCA PTR
1110	02212	3024	TAD 07771
1111	02213	1426	DCA TEMP2
1112	02214	2026	TAD I PTR /MOVE FAKE LINE INTO "LINE"
1113	02215	3423	ISZ PTR
1114	02216	2023	DCA I TEMP
1115	02217	2024	ISZ TEMP
1116	02220	5213	ISZ TEMP2
1117	02221	2033	JMP .-5
1118	02222	5226	ISZ RUNFLAG /SET RUN FLAG
1119	02223	3033	JMP GFN
1120	02224	1052	DCA RUNFLAG /CLEAR THE RUN FLAG
1121	02225	3321	TAD DEVNUM /SAVE CURRENT DEVICE NUM
1122	02226	7201	DCA OLDDEV /INCASE WE CHANGE
1123	02227	4535	GFN, CLA IAC /SET SAVFLAG
1124	02230	1265	JMS I [GETFN /GET THE DEV:NAME,EX
1125	02231	3236	TAD XNAME /SET UP ENTER
1126	02232	1052	DCA SAVBLK /POINTER TO FILE NAME
1127	02233	6212	TAD DEVNUM /GET DEVICE NUMBER
1128	02234	4456	CIF 10
1129	02235	0003	JMS I 0200 /ENTER FILE
1130	02236	0000	3
1131	02237	0000	SAVBLK, 0 /STARTING BLOCK GOES HERE
1132	02240	5776	0 /SIZE GOES HERE
1133	02241	1236	JMP IOERR
1134	02242	3775	TAD SAVBLK /PUT BLOCK NUMBER
1135	02243	1237	DCA WRBLOK /INTO WRITE
1136	02244	3774	TAD SAVBLK+1 /PUT SIZE
1137	02245	3266	DCA OUSIZE /SOMEWHERE TOO
1138	02246	6212	DCA OULEN /ZERO BLOCK COUNT
1139	02247	4456	CIF 10
1140	02250	0011	JMS I 0200 /DUMP USR
1141	02251	4555	11
1142	02252	4773	JMS I [SWAP /AND NOW OS8
1143	02253	4555	JMS PUTFIL /DO THE SAVE
1144	02254	1033	JMS I [SWAP /GET OS8
1145	02255	3772	TAD RUNFLAG /SET NO SWAP BIT IF RUN
1146	02256	6212	DCA 7746
1147	02257	4455	CIF 10 /GET THE USR
1148	02260	0010	JMS I 07700
1149	02261	1052	10
1150	02262	6212	TAD DEVNUM /GET DEVICE NUMBER
1151	02263	4456	CIF 10
			JMS I 0200 /CLOSE THE FILE

```

1152 02264 0004      4
1153 02265 0041  XNAME, NAME
1154 02266 0000  OULEN, 0      /SIZE
1155 02267 5776'    JMP      IOERR
1156 02270 1033    TAD      RUNFLAG /WAS IT A RUN ?
1157 02271 7640    SZA CLA
1158 02272 5771'    JMP      DORUN /YES
1159 02273 1321    TAD      OLDDEV /IS OLD DEVICE
1160 02274 7041    CIA      /THE SAME AS
1161 02275 1052    TAD      DEVNUM /THE NEW ONE ??
1162 02276 7650    SNA CLA
1163 02277 5314    JMP      HNDLOK /YES, THE HANDLER IS OK
1164 02300 1321    TAD      OLDDEV /RESTORE DEVICE NUMBER
1165 02301 3052    DCA      DEVNUM
1166 02302 1144    TAD      [HANDLR+1
1167 02303 3310    DCA      DEVN /SET UP HANDLER LOAD ADDR
1168 02304 1052    TAD      DEVNUM
1169 02305 6212    CIF      10
1170 02306 4456    JMS I    0200
1171 02307 0001    1
1172 02310 0000  DEVN, 0
1173 02311 5776'    JMP      IOERR
1174 02312 1310    TAD      DEVN /RESET THE HANDLER ADDRESS
1175 02313 3051    DCA      DEVHAN
1176 02314 6212  HNDLOK, CIF 10 /GET RID OF THE USR
1177 02315 4456    JMS I    0200
1178 02316 0011    11
1179 02317 4555    JMS I    [SWAP /REMOVE OS8 AGAIN
1180 02320 5553    JMP I    [CMDDONE
1181 02321 0000  OLDDEV, 0
1182 02322 0000  FINDLN, 0      /FIND A LINE
1183 02323 1154    TAD      [TXTAREA
1184 02324 3023    DCA      TEMP
1185 02325 1023  SEARCH, TAD      TEMP /COMPARE THE NUMBER OF
1186 02326 3026    DCA      PTR /THIS LINE WITH THE SPOT
1187 02327 1423    TAD I    TEMP /IN THE TEXT AREA.
1188 02330 1103    TAD      07701
1189 02331 7450    SNA
1190 02332 5722    JMP I    FINDLN /NEW LINE GOES AT EOF
1191 02333 1120    TAD      077
1192 02334 7141    CLL CIA
1193 02335 1035    TAD      LINENO
1194 02336 7450    SNA
1195 02337 5347    JMP      SAME1ST
1196 02340 7620    SNL CLA
1197 02341 5722    JMP I    FINDLN /INSERT NEW LINE
1198 02342 2023    ISZ      TEMP
1199 02343 7410    SKP
1200 02344 4560    JMS I    [FINCR
1201 02345 4534  CONTIN, JMS I    [PASS /IF ITS GREATER KEEP SEARCHING.
1202 02346 5325    JMP      SEARCH
1203 02347 2023  SAME1ST, ISZ      TEMP /FIRST HALF OF LINE NUM SAME
1204 02350 7410    SKP
1205 02351 4560    JMS I    [FINCR
1206 02352 1423    TAD I    TEMP

```

1207	02353	7141	CLL CIA	/CHECK SECOND HALF
1208	02354	1036	TAD	LINENO+1
1209	02355	7450	SNA	
1210	02356	5365	JMP	SAME2ND /REPLACE OLD WITH NEW
1211	02357	7630	SZL CLA	
1212	02360	5345	JMP	CONTIN
1213	02361	4533	JMS I	[FDECK
1214	02362	1023	TAD	TEMP
1215	02363	3026	DCA	PTR
1216	02364	5722	JMP I	FINDLN /INSERT NEW LINE
1217	02365	4534	SAME2ND, JMS I	[PASS
1218	02366	5722	JMP I	FINDLN

1219	02371	2400	
1220	02372	7746	
1221	02373	2000	
1222	02374	2111	
1223	02375	2064	
1224	02376	2715	
1225	02377	4721	
1226		2400	PAGE
1227	02400	1377	DORUN, TAD (INFO+11/SET UP SOME OF INFO BLOCK
1228	02401	3010	DCA X10
1229	02402	6211	CDF 10
1230	02403	1051	TAD DEVHAN /SAVE DEVICE HANDLER ADDRESS (DSK:)
1231	02404	3410	DCA I X10
1232	02405	7126	CLL CML RTL /SAVE DEVICE NUMBER
1233	02406	3410	DCA I X10
1234	02407	6201	CDF
1235	02410	1776	TAD SAVBLK /SAVE STARTING BLOCK
1236	02411	6211	CDF 10
1237	02412	3410	DCA I X10
1238	02413	1045	TAD FNAME /SAVE FILE NAME
1239	02414	3410	DCA I X10
1240	02415	1046	TAD FNAME+1
1241	02416	3410	DCA I X10
1242	02417	1047	TAD FNAME+2
1243	02420	3410	DCA I X10
1244	02421	1050	TAD FNAME+3
1245	02422	3410	DCA I X10
1246	02423	6201	CDF
1247	02424	4315	JMS XMOVE /MOVE THIS PAGE INTO FIELD 1
1248	02425	6201	CDF
1249	02426	2400	DORUN
1250	02427	6211	CDF 10
1251	02430	7600	-200
1252	02431	6213	CDF CIF 10 /GO TO THE FIELD ONE COPY
1253	02432	4315	JMS XMOVE /MOVE THE HANDLER INTO FIELD 1
1254	02433	6201	CDF
1255	02434	4400	HANDLR
1256	02435	6211	CDF 10
1257	02436	7400	-400
1258	02437	1775	TAD INFO+1 /GET START OF BCOMP.SV
1259	02440	3245	DCA BCBLOK
1260	02441	6202	CIF
1261	02442	4774	JMS 7607 /READ IN THE COMPILER
1262	02443	3700	BCSIZE
1263	02444	0000	0
1264	02445	0000	BCBLOK, 0
1265	02446	5273	JMP WHUPS
1266	02447	4315	JMS XMOVE /MOVE BACK THE HANDLER
1267	02450	6211	CDF 10
1268	02451	4400	HANDLR
1269	02452	6201	CDF
1270	02453	7400	-400
1271	02454	7201	CLA IAC /OPEN THE TEMP FILE
1272	02455	4773	JMS 200
1273	02456	0003	3

```

1274 02457 2547 TMPBLK, TMPFIL
1275 02460 0000 0
1276 02461 5273 JMP WHUPS
1277 02462 4773 JMS 200 /RESET SYSTEM TABLES
1278 02463 0013 13 /AND REMOVE TENTATIVE FILES
1279 02464 1257 TAD TMPBLK /SAVE ITS START
1280 02465 3772 DCA INFO+10
1281 02466 1260 TAD TMPBLK+1
1282 02467 3777 DCA INFO+11 /AND ITS MAX LENGTH
1283 02470 3771 DCA 7644 /KILL R SWITCH
1284 02471 6203 CIF CDF
1285 02472 5770 JMP BCBEGR /GO START THE COMPILER
1286 02473 1367 WHUPS, TAD (123 /PRINT SY ERROR
1287 02474 4307 JMS TTYO
1288 02475 1366 TAD (131
1289 02476 4307 JMS TTYO
1290 02477 1365 TAD (15
1291 02500 4307 JMS TTYO
1292 02501 1364 TAD (12
1293 02502 4307 JMS TTYO
1294 02503 6041 TSF /WAIT FOR FLAG
1295 02504 5303 JMP .-1
1296 02505 6203 CDF CIF
1297 02506 5775 JMP 7605 /RETURN TO OS8
1298 02507 0000 TTYO, 0
1299 02510 6041 TSF
1300 02511 5310 JMP .-1
1301 02512 6046 TLS
1302 02513 7200 CLA
1303 02514 5707 JMP I TTYO
1304 02515 0000 XMOVE, 0
1305 02516 6214 RDF /GET CALLING FIELD
1306 02517 1363 TAD (6203 /PLUS CIF CDF 0
1307 02520 3340 DCA MOVRTN /FOR RETURN
1308 02521 4343 JMS GMOVE /GET FROM FIELD
1309 02522 3331 DCA MFFLD
1310 02523 4343 JMS GMOVE /GET ADDRESS
1311 02524 3342 DCA MFPTR
1312 02525 4343 JMS GMOVE /GET TO FIELD
1313 02526 3333 DCA MTFLD
1314 02527 4343 JMS GMOVE /GET COUNT
1315 02530 3343 DCA MCNT
1316 02531 7402 MFFLD, HLT
1317 02532 1742 TAD I MFPTR /MOVE IT
1318 02533 7402 MTFLD, HLT
1319 02534 3742 DCA I MFPTR
1320 02535 2342 ISZ MFPTR
1321 02536 2343 ISZ MCNT
1322 02537 5331 JMP MFFLD
1323 02540 7402 MOVRTN, HLT
1324 02541 5713 JMP I XMOVE
1325 02542 0000 MFPTR, 0
1326 MCNT,
1327 02543 0000 GMOVE, 0
1328 02544 1715 TAD I XMOVE /GET ARG FOR MOVE

```

1329	02545	2315	ISZ	XMOVE
1330	02546	5743	JMP I	GMOVE
1331	02547	0201	TMPFIL, FILENAME BASIC.TM	
1332	02550	2311		
1333	02551	0300		
1334	02552	2415		

1335	02563	6203			
1336	02564	0012			
1337	02565	0015			
1338	02566	0131			
1339	02567	0123			
1340	02570	7001			
1341	02571	7644			
1342	02572	7614			
1343	02573	0200			
1344	02574	7607			
1345	02575	7605			
1346	02576	2236			
1347	02577	7615			
1348		2600			
1349	02600	0000	GETCH,	PAGE 0	/GET A CHARACTER FROM THE TTY
1350	02601	1034		TAD OLDFLAG	/INPUT FROM A FILE ?
1351	02602	7640		SZA CLA	
1352	02603	5211		JMP FILEIN	/YES
1353	02604	6031		KSF	
1354	02605	5204		JMP	.-1
1355	02606	6036		KRB	
1356	02607	0062		AND	0177
1357	02610	5600		JMP I	GETCH
1358	02611	2360	FILEIN,	ISZ COUNT	/ANYTHING IN BUFFER ?
1359	02612	5236		JMP NOREAD	/YES, NO READ
1360	02613	1070		TAD 07200	/SET BUFFER COUNT
1361	02614	3360		DCA COUNT	
1362	02615	1140		TAD [DSKBUF	/SET BUFFER POINTER
1363	02616	3021		DCA RDPTR	
1364	02617	1357		TAD RDJMP	/RESET JUMP
1365	02620	3236		DCA NOREAD	
1366	02621	4555		JMS I [SWAP	/GET OS8
1367	02622	1343		TAD RDSIZE	/ANY ROOM LEFT ?
1368	02623	7450		SNA	
1369	02624	5320		JMP INERR	/BAD END OF IFILE
1370	02625	7001		IAC	
1371	02626	3343		DCA RDSIZE	
1372	02627	4451		JMS I DEVHAN	/READ NEXT BLOCK
1373	02630	0200		200	
1374	02631	4000		DSKBUF	
1375	02632	0000	RDBLOK,	0	
1376	02633	5320		JMP INERR	
1377	02634	2232		ISZ RDBLOK	/BUMP BLOCK NUMBER
1378	02635	4555		JMS I [SWAP	/AWAY WITH OS8
1379	02636	7402	NOREAD,	HLT	/3W UNPACK JUMP
1380	02637	5262		JMP INCHR1	
1381	02640	5257		JMP INCHR2	
1382	02641	1357	INCHR3,	TAD RDJMP	/RESET SWITCH
1383	02642	3236		DCA NOREAD	
1384	02643	1421		TAD I RDPTR	/GET LOW 4 BITS
1385	02644	2021		ISZ RDPTR	/BUMP POINTER
1386	02645	0124		AND 07400	/MASK IT
1387	02646	7112		CLL RTR	/SHIFT RIGHT 4
1388	02647	7012		RTR	
1389	02650	3023		DCA TEMP	/SAVE

Address	Hex	Dec	Label	Op	Opnd	Comment
1390	02651	1420		TAD I	RDTMP	/GET HIGH 4 BITS
1391	02652	0124		AND	07400	
1392	02653	1023		TAD	TEMP	/COMBINE THEM
1393	02654	7112		CLL RTR		/SHIFT RIGHT 4
1394	02655	7012		RTR		
1395	02656	5264		JMP	AND177	/GO FINISH
1396	02657	1021	INCHR2,	TAD	RDPTR	/SAVE ADDR OF FIRST WORD
1397	02660	3020		DCA	RDTMP	
1398	02661	2021		ISZ	RDPTR	/BUMP POINTER
1399	02662	1421	INCHR1,	TAD I	RDPTR	/GET NEXT CHAR
1400	02663	2236		ISZ	NOREAD	/BUMP SWITCH
1401	02664	0062	AND177,	AND	0177	/MASK 7 BITS
1402	02665	1116		TAD	07746	/CHECK FOR ^Z
1403	02666	7450		SNA		
1404	02667	5272		JMP	ENDOLD	/EOF
1405	02670	1066		TAD	032	/RESTORE CHAR
1406	02671	5600		JMP I	GETCH	
1407	02672	3034	ENDOLD,	DCA	OLDFLAG	/KILL OLD FLAG
1408	02673	1032		TAD	CHNFLAG	/WAS IT A RETURN FROM RUN ?
1409	02674	7650		SNA CLA		
1410	02675	5553		JMP I	[CMDDONE/NO,	JUST AN OLD COMMAND
1411	02676	3032		DCA	CHNFLAG	/KILL FLAG
1412	02677	1377		TAD	(INFO-4164	
1413	02700	3010		DCA	X10	/PICK UP NAME FROM INFO BLOCK
1414	02701	1410		TAD I	X10	/(WHICH IS SWAPPED OUT)
1415	02702	3045		DCA	FNAME	
1416	02703	1410		TAD I	X10	
1417	02704	3046		DCA	FNAME+1	
1418	02705	1410		TAD I	X10	
1419	02706	3047		DCA	FNAME+2	
1420	02707	1410		TAD I	X10	
1421	02710	3050		DCA	FNAME+3	
1422	02711	5553		JMP I	[CMDDONE/DONE WITH RETURN	
1423	02712	3050	OLDBAD,	DCA	FNAME+3	/TRY WITHOUT EXT
1424	02713	2023		ISZ	TEMP	/OR HAVE WE ALREADY ?
1425	02714	5334		JMP	OLDTRY	/NO, NOT YET
1426			IOERR,			
1427	02715	6212	INERRX,	CIF	10	
1428	02716	4456		JMS I	0200	/DISMISS USR
1429	02717	0011		11		
1430			OUERR,			
1431	02720	3034	INERR,	DCA	OLDFLAG	/KILL OLD STATUS
1432	02721	4555		JMS I	[SWAP	/OUT WITH OS8
1433	02722	4576		JMS I	[TYPE	/PRINT MESSAGE
1434	02723	2162		BADFIL		
1435	02724	5556		JMP I	[MAINLUP	
1436	02725	1154	OLD,	TAD	[TXTAREA/SCRATCH FILE	
1437	02726	4562		JMS I	[SETEOF	
1438	02727	3037		DCA	EOFLIN	/ZERO LAST LINE NUM
1439	02730	3040		DCA	EOFLIN+1	
1440	02731	4535		JMS I	[GETFN	/GET FILE NAME
1441	02732	7144		CLL CMA	RAL	/SET RETRY COUNT
1442	02733	3023		DCA	TEMP	
1443	02734	1152	OLDTRY,	TAD	[FNAME	/POINTER TO FILE NAME
1444	02735	3342		DCA	SB	/INTO LOOKUP CALL

1445	02736	1052	TAD	DEVNUM	/GET DEVICE NUMBER
1446	02737	6212	CIF	10	
1447	02740	4456	JMS I	0200	/LOOKUP FILE
1448	02741	0002	2		
1449	02742	0000	SB,	0	/START GOES HERE
1450	02743	0000	RDSIZE,	0	/SIZE GOES HERE
1451	02744	5312	JMP	OLDBAD	/BAD FILE
1452	02745	1342	TAD	SB	/MOVE BLOCK
1453	02746	7450	SNA		
1454	02747	2343	ISZ	RDSIZE	/SET COUNT TO 4095 IF NOT D.A.
1455	02750	3232	DCA	RDBLOK	
1456	02751	7201	CLA IAC		/SET SWITCH
1457	02752	3034	DCA	OLDFLAG	/INPUT COMING FROM FILE
1458	02753	7240	CLA CMA		/SET COUNT TO INITIALIZE READ
1459	02754	3360	DCA	COUNT	
1460	02755	4555	JMS I	ISWAP	/MOVE OSB
1461	02756	5556	JMP I	IMAINLUP/DO OLD	
1462	02757	5237	RDJMP,	JMP	NOREAD+1
1463	02760	0000	COUNT,	0	
1464	02761	0000	SETEOF,	0	/SET NEW EOF
1465	02762	3376	DCA	EOFADR	/SAVE ADDR
1466	02763	6214	RDF		/GET FIELD
1467	02764	1110	TAD	06201	
1468	02765	3374	DCA	EOFFLD	/SAVE IT
1469	02766	1120	TAD	077	/STORE EOF
1470	02767	3776	DCA I	EOFADR	
1471	02770	5761	JMP I	SETEOF	
1472	02771	0000	GETEOF,	0	/RETRIEVE EOF INFO
1473	02772	1376	TAD	EOFADR	/FIRST ADDR
1474	02773	3023	DCA	TEMP	
1475	02774	6201	EOFFLD,	COF	/THEN THE FIELD
1476	02775	5771	JMP I	GETEOF	
1477	02776	5000	EOFADR,	TXAREA	

```

1478 02777 3420
1479      3000
1480      PAGE
1481 03000 0000 LINE, /THE TELETYPE LINE BUFFER.
1482 03001 0000 WSNAME, 01010145164541334314264152441177016400
1483 03002 0145
1484 03003 6454
1485 03004 3343
1486 03005 4264
1487 03006 5244
1488 03007 1770
1489 03010 6400
1490 03011 5227 START, JMP RBASIC /IT WAS RAN
1491 03012 6213 CDF CIF 10
1492 03013 7201 CLA IAC
1493 03014 3777 DCA 7746 /NO SWAP
1494 03015 6201 CDF
1495 03016 4776 JMS I (7700 /FETCH USR
1496 03017 0010 10
1497 03020 6212 CIF 10
1498 03021 4775 JMS I (200 /RESET SYSTEM TABLES
1499 03022 0013 13
1500 03023 6046 TLS /SET TTY FLAG
1501 03024 4300 JMS GETDAT /SET UP TITLE
1502 03025 2032 ISZ CHNFLAG /TELL ABOUT RETURN FROM RUN
1503 03026 5774 JMP OLD /READ IN OLD WORK SPACE
1504 03027 6046 RBASIC, TLS
1505 03030 1120 TAD 077
1506 03031 3554 DCA I [TXTAREA
1507 03032 4300 JMS GETDAT /SET UP TITLE
1508 03033 1411 FINDSV, TAD I X11 /LOOK UP SOME SAVE FILES
1509 03034 7450 SNA
1510 03035 5254 JMP LUBUF /GO FIND BASIC.UF
1511 03036 3243 DCA XXXXSV /SAVE POINTER TO NAME
1512 03037 7201 CLA IAC
1513 03040 6212 CIF 10
1514 03041 4775 JMS I (200
1515 03042 0002 2
1516 03043 0000 XXXXSV, 0
1517 03044 0000 0
1518 03045 5275 JMP NG
1519 03046 7001 IAC
1520 03047 6211 CDF 10
1521 03050 1243 TAD XXXXSV
1522 03051 3410 DCA I X10 /SAVE BLOCK PLUS 1
1523 03052 6201 CDF /IN INFO AREA
1524 03053 5233 JMP FINDSV
1525 03054 7201 LUBUF, CLA IAC /FIND BASIC.UF
1526 03055 6212 CIF 10
1527 03056 4775 JMS I (200
1528 03057 0002 2
1529 03060 3333 BUFN
1530 03061 0000 0
1531 03062 5265 JMP .+3 /OK IF MISSING
1532 03063 1260 TAD .-3

```

*Not
quite
same*

1533	03064	7001	IAC		/SAVE BLOCK +1
1534	03065	6211	CDF	10	
1535	03066	3410	DCA I	X10	
1536	03067	6201	CDF		
1537	03070	7201	CLA IAC		/TYPE WITH NO CARRIAGE RETURN
1538	03071	4576	JMS I	[TYPE	/"OLD OR NEW -- "
1539	03072	3350	OLDNEW		
1540	03073	4555	JMS I	[SWAP	
1541	03074	5773'	JMP	MAINLUP	
1542	03075	4576	JMS I	[TYPE	/PART OF SYSTEM MISSING
1543	03076	3337	MISING		
1544	03077	5772'	JMP	7605	
1545	03100	0000	GETDAT, 0		/PUT OS8 DATE INTO THE TITLE
1546	03101	6211	CDF	10	
1547	03102	1771'	TAD	7666	/GET DATE WORD
1548	03103	6201	CDF		
1549	03104	3024	DCA	TEMP2	/SAVE IT
1550	03105	1024	TAD	TEMP2	
1551	03106	7450	SNA		
1552	03107	5700	JMP I	GETDAT	/NO DATE
1553	03110	0124	AND	07400	/GET MONTH
1554	03111	7106	CLL RTL		/SHIFT SOME
1555	03112	7006	RTL		
1556	03113	7006	RTL		
1557	03114	1370	TAD	(MONTHS-3	
1558	03115	3012	DCA	X12	
1559	03116	1367	TAD	(DATE-1	/SET UP POINTER TO DATE
1560	03117	3013	DCA	X13	
1561	03120	1024	TAD	TEMP2	/GET DAY
1562	03121	7012	RTR		
1563	03122	7010	RAR		
1564	03123	0100	AND	037	
1565	03124	1366	TAD	(NUMS-1	/GET NUMBERS
1566	03125	3023	DCA	TEMP	
1567	03126	1423	TAD I	TEMP	
1568	03127	3413	DCA I	X13	/INTO DATE
1569	03130	1412	TAD I	X12	/GET MONTH CHARS
1570	03131	3413	DCA I	X13	
1571	03132	1412	TAD I	X12	
1572	03133	3413	DCA I	X13	
1573	03134	1024	TAD	TEMP2	/GET YEAR
1574	03135	0365	AND	(7	
1575	03136	1364	TAD	(21	
1576	03137	7106	CLL RTL		
1577	03140	7006	RTL		
1578	03141	7006	RTL		
1579	03142	2013	ISZ	X13	/THE WORD WITH -7 IS THERE
1580	03143	3413	DCA I	X13	/STORE LAST DIGIT OF YEAR
1581	03144	5700	JMP I	GETDAT	

1582	03164	0021		
1583	03165	0007		
1584	03166	3227		
1585	03167	1324		
1586	03170	3175		
1587	03171	7666		
1588	03172	7605		
1589	03173	0604		
1590	03174	2725		
1591	03175	0200		
1592	03176	7700		
1593	03177	7746		
1594		3200		
1595	03200	1653	PAGE MONTHS, 165314257	/-JAN
1596	03201	4257		
1597	03202	1647	164714643	/-FEB
1598	03203	4643		
1599	03204	1656	165614263	/-MAR
1600	03205	4263		
1601	03206	1642	164216263	/-APR
1602	03207	6263		
1603	03210	1656	165614272	/-MAY
1604	03211	4272		
1605	03212	1653	165316657	/-JUN
1606	03213	6657		
1607	03214	1653	165316655	/-JUL
1608	03215	6655		
1609	03216	1642	164216650	/-AUG
1610	03217	6650		
1611	03220	1664	166414661	/-SEP
1612	03221	4661		
1613	03222	1660	166014465	/-OCT
1614	03223	4465		
1615	03224	1657	165716067	/-NOV
1616	03225	6067		
1617	03226	1645	164514644	/-DEC
1618	03227	4644		
1619	03230	2122	NUMS, 21221212312124121251212612127121301213112132	
1620	03231	2123		
1621	03232	2124		
1622	03233	2125		
1623	03234	2126		
1624	03235	2127		
1625	03236	2130		
1626	03237	2131		
1627	03240	2132		
1628	03241	2221	22211222212223122241222512226122271223012231	
1629	03242	2222		
1630	03243	2223		
1631	03244	2224		
1632	03245	2225		
1633	03246	2226		
1634	03247	2227		
1635	03250	2230		
1636	03251	2231		

```

1637 03252 2232      223212321;232212323;232412325;232612327;2330
1638 03253 2321
1639 03254 2322
1640 03255 2323
1641 03256 2324
1642 03257 2325
1643 03260 2326
1644 03261 2327
1645 03262 2330
1646 03263 2331      2331123321242112422
1647 03264 2332
1648 03265 2421
1649 03266 2422
1650 03267 3277      NAMLIST, BASICN
1651 03270 3303      BCOMPN
1652 03271 3307      BLOADN
1653 03272 3313      BRTSN
1654 03273 3317      BAFN
1655 03274 3323      BSFN
1656 03275 3327      BFFN
1657 03276 0000      0
1658 03277 0201      BASICN, FILENAME BASIC.SV
1659 03300 2311
1660 03301 0300
1661 03302 2326
1662 03303 0203      BCOMPN, FILENAME BCOMP.SV
1663 03304 1715
1664 03305 2000
1665 03306 2326
1666 03307 0214      BLOADN, FILENAME BLOAD.SV
1667 03310 1701
1668 03311 0400
1669 03312 2326
1670 03313 0222      BRTSN, FILENAME BRTS.SV
1671 03314 2423
1672 03315 0000
1673 03316 2326
1674 03317 0201      BAFN, FILENAME BASIC.AF
1675 03320 2311
1676 03321 0300
1677 03322 0106
1678 03323 0201      BSFN, FILENAME BASIC.SF
1679 03324 2311
1680 03325 0300
1681 03326 2306
1682 03327 0201      BFFN, FILENAME BASIC.FF
1683 03330 2311
1684 03331 0300
1685 03332 0606
1686 03333 0201      BUFN, FILENAME BASIC.UF
1687 03334 2311
1688 03335 0300
1689 03336 2506
1690 03337 5257      MISING, 5257;4460;5661;5546;6546;0164;7264;6546;5600
1691 03340 4460

```

1692	03341	5661
1693	03342	5546
1694	03343	6546
1695	03344	0164
1696	03345	7264
1697	03346	6546
1698	03347	5600
1699	03350	5746
1700	03351	7001
1701	03352	6063
1702	03353	0160
1703	03354	5545
1704	03355	1616
1705	03356	0000
1706		
1707	00133	0547
1708	00134	0533
1709	00135	1600
1710	00136	1242
1711	00137	1444
1712	00140	4000
1713	00141	1467
1714	00142	0041
1715	00143	1215
1716	00144	4401
1717	00145	1332
1718	00146	1251
1719	00147	3004
1720	00150	3003
1721	00151	0044
1722	00152	0045
1723	00153	0600
1724	00154	5000
1725	00155	2112
1726	00156	0604
1727	00157	0731
1728	00160	0563
1729	00161	3001
1730	00162	2761
1731	00163	2322
1732	00164	2771
1733	00165	3000
1734	00166	1200
1735	00167	0200
1736	00170	1553
1737	00171	1136
1738	00172	3002
1739	00173	0356
1740	00174	0400
1741	00175	0525
1742	00176	0471
1743	00177	2600

OLDNEW, 5746170011606310160155451161610

SS

ALT	0234	EOFADR	2776	MOVE	0731	037	0100
AND177	2664	EOFFLD	2774	MOVEFN	1657	0400	0126
ARROW	0315	EOFLIN	0037	MOVENT	0741	06171	0105
ASKFN	2167	EXPAND	1000	MOVLUP	0743	06201	0110
ASKNAM	1675	FDECR	0547	MOVRTN	2540	06211	0106
ASKNM	1713	FILEIN	2611	MSGALT	1116	072	0067
BADFIL	2162	FINCR	0563	MSGBIG	1105	07200	0070
BAFN	3317	FINDLN	2322	MSGRDY	1123	07201	0064
BASICN	3277	FINDSV	3033	MSGT00	0571	07400	0124
BCBEGN	7001	FNAME	0045	MSGWHA	1126	07405	0102
BCBLOK	2445	FRSTDI	1506	MTFLD	2533	07600	0071
BCOMPN	3303	GETCH	2600	M40	1660	07607	0054
BCSIZE	3700	GETDAT	3100	NAME	0041	07634	0114
BFFN	3327	GETEOF	2771	NAMLST	3267	077	0120
BLOADN	3307	GETFIL	1444	NAMLUP	1736	07700	0055
BRTSN	3313	GETFN	1600	NCHARS	0030	07701	0103
BSFN	3323	GETHAN	1642	NDIGS	0027	07706	0065
BSKIP	1616	GETLIN	0200	NEW	1763	07741	0101
BUFN	3333	GETNAM	1724	NG	3075	07745	0121
BYE	1101	GETNC	1332	NOCOMD	0614	07746	0116
BYEBYE	1076	GFN	2226	NOEXT	1361	07760	0125
CARRET	0334	GFRET	1542	NOFUSR	1615	07761	0060
CHNFLA	0032	GMOVE	2543	NOREAD	2636	07764	0061
CMDDON	0600	HANDLR	4400	NORM	1136	07766	0113
COFLAG	0031	HEADIN	1251	NOSKIP	1623	07770	0107
COMLUP	1202	HNDLOK	2314	NOTLAS	0645	07771	0123
COMMAN	1200	IGNORE	0203	NUMDUN	1425	07772	0104
COMTBL	1220	INCHR1	2662	NUMS	3230	07774	0112
CONTIN	2345	INCHR2	2657	NZDIGI	1523	PASS	0533
CONV	1270	INCHR3	2641	OLD	2725	PFCTLZ	2020
COREOV	1073	INERR	2720	OLDBAD	2712	PFL00P	2014
COUNT	2760	INERRX	2715	OLDDEV	2321	PFTMP	2030
CRLF	0356	INFO	7604	OLDFLA	0034	PH1	2104
CRSWIT	0547	INSERT	0646	OLDNEW	3350	PH2	2101
CTRLO	1553	IOERR	2715	OLDTRY	2734	PH3	2036
CTRLOX	1566	ISTEXT	0264	OS8RES	3200	PTR	0026
DATE	1325	JMPINS	2110	QUERR	2720	PTRFLD	0746
DEV	1647	LASTDI	1505	QULEN	2266	PTR2FL	1467
DEVHAN	0051	LASTWD	1070	QUSIZE	2111	PUTCH	2031
DEVN	2310	LEFTTX	1526	Q10	0111	PUTFIL	2000
DEVNUM	0052	LINE	3000	Q12	0122	PUTJMP	2033
DIGIT	1507	LINENO	0035	Q13	0075	RBASIC	3027
DIG1	0463	LINFTX	1550	Q137	0115	ROBLOK	2632
DIG123	0467	LIST	1400	Q14	0127	ROJMP	2757
DIG2	0453	LISTNH	1411	Q17	0072	ROPTR	0021
DIG23	0461	LNDIG	0400	Q177	0062	RDSIZE	2743
DIG3	0450	LSTLUP	1436	Q200	0056	ROTMP	0020
DIG4	0442	LUBUF	3054	Q201	0057	RENAME	1754
DIG456	0446	LWREOF	0727	Q212	0077	RIGHTY	0307
DIG5	0432	MAINLU	0604	Q215	0076	RITETX	1533
DIG56	0440	MCNT	2543	Q232	0063	RUN	2200
DIG6	0427	MFFLD	2531	Q237	0074	RUNDO	2432
DORUN	2400	MFPTR	2542	Q260	0073	RUNFLA	0033
DSKBUF	4000	MISING	3337	Q32	0066	SAME1S	2347
ENDOLD	2672	MONTHS	3200	Q36	0117	SAME2N	2365

SAVBLK 2236 ZEROTX 1544
SAVE 2223
SAVFLA 1723
SB 2742
SCRATC 1131
SEARCH 2325
SETEOF 2761
SIZE 0022
START 3011
SWAP 2112
SWAPT1 0003
SWAPT2 0004
SWAPT3 0005
SWAPT4 0006
SWAP2 2127
SWPFLD 2147
SWPLUP 2141
SWPNUM 0053
TCRLF 0521
TEMP 0023
TEMP2 0024
TESTCH 1346
TITLE 1312
TLOOP 0476
TMPBLK 2457
TMPFIL 2547
TMPFLD 0704
TMP2FL 1037
TOWARD 0025
TTYO 2507
TTYOUT 0525
TWOFLD 0706
TWO2FL 1041
TXTARE 5000
TYPE 0471
USEDISK 1636
VERS 0007
VERSION 0100
WHAT 1215
WHUPS 2473
WRBLOK 2064
WSNAME 3000
WSSAVE 1242
XMOVE 2515
XNAME 2265
XTITLE 1265
XXXXSV 3043
X10 0010
X11 0011
X12 0012
X13 0013
X14 0014
X15 0015
X16 0016
X17 0017

[illegible]

[illegible]

MONTHS	1557	1595#								
MOVE	413	450#	533							
MOVEFN	891#									
MOVENT	458#									
MOVLUP	460#	471								
MOVRTN	1307	1323#								
MSGALT	142	553#								
MSGBIG	535	544#								
MSGRDY	364	558#								
MSGTOO	171	353#								
MSGWHA	561#	606								
MTFLD	1313	1318#								
M40	860	892#								
NAME	57#	702	866	868	894	898	900	902	911	913
	915	917	929	930	931	933	934	1153		
NAMLST	32	1650#								
NAMLUP	704	938#	950							
NCHARS	48#	115	158	165	173	188	196	199	208	683
	684	727	857	925						
NDIGS	47#	114	201	203	211	240	241	570	588	728
	788	798	801							
NEW	622	959#								
NG	1518	1542#								
NOCOMD	368	373#								
NOEXT	699	705#								
NOFUSR	848	857#								
NOREAD	1359	1365	1379#	1383	1400	1462				
NORM	221	569#	574	577	589	741				
NOSKIP	863#	926								
NOTLAS	389	398#								
NUMDUN	730	735	739#							
NUMS	1565	1619#								
NZDIGI	797	801#								
OLD	612	1436#	1503							
OLDBAD	1423#	1451								
OLDEV	1121	1159	1164	1181#						
OLDFLA	52#	153	190	205	369	1350	1407	1431	1457	
OLDNEW	1539	1699#								
OLDTRY	1425	1443#								
OS8RES	21#	1050	1053	1056						
OUERR	1025	1028	1430#							
OULEN	1030	1137	1154#							
OUSIZE	1026	1045#	1136							
O10	97#	440	445	468	784	809				
O12	106#	233								
O13	85#	818								
O137	101#	193								
O14	111#	833								
O17	82#	795	1005							
O177	74#	829	1356	1401						
O200	70#	881	955	1128	1139	1151	1170	1177	1428	1447
O201	71#	932								
O212	87#	226	823							
O215	86#	224	820							
O232	75#	988								
O237	84#	302	309	815						
O260	83#	802								
O32	78#	1405								
O36	103#	144								
O37	88#	479	1564							

[illegible]

[illegible]

X	1511		1521								
XSV		1516#									
X10	31#	1228	1231	1233	1237	1239	1241	1243	1245	1413	
	1414	1416	1418	1420	1522	1535					
X11	32#	1508									
X12	33#	1558	1569	1571							
X13	34#	1560	1568	1570	1572	1579	1580				
X14	35#										
X15	36#										
X16	37#										
X17	38#										
ZEROTX	814	818#									
L0133	1213										
L0134	1201	1217									
L0135	1123	1440									
L0136	1107										
L0137	984										
L0140	978	1032	1362								
L0141	976										
L0142	934										
L0143	924										
L0144	878	1166									
L0145	729	734	938	944							
L0146	723	1104									
L0147	719	1101									
L0150	718	1100									
L0151	702										
L0152	635	1443									
L0153	568	726	749	751	958	965	1180	1410	1422		
L0154	564	739	959	973	1183	1436	1506				
L0155	540	849	957	964	1020	1031	1141	1143	1179	1366	
	1378	1432	1460	1540							
L0156	536	607	1435	1461							
L0157	533										
L0160	490	1200	1205								
L0161	457										
L0162	449	505	565	960	1437						
L0163	398	743									
L0164	390	481									
L0165	375	455	458								
L0166	371										
L0167	366	922									
L0170	293	748									
L0171	221	741									
L0172	175	215	377	599	686	1105					
L0173	169	207	314	362	634	646					
L0174	163	737									
L0175	157	194	225	227	752						
L0176	141	170	363	534	605	644	920	1433	1538	1542	
L0177	117										
L1376	660										
L1377	652										
L1576	731										
L1577	720										
L1773	876										
L1774	874										
L2377	1102										
L2563	1306										
L2564	1292										

±L2567	1286		
±L2577	1227		
±L2777	1412		
±L3164	1575		
±L3165	1574		
±L3166	1565		
±L3167	1559		
±L3170	1557		
±L3175	1498	1514	1527
±L3176	1495		

```
1      /OS/8 BASIC COMPILER
2      /
3      /DEC-S8-LBASA-A-LA
4      /
5      /COPYRIGHT,1972
6      /
7      /DIGITAL EQUIPMENT CORPORATION
8      /MAYNARD,MASSACHUSETTS 01754
9      /
10     /AUGUST 19, 1972
11     /
12     /HANK MAURER
13     /
14     /
15     /
16     /
17     0100      VERSION=100
```


18		0006	*6	
19	00006	6635	XABORT, ABORT	/ADDR OF ABORT ROUTINE
20	00007	0100	VERS, VERSION	/VERSION NUMBER
21	00010	7603	X10, INFO-1	/AUTO INDEX REGISTERS
22	00011	7325	X11, NAMLST-1	
23	00012	7577	X12, INFO-5	
24	00013	7773	X13, BOSINFO-1	
25	00014	1133	OSTACK, STACKO-1	/OPERAND STACK POINTER
26	00015	7117	STACK, STACKA-1	/GENERAL STACK POINTER
27	00016	2571	NEXT, FREE-1	/NEXT FREE LOCATION
28	00017	0000	CHRPTR, 0	/INPUT BUFFER POINTER
29	00020	0000	NCHARS, 0	/SIZE OF INPUT LINE
30	00021	7774	TEMP, -4	
31	00022	0000	TEMP2, 0	
32	00023	0000	DECPT, 0	/SET 1 IF .
33	00024	0000	NDIGIT, 0	/NUM DIGITS RIGHT OF .
34	00025	0000	EXPON, 0	/EXPONENT FOR NUM CONV
35	00026	0000	TYPE, 0	/TYPE OF CURRENT OPERAND
36	00027	0000	SYMBOL, 0	/SYMBOL NUMBER OF CUR. OPERAND
37	00030	0000	LEFT, 0	/LEFT SIDE SWITCH
38	00031	0000	OLDOP, 0	/OLD OPERATOR
39	00032	0000	NEWOP, 0	/NEW OPERATOR
40	00033	0000	TMPCNT, 0	/TEMP COUNTER
41	00034	0003	TMPLVL, 3	/TEMP LEVEL
42	00035	0000	STMPCT, 0	/TEMP COUNT (STRINGS)
43	00036	0001	STMPLV, 1	/TEMP LEVEL (STRINGS)
44	00037	0000	STPTR, 0	/POINTER TO S.T. ENTRY
45	00040	7377	VARCNT, -401	/NUMBER OF POSSIBLE NUMERIC
46				/VARIABLES, LITERALS, AND TEMPS
47	00041	7377	SVCNT, -401	/SAME FOR STRING VARS
48	00042	7737	ACNT, -41	/ARRAY COUNTER
49	00043	7737	SACNT, -41	/STRING ARRAY COUNTER
50	00044	0000	LOCTRH, 0	/HIGH ORDER LOCATION COUNTER
51	00045	0000	LOCTRL, 0	/LOW ORDER " "
52	00046	0000	BLOCK, 0	/START BLOCK OF TEMP FILE
53	00047	0000	HIFLD, 0	/HIGHEST CORE FIELD
54	00050	0000	BRTS, 0	/START OF BRTS.SV
55	00051	0000	DLSIZE, 0	/NEG. SIZE OF DATA LIST
56	00052	0000	ABORTX, 0	/START OF EDITOR
57	00053	0000	LINEH, 0	/LINE NUMBER (HIGH)
58	00054	0000	LINEL, 0	/LINE NUMBER (LOW)
59	00055	0000	MODE, 0	/INTERPRETER MODE
60	00056	0000	TYPE1, 0	/TYPE AFTER JMS GETA1
61	00057	0000	SYMBL1, 0	/SYM # AFTER JMS GETA1
62	00060	0000	OLDSTK, 0	/STACK SAVER FOR DEF
63	00061	0000	ARGCNT, 0	/ARG COUNTER FOR DEF
64			PCRLF,	/CR SWITCH FOR PRINT STMT
65			DACNT,	/ARG COUNT FOR UDEF STMT
66			FORJMP,	/FOR LOOP JUMP INSTR
67			NOSN,	/STMT NUMBER PRESENT SWITCH
68			COLON,	/: SWITCH FOR GETFN ROUTINE
69	00062	0000	JAROND, 0	/END OF DEF ADDR GOES HERE (INDIRECTLY)
70	00063	0000	IFNREG, 0	/CONTENTS OF IFN REG
71	00064	0000	SSREG1, 0	/EXECUTION TIME CONTENTS
72	00065	0000	SSREG2, 0	/OF THE SS REGISTERS

73	00066	7117	STKLVL, STACKA-1	/STACK BASE LEVEL
74	00067	0000	FINDEX, 0	/FOR LOOP INDEX
75	00070	0000	SETFLD, 0	/FIELD CHANGE RTNE FOR LUKUP2
76	00071	6211	LUFLO, CDF	/FIELD OF ENTRY FOR LUKUP2
77	00072	5470	JMP I	
78	00073	5706	QERMSG, ERMSG	
79	00074	0547	QLODSN, LODSN	/SUBROUTINE POINTERS
80	00075	6736	QCHKWD, CHKWD	
81	00076	5542	QMODSET, MODSET	
82	00077	3632	QSNUM, SNUM	
83	00100	6446	QOUTWRD, OUTWRD	
84	00101	4021	QSAVECP, SAVECP	
85	00102	4035	QGETC, GETC	
86	00103	4010	QGETCWB, GETCWB	
87	00104	4027	QRESTCP, RESTCP	
88	00105	2600	QEXPR, EXPR	
89	00106	3400	QOUTOPR, OUTOPR	
90	00107	0201	QNEWLIN, NEWLIN	
91	00110	0205	QREMARK, REMARK	
92	00111	3517	QGETA1, GETA1	
93	00112	4303	QLOADSS, LOADSS	
94	00113	6555	QCHECKC, CHECKC	
95	00114	5600	QGETNAM, GETNAM	
96	00115	4102	QCOMARP, COMARP	
97	00116	2400	QLOOKUP, LOOKUP	
98	00117	2200	QLUKUP2, LUKUP2	
99	00120	4117	QLOAD, LOAD	
100	00121	4062	QPUSH, PUSH	
101	00122	4052	QPOP, POP	
102	00123	4073	QPUSHO, PUSHO	
103	00124	4251	QSAVAC, SAVAC	
104	00125	4000	QBACK1, BACK1	
105	00126	5000	QNUMBER, NUMBER	
106	00127	5455	QSTRING, STRING	
107	00130	3600	QLETTER, LETTER	
108	00131	3616	QDIGIT, DIGIT	
109	00132	3336	QNOREGS, NOREGS	
110	00133	0400	Q400, 400	
111			NAME1,	/VARIABLE OR FUNCT NAME
112	00134	0000	WORD1, 0	/3 WORD LITERAL BUFFER
113			NAME2,	
114	00135	0000	WORD2, 0	
115			NAME3,	
116	00136	0000	WORD3, 0	
117	00137	0000	ACO, 0	/FAC OVERFLOW WD
118	00140	0000	OP1, 0	/4 WORD ARG FOR "NUMBER"
119	00141	0000	OP2, 0	
120	00142	0000	OP3, 0	
121	00143	0000	OPO, 0	

```

122      7604      INFO=7604      /INFORMATION AREA
123      /INFO      STARTING BLOCK +1 OF BASIC,SV
124      /INFO+1      STARTING BLOCK +1 OF BCOMP,SV
125      /INFO+2      STARTING BLOCK +1 OF BLOAD,SV
126      /INFO+3      STARTING BLOCK +1 OF BRTS,SV
127      /INFO+4      STARTING BLOCK +1 OF BASIC,AF
128      /INFO+5      STARTING BLOCK +1 OF BASIC,SF
129      /INFO+6      STARTING BLOCK +1 OF BASIC,FF
130      /INFO+7      STARTING BLOCK +1 OF BASIC,UF
131      /INFO+10     STARTING BLOCK OF BASIC,TM
132      /INFO+11     SIZE IN BLOCKS OF BASIC,TM
133      /INFO+12     INPUT HANDLER ENTRY ADDRESS
134      /INFO+13     SIZE AND DEVICE NUMBER OF INPUT FILE
135      /INFO+14     STARTING BLOCK OF INPUT FILE
136      /INFO+15     THROUGH
137      /INFO+20     NAME OF WORKSPACE
138      /
139      /
140      7774      BOSINFO=7774      /BOS PARAMETER AREA
141      1600      EDTSI2=1600      /SIZE OF BASIC,SV
142      3012      EDTBGN=3012      /RESTART FOR EDITOR
143      1712      ERMSG2=1712      /POST PROCESSOR ERROR SWITCH
144      7570      EOST=7570      /UPPER LIMIT FOR SYMBOL TABLE
145      4400      INDEVH=4400      /INPUT DEVICE HANDLER
146      7000      LINE=7000      /LINE BUFFER
147      0120      LINMAX=120      /MAXIMUM BASIC STMT
148      7120      STACKA=7120      /MAIN STACK
149      0060      STAKSZ=60      /SIZE OF MAIN STACK
150      /OPERAND STACK DEFINED IN=LINE
151      7200      INBUF=7200      /INPUT BUFFER
152      /
153      /
154      /FIELD ONE STUFF
155      /
156      /
157      0000      OUBUF=0      /OUTPUT BUFFER
158      0400      VARST=400      /VARIABLE SYMBOL TABLE
159      1036      SVARST=VARST+436 /STRING VAR SYMBOL TABLE
160      2132      ARAYST=SVARST+1074 /ARRAY SYMBOL TABLE
161      2332      SARYST=ARAYST+200 /STRING ARRAY SYMBOL TABLE
162      2532      SNUMS=SARYST+200 /STMT NUMBER BUCKETS
163      2556      TEMPS=SNUMS+24 /NUMERIC TEMP BUCKET
164      2560      STEMP=TEMPS+2 /STRING TEMP BUCKET
165      2562      LITRL=STEMP+2 /NUMERIC LITERAL BUCKET
166      2564      SLITRL=LITRL+2 /STRING LITERAL BUCKET
167      2566      DATLST=SLITRL+2 /DATA LIST
168      2570      FUNCTN=DATLST+2 /FUNCTION LIST
169      2572      FREE=FUNCTN+2 /START OF FREE CORE

```

170	/	INTERPRETER OPCODES
171	/	
172	/	MEMORY REFERENCE SET
173	0000	FADD=0000
174	0400	FSUB=0400
175	1000	FMPY=1000
176	1400	FDIV=1400
177	2000	FLDA=2000
178	2400	FSTA=2400
179	3000	FISUB=3000
180	3400	FIDIV=3400
181	4000	LSS1=4000
182	4400	LSS2=4400
183	5400	JEOF=5400
184	6000	LOADSN=6000
185	/	
186	/	JOC CLASS
187	5000	JSUB=5000
188	5001	JUMP=5001
189	5002	JGE=5002
190	5003	JNE=5003
191	5004	JGT=5004
192	5005	JLT=5005
193	5006	JEQ=5006
194	5007	JLE=5007
195	5010	JFOR=5010
196	/	
197	/	ARRAY CLASS
198	6400	AISUB=6400
199	6440	AFADD=6440
200	6500	AFSUB=6500
201	6540	AFMPY=6540
202	6600	AFDIV=6600
203	6640	AFLDA=6640
204	6700	AFSTA=6700
205	6740	AIDIV=6740
206	/	
207	/	STRING CLASS
208	0000	SCON=FADD
209	0400	SCOMP=FSUB
210	1000	SREAD=FMPY
211	2000	SLOAD=FLDA
212	2400	SSTORE=FSTA
213	6400	SACON=AI SUB
214	6440	SACOMP=AFADD
215	6500	SAREAD=AFSUB
216	6640	SALOAD=AFLDA
217	6700	SASTOR=AFSTA
218	/	
219	/	OPERATE CLASS
220	7401	SETJF=7401
221	7421	RND0=7421
222	7441	STOP=7441
223	7461	SRDL=7461
224	7414	CHN=7414

225	7521	NRDL=7521
226	7434	CLOSEF=7434
227	7474	OPENAV=7474
228	7454	OPENAF=7454
229	7534	OPENNV=7534
230	7514	OPENNF=7514
231	7501	CLRFN=7501
232	7402	FILENO=7402
233	7403	FNEG=7403
234	7404	RET=7404
235	7405	REST=7405
236	7406	LSS1AC=7406
237	7407	LSS2AC=7407
238	7410	FESC=7410
239	7411	READ=7411
240	7412	WRITE=7412
241	7413	SWRITE=7413
242	7561	SMODE=7561
243	7541	NMODE=7541
244	7416	FUNC1=7416
245	7417	FUNC2=7417
246	7400	FUNC3=7400
247	7415	FUNC4=7415
248	7540	USE=7540

```

249      / ASSEMBLE LINE
250      0201      *WORD1+45      /ORG PAST BIGGEST STRING LIT
251      00201    4502    NEWLIN, JMS I  QGETC      /ANY CHARS LEFT ?
252      00202    5205      JMP      REMARK      /NO, LINE ENDED OK
253      00203    4473      JMS I  QERMSG      /EXTRA CHARACTERS
254      00204    3003      3003
255      00205    3062    REMARK, DCA      NOSN      /CLEAR STMT NUMBER SWITCH
256      00206    1034      TAD      TEMPLVL      /RESET TEMP LEVELS
257      00207    3033      DCA      TMPCNT      /FOR NUMERIC
258      00210    1036      TAD      STMPVL      /AND STRING
259      00211    3035      DCA      STMPCT      /TEMPORARIES
260      00212    1377      TAD      (STACK0-1
261      00213    3014      DCA      OSTACK      /RESET STACK POINTERS
262      00214    1066      TAD      STKLVL      /((CHANGED BY FOR LOOPS)
263      00215    3015      DCA      STACK
264      00216    1376      TAD      (LINE-1 /GET THE NEXT LINE
265      00217    3010      DCA      X10
266      00220    1375      TAD      (-LINMAX/MAX SIZE
267      00221    3021      DCA      TEMP
268      00222    4774      GETLIN, JMS      ICHAR      /GET NEXT CHAR
269      00223    5235      JMP      GOTCR      /CR
270      00224    3410      DCA I  X10      /PUT INTO LINE BUFFER
271      00225    2021      ISZ      TEMP      /BUMP MAX COUNTER
272      00226    5222      JMP      GETLIN
273      00227    4473      JMS I  QERMSG      /LINE TOO LONG
274      00230    1424      1424
275      00231    4774      JMS      ICHAR      /SKIP REST OF LINE
276      00232    5235      JMP      GOTCR
277      00233    7200      CLA
278      00234    5231      JMP      .-3
279      00235    1010    GOTCR, TAD      X10      /COMPUTE SIZE
280      00236    7040      CMA
281      00237    1376      TAD      (LINE-1 /OF LINE
282      00240    3020      DCA      NCHARS
283      00241    1376      TAD      (LINE-1 /SETUP LINE POINTER
284      00242    3017      DCA      CHRPTR
285      00243    1045      TAD      LOCTRL      /PUT LOCATION COUNTER
286      00244    7421      7421      /INTO MQ
287      00245    7330      CLA CLL CML RAR /ALLOW DEFINITION
288      00246    4477      JMS I  QSNUM      /GET THE STATEMENT NUMBER
289      00247    5267      JMP      NOSNUM      /NO STMT NUMBER ON THIS LINE
290      00250    2062      ISZ      NOSN      /SET STMT NUMBER PRESENT
291      00251    4476      JMS I  QMODSET      /IN N MODE AT ALL LABELS
292      00252    4532      JMS I  QNOREGS      /FORGET REG CONTENTS
293      00253    1134      TAD      WORD1      /SAVE NEW LINE NUMBER
294      00254    3053      DCA      LINEH
295      00255    1135      TAD      WORD2
296      00256    3054      DCA      LINEL
297      00257    4070      JMS      SETFLD      /GET TO FIELD OF ENTRY
298      00260    1422      TAD I  TEMP2      /GET DEFINED/REFNCED BITS
299      00261    1044      TAD      LOCTRH      /ADD IN HIGH ORDER LOCATION CTR
300      00262    3422      DCA I  TEMP2      /PUT IT AWAY
301      00263    2022      ISZ      TEMP2
302      00264    1045      TAD      LOCTRL      /NOW PUT IN LOW ORDER LOCATION
303      00265    3422      DCA I  TEMP2

```

```

304 00266 6201 CDF
305 00267 4773 NOSNUM, JMS KBOCHK /CHECK FOR "C OR "O
306 00270 1372 TAD (KEYWRD-1
307 00271 3010 DCA X10 /SET UP FOR KEYWORD SEARCH
308 00272 4501 JMS I QSAVECP /SAVE CHAR POS
309 00273 1410 KWLOOP, TAD I X10 /GET NEXT CHAR OF KEYWORD
310 00274 7500 SMA
311 00275 5314 JMP GOTKW /OK, THIS IS THE KW
312 00276 3021 DCA TEMP
313 00277 4502 JMS I QGETC /GET NEXT CHAR FROM STMT
314 00300 5304 JMP NOGOOD /THIS ISN'T IT
315 00301 1021 TAD TEMP /IS THIS CHAR OK ?
316 00302 7650 SNA CLA
317 00303 5273 JMP KWLOOP /YES, CONTINUE LOOKING
318 00304 4504 NOGOOD, JMS I QRESTCP /BACK TO START OF STMT
319 00305 1410 TAD I X10 /SKIP OVER REST OF KEYWORD
320 00306 7710 SPA CLA
321 00307 5305 JMP .-2
322 00310 1410 TAD I X10 /IS THIS END OF LIST ?
323 00311 7440 SZA
324 00312 5276 JMP KWLOOP+3 /NO, KEEP LOOKING
325 00313 5316 JMP LET /TREAT AS LET STMT
326 00314 3021 GOTKW, DCA TEMP /SAVE ADDR OF ROUTINE
327 00315 5421 JMP I TEMP /GO PROCESS THE STMT

```

```

328      / LET STATEMENT PROCESSOR
329      00316 4474 LET,      JMS I  QLODSN  /LOAD THE STMT NUMBER
330      00317 7130      CLL CML RAR  /COMPILE LEFT SIDE
331      00320 4505      JMS I  QEXPR  /GET EXPRESSION
332      00321 5205      JMP      REMARK
333      00322 4513      JMS I  QCHECKC /LOOK FOR =
334      00323 7703      -75
335      00324 5340      JMP      BADLET  /BAD IF MISSING
336      00325 4505      JMS I  QEXPR  /GET RIGHT SIDE
337      00326 5205      JMP      REMARK
338      00327 7240      CLA CMA      /GET TYPE OF
339      00330 1014      TAD      OSTACK /RIGHT SIDE
340      00331 3021      DCA      TEMP  /OF EQUAL SIGN
341      00332 1421      TAD I  TEMP  /SO THAT WE GENERATE
342      00333 7710      SPA CLA
343      00334 7144      CLL CMA RAL  /THE CORRECT STORE
344      00335 1371      TAD      (ASSIGN-1
345      00336 4506      JMS I  QOUTOPR /GENERATE STORE
346      00337 5201      JMP      NEWLIN
347      00340 4473 BADLET, JMS I  QERMSG  /BAD LET STMT
348      00341 1423      1423
349      00342 5205      JMP      REMARK
350      00343 1370 END,    TAD      (STOP /OUTPUT STOP OPCODE
351      00344 4500      JMS I  QOUTWRD
352      00345 4767'     JMS      QUDUMP  /DUMP BUFFER
353      00346 4766      JMS I  (7607  /READ IN POST PROCESSOR
354      00347 1000      1000      /EIGHT PAGES
355      00350 0400 POSTX, 400      /FROM 400
356      00351 0000 LDRBLK, 0      /FROM THIS BLOCK
357      00352 5406      JMP I  XABORT
358      00353 1473      TAD I  QERMSG  /SET POST PROCESSOR ERROR SWITCH
359      00354 3765'     DCA      ERMMSG2
360      00355 5750      JMP I  POSTX  /START IT UP
361      00356 0050 STAR,  5010;XMUL;XMUL
362      00357 0000
363      00360 5762
364      00361 5762
365      00362 0060 UPAROW, 6011;EXPRTN-1
366      00363 0001
367      00364 3543

```



```

368 / RESTORE, PRINT, AND INPUT PROCESSORS
369 00365 1712
370 00366 7607
371 00367 6471
372 00370 7441
373 00371 6066
374 00372 6242
375 00373 2346
376 00374 6600
377 00375 7660
378 00376 6777
379 00377 1133
380 00400 0400
381 00400 4474 INPUT, JMS I QLODSN /OUTPUT STMT NUM
382 00401 4777 JMS GETFN /LOOK FOR #<FILE NUM EXPR>:
383 00402 7130 INPUTL, CLL CML RAR /PROCESS INPUT STMT
384 00403 4505 JMS I QEXPR /GET EXPR
385 00404 5510 JMP I QREMARK
386 00405 4511 JMS I QGETA1 /GET TOP OF STACK
387 00406 1056 TAD TYPE1 /LOOK AT THE TYPE
388 00407 7710 SPA CLA
389 00410 5231 JMP RSTRNG /READ STRING
390 00411 4476 JMS I QMODSET /SET MODE
391 00412 1376 TAD (READ /OUTPUT READ COMMAND
392 00413 4500 JMS I QOUTWRD
393 00414 7132 CLL CML RTR /IS IT DIMENSIONED ?
394 00415 0056 AND TYPE1
395 00416 7650 SNA CLA
396 00417 5222 JMP ,+3 /NO
397 00420 4512 JMS I QLOADSS /SET UP SS REGS
398 00421 1375 TAD (AFSTA-FSTA
399 00422 1374 TAD (FSTA /USE SCALAR STORE
400 00423 1057 FININP, TAD SYMBL1 /PLUS SYMBOL NUMBER
401 00424 4500 JMS I QOUTWRD /OUTPUT INSTR
402 00425 4513 JMS I QCHECKC /LOOK FOR ,
403 00426 7724 -54
404 00427 5507 JMP I QNEWLIN /END OF INPUT
405 00430 5202 JMP INPUTL /YES, LOOP
406 00431 7130 RSTRNG, CLL CML RAR /SET MODE
407 00432 4476 JMS I QMODSET /TO STRING
408 00433 7132 CLL CML RTR /SUBSCRIPTED ?
409 00434 0056 AND TYPE1
410 00435 7650 SNA CLA
411 00436 5241 JMP ,+3 /NO
412 00437 4512 JMS I QLOADSS /LOAD SS REG
413 00440 1373 TAD (SAREAD-SREAD
414 00441 1372 TAD (SREAD /STRING READ
415 00442 5223 JMP FININP /USE SOME COMMON CODE
416 00443 4474 PRINT, JMS I QLODSN /OUTPUT STMT NUM
417 00444 4777 JMS GETFN /GET FILE NUMBER
418 00445 3505 DCA I QEXPR /USE ENTRY AS SWITCH
419 00446 3062 PRINTL, DCA PCRLF /CLEAR THE FLAG
420 00447 4502 JMS I QGETC /LOOK FOR A CHAR
421 00450 5334 JMP PRTEHD /NONE LEFT, END PRINT
422 00451 1371 TAD (-73 /?

```

```

423 00452 7450      SNA
424 00453 5261      JMP      NOCR      /YES, DON'T SPACE OUTPUT
425 00454 1370      TAD      (73-54  / , ?
426 00455 7640      SZA CLA
427 00456 5264      JMP      TABPNT /LOOK FOR TAB OR PNT
428 00457 1367      TAD      (FUNC3+20
429 00460 4500      JMS I    QOUTWRD /OUTPUT FUNC3+20 (COMMA)
430 00461 3505      NOCR,   DCA I    QEXPR  /CLEAR THE SWITCH
431 00462 7201      CLA IAC      /SET NO CRLF FLAG
432 00463 5246      JMP      PRINTL
433 00464 1505      TABPNT, TAD I    QEXPR  /WAS LAST THING AN EXPR ?
434 00465 7640      SZA CLA
435 00466 5507      JMP I      QNEWLIN /YES, CAN'T HAVE TWO IN A ROW
436 00467 4525      JMS I      QBACK1 /PUT THAT CHAR BACK
437 00470 4501      JMS I      QSAVECP /SAVE CHAR POS
438 00471 4475      JMS I      QCHKWD  /LOOK FOR "TAB("
439 00472 3132      WTAB
440 00473 5314      JMP      TRYPNT /NO TAB
441 00474 1366      TAD      (FUNC3+100
442 00475 3347      PFCALL, DCA      PRFUN  /SAVE PRINT FUNCTION
443 00476 4505      JMS I      QEXPR  /GET ARG
444 00477 5510      JMP I      QREMARK
445 00500 4520      JMS I      QLOAD  /LOAD ARG
446 00501 1056      TAD      TYPE1  /MUST BE NUMERIC
447 00502 7700      SMA CLA
448 00503 5307      JMP      .+4      /OK, IT IS
449 00504 4473      BADPF, JMS I      QERMSG  /PRINT ERROR
450 00505 0622      0622          /BAD FUNCTION REFERENCE
451 00506 5510      JMP I      QREMARK
452 00507 4513      JMS I      QCHECKC /LOOK FOR )
453 00510 7727      -51
454 00511 5304      JMP      BADPF  /BAD FUN REFERENCE
455 00512 1347      TAD      PRFUN  /OUTPUT FUNCTION CALL
456 00513 5332      JMP      PUT1
457 00514 4504      TRYPNT, JMS I      QRESTCP /RESTORE CHAR POS
458 00515 4475      JMS I      QCHKWD  /LOOK FOR PNT(
459 00516 5561      WPNT
460 00517 5322      JMP      PEXP   /NO
461 00520 1365      TAD      (FUNC3+120
462 00521 5275      JMP      PFCALL /GO DO FUN CALL
463 00522 4504      PEXP,   JMS I      QRESTCP /RESTORE CHAR POS
464 00523 4505      JMS I      QEXPR  /GET EXPR TO BE PRINTED
465 00524 5510      JMP I      QREMARK
466 00525 4520      JMS I      QLOAD  /PUT THING INTO FAC (OR SAC)
467 00526 7130      CLL CML  RAR
468 00527 0056      AND      TYPE1  /GET TYPE BIT
469 00530 7106      CLL RTL      /INTO AC 11
470 00531 1364      TAD      (WRITE  /SWRITE=WRITE+1
471 00532 4500      PUT1,   JMS I      QOUTWRD
472 00533 5246      JMP      PRINTL
473 00534 1062      PRTEnd, TAD      PCRLF  /DID PRINT END WITH
474 00535 7640      SZA CLA      / , OR ?
475 00536 5507      JMP I      QNEWLIN /YES, NO CR LF
476 00537 1363      TAD      (FUNC3+40
477 00540 4500      PUT2,   JMS I      QOUTWRD /CALL TO CRLF ROUTINE

```

478	00541	5507		JMP I	QNEWLIN	/END OF PRINT
479	00542	4474	RESTOR,	JMS I	QLODSN	/OUTPUT LOAD STMT NUMBER
480	00543	7201		CLA IAC		/NO COLON NEEDED
481	00544	4777		JMS	GETFN	/LOAD FILE REG
482	00545	1362		TAD	(REST	/OUTPUT RESTORE OP
483	00546	5340		JMP	PUT2	
484			PRFUN,			
485	00547	0000	LODSN,	0		/OUTPUT STMT NUMBER INTO CODE
486	00550	1062		TAD	NOSN	/ANY STMT NUMBER ?
487	00551	7650		SNA CLA		
488	00552	5747		JMP I	LODSN	/NO, JUST RETURN
489	00553	1134		TAD	WORD1	/NOW OUTPUT "LOAD STMT NUM REG"
490	00554	1361		TAD	(LOADSN	
491	00555	4500		JMS I	QOUTWRD	
492	00556	1135		TAD	WORD2	
493	00557	4500		JMS I	QOUTWRD	
494	00560	5747		JMP I	LODSN	

495 / DIM PROCESSOR

496 00561 6000
 497 00562 7405
 498 00563 7440
 499 00564 7412
 500 00565 7520
 501 00566 7500
 502 00567 7420
 503 00570 0017
 504 00571 7705
 505 00572 1000
 506 00573 5500
 507 00574 2400
 508 00575 4300
 509 00576 7411
 510 00577 2111

511 00600 0600

512 00600 4514

513 00601 5301

514 00602 1026

515 00603 7006

516 00604 7700

517 00605 7420

518 00606 5301

519 00607 4777

520 00610 1025

521 00611 3342

522 00612 4515

523 00613 5301

524 00614 5224

525 00615 4267

526 00616 5232

527 00617 7130

528 00620 3026

529 00621 4516

530 00622 6211

531 00623 5257

532 00624 4777

533 00625 4513

534 00626 7727

535 00627 5301

536 00630 4267

537 00631 1376

538 00632 1376

539 00633 3021

540 00634 4516

541 00635 6211

542 00636 1437

543 00637 0376

544 00640 7450

545 00641 5247

546 00642 7500

547 00643 1021

548 00644 7640

549 00645 5301

PAGE
 DIM, JMS I QGETNAM /GET VAR NAME
 JMP DIMERR
 TAD TYPE /CHECK TYPE
 RTL /MOVE BITS TO BE TESTED
 SMA CLA /IF FUNC BIT SET THEN ERROR
 SNL /IF DIM BIT NOT SET THEN ERROR
 JMP DIMERR /NO DIMENSIONS
 JMS SMLNUM /GET DIMENSION
 TAD EXPON /SAVE IT
 DCA DIM1
 JMS I QCOMARP /, OR) ??
 JMP DIMERR /NEITHER IS BAD
 JMP TWODIM /, THERE'S ANOTHER DIMENSION
 JMS CHKSOM /CHECK SIZE IF STRING
 JMP CHKDIM /NUMERIC VECTOR, CHECK PREV REF
 CLL CML RAR /THIS WAS A STRING SIZE DIM
 DCA TYPE /PERFORM THE SPECIAL CASE
 JMS I QLOOKUP
 CDF 10 /OF NOT CHECKING PREVIOUS REFS
 JMP FINDIM
 TWODIM, JMS SMLNUM /GET SECOND
 JMS I QCHECKC /LOOK FOR)
 -51
 JMP DIMERR
 JMS CHKSOM /CHECK SIZE IF STRING ARRAY
 TAD (7000 /NUMERIC ARRAY
 CHKDIM, TAD (7000 /GET NUMBER OF DIMS
 DCA TEMP
 JMS I QLOOKUP /FIND ST ENTRY
 CDF 10
 TAD I STPTR /LOOK AT DIM BITS
 AND (7000 /PREVIOUSLY REFERENCED ?
 SNA
 JMP UNREFD /NO
 SMA /IF MINUS, CAUSE ERROR
 TAD TEMP /COMPARE NUMBER
 SZA CLA
 JMP DIMERR /NUMBER OF DIMS DON'T MATCH

```

550 00646 3021      DCA      TEMP      /ZERO TEMP
551 00647 7130 UNREFD, CLL CML RAR      /PUT IN DIMENSIONED BIT
552 00650 1021      TAD      TEMP      /AND NUMBER OF DIMENSIONS
553 00651 7041      CIA      /NEGATE WHOLE MESS (4000=-4000)
554 00652 1437      TAD I   STPTR      /TOGETHER WITH SYM NUMBER
555 00653 3437      DCA I   STPTR
556 00654 2037      ISZ      STPTR
557 00655 1342      TAD      DIM1      /NOW FIRST DIMENSION (IF 2)
558 00656 3437      DCA I   STPTR
559 00657 2037 FINDIM, ISZ      STPTR
560 00660 1025      TAD      EXPON      /NOW SECOND (IF 2, OTHERWISE FIRST)
561 00661 3437      DCA I   STPTR
562 00662 6201      CDF
563 00663 4513      JMS I   QCHECKC /LOOK FOR ,
564 00664 7724      -54
565 00665 5507      JMP I   QNEWLIN /NONE, ASSUME END OF DIM
566 00666 5200      JMP      DIM      /GET NEXT ELEMENT
567 00667 0000 CHKSDM, 0      /CHECK SIZE OF STRINGS
568 00670 1026      TAD      TYPE      /WAS THIS A STRING DIM ?
569 00671 7700      SMA CLA
570 00672 5667      JMP I   CHKSDM /NO, RETURN IMMEDIATE
571 00673 2267      ISZ      CHKSDM /YES, SKIP ON RETURN
572 00674 1025      TAD      EXPON      /SIZE MUST BE < 73
573 00675 7100      CLL
574 00676 1375      TAD      (-111
575 00677 7620      SNL CLA
576 00700 5667      JMP I   CHKSDM /OK, SIZE < 73
577 00701 4473 DIMERR, JMS I   QERMSG /GIVE ERROR
578 00702 0411      0411
579 00703 5510      JMP I   QREMARK /ABORT STMT

```

```

580      / NEXT PROCESSOR
581      00704 4514 NEXTX, JMS I QGETNAM /GET INDEX VARIABLE
582      00705 5362      JMP      BADNXT
583      00706 4516      JMS I   QLOOKUP
584      00707 1026      TAD      TYPE      /MUST BE NUMERIC
585      00710 7710      SPA CLA
586      00711 5362      JMP      BADNXT /IT ISN'T
587      00712 4476      JMS I   QMODSET /N MODE
588      00713 1374 NEXTL, TAD      (-STACKA-3
589      00714 1015      TAD      STACK /ANY FOR'S LEFT ?
590      00715 7710      SPA CLA /((OK IF STACKA ABOVE 4000)
591      00716 5362      JMP      BADNXT /NO
592      00717 4522      JMS I   QPOP      /GET LABEL ADDR
593      00720 3021      DCA      TEMP
594      00721 4522      JMS I   QPOP      /GET LABEL FIELD
595      00722 3342      DCA      LUPFLD
596      00723 4522      JMS I   QPOP      /GET STEP VAR
597      00724 1370      TAD      XLOAD /LOAD IT
598      00725 4500      JMS I   QOUTWRD
599      00726 4522      JMS I   QPOP      /GET INDEX VAR
600      00727 3067      DCA      FINDEX
601      00730 1067      TAD      FINDEX /ADD IT TO STEP (FADD=0)
602      00731 4500      JMS I   QOUTWRD
603      00732 1342      TAD      LUPFLD /CREATE JUMP TO LOOP
604      00733 0373      AND      (70
605      00734 7106      CLL RTL
606      00735 1372      TAD      (JUMP
607      00736 4500      JMS I   QOUTWRD
608      00737 7144      CLL CMA RAL /GET LABEL DEFINITION ADDR
609      00740 1021      TAD      TEMP
610      00741 4500      JMS I   QOUTWRD /OUTPUT IT AS LOW PART OF JUMP
611
612      00742 7402 DIM1, LUPFLO, HLT
613      00743 7130      CLL CML RAR /SET LABEL DEFINED BIT
614      00744 1044      TAD      LOCTRH /DEFINE END OF LOOP LABEL
615      00745 3421      DCA I   TEMP
616      00746 2021      ISZ     TEMP
617      00747 1045      TAD      LOCTRL
618      00750 3421      DCA I   TEMP
619      00751 6201      CDF
620      00752 1015      TAD      STACK /BACK OFF STACK LEVEL
621      00753 3066      DCA      STKLVL
622      00754 4532      JMS I   QNOREGS /FORGET REGS
623      00755 1027      TAD      SYMBOL /IS THIS THE RIGHT NEXT ?
624      00756 7041      CIA
625      00757 1067      TAD      FINDEX
626      00760 7650      SNA CLA
627      00761 5507      JMP I   QNEWLIN /YES, FINISHED
628      00762 4473 BADNXT, JMS I   QERMSG /NEXT WITHOUT FOR
629      00763 1606      1606
630      00764 5510      JMP I   QREMARK
631      00765 0040 UMOPR, 4011;UMRTNE=1
632      00766 0001
633      00767 3534
634      00770 2000 XLOAD, FLDA;AFLDA

```

/OS/8 BASIC COMPILER

PAL8-V8 10/30/72 PAGE 9-1

635 00771 6640

```

636 / UDEF PROCESSOR (DEFINE USER FUNCTION)
637 00772 5001
638 00773 0070
639 00774 0655
640 00775 7667
641 00776 7000
642 00777 5400
643 1000
644 01000 2333 UDEF, PAGE
645 01001 4530 ISZ NFUNS /ROOM FOR ANOTHER FUN ?
646 01002 5251 JMS I QLETTER /GET FIRST LETTER
647 01003 7106 JMP DEFBAD /ERROR IN DEFINE
648 01004 7006 CLL RTL /PUT INTO HIGH ORDER
649 01005 7006 RTL
650 01006 3134 DCA NAME1 /SAVE CHAR 1
651 01007 4530 JMS I QLETTER /GET SECOND LETTER
652 01010 5251 JMP DEFBAD /ERROR
653 01011 1134 TAD NAME1 /COMBINE THE TWO CHARS
654 01012 7041 CIA
655 01013 3727 DCA I FUNPTR /SAVE IN FUN TABLE
656 01014 2327 ISZ FUNPTR
657 01015 4530 JMS I QLETTER /GET THIRD LETTER
658 01016 5251 JMP DEFBAD
659 01017 7041 CIA /SAVE NEG OF THIRD LETTER
660 01020 3727 DCA I FUNPTR
661 01021 2327 ISZ FUNPTR /BUMP POINTER
662 01022 1331 TAD M5 /NUMERIC ARG COUNT
663 01023 3021 DCA TEMP / (MAX OF 4 ARGS)
664 01024 7146 CLL CMA RTL /STRING ARG COUNT
665 01025 3022 DCA TEMP2 / (MAX OF 2 ARGS)
666 01026 4513 JMS I QCHECKC /IS IT A STRING FUN ?
667 01027 7734 -44
668 01030 7610 SKP CLA
669 01031 7130 CLL CML RAR /YES, SET TYPE OF FUNCTION
670 01032 3056 DCA TYPE1
671 01033 4513 JMS I QCHECKC /LOOK FOR (
672 01034 7730 -50
673 01035 5251 JMP DEFBAD /ERROR IF MISSING
674 01036 4514 DALOOP, JMS I QGETNAM /GET AN ARG
675 01037 5251 JMP DEFBAD
676 01040 1026 TAD TYPE /LOOK AT ITS TYPE
677 01041 7104 CLL RAL /SHIFT TYPE BIT INTO LINK
678 01042 7640 SZA CLA
679 01043 5251 JMP DEFBAD /OTHER BITS MUST BE OFF
680 01044 7430 SZL
681 01045 5254 JMP STRARG /STRING ARG
682 01046 1021 TAD TEMP /GET ARG NUMBER
683 01047 2021 ISZ TEMP /INCREMENT IT
684 01050 5260 JMP DAPUSH /GO SAVE IT
685 01051 4473 DEFBAD, JMS I QERMSG /BAD USER DEF
686 01052 2504 2504
687 01053 5510 JMP I GREMARK
688 01054 1022 STRARG, TAD TEMP2 /GET ARG NUMBER
689 01055 2022 ISZ TEMP2 /AND INCREMENT IT
690 01056 5261 JMP DAPUSH+1

```



```

691 01057 5251      JMP      DEFBAD  /TOO MANY STRING ARGS
692 01060 1330 DAPUSH, TAD      Q2      /ADJUST ARG NUMBER
693 01061 1330      TAD      Q2      /ADD 4 FOR NUM, 2 FOR STRING
694 01062 7510      SPA
695 01063 7332      CLA CLL CML RTR /FIRST ARG STAYS IN AC
696 01064 1026      TAD      TYPE    /ADD IN TYPE BIT
697 01065 4521      JMS I    QPUSH   /SAVE IT ON STACK
698 01066 4515      JMS I    QCOMARP /LOOK FOR , OR )
699 01067 5251      JMP      DEFBAD  /ERROR IF NEITHER
700 01070 5236      JMP      QALoop   /, GET NEXT ARG
701 01071 1022      TAD      TEMP2   /GET TOTAL NUMBER OF ARGS
702 01072 1021      TAD      TEMP
703 01073 1332      TAD      Q10     /ADJUST COUNT
704 01074 7041      CIA          /NEGATED
705 01075 3062      DCA          DACNT
706 01076 1727      TAD I    FUNPTR  /GET FUNCTION CODE
707 01077 2327      ISZ      FUNPTR  /BUMP POINTER
708 01100 3134      DCA      WORD1   /MAKE IT THE SEARCH OBJECT
709 01101 4726      JMS I    XSTCHEK /MAKE SURE THERE'S ROOM
710 01102 7560      EOST-10
711 01103 4517      JMS I    QLUKUP2 /ENTER NEW FUNCTION
712 01104 2570      FUNCTN
713 01105 7777      -1
714 01106 1062      TAD      DACNT   /PUT IN ARG COUNT
715 01107 4070      JMS      SETFLD  /(FIRST SET THE FIELD)
716 01110 3416      DCA I    NEXT
717 01111 6201 DAPUT, CDF
718 01112 4522      JMS I    QPOP    /GET ARG TYPE (LAST TO FIRST)
719 01113 4070      JMS      SETFLD  /SET THE FIELD
720 01114 3416      DCA I    NEXT    /SAVE IT
721 01115 2062      ISZ      DACNT   /ANY MORE ?
722 01116 5311      JMP      DAPUT   /YES
723 01117 1056      TAD      TYPE1   /PUT IN TYPE OF FUNCTION
724 01120 3416      DCA I    NEXT
725 01121 6201      CDF
726 01122 4513      JMS I    QCHECKC /LOOK FOR A COMMA
727 01123 7724      -54
728 01124 5507      JMP I    QNEWLIN /NO COMMA, END OF LINE
729 01125 5200      JMP      UDEF     /GET NEXT DEFINITION
730 01126 2317 XSTCHEK, STCHEK
731 01127 6162 FUNPTR, ENDFNS
732 01130 0002 Q2, 2 /THESE FOUR WORDS
733 01131 7773 M5, -5 /PREVENT ERRONEOUS "SAVES"
734 01132 0010 Q10, 10 /BY THE ROUTINE SAVAC
735 01133 7757 NFUNS, -21 /WHEN THE OP STACK IS EMPTY
736 STACKO, /OPERAND STACK
737 0044 STOKSZ=UDEF+200=STACKO

```

```

738      / DEF PROCESSOR
739      PAGE
740      01200 4532 DEF, JMS I QNOREGS /FORGET REGS
741      01201 4514 JMS I QGETNAM /GET FUN NAME
742      01202 5253 JMP BADDEF /NO GOOD
743      01203 1026 TAD TYPE /SAVE ITS TYPE
744      01204 3022 DCA TEMP2
745      01205 3061 DCA ARGCNT /ZERO ARG COUNT
746      01206 1026 TAD TYPE /TYPE MUST BE 3000 OR 7000
747      01207 7006 RTL /MOVE BITS TO BE TESTED
748      01210 7710 SPA CLA /FUN BIT OFF IS AN ERROR
749      01211 7420 SNL /DIM BIT OFF IS AN ERROR
750      01212 5253 JMP BADDEF
751      01213 4476 JMS I QMODSET /ENTER N MODE
752      01214 1027 TAD SYMBOL /SAVE FUNCTION NAME
753      01215 3246 DCA FUNNAM
754      01216 4514 ARG LUP, JMS I QGETNAM /GET ARG NAME
755      01217 5253 JMP BADDEF
756      01220 7150 CLL CMA RAR /LOOK AT TYPE
757      01221 0026 AND TYPE
758      01222 7640 SZA CLA
759      01223 5253 JMP BADDEF /ARG WAS AN ARRAY OR FUNC
760      01224 4516 JMS I QLOOKUP /ENTER INTO S.T.
761      01225 1037 TAD STPTR /SAVE ST ADDRESS
762      01226 4521 JMS I QPUSH
763      01227 1027 TAD SYMBOL /AND SYMBOL NUMBER
764      01230 4521 JMS I QPUSH
765      01231 1026 TAD TYPE /AND ARG TYPE
766      01232 4521 JMS I QPUSH
767      01233 2061 ISZ ARGCNT /BUMP ARG COUNT
768      01234 4515 JMS I QCOMARP /LOOK FOR , OR )
769      01235 5253 JMP BADDEF
770      01236 5216 JMP ARG LUP /, GET NEXT ARG
771      01237 1246 TAD FUNNAM /ENTER FUNCTION
772      01240 3134 DCA WORD1
773      01241 1061 TAD ARGCNT /FIRST GET ENOUGH ROOM
774      01242 7041 CIA
775      01243 1377 TAD (EOST-3
776      01244 3246 DCA FUNNAM
777      01245 4776 JMS STCHK /CHECK IT
778      01246 0000 FUNNAM, 0
779      01247 4517 JMS I QLOOKUP2 /LOOK UP FUNCTION
780      01250 2570 FUNCTN
781      01251 7777 -1
782      01252 5256 JMP OKFUN /OK, NOT MULTIPLY DEFINED
783      01253 4473 BADDEF, JMS I QERMSG /BAD DEFINE
784      01254 0405 0405
785      01255 5510 JMP I QREMARK
786      01256 1016 OKFUN, TAD NEXT /SAVE "NEXT"
787      01257 3012 DCA X12
788      01260 1016 TAD NEXT /INCREMENT NEXT BY
789      01261 1061 TAD ARGCNT /NUMBER OF ARGS
790      01262 1375 TAD (4 /PLUS 4
791      01263 3016 DCA NEXT
792      01264 4070 JMS SETFLD /GET ROOM FOR LABEL

```

793	01265	7130	CLL CML RAR	/FOR JUMP AROUND
794	01266	3416	DCA I NEXT	/SET DEFINED BIT
795	01267	1016	TAD NEXT	/SAVE ADDR
796	01270	3062	DCA JAROND	/FOR LATER
797	01271	2016	ISZ NEXT	
798	01272	6201	COF	
799	01273	1071	TAD LUFLD	/SAVE FIELD OF FUN BLOCK
800	01274	3336	DCA FUNFLD	
801	01275	1071	TAD LUFLD	/ALSO FIELD OF LABEL
802	01276	3774	DCA JARFLD	
803	01277	1071	TAD LUFLD	/GET FIELD
804	01300	0373	AND (70	/ISOLATE BITS
805	01301	7106	CLL RTL	/INTO JUMP INSTR
806	01302	1372	TAD (JUMP	
807	01303	4500	JMS I QOUTWRD	/OUTPUT IT
808	01304	1062	TAD JAROND	/OUTPUT LOW PART
809	01305	4500	JMS I QOUTWRD	/OF JUMP ADDR
810	01306	1015	TAD STACK	/SAVE STACK
811	01307	3060	DCA OLDSTK	
812	01310	1061	TAD ARGCNT	/GET COUNT
813	01311	7040	CMA	
814	01312	3021	DCA TEMP	
815	01313	1061	TAD ARGCNT	/TWICE
816	01314	7041	CIA	
817	01315	3061	DCA ARGCNT	
818	01316	1061	TAD ARGCNT	/STORE COUNT FIRST
819	01317	5336	JMP FUNFLD	
820	01320	6201	CHGARG, COF	
821	01321	4522	JMS I QPOP	/GET ARG TYPE
822	01322	3026	DCA TYPE	
823	01323	1026	TAD TYPE	
824	01324	4771	JMS GENTMP	/GENERATE A TEMPORARY
825	01325	4522	SWTARG, JMS I QPOP	/PURGE SYMBOL NUMBER
826	01326	7200	CLA	
827	01327	4522	JMS I QPOP	/GET ST ADDR OF
828	01330	3037	DCA STPTR	/OF DUMMY ARG
829	01331	6211	COF 10	
830	01332	1027	TAD SYMBOL	/PUT IN TEMP SYMBOL NUMBER
831	01333	3437	DCA I STPTR	/TO FAKE EXPR
832	01334	1026	TAD TYPE	/CREATE ARG DESCRIPTOR
833	01335	1027	TAD SYMBOL	/FOR FUNC BLOCK
834	01336	7402	FUNFLD, HLT	
835	01337	3412	DCA I X12	/AND PUT IT INTO F.B.
836	01340	2021	ISZ TEMP	/MORE ARGS?
837	01341	5320	JMP CHGARG	/YUP
838	01342	7130	CLL CML RAR	
839	01343	0022	AND TEMP2	/SAVE TYPE OF FUNCTION
840	01344	3412	DCA I X12	
841	01345	7130	CLL CML RAR	/SET DEFINED BIT
842	01346	1044	TAD LOCTRH	/AND LOCATION COUNTER
843	01347	3412	DCA I X12	/AT START OF FUNCTION
844	01350	1045	TAD LOCTRL	
845	01351	3412	DCA I X12	
846	01352	6201	COF	
847	01353	1015	TAD STACK	/SAVE BOTTOM OF STACK

848	01354	3013	DCA	X13	
849	01355	1060	TAD	OLDSTK	/RESTORE TO TOP
850	01356	3015	DCA	STACK	
851	01357	4513	JMS I	QCHECKC	/FIND =
852	01360	7703	-75		
853	01361	5253	JMP	BADDEF	
854	01362	4505	JMS I	QEXPR	/COMPILE FUNCTION
855	01363	5510	JMP I	QREMARK	
856	01364	4520	JMS I	QLOAD	/GET IT INTO AC
857	01365	1013	TAD	X13	/RESTORE STACK
858	01366	3015	DCA	STACK	/TO BOTTOM
859	01367	5770	JMP	RESARG	/FINISH DEF

```

860      / DEF PROCESSOR (FINALE)
861      01370 1400
862      01371 4200
863      01372 5001
864      01373 0070
865      01374 1413
866      01375 0004
867      01376 2317
868      01377 7565
869      1400
870      01400 1413  RESARG, TAD I  X13  /GET ST ADDR
871      01401 3037      DCA  STPTR
872      01402 1413      TAD I  X13  /PUT BACK CORRECT SYM #
873      01403 6211      CDF  10
874      01404 3437      DCA I  STPTR
875      01405 6201      CDF
876      01406 2013      ISZ  X13  /SKIP OTHER STUFF
877      01407 2061      ISZ  ARG CNT
878      01410 5200      JMP  RESARG /RESTORE NEXT
879      01411 1377      TAD  (RET  /OUTPUT RETURN CODE
880      01412 4500      JMS I  QOUTWRD
881      01413 7402  JARFLD, HLT
882      01414 7130      CLL CML RAR  /SET LABEL DEFINED BIT
883      01415 1044      TAD  LOCTRH /STICK IN ADDR
884      01416 3462      DCA I  JAROND /OF END OF FUNCT
885      01417 2062      ISZ  JAROND /PLUS ONE
886      01420 1045      TAD  LOCTRL /STORE LOW ADDR
887      01421 3462      DCA I  JAROND
888      01422 6201      CDF
889      01423 1033      TAD  TMCNT  /SAVE NEW TEMP LEVELS
890      01424 3034      DCA  TMPLVL
891      01425 1035      TAD  STMPCT
892      01426 3036      DCA  STMPLV
893      01427 4532      JMS I  QNOREGS /FORGET REGS
894      01430 5507      JMP I  QNEWLIN /END OF DEF

```

```

895 / DATA STATEMENT PROCESSOR
896 01431 4526 DATA, JMS I QNUMBER /LOOK FOR NUMBER
897 01432 5241 JMP DSTRNG /MUST BE A STRING
898 01433 4255 JMS DENTRY /MAKE AN ENTRY
899 01434 7775 -3 /3 WORDS LONG
900 01435 4513 MORDAT, JMS I QCHECKC /LOOK FOR ,
901 01436 7724 -54
902 01437 5507 JMP I QNEWLIN /END OF DATA
903 01440 5231 JMP DATA /DO NEXT ELEMENT
904 01441 4527 DSTRNG, JMS I QSTRING /LOOK FOR STRING
905 01442 5507 JMP I QNEWLIN /BAD
906 01443 1134 TAD WORD1 /COMPUTE SIZE
907 01444 7001 IAC
908 01445 7170 CLL CML CMA RAR
909 01446 3253 DCA DSSIZE /INCLUDING CHAR COUNT
910 01447 1134 TAD WORD1 /NEGATE COUNT
911 01450 7041 CIA
912 01451 3134 DCA WORD1
913 01452 4255 JMS DENTRY /CREATE ENTRY
914 01453 0000 DSSIZE, 0
915 01454 5235 JMP MORDAT /GO DO MORE
916 01455 0000 DENTRY, 0 /MAKE AN ENTRY IN DATA LIST
917 01456 1655 TAD I DENTRY /GET SIZE
918 01457 3021 DCA TEMP
919 01460 2255 ISZ DENTRY
920 01461 1021 TAD TEMP /INCREMENT SIZE COUNT
921 01462 1051 TAD DLSIZE
922 01463 3051 DCA DLSIZE
923 01464 1376 TAD (EOST /HOW MUCH DO WE NEED ?
924 01465 1021 TAD TEMP
925 01466 3270 DCA .+2
926 01467 4775 JMS STCHEK /ASK FOR IT
927 01470 0000 0
928 01471 1774 TAD FREFLD /GET FIELD OF FREE SPACE
929 01472 3071 DCA LUFLD /SAVE IT IN SETFLD SUBROUTINE
930 01473 6211 DATFLD, CDF 10
931 01474 1016 TAD NEXT /HOOK IN NEW ENTRY
932 01475 7001 IAC
933 01476 3725 DCA I DATPTR
934 01477 2325 PATCH3, 187 DATPTR /POINTER THEN FIELD
935 01500 1071 TAD LUFLD
936 01501 3725 DCA I DATPTR
937 01502 4070 JMS SETFLD
938 01503 1021 TAD TEMP /SAVE SIZE OF ENTRY
939 01504 3416 DCA I NEXT
940 01505 1373 TAD (WORD1-1/MAKE READY TO MOVE
941 01506 3010 DCA X10
942 01507 6201 DELOOP, CDF
943 01510 1410 TAD I X10 /GET WORD
944 01511 4070 JMS SETFLD
945 01512 3416 DCA I NEXT /SAVE IT
946 01513 2021 ISZ TEMP /MORE ?
947 01514 5307 JMP DELOOP
948 01515 3416 DCA I NEXT /SAVE ROOM FOR POINTER&CDF
949 01516 1016 TAD NEXT /THIS IS NOW LAST ENTRY

```

JMP .+4 8K only

950 01517 3325 DCA DATPTR
951 01520 1071 PATCH4, TAD ~~LUPLO~~ *8K; VAP .+3*
952 01521 3273 DCA DATFLD /AND THIS IS ITS FIELD
953 01522 3416 DCA I NEXT
954 01523 6201 ~~CDF~~
955 01524 5655 JMP I DENTRY
956 01525 2566 DATPTR, DATLST

```
957 / READ PROCESSOR
958 01526 4474 READX, JMS I QLODSN /OUTPUT STMT NUMBER
959 01527 7130 CLL CML RAR /GET VAR TO READ
960 01530 4505 JMS I QEXPR /SAME AS LEFT SIDE OF LET
961 01531 5510 JMP I QREMARK
962 01532 4511 JMS I QGETA1 /GET VAR INFO FROM STACK
963 01533 1056 TAD TYPE1 /SET MODE
964 01534 4476 JMS I QMODSET
965 01535 1056 TAD TYPE1 /WHAT TYPE ?
966 01536 7710 SPA CLA
967 01537 1372 TAD (SRDL-NRDL
968 01540 1371 TAD (NRDL /STRING OR NUMERIC
969 01541 4500 JMS I QOUTWRD
970 01542 7132 CLL CML RTR /SUBSCRIPTS ?
971 01543 0056 AND TYPE1
972 01544 7650 SNA CLA
973 01545 5350 JMP .+3 /NO
974 01546 4512 JMS I QLOADSS /YES, LOAD SS REGS
975 01547 1370 TAD (AFSTA-FSTA
976 01550 1367 TAD (FSTA /ARRAY OR SCALAR STORE
977 01551 1057 TAD SYMBL1
978 01552 4500 JMS I QOUTWRD
979 01553 4513 JMS I QCHECKC /ANY MORE ?
980 01554 7724 -54 /CHECK FOR COMMA
981 01555 5507 JMP I QNEWLIN /NO
982 01556 5327 JMP READX+1 /YUP
983 01557 0040 AMPSND, 40;1;AMPRTN=1;4000;XADD;XADD
984 01560 0001
985 01561 6506
986 01562 4000
987 01563 5162
988 01564 5162
```



```

989      / FOR PROCESSOR
990      01567 2400
991      01570 4300
992      01571 7521
993      01572 7740
994      01573 0133
995      01574 2266
996      01575 2317
997      01576 7570
998      01577 7404
999      1600
1000     01600 4474  FOR,    PAGE
1001     01601 4514      JMS I  QLODSN /OUTPUT STMT NUMBER
1002     01602 5777'    JMS I  QGETNAM /GET INDEX VARIABLE
1003     01603 1026      JMP    BADFOR /BAD
1004     01604 7640      TAD     TYPE  /MUST BE NUMBER
1005     01605 5777'    SZA CLA
1006     01606 4516      JMP    BADFOR /ITS NOT
1007     01607 1027      JMS I  QLOOKUP /ST SEARCH
1008     01610 3067      TAD     SYMBOL /SAVE INDEX VAR
1009     01611 4513      DCA     FINDEX /FOR LATER
1010     01612 7703      JMS I  QCHECKC /FIND =
1011     01613 5777'    -75
1012     01614 1017      JMP    BADFOR
1013     01615 3336      TAD     CHRPTR /SAVE CHAR POSITION
1014     01616 1020      DCA     FORCP  /IN A SPECIAL PLACE
1015     01617 3335      TAD     NCHARS
1016     01620 7410      DCA     FORNC
1017     01621 4504      SKP
1018     01622 4502      FINDTO, JMS I  QRESTCP /RESTORE CHAR POS
1019     01623 5777'    JMS I  QGETC  /SKIP A CHAR
1020     01624 7200      JMP    BADFOR
1021     01625 4501      CLA
1022     01626 4475      JMS I  QSAVECP /SAVE THIS POSITION
1023     01627 6273      JMS I  QCHKWD /LOOK FOR "TO"
1024     01630 5221      WTD
1025     01631 4776'    JMP    FINDTO /KEEP GOING
1026     01632 3333      JMS    FSUB2  /LOAD LIMIT AND SAVE IN TEMP
1027     01633 4475      DCA     FLIMIT /SAVE LIMIT VAR
1028     01634 5156      JMS I  QCHKWD /LOOK FOR "STEP"
1029     01635 5327      WSTEP
1030     01636 4776'    JMP    STEP1  /USE 1.0 FOR THE STEP
1031     01637 3334      JMS    FSUB2  /LOAD STEP AND SAVE IN TEMP
1032     01640 1375      DCA     FSTEP  /SAVE STEP VAR
1033     01641 4500      TAD     (SETJF /OUTPUT SETJF
1034     01642 1374      JMS I  QOUTWRD
1035     01643 3062      TAD     (JFOR  /STEP IS VARIABLE, USE JFOR
1036     01644 4502      DCA     FORJMP /SAVE CORRECT JUMP
1037     01645 7410      JMS I  QGETC  /ANY MORE CHARS ?
1038     01646 5777'    SKP
1039     01647 1335      JMP    BADFOR /YES, ERROR
1040     01650 3020      TAD     FORNC  /RESTORE CHAR POSITION
1041     01651 1336      DCA     NCHARS /FROM SPECIAL PLACE
1042     01652 3017      TAD     FORCP
1043     01653 4773'    DCA     CHRPTR
                        JMS    FSUB1  /COMPILE INITIAL VALUE INTO FAC

```

```

1044 01654 4772' JMS STCHK /CHECK FOR ROOM
1045 01655 7570 EOST
1046 01656 1771' TAD FREFLD /SAVE FIELD OF LABELS
1047 01657 3260 DCA FORFLD
1048 01660 7402 FORFLD, HLT
1049 01661 7130 CLL CML RAR /SET LABEL DEFINED BIT
1050 01662 1044 TAD LOCTRH /DEFINE THE LOOP LABEL
1051 01663 3416 DCA I NEXT
1052 01664 1045 TAD LOCTRL
1053 01665 3416 DCA I NEXT
1054 01666 7130 CLL CML RAR /SET LABEL DEFINED BIT
1055 01667 3416 DCA I NEXT /FOR END OF LOOP LABEL
1056 01670 6201 CDF
1057 01671 1333 TAD FLIMIT /TEST FOR DONE
1058 01672 1343 TAD XSUB /BY SUBTRACTING THE LIMIT
1059 01673 4500 JMS I QOUTWRD
1060 01674 1260 TAD FORFLD /OUTPUT JUMP TO DONE
1061 01675 0370 AND (70
1062 01676 7106 CLL RTL /SHIFT FIELD BITS
1063 01677 1062 TAD FORJMP /USE PROPER JUMP INS
1064 01700 4500 JMS I QOUTWRD
1065 01701 1016 TAD NEXT /OUTPUT LOW PART OF JMP
1066 01702 4500 JMS I QOUTWRD
1067 01703 1333 TAD FLIMIT /FADD FLIMIT (FADD=0)
1068 01704 4500 JMS I QOUTWRD
1069 01705 1067 TAD FINDEX /FSTA INDEX
1070 01706 1367 TAD (FSTA
1071 01707 4500 JMS I QOUTWRD
1072 01710 1067 TAD FINDEX /PUT STUFF ONTO STACK
1073 01711 4521 JMS I QPUSH
1074 01712 1334 TAD FSTEP
1075 01713 4521 JMS I QPUSH
1076 01714 1260 TAD FORFLD
1077 01715 4521 JMS I QPUSH
1078 01716 1016 TAD NEXT
1079 01717 4521 JMS I QPUSH
1080 01720 2016 ISZ NEXT /BUMP NEXT AGAIN
1081 01721 1033 TAD TMPCNT /RESERVE THESE TEMPS
1082 01722 3034 DCA TMLVL
1083 01723 4532 JMS I QNOREGS /FORGET REGISTERS
1084 01724 1015 TAD STACK /SET NEW STACK LEVEL
1085 01725 3066 DCA STKLVL
1086 01726 5510 JMP I QREMARK
1087 01727 1366 STEP1, TAD (3 /1.0 IS SLOT #3
1088 01730 3334 DCA FSTEP
1089 01731 1365 TAD (JGT /USE JGT
1090 01732 5243 JMP SAVEJF /GO DO THE REST
1091 01733 0000 FLIMIT, 0 /FOR LOOP UPPER LIMIT
1092 01734 0000 FSTEP, 0 /FOR LOOP STEP
1093 01735 0000 FORNC, 0 /FOR STMT CHAR POSITION
1094 01736 0000 FORCP, 0
1095 01737 7654 WTHEN, -124;-110;-105;-116
1096 01740 7670
1097 01741 7673
1098 01742 7662

```

/OS/8 BASIC COMPILER

PAL8-V8 10/30/72 PAGE 15-2

1099 01743 0400 XSUB, FSUB;AFSUB
1100 01744 6500

```
1101      / USE PROCESSOR
1102 01745 1364 USEX, TAD (USE /OUTPUT USE OPERATOR
1103 01746 4500 JMS I QOUTWRD
1104 01747 4514 JMS I QGETNAM /GET ARRAY NAME
1105 01750 5354 JMP USEERR /ERROR
1106 01751 1026 TAD TYPE /CHECK TYPE
1107 01752 7700 SMA CLA / (MUST BE NUMERIC)
1108 01753 5356 JMP .+3 /IT WAS
1109 01754 4473 USEERR, JMS I QERMSG /ERROR IN USE STMT
1110 01755 2525 2525
1111 01756 7132 CLL CML RTR /SET DIM BIT
1112 01757 3026 DCA TYPE
1113 01760 4516 JMS I QLOOKUP /LOOKUP SYMBOL
1114 01761 1027 TAD SYMBOL /OUTPUT ARRAY NUMBER
1115 01762 4500 JMS I QOUTWRD
1116 01763 5510 JMP I QREMARK
```

```

1117      / IF AND IFEND PROCESSORS
1118      01764 7540
1119      01765 5004
1120      01766 0003
1121      01767 2400
1122      01770 0070
1123      01771 2266
1124      01772 2317
1125      01773 3314
1126      01774 5010
1127      01775 7401
1128      01776 3326
1129      01777 3323
1130      2000
1131      02000 4474 IF, PAGE JMS I QLODSN /OUTPUT STMT NUMBER
1132      02001 4505 JMS I QEXPR /GET LEFT EXPRESSION
1133      02002 5510 JMP I QREMARK
1134      02003 4502 JMS I QGETC /GET RELATIONAL OPERATOR
1135      02004 5262 JMP BADIF /ERROR IF NONE
1136      02005 7106 CLL RTL
1137      02006 7006 RTL /MOVE TO LEFT HALF
1138      02007 7006 RTL
1139      02010 3021 DCA TEMP /AND SAVE IT
1140      02011 4502 JMS I QGETC /GET 2 CHAR RELATIONALS
1141      02012 5262 JMP BADIF
1142      02013 1021 TAD TEMP /COMBINE THE 2
1143      02014 3022 DCA TEMP2
1144      02015 1377 TAD (IFOPS-1/SETUP POINTER
1145      02016 3010 DCA X10
1146      02017 1410 IFLUP1, TAD I X10 /GET JUMP OPCODE
1147      02020 7450 SNA
1148      02021 5271 JMP IFLUP2-1/NOT A 2 CHAR RELATIONAL
1149      02022 3311 DCA RELOPR /SAVE IT
1150      02023 1410 TAD I X10 /COMPARE CHARS
1151      02024 1022 TAD TEMP2
1152      02025 7640 SZA CLA
1153      02026 5217 JMP IFLUP1 /NOT THIS OOE
1154      02027 4505 GOTREL, JMS I QEXPR /GET RIGHT HALF
1155      02030 5510 JMP I QREMARK
1156      02031 7240 CLA CMA /GET TYPE OF RIGHT SIDE
1157      02032 1014 TAD OSTACK
1158      02033 3021 DCA TEMP
1159      02034 1421 TAD I TEMP
1160      02035 7710 SPA CLA
1161      02036 5265 JMP STRCMP /STRING, DO STRING COMPARE
1162      02037 1376 TAD (MINUS /NUMERIC, DO A SUBTRACT
1163      02040 4506 JMS I QOUTOPR
1164      02041 4501 NUMCMP, JMS I QSAVECP /SAVE CHAR POSITION
1165      02042 4475 JMS I QCHKWD /LOOK FOR "THEN"
1166      02043 1737 WTHEN
1167      02044 5255 JMP NOTHEN /NOT THEN
1168      02045 4477 GETIFN, JMS I QSNUM /GET STATEMENT NUMBER
1169      02046 5366 JMP BADGO2
1170      02047 1021 TAD TEMP /OUTPUT JUMP
1171      02050 1311 TAD RELOPR

```

4B

Not a digit.
Pointer still at
same position

```

1172 02051 4500 JMS I QOUTWRD
1173 02052 1022 TAD TEMP2 /TWO WORDS
1174 02053 4500 JMS I QOUTWRD
1175 02054 5507 JMP I QNEWLIN
1176 02055 4504 NOTHEN, JMS I QRESTCP /BACKUP CHAR POS
1177 02056 4475 JMS I QCHKWD /LOOK FOR "GOTO"
1178 02057 6271 Wgoto
1179 02060 7410 SKP
1180 02061 5245 JMP GETIFN /OK, GO GET STMT NUMBER
1181 02062 4473 BADIF, JMS I QERMSG /BAD IF STMT
1182 02063 1106
1183 02064 5510 JMP I QREMARK
1184 02065 1375 STRCMP, TAD (SCOMPR-1
1185 02066 4506 JMS I QOUTOPR /OUTPUT STRING COMPARE
1186 02067 4476 JMS I QMODSET /BACK TO N MODE
1187 02070 5241 JMP NUMCMP /REST IS LIKE NUMERIC COMPARES
1188 02071 4525 JMS I QBACK1 /PUT BACK NON OPERATOR
1189 02072 1410 IFLUP2, TAD I X10 /GET CONDITIONAL JUMP
1190 02073 7450 SNA
1191 02074 5262 JMP BADIF /RELATIONAL INCORRECT
1192 02075 3311 DCA RELOPR
1193 02076 1410 TAD I X10 /COMPARE OPERATORS
1194 02077 1021 TAD TEMP
1195 02100 7650 SNA CLA
1196 02101 5227 JMP GOTREL /GOTIT
1197 02102 5272 JMP IFLUP2
1198 02103 4474 IFEND, JMS I QLODSN /OUTPUT STMT NUMBER
1199 02104 7201 CLA IAC / (NO COLON)
1200 02105 4311 JMS GETFN /GET FILE NUMBER
1201 02106 1374 TAD (JEOP /SETUP CORRECT JUMP
1202 02107 3311 DCA RELOPR
1203 02110 5241 JMP NUMCMP /GO FIND "THEN" OR "GOTO"
1204
1205 02111 0000 GETFN, 0 /GET FILE NUMBER
1206 02112 3062 DCA COLON /SAVE COLON SWITCH
1207 02113 4513 JMS I QCHECKC /LOOK FOR #
1208 02114 7735 -43
1209 02115 5340 JMP TTYFIL /NONE, MUST BE TTY
1210 02116 4505 JMS I QEXPR /GET FILE EXPR
1211 02117 5510 JMP I QREMARK /ERROR
1212 02120 1062 TAD COLON /DO WE NEED A COLON ?
1213 02121 7640 SZA CLA
1214 02122 5326 JMP .+4 /NO, SKIP THIS TEST
1215 02123 4513 JMS I QCHECKC /YES, LOOK FOR IT
1216 02124 7706 -72
1217 02125 5331 JMP BADFN /NOT THERE, BAD
1218 02126 4520 JMS I QLOAD /LOAD IT
1219 02127 1056 TAD TYPE1 /TYPE MUST BE NUMERIC
1220 02130 7710 SPA CLA
1221 02131 4473 BADFN, JMS I QERMSG /NOPE, IT ISN'T
1222 02132 0616
1223 02133 7201 CLA IAC /SET IFNREG TO "NOT TTY"
1224 02134 3063 DCA IFNREG /SAVE NEW IFNREG
1225 02135 1373 TAD (FILENO /OUTPUT SET IFN COMMAND
1226 02136 4500 JMS I QOUTWRD

```

4B

1227	02137	5711	JMP I	GETFN	
1228	02140	1063	TTYFIL, TAD	IFNREG	/IS IFNREG 0 ?
1229	02141	7650	SNA CLA		
1230	02142	5711	JMP I	GETFN	/IF YES, QUIT
1231	02143	1372	TAD	(CLRFN	/OTHERWISE ZERO AC
1232	02144	4500	JMS I	QOUTWRD	
1233	02145	3063	DCA	IFNREG	/SET IFNREG TO TTY
1234	02146	5711	JMP I	GETFN	/RETURN

4B

```
1235 / GOTO AND GOSUB
1236 02147 4477 GOTO, JMS I QSNUM /GET NUMBER
1237 02150 5366 JMP BADG02
1238 02151 4476 JMS I QMODSET /ALL GOTO'S IN NMODE
1239 02152 7201 CLA IAC /JUMP=JSUB+1
1240 02153 5360 JMP ,+5
1241 02154 4474 GOSUB, JMS I QLODSN /OUTPUT STMT NUM LOAD
1242 02155 4477 JMS I QSNUM /GET NUMBER
1243 02156 5366 JMP BADG02
1244 02157 4476 JMS I QMODSET /ALL GOTO'S IN NMODE
1245 02160 1371 TAD (JSUB /GET GOSUB OPCODE
1246 02161 1021 TAD TEMP /PLUS ADDRESS
1247 02162 4500 JMS I QOUTWRD /OUTPUT IT
1248 02163 1022 TAD TEMP2 /BOTH WORDS
1249 02164 4500 JMS I QOUTWRD
1250 02165 5507 JMP I QNEWLIN
1251 02166 4473 BADG02, JMS I QERMSG /BAD GOTO OR GOSUB
1252 02167 1615 1615 /NUMBER MISSING
1253 02170 5510 JMP I QREMARK
```

4B


```

1254      / TABLE SEARCH FOR LITERALS, STMT NUMBERS, TEMPS, ETC.
1255      02171  5000
1256      02172  7501
1257      02173  7402
1258      02174  5400
1259      02175  6566
1260      02176  2755
1261      02177  4140
1262      2200
1263      02200  0000  LUKUP2, 0
1264      02201  1600      TAD I   LUKUP2  /GET THE BUCKET START
1265      02202  3342      DCA     OLDN3  /SAVE IT AS THE PREVIOUS ENTRY
1266      02203  2200      ISZ     LUKUP2
1267      02204  1600      TAD I   LUKUP2  /GET THE ENTRY SIZE
1268      02205  2200      ISZ     LUKUP2
1269      02206  3346      DCA     N3SIZE
1270      02207  1377      TAD     (6211  /PRIME THE FIELD SETTER
1271      02210  3071      DCA     LUFLD
1272      02211  4070      JMS     SETFLD  /NOW SET THE FIELD
1273      02212  1742  LOOK2,  TAD I   OLDN3  /GET ADDR OF NEXT ENTRY
1274      02213  3343      DCA     NEWN3  /SAVE IT
1275      02214  2342  PATCH1, ISZ     OLDN3  /GET TO FIELD OF NEW ENTRY
1276      02215  1742      TAD I   OLDN3  /GET INTO AC
1277      02216  3233      DCA     NEWFLD  /AND SAVE IT
1278      02217  1343      TAD     NEWN3
1279      02220  7450      SNA
1280      02221  5257      JMP     HOOKIN  /IF 0 ITS END OF LIST
1281      02222  7001  PATCH5, IAC
1282      02223  3010      DCA     X10     /START OF VALUE INFO
1283      02224  1376      TAD     (WORD1-1/SETUP POINTER TO VALUE
1284      02225  3011      DCA     X11
1285      02226  1346      TAD     N3SIZE  /AND TEMP OF ENTRY SIZE
1286      02227  3344      DCA     LTEMP
1287      02230  6201  CHKVAL, CDF
1288      02231  1411      TAD I   X11
1289      02232  7141      CIA CLL  /COMPARE THIS WORD
1290      02233  6211  NEWFLD, CDF  10     /FIELD OF NEW ENTRY
1291      02234  1410      TAD I   X10
1292      02235  7640      SZA CLA
1293      02236  5250      JMP     NOTSAM  /NOT THIS ONE
1294      02237  2344      ISZ     LTEMP  /INCR SIZE COUNT
1295      02240  5230      JMP     CHKVAL  /MORE STUFF
1296      02241  1410      TAD I   X10     /GET SYMBOL NUMBER
1297      02242  6201  L6201, CDF
1298      02243  3027      DCA     SYMBOL
1299      02244  1233      TAD     NEWFLD  /MAKE ENTRY ADDRESSABLE
1300      02245  3071      DCA     LUFLD  /THROUGH SETFLD
1301      02246  2200      ISZ     LUKUP2  /BUMP RETURN
1302      02247  5600      JMP I   LUKUP2
1303      02250  7430  NOTSAM, SZL
1304      02251  5257      JMP     HOOKIN  /NEW SYMBOL < CURRENT
1305      02252  1343      TAD     NEWN3  /GO TO NEXT ENTRY
1306      02253  3342      DCA     OLDN3  /(MOVE POINTER)
1307      02254  1233      TAD     NEWFLD  /(AND FIELD)
1308      02255  3071      DCA     LUFLD

```

4B

BK: JMP . + 3

BK: NOP

```

1309 02256 5212 JMP LOOK2
1310 02257 7144 HOOKIN, CLL CMA RAL /HOW MANY WORDS NEEDED ?
1311 02260 1346 TAD N3SIZE
1312 02261 1375 TAD (EOST
1313 02262 3264 DCA ,+2
1314 02263 4317 JMS STCHEK /MAKE SURE
1315 02264 0000 0 /WE GOT ENOUGH
1316 02265 1343 TAD NEWN3 /HOOK IN NEW ENTRY
1317 02266 6211 FREFLD, CDF 10 /CHANGE TO FREE FIELD
1318 02267 3416 DCA I NEXT
1319 02270 1233 PATCH2, TAD NEWFLD /HOOK IN FIELD
1320 02271 3416 DCA I NEXT
1321 02272 4070 JMS SETFLD /BACK TO FIELD OF OLD
1322 02273 1266 TAD FREFLD /PUT FIELD OF NEW
1323 02274 3742 DCA I OLDN3
1324 02275 7240 CLA CMA /BACK UP OLDN3
1325 02276 1342 TAD OLDN3 /SO THAT IT POINTS TO POINTER
1326 02277 3342 DCA OLDN3
1327 02300 7240 CLA CMA
1328 02301 1016 TAD NEXT /PUT POINTER TO NEW ENTRY
1329 02302 3742 DCA I OLDN3 /INTO OLD
1330 02303 1266 TAD FREFLD /SAVE ENTRY FIELD
1331 02304 3071 DCA LUFLO /FOR POSSIBLE POST PROCESSING
1332 02305 1376 TAD (WORD1-1 /PREPARE TO STICK IN THE VALUE
1333 02306 3011 DCA X11
1334 02307 6201 ENTERV, CDF
1335 02310 1411 TAD I X11 /MOVE IN THE VALUE
1336 02311 6211 FFLD2, CDF 10
1337 02312 3416 DCA I NEXT
1338 02313 2346 ISZ N3SIZE /INCR SIZE COUNT
1339 02314 5307 JMP ENTERV
1340 02315 6201 CDF
1341 02316 5600 JMP I LUKUP2
1342 02317 0000 STCHEK, 0 /CHECK FOR ENOUGH ROOM
1343 02320 1016 TAD NEXT /CHECK FOR OVERFLOW
1344 02321 7141 CIA CLL
1345 02322 6201 CDF
1346 02323 1717 TAD I STCHEK /THIS IS LIMIT
1347 02324 2317 ISZ STCHEK
1348 02325 7630 SZL CLA
1349 02326 5717 JMP I STCHEK
1350 02327 1266 TAD FREFLD /BUMP FREE FIELD
1351 02330 1374 TAD (10
1352 02331 3266 DCA FREFLD
1353 02332 1266 TAD FREFLD /PUT IN TWO PLACES
1354 02333 3311 DCA FFLD2
1355 02334 3016 DCA NEXT /START POINTER AT 0
1356 02335 2345 ISZ NFLDS /GONE TOO FAR ?
1357 02336 5717 JMP I STCHEK /NO
1358 02337 4473 STOVER, JMS I QERMSG /S.T. FULL
1359 02340 2324
1360 02341 5406 JMP I XABORT /ABORT COMPILATION
1361 02342 0000 OLDN3, 0 /ADDR OF PREVIOUS ENTRY
1362 02343 0000 NEWN3, 0 /ADDR OF NEW ENTRY
1363 02344 0000 LTEMP, 0

```

1364	02345	0000	NFLDS, 0			/= COUNT OF AVAILABLE FIELDS
1365			N3SIZE, 0			/SIZE OF ENTRY
1366	02346	0000	KBDCHK, 0			/CHECK FOR ^C OR ^D
1367	02347	6031	KSF			
1368	02350	5746	JMP I	KBDCHK		/NO CHAR
1369	02351	6036	KRB			
1370	02352	0373	AND	(177		/REMOVE PARITY BIT
1371	02353	1372	TAD	(-3		/^C ??
1372	02354	7450	SNA			
1373	02355	5406	JMP I	XABORT		/YES, EXIT TO OS8
1374	02356	1371	TAD	(3-17		/^D ??
1375	02357	7640	SZA CLA			
1376	02360	5746	JMP I	KBDCHK		/NO, RETURN
1377	02361	3770	DCA	TTX+1		/NOP TTY OUTPUT ROUTINE
1378	02362	5746	JMP I	KBDCHK		

```

1379 / SYMBOL TABLE LOOKUP
1380 02370 3741
1381 02371 7764
1382 02372 7775
1383 02373 0177
1384 02374 0010
1385 02375 7570
1386 02376 0133
1387 02377 6211
1388 2400
1389 02400 0000 LOOKUP, 0 /LOOK UP SYMBOL
1390 02401 1134 TAD NAME1 /GET NAME1*11+NAME2
1391 02402 7106 CLL RTL
1392 02403 1134 TAD NAME1
1393 02404 7104 CLL RAL
1394 02405 1134 TAD NAME1
1395 02406 1135 TAD NAME2
1396 02407 3134 DCA NAME1 /THIS IS IT
1397 02410 1026 TAD TYPE /WHAT KIND SYMBOL ?
1398 02411 7106 CLL RTL /MOVE TYPE BITS
1399 02412 7006 RTL /INTO AC 9,10,11
1400 02413 1216 TAD JTABLE
1401 02414 3215 DCA .+1
1402 02415 0000 VCPTR, 0 /GO THERE
1403 02416 5617 JTABLE, JMP I .+1
1404 02417 2427 LUVAR
1405 02420 2451 LURETN
1406 02421 2463 LUARRAY
1407 02422 2451 LURETN
1408 02423 2456 LUSTRG
1409 02424 2451 LURETN
1410 02425 2516 LUSARY
1411 02426 2451 LURETN
1412 02427 1377 LUVAR, TAD (VARCNT /POINTER TO VAR COUNT
1413 02430 3215 DCA VCPTR
1414 02431 1376 TAD (VARST-13
1415 02432 1134 DOLU, TAD NAME1
1416 02433 3037 DCA STPTR /ST POINTER
1417 02434 6211 CDF 10 /THATS WHERE ST IS
1418 02435 1437 TAD I STPTR /IS THIS VAR DEFINED YET ?
1419 02436 7500 SMA
1420 02437 5253 JMP GOTSYM /YES
1421 02440 1375 TAD (4401 /GET 401 INTO AC
1422 02441 6201 CHEKST, CDF
1423 02442 1615 TAD I VCPTR /PLUS VAR COUNT
1424 02443 6211 CDF 10
1425 02444 3027 DCA SYMBOL /THATS THE NEW SYMBOL NUMBER
1426 02445 1027 TAD SYMBOL /PUT SYMBOL NUMBER
1427 02446 3437 DCA I STPTR /INTO S.T. ENTRY
1428 02447 6201 CDF
1429 02450 2615 ISZ I VCPTR /BUMP SYMBOL NUMBER
1430 02451 5600 LURETN, JMP I LOOKUP
1431 02452 5774 JMP STOVER /S.T. OVERFLOW
1432 02453 3027 GOTSYM, DCA SYMBOL /PUT NUMBER INTO SYMBOL
1433 02454 6201 CDF

```

```

1434 02455 5600 JMP I LOOKUP
1435 02456 1373 LUSTRG, TAD (SVCNT /POINTER TO STRING VAR COUNT
1436 02457 3215 DCA VCPTR
1437 02460 1372 TAD (SVARST-26
1438 02461 1134 TAD NAME1 /TWO WORDS PER ENTRY
1439 02462 5232 JMP DOLU
1440 02463 1371 LUARRAY, TAD (ACNT /ARRAY VAR COUNT
1441 02464 3215 DCA VCPTR
1442 02465 1370 TAD (ARAYST /ARRAY SYMBOL TABLE
1443 02466 3037 DCA STPTR
1444 02467 6211 CDF 10
1445 02470 1437 FINDA, TAD I STPTR /SEARCH TABLE
1446 02471 7450 SNA
1447 02472 5311 JMP NEWARY /NEW ENTRY
1448 02473 7041 CIA
1449 02474 1134 TAD NAME1 /IS THIS IT ?
1450 02475 2037 ISZ STPTR
1451 02476 7650 SNA CLA
1452 02477 5304 JMP GOTARY /YES
1453 02500 2037 ISZ STPTR
1454 02501 2037 ISZ STPTR
1455 02502 2037 ISZ STPTR /GO TO NEXT ENTRY
1456 02503 5270 JMP FINDA
1457 02504 1367 GOTARY, TAD (37 /GET NUMBER
1458 02505 0437 AND I STPTR
1459 02506 3027 DCA SYMBOL /INTO SYMBOL
1460 02507 6201 CDF
1461 02510 5600 JMP I LOOKUP
1462 02511 1134 NEWARY, TAD NAME1 /PUT IN NEW ENTRY
1463 02512 3437 DCA I STPTR
1464 02513 2037 ISZ STPTR
1465 02514 1373 TAD (41 /PUT IN NUMBER
1466 02515 5241 JMP CHEKST /GO DO THE REST
1467 02516 1366 LUSARY, TAD (SACNT /STRING ARRAY COUNT
1468 02517 3215 DCA VCPTR
1469 02520 1365 TAD (SARYST /USE STRING ARRAY TABLE
1470 02521 5266 JMP FINDA-2 /GO DO SEARCH

```

```

1471      / FILE AND CLOSE PROCESSORS
1472      FILE, JMS I QLODSN /OUTPUT STMT NUMBER
1473      TAD (FOPENS /POINTER TO FILE OPENS
1474      DCA FILESW
1475      JMS I QCHECKC /LOOK FOR "V"
1476      -126
1477      SKP /NOT V
1478      ISZ FILESW /YUP, INCR FILESW
1479      JMS I QCHECKC /LOOK FOR "N"
1480      -116
1481      JMP .+3
1482      ISZ FILESW /INCR FILESW BY TWO IF "N"
1483      ISZ FILESW
1484      JMS GETFN /GET FILE NUMBER
1485      JMS I QEXPR /GET DEVICE/FILE DESCRIPTOR
1486      JMP I QREMARK
1487      JMS I QLOAD /LOAD INTO SAC
1488      TAD TYPE1 /TYPE MUST BE STRING
1489      SPA CLA
1490      JMP .+3 /IT WERE
1491      JMS I QERMSG /IT WEREN'T
1492      0616
1493      TAD I FILESW /GET CORRECT OPEN
1494      JMS I QOUTWRD
1495      JMP I QNEWLIN
1496      FOPENS, OPENAF;OPENAV;OPENNF;OPENNV
1497      02553 7474
1498      02554 7514
1499      02555 7534
1500      02556 0000 FILESW, 0
1501      02557 0040 PLUS, 40/0;XADD;XADD
1502      02560 0000
1503      02561 5162
1504      02562 5162

```

/ EXPRESSION ANALYZER

1505				
1506	02563	2111		
1507	02564	2552		
1508	02565	2332		
1509	02566	0043		
1510	02567	0037		
1511	02570	2132		
1512	02571	0042		
1513	02572	1010		
1514	02573	0041		
1515	02574	2337		
1516	02575	4401		
1517	02576	0365		
1518	02577	0040		
1519		2600		
1520	02600	0000	EXPR,	PAGE 0 /POLISHIZE EXPRESSION
1521	02601	3021		DCA TEMP /SAVE LEFT
1522	02602	1030		TAD LEFT /SO WE CAN PUSH OLD VALUE
1523	02603	4521		JMS I QPUSH /OF IT
1524	02604	1021		TAD TEMP /NOW SET NEW VALUE
1525	02605	3030		DCA LEFT /OF THAT SWITCH
1526	02606	1200		TAD EXPR
1527	02607	4521		JMS I QPUSH /SAVE RETURN ADDR
1528	02610	4521		JMS I QPUSH /MARK STACK
1529	02611	1030		TAD LEFT /IS THIS LEFT SIDE ?
1530	02612	7710		SPA CLA
1531	02613	5244		JMP OPRAND+1 /YES, NO UNARY MINUS
1532	02614	4502	UNOPR,	JMS I QGETC /LOOK FOR UNARY OPERATOR
1533	02615	5352		JMP MISARG /THERE HAS TO BE AN OPERAND
1534	02616	1377		TAD (-53 /UNARY+(NOP)
1535	02617	7450		SNA
1536	02620	5214		JMP UNOPR
1537	02621	1376		TAD (53-55 /UNARY -
1538	02622	7440		SZA
1539	02623	5227		JMP NOTMIN /NOT UNARY MINUS
1540	02624	1375		TAD (UNOPR /PUSH UNARY MINUS
1541	02625	4521		JMS I QPUSH
1542	02626	5214		JMP UNOPR
1543	02627	1374	NOTMIN,	TAD (55-50 /LOOK FOR (
1544	02630	7640		SZA CLA
1545	02631	5243		JMP OPRAND /NOT A SUB EXPRESSION
1546	02632	4505		JMS I QEXPR /COMPILE SUB EXPRESSION
1547	02633	5773		JMP BADEXP /BAD SUB EXPRESSION
1548	02634	4513		JMS I QCHECKC /LOOK FOR)
1549	02635	7727		-51
1550	02636	7410		SKP /ERROR
1551	02637	5263		JMP OPRBR /GOTIT
1552	02640	4473		JMS I QERMSG /PARENTHESIS MIS MATCH
1553	02641	1520		1520
1554	02642	5773		JMP BADEXP
1555	02643	4525	OPRAND,	JMS I QBACK1 /PUT BACK NON UNARY OP
1556	02644	4514		JMS I QGETNAM /LOOK FOR VARIABLE REF
1557	02645	5772		JMP NOTVAR /NOPE.
1558	02646	4516		JMS I QLOOKUP /SYMBOL TABLE SEARCH
1559	02647	1027		TAD SYMBOL /SAVE SYMBOL NUMBER

1560	02650	3022	DCA	TEMP2	/BECAUSE SAVAC MIGHT KILL IT
1561	02651	4524	JMS I	QSAVAC	/GENERATE FSTA (MAYBE)
1562	02652	7775	-3		
1563	02653	1026	TAD	TYPE	/WAS THIS A FUNCTION OR ARRAY ?
1564	02654	0371	AND	(3000	
1565	02655	7440	SZA		
1566	02656	5771	JMP	FUNSS	/YES, GO PROCESS IT
1567	02657	1026	TAD	TYPE	/MAKE OPERAND STACK ENTRY
1568	02660	4523	JMS I	QPUSHO	
1569	02661	1022	TAD	TEMP2	/FIRST TYPE THEN SYMBOL #
1570	02662	4523	JMS I	QPUSHO	
1571	02663	1030	OPR8R,	TAD	LEFT /LEFT SIDE ?
1572	02664	7700	SMA CLA		/YES, NO OPERATORS LEGAL
1573	02665	4502	JMS I	QGETC	/LOOK FOR OPERATOR
1574	02666	5341	JMP	ENDEXP	/END OF EXPR
1575	02667	1370	TAD	(-52	/** IS SPECIAL CASE
1576	02670	7440	SZA		
1577	02671	5301	JMP	NOSTAR	/NOT *
1578	02672	4502	JMS I	QGETC	/LOOK FOR SECOND *
1579	02673	5301	JMP	NOSTAR	
1580	02674	1370	TAD	(-52	
1581	02675	7650	SNA CLA		
1582	02676	1367	TAD	(136-52	/** -> -
1583	02677	7450	SNA		
1584	02700	4525	JMS I	QBACK1	/PUT IT BACK
1585	02701	1366	NOSTAR,	TAD	(52 /RESTORE CHAR
1586	02702	3021	DCA	TEMP	
1587	02703	1365	TAD	(OPR8RS-1	
1588	02704	3010	DCA	X10	/PTR TO LIST
1589	02705	1410	OPRLUP,	TAD I	X10 /GET OPERATOR PTR
1590	02706	7450	SNA		
1591	02707	5336	JMP	ENDEXP-3	/END OF LIST
1592	02710	3032	DCA	NEWOP	/SAVE IT IN CASE
1593	02711	1410	TAD I	X10	/COMPARE
1594	02712	1021	TAD	TEMP	
1595	02713	7640	SZA CLA		
1596	02714	5305	JMP	OPRLUP	/KEEP LOOKING
1597	02715	4522	GOTOPR,	JMS I	QPOP /GET STACK TOP
1598	02716	7450	SNA		
1599	02717	5327	JMP	PUSH2	/EMPTY
1600	02720	3031	DCA	OLDOP	
1601	02721	1431	TAD I	OLDOP	/COMPARE PREC.
1602	02722	7041	CIA		
1603	02723	1432	TAD I	NEWOP	/NEW-OLD
1604	02724	7750	SPA SNA	CLA	
1605	02725	5333	JMP	OUTOLD	/OLD>NEW
1606	02726	1031	TAD	OLDOP	
1607	02727	4521	PUSH2,	JMS I	QPUSH /OLD < NEW
1608	02730	1032	TAD	NEWOP	/GO PUSH BOTH
1609	02731	4521	JMS I	QPUSH	
1610	02732	5214	JMP	UNOPR	/GO LOOK FOR NEXT OPERAND
1611	02733	1031	OUTOLD,	TAD	OLDOP /OUTPUT CODE FOR OLD OPR8R
1612	02734	4506	JMS I	QOUTOPR	
1613	02735	5315	JMP	GOTOPR	/LOOK AT NEXT TOP OF STACK
1614	02736	4525	JMS I	QBACK1	/PUT BACK NON OPERATOR

1615	02737	7410	SKP	
1616	02740	4506	JMS I	QOUTOPR /OUTPUT OPERATOR
1617	02741	4522	ENDEXP, JMS I	QPOP /LOOK FOR STACK MARK
1618	02742	7440	SZA	
1619	02743	5340	JMP	ENDEXP-1/NOT THIS
1620	02744	4522	JMS I	QPOP /GET RETURN ADDR
1621	02745	7001	IAC	
1622	02746	3021	DCA	TEMP
1623	02747	4522	JMS I	QPOP /GET LEFT SIDE SWITCH
1624	02750	3030	DCA	LEFT
1625	02751	5421	JMP I	TEMP /RETURN
1626	02752	4473	MISARG, JMS I	QERMSG /MISSING OPERAND
1627	02753	1517		1517
1628	02754	5773	JMP	BADEXP
1629	02755	0040	MINUS, 40	0
1630	02756	0000		
1631	02757	3763		
1632	02760	1743		
1633	02761	0050	SLASH, 50	0
1634	02762	0000		
1635	02763	5557		
1636	02764	6756		

```

1637      / EXPRESSION ANALYZER (HANDLE SUBSCRIPTS)
1638      02765  6047
1639      02766  0052
1640      02767  0064
1641      02770  7726
1642      02771  3000
1643      02772  3136
1644      02773  3122
1645      02774  0005
1646      02775  0765
1647      02776  7776
1648      02777  7725
1649      3000
1650      03000  0377  FUNSS,  PAGE      (1000  /IS IT FUN CALL ?
1651      03001  7650      SNA CLA
1652      03002  5205      JMP      ,+3      /NO
1653      03003  4524      JMS I   QSAVAC  /YES, SAVE AC
1654      03004  7777      -1
1655      03005  1026      TAD      TYPE    /SAVE TYPE
1656      03006  4521      JMS I   QPUSH
1657      03007  1022      TAD      TEMP2   /AND SYMBOL NUMBER
1658      03010  4521      JMS I   QPUSH
1659      03011  1037      TAD      STPTR   /AND SYMBOL TABLE PTR
1660      03012  4521      JMS I   QPUSH
1661      03013  7410      SKP
1662      03014  4522  SSLOOP, JMS I   QPOP    /GET ARG/SS COUNT
1663      03015  7001      IAC
1664      03016  4521      JMS I   QPUSH    /INCREMENT IT
1665      03017  4505      JMS I   QEXPR   /GET NEXT ARG/SS
1666      03020  5315      JMP      BADFSS
1667      03021  4511      JMS I   QGETA1  /IS THIS ARG(SS) AN ARRAY REF ?
1668      03022  7132      CLL CML  RTR
1669      03023  0056      AND      TYPE1   /CHECK THE TYPE
1670      03024  7650      SNA CLA
1671      03025  5242      JMP      NOTSSD  /NOT AN ARRAY REFERENCE
1672      03026  4512      JMS I   QLOADSS /LOAD THE SS REGS
1673      03027  4524      JMS I   QSAVAC  /SAVE AC IF NEEDED
1674      03030  7777      -1
1675      03031  1056      TAD      TYPE1   /SET THE MODE
1676      03032  4476      JMS I   QMODSET
1677      03033  1376      TAD      (AFLOA  /LOAD THIS ARG/SS
1678      03034  1057      TAD      SYMBL1
1679      03035  4500      JMS I   QOUTWRD
1680      03036  1133      TAD      Q400    /SET THE IN-AC BIT
1681      03037  1055      TAD      MODE    /WE JUST CALLED MODSET
1682      03040  3414      DCA I   OSTACK  /CHANGE THIS STACK ENTRY
1683      03041  7410      SKP
1684      03042  2014  NOTSSD, ISZ      OSTACK /FIX UP OSTACK
1685      03043  2014      ISZ      OSTACK
1686      03044  4515      JMS I   QCOMARP  /LOOK FOR , OR )
1687      03045  5315      JMP      BADFSS  /NEITHER IS BAD
1688      03046  5214      JMP      SSLOOP  / , MEANS MORE ARGS/SS
1689      03047  4522      JMS I   QPOP    /GET # OF ARG/SS
1690      03050  3021      DCA      TEMP   /GET ARG/SS COUNT
1691      03051  4522      JMS I   QPOP    /RESTORE S.T. ADDR

```

1692	03052	3037	DCA	STPTR	
1693	03053	4522	JMS I	QPOP	
1694	03054	3027	DCA	SYMBOL	/GET BACK THE SYMBOL #
1695	03055	4522	JMS I	QPOP	
1696	03056	3026	DCA	TYPE	/GET BACK THE TYPE
1697	03057	1026	TAD	TYPE	/IS IT AN ARRAY OR FUN REF ?
1698	03060	0377	AND	(1000	
1699	03061	7640	SZA	CLA	
1700	03062	5775	JMP	DOCALL	/FUNCTION REFERENCE
1701	03063	1021	TAD	TEMP	/MOVE SS COUNT
1702	03064	7112	CLL	RTR	/INTO THE CORRECT
1703	03065	7012	RTR		/FIELD
1704	03066	3022	DCA	TEMP2	/AND SAVE IT
1705	03067	6211	CDF	10	
1706	03070	1437	TAD I	STPTR	/ANY PREV REFERENCE ?
1707	03071	0374	AND	(3000	
1708	03072	7440	SZA		
1709	03073	5300	JMP	NOTNEW	/YES, GO CHECK NUMBERS
1710	03074	1022	TAD	TEMP2	/IF NONE, PUT IN NUMBER
1711	03075	1437	TAD I	STPTR	
1712	03076	3437	DCA I	STPTR	
1713	03077	5304	JMP	NDOK	/THATS ALL
1714	03100	7041	NOTNEW, CIA		/COMPARE NUMBER OF SS
1715	03101	1022	TAD	TEMP2	/WITH ANY PREVIOUS
1716	03102	7640	SZA	CLA	
1717	03103	5320	JMP	BADFSS+3	/THEY DON'T MATCH
1718	03104	6201	NDOK, CDF		
1719	03105	1026	TAD	TYPE	/PUT TYPE
1720	03106	1021	TAD	TEMP	/AND DIM COUNT
1721	03107	4523	ONSTAK, JMS I	QUSHO	/ONTO ARGUMENT STACK
1722	03110	1027	TAD	SYMBOL	
1723	03111	4523	JMS I	QUSHO	/AND SYMBOL NUMBER
1724	03112	4524	JMS I	QSAVAC	/SAVE FIRST SS IF LEFT IN AC
1725	03113	7773	=5		
1726	03114	5773	JMP	OPRBR	/GO GET AN OPERATOR
1727	03115	1372	BADFSS, TAD	(=4	/PURGE STACK JUNK
1728	03116	1015	TAD	STACK	
1729	03117	3015	DCA	STACK	
1730	03120	4473	JMS I	QERMSG	/PUT ERROR MESSAGE
1731	03121	2323	2323		
1732	03122	4522	BADEXP, JMS I	QPOP	/LOOK FOR STACK MARK
1733	03123	7640	SZA	CLA	
1734	03124	5322	JMP	BADEXP	/NOT YET
1735	03125	4522	JMS I	QPOP	/RETURN ADDR
1736	03126	3021	DCA	TEMP	
1737	03127	4522	JMS I	QPOP	/SS LOAD SWITCH
1738	03130	3030	DCA	LEFT	
1739	03131	5421	JMP I	TEMP	/TAKE ERROR EXIT
1740	03132	7654	WTAB, -124;-101;-102;-50		
1741	03133	7677			
1742	03134	7676			
1743	03135	7730			
1744	03136	1030	NOTVAR, TAD	LEFT	/LEFT SIDE ?
1745	03137	7710	SPA	CLA	
1746	03140	5771	JMP	MISARG	/YES, NO LITERALS LEGAL

1747	03141	4526	JMS I	QNUMBER /LOOK FOR LITERAL
1748	03142	5350	JMP	NOTNUM /NOT A NUMBER
1749	03143	4517	JMS I	QLUKUP2 /SEARCH LITERAL TABLE
1750	03144	2562	LITRL	
1751	03145	7775	-3	
1752	03146	4770'	JMS	NEWVAR /IF NEW, GIVE IT NUMBER
1753	03147	5307	JMP	ONSTAK /GO PUT IT ONTO THE STACK
1754	03150	4527	NOTNUM, JMS I	QSTRING /LOOK FOR STRING LITERAL
1755	03151	5771'	JMP	MISARG /NO, MISSING ARG
1756	03152	1134	TAD	WORD1 /GET =NUMBER WORDS - 1
1757	03153	7001	IAC	
1758	03154	7170	CLL CML	CMA RAR
1759	03155	3360	DCA	.+3 /FOR LOOKUP
1760	03156	4517	JMS I	QLUKUP2 /LOOK UP LITERAL
1761	03157	2564	SLITRL	
1762	03160	0000	0	
1763	03161	4767'	JMS	NWSVAR /IF NEW, GIVE IT NUMBER
1764	03162	7130	CLL CML	RAR /SET TYPE BIT FOR STRING
1765	03163	5307	JMP	ONSTAK /PUT INFO ONTO STACK

```

1766      / EXPRESSION ANALYZER (HANDLE FUNCTION CALLS)
1767      03167 4236
1768      03170 4223
1769      03171 2752
1770      03172 7774
1771      03173 2663
1772      03174 3000
1773      03175 3200
1774      03176 6640
1775      03177 1000
1776      3200
1777      03200 1030      DOCALL, TAD      LEFT      /IS THIS LEFT SIDE ?
1778      03201 7700      SMA CLA      /IF YES, FUN ILLEGAL
1779      03202 4210      JMS      OUTCAL      /GENERATE CALL
1780      03203 7410      SKP      /SKIP IF ERROR
1781      03204 5777      JMP      OPRBR      /GO LOOK FOR OPERATOR
1782      03205 4473      JMS I      QERMSG      /BAD FUNCTION REFERENCE
1783      03206 0622      0622
1784      03207 5776      JMP      BADEXP
1785      03210 0000      OUTCAL, 0      /GENERATE FUN CALL; TYPE,
1786      /SYMBOL AND TEMP ARE INPUTS
1787      03211 1027      TAD      SYMBOL      /SAVE FUNCTION NUMBER AROUND SAVAC
1788      03212 3336      DCA      FUNNUM
1789      03213 4524      JMS I      QSAVAC      /SAVE SECOND FROM TOP
1790      03214 7775      -3
1791      03215 1336      TAD      FUNNUM      /SETUP FOR FINDING FUNCTION
1792      03216 3134      DCA      WORD1      /INFO BLOCK
1793      03217 4517      JMS I      QLUKUP2      /ON THE FUNCTION LIST
1794      03220 2570      FUNCTN
1795      03221 7777      -1
1796      03222 5610      JMP I      OUTCAL      /UNDEFINED FUNCTION
1797      03223 1027      TAD      SYMBOL      /CHECK NUMBER OF ARGS
1798      03224 1021      TAD      TEMP
1799      03225 7640      SZA CLA
1800      03226 5610      JMP I      OUTCAL
1801      03227 4520      MOVARG, JMS I      QLOAD      /GET TOP OF STACK INTO AC
1802      03230 4070      JMS      SETFLO      /GET FIELD OF FORMAL-PARAMS
1803      03231 1410      TAD I      X10      /GET FIRST ONE
1804      03232 6201      CDF
1805      03233 3021      DCA      TEMP
1806      03234 7130      CLL CML      /COMPARE TYPE OF ARG
1807      03235 0056      AND      TYPE1      /WITH THAT OF FORMAL PARAMETER
1808      03236 1021      TAD      TEMP
1809      03237 7710      SPA CLA      /THEY MUST MATCH
1810      03240 5610      JMP I      OUTCAL      /((THEY DON'T)
1811      03241 7132      CLL CML      /SHOULD WE LEAVE IT IN THE AC ?
1812      03242 0021      AND      TEMP
1813      03243 7640      SZA CLA
1814      03244 5253      JMP      OKINAC      /YES, SAVES AN INSTRUCTION
1815      03245 1056      TAD      TYPE1      /SET MODE
1816      03246 4476      JMS I      QMODSET      /APPROPRIATELY
1817      03247 7150      CLL CMA      RAR      /3777
1818      03250 0021      AND      TEMP      /GET SYM NUMBER
1819      03251 1375      TAD      (FSTA      /STORE VALUE IN FORM PARAM
1820      03252 4500      JMS I      QOUTWRD

```

```

1821 03253 2027 OKINAC, ISZ SYMBOL /MORE ARGS ?
1822 03254 5227 JMP MOVARG
1823 03255 4070 JMS SETFLD
1824 03256 1410 TAD I X10 /GET TYPE OF FUNCTION
1825 03257 3056 DCA TYPE1 /((ITS RESULT THAT IS)
1826 03260 6201 CDF
1827 03261 1026 TAD TYPE /IS TYPE OF FUNCTION
1828 03262 1056 TAD TYPE1 /SAME AS TYPE OF CALL
1829 03263 7710 SPA CLA
1830 03264 5610 JMP I OUTCAL /NO, ERROR
1831 03265 4476 JMS I QMODSET /ALL CALLS IN N MODE
1832 03266 1134 TAD WORD1 /CHECK FOR USER FUNCTION
1833 03267 7500 SMA
1834 03270 5303 JMP CALLUF /YES, DO SPECIAL CALL
1835 03271 2210 FINCAL, ISZ OUTCAL /FIX RETURN
1836 03272 4500 JMS I QOUTWRD /OUTPUT CODE
1837 03273 1133 TAD Q400 /SET TOP OF STACK
1838 03274 1056 TAD TYPE1
1839 03275 3414 DCA I OSTACK /TO AC
1840 03276 3414 DCA I OSTACK /SYMBOL NUMBER IS MEANINGLESS
1841 03277 7130 CLL CML RAR
1842 03300 0056 AND TYPE1 /INTERPRETER MODE SAME
1843 03301 3055 DCA MODE /AS FUNCTION TYPE
1844 03302 5610 JMP I OUTCAL /ON RETURN
1845 03303 4532 CALLUF, JMS I QNOREGS /FORGET REGS ON USER FUNC
1846 03304 1071 TAD LUFLO /OUTPUT JSUB
1847 03305 0374 AND (70 /WITH POINTER TO
1848 03306 7106 CLL RTL /DOUBLE WORD
1849 03307 1373 TAD (JSUB /VALUE OF LOCATION
1850 03310 4500 JMS I QOUTWRD /COUNTER FOR THE
1851 03311 1010 TAD X10 /START OF THE
1852 03312 7001 IAC /USER "DEF"INED FUNC
1853 03313 5271 JMP FINCAL
1854 03314 0000 FSUB1, 0 /FOR SUBROUTINE #1
1855 03315 4505 JMS I QEXPR /GET AN EXPRESSION
1856 03316 5323 JMP BADFOR
1857 03317 4520 JMS I QLOAD /LOAD VALUE
1858 03320 1056 TAD TYPE1 /MUST BE NUMERIC
1859 03321 7700 SMA CLA
1860 03322 5714 JMP I FSUB1 /OK
1861 03323 4473 BADFOR, JMS I QERMSG /BAD FOR LOOP PARAMETERS
1862 03324 0620 0620
1863 03325 5510 JMP I QREMARK
1864 03326 0000 FSUB2, 0 /FOR SUBROUTINE #2
1865 03327 4314 JMS FSUB1 /GET EXPR AND LOAD IT
1866 03330 4772 JMS GENTMP /MAKE A TEMP FOR IT
1867 03331 1027 TAD SYMBOL /STORE EXPR IN TEMP
1868 03332 1375 TAD (FSTA
1869 03333 4500 JMS I QOUTWRD
1870 03334 1027 TAD SYMBOL /RETURN SLOT #
1871 03335 5726 JMP I FSUB2
1872
1873 03336 0000 FUNNUM, NOREGS, 0 /FORGET REGISTORS
1874 03337 7201 CLA IAC /FILE NUMBER REG
1875 03340 3063 DCA IFNREG

```

/OS/8 BASIC COMPILER

PAL8-V8 10/30/72 PAGE 24-2

1876		/	CMA	/SUBSCRIPT REG #1
1877		/	DCA	SSREG1
1878		/	CMA	/SUBSCRIPT REG #2
1879		/	DCA	SSREG2
1880	03341	5736	JMP I	NOREGS
1881	03342	4474	CLOSE, JMS I	QLODSN /OUTPUT STMT NUMBER
1882	03343	7201	CLA IAC	/NO COLON NEEDED AFTER FILE NUM
1883	03344	4771	JMS	GETFN /GET FILE NUM
1884	03345	1370	TAD	(CLOSEF /OUTPUT CLOSE
1885	03346	4500	JMS I	QOUTWRD
1886	03347	5507	JMP I	QNEWLIN

1887 / CODE GENERATOR

1888 03370 7434
 1889 03371 2111
 1890 03372 4200
 1891 03373 5000
 1892 03374 0070
 1893 03375 2400
 1894 03376 3122
 1895 03377 2663

1896 3400

1897 03400 0000

1898 03401 3010

1899 03402 1410

1900 03403 7540

1901 03404 5314

1902 03405 3026

1903 03406 1377

1904 03407 3355

1905 03410 1410

1906 03411 3354

1907 03412 1410

1908 03413 3356

1909 03414 1026

1910 03415 4476

1911 03416 7144

1912 03417 1014

1913 03420 3014

1914 03421 1014

1915 03422 3010

1916 03423 1410

1917 03424 3357

1918 03425 1410

1919 03426 3360

1920 03427 1357

1921 03430 0376

1922 03431 3021

1923 03432 1357

1924 03433 0133

1925 03434 7640

1926 03435 5256

1927 03436 7132

1928 03437 0357

1929 03440 7650

1930 03441 5245

1931 03442 4512

1932 03443 2355

1933 03444 2356

1934 03445 4317

1935 03446 1056

1936 03447 0133

1937 03450 7640

1938 03451 5306

1939 03452 1755

1940 03453 1360

1941 03454 4500

PAGE

OUTOPR, 0

DCA X10 /OUTPUT CODE FOR OPERATOR

TAD I X10 /SAVE POINTER TO SKELETON

SMA SZA /GET CONTROL WORD

JMP SPCIAL /TREAT AS SPECIAL CASE

DCA TYPE /ITS THE TYPE ALLOWANCE

TAD (XLOAD /GET SKEL ADDRS

DCA CASEMM /FOR THE THREE CASES

TAD I X10

DCA CASEMA

TAD I X10

DCA CASEAM

TAD TYPE /ENTER CORRECT MODE

JMS I QMODSET

CLL CMA RAL /GET THE SECOND OPERAND

TAD OSTACK

DCA OSTACK

TAD OSTACK

DCA X10 /BY BACKING UP THE STACK

TAD I X10 /TYPE

DCA TYPE2

TAD I X10

DCA SYMBL2 /SYMBOL NUMBER

TAD TYPE2

AND (3

DCA TEMP /SS COUNT

TAD TYPE2 /LOOK AT OPERAND 2

AND Q400

SZA CLA

JMP MAC /MUST BE CASE M,AC

CLL CML RTR /ITS IN MEMORY, IS IT SS'D

AND TYPE2

SNA CLA

JMP A20K /NO, ITS SCALAR

JMS I QLOADSS /LOAD NECESSARY SS REGS

ISZ CASEMM /FIXUP THE SKELETON POINTERS

ISZ CASEAM

JMS GETA1 /GET STUF FOR ARG1

TAD TYPE1 /LOOK AT IT

AND Q400

SZA CLA

JMP ACM /ITS CASE AC,M

TAD I CASEMM /ITS CASE M,M LOAD OPERAND 2

TAD SYMBL2

JMS I QOUTWRD

1942	03455	7410	SKP		
1943	03456	4317	MAC, JMS	GETA1	/GET STUF FRO ARG1
1944	03457	7132	CLL CML	RTR	/IS IT SS'D ?
1945	03460	0056	AND	TYPE1	
1946	03461	7650	SNA CLA		
1947	03462	5265	JMP	A10K	/NO, ITS SCALAR
1948	03463	4512	JMS I	QLOADSS	/LOAD THE SS REGS
1949	03464	2354	ISZ	CASEMA	/BUMP SKELETON ADDR
1950	03465	1754	A10K, TAD I	CASEMA	/GET CORRECT INSTRUCTION
1951	03466	1057	TAD	SYMBL1	/PLUS SYMBOL NUMBER
1952	03467	4500	TYPCHK, JMS I	QOUTWRD	/OUTPUT IT
1953	03470	7130	CLL CML	RAR	/TYPES OF OPERANDS MUST MATCH
1954	03471	0056	AND	TYPE1	
1955	03472	1357	TAD	TYPE2	
1956	03473	7710	SPA CLA		
1957	03474	5311	JMP	MIXED	/THEY DON'T
1958	03475	1026	TAD	TYPE	/TYPE OF OPERATOR
1959	03476	1056	TAD	TYPE1	/MUST MATCH
1960	03477	7710	SPA CLA		/THAT OF OPERANDS
1961	03500	5311	JMP	MIXED	/THEY DON'T
1962	03501	1133	TAD	Q400	/GENERATE STACK ENTRY
1963	03502	1026	TAD	TYPE	
1964	03503	3414	DCA I	OSTACK	
1965	03504	3414	DCA I	OSTACK	/THIS IS SAFE
1966	03505	5600	JMP I	OUTOPR	
1967	03506	1756	ACM, TAD I	CASEAM	/ITS CASE AC,M
1968	03507	1360	TAD	SYMBL2	/GEN OPERATION FOR OPERAND 2
1969	03510	5267	JMP	TYPCHK	/GO FINISH IT UP
1970	03511	4473	MIXED, JMS I	GERMSG	/MIXED TYPES
1971	03512	1524	1524		
1972	03513	5600	JMP I	OUTOPR	
1973	03514	1410	SPCIAL, TAD I	X10	/GET ADDR OF SPECIAL RTNE
1974	03515	3021	DCA	TEMP	/(PLUS 1 FROM THE TYPE WORD)
1975	03516	5421	JMP I	TEMP	/HANDLE SPECIAL CASE
1976	03517	0000	GETA1, 0		/GET STUFF FOR ARG 1
1977	03520	7144	CLL CMA	RAL	/BACK UP STACK
1978	03521	1014	TAD	OSTACK	
1979	03522	3014	DCA	OSTACK	
1980	03523	1014	TAD	OSTACK	
1981	03524	3011	DCA	X11	
1982	03525	1411	TAD I	X11	/GET TYPE1
1983	03526	3056	DCA	TYPE1	
1984	03527	1411	TAD I	X11	/GET SYMBL1
1985	03530	3057	DCA	SYMBL1	
1986	03531	1056	TAD	TYPE1	/GET SS COUNT
1987	03532	0376	AND	(3	
1988	03533	3021	DCA	TEMP	
1989	03534	5717	JMP I	GETA1	
1990	03535	4524	UMRTNE, JMS I	QSAVAC	/SAVE CURRENT AC IF NEEDED
1991	03536	7775	-3		
1992	03537	4520	JMS I	QLOAD	/GET ARG IN AC
1993	03540	3026	DCA	TYPE	/TYPE MUST BE NUMERIC
1994	03541	3357	DCA	TYPE2	
1995	03542	1375	TAD	(FNEG	/DO NEGATE
1996	03543	5267	JMP	TYPCHK	

1997	03544	3026	EXPRTN, DCA	TYPE	/SET FUNC TYPE
1998	03545	7126	CLL CML	RTL	/SET NUMBER OF ARGS
1999	03546	3021	DCA	TEMP	
2000	03547	1374	TAD	(FUNC1+60	
2001	03550	3027	DCA	SYMBOL	/EXP2
2002	03551	4773	JMS	OUTCAL	/OUTPUT FUNCTION CALL
2003	03552	5311	JMP	MIXED	/ERROR
2004	03553	5600	JMP I	OUTOPR	/DONE
2005	03554	0000	CASEMA, 0		
2006	03555	0000	CASEMM, 0		
2007	03556	0000	CASEAM, 0		
2008	03557	0000	TYPE2, 0		
2009	03560	0000	SYMBL2, 0		
2010	03561	4474	RETURN, JMS I	QLODSN	/OUTPUT STMT NUM LOAD
2011	03562	4476	JMS I	QMODSET	/ALWAYS RETURN IN N MODE
2012	03563	1372	TAD	(RET-RNDO	
2013	03564	1371	RANDOM, TAD	(RNDO-STOP	
2014	03565	1370	STOPX, TAD	(STOP	/RETURN, RANDOMIZE, OR STOP
2015	03566	4500	JMS I	QOUTWRD	
2016	03567	5507	JMP I	QNEWLIN	

```

2017          / LETTER AND DIGIT SCANNERS
2018 03570 7441
2019 03571 7760
2020 03572 7763
2021 03573 3210
2022 03574 7476
2023 03575 7403
2024 03576 0003
2025 03577 0770
2026          3600
2027 03600 0000  LETTER, 0  PAGE 0  /SKIP ON LETTER
2028 03601 4502          JMS I  QGETC
2029 03602 5600          JMP I  LETTER /NO LETTER
2030 03603 1377          TAD    (-133 /MUST BE .LT. 133
2031 03604 7500          SMA
2032 03605 5214          JMP    NOLETR
2033 03606 1376          TAD    (133-100/MUST BE .GT. 100
2034 03607 7510          SPA
2035 03610 5214          JMP    NOLETR
2036 03611 0375          AND    (77 /RESTORE 6 BITS
2037 03612 2200          ISZ    LETTER /BUMP RETURN ADDR
2038 03613 5600          JMP I  LETTER
2039 03614 4525  NOLETR, JMS I  QBACK1 /PUT CHAR BACK
2040 03615 5600          JMP I  LETTER
2041 03616 0000  DIGIT, 0  /SKIP ON DIGIT
2042 03617 4502          JMS I  QGETC
2043 03620 5616          JMP I  DIGIT /NO DIGIT
2044 03621 1374          TAD    (-72 /MUST BE .LT. 72
2045 03622 7100  07100, CLL    / (USED AS LITERAL BY "TTY")
2046 03623 1373          TAD    (72-60 /MUST BE .GE. 60
2047 03624 7420          SNL
2048 03625 5230          JMP    NODIGT /NOPE
2049 03626 2216          ISZ    DIGIT /RETURN DIGIT MINUS 60
2050 03627 5616          JMP I  DIGIT
2051 03630 4525  NODIGT, JMS I  QBACK1 /PUT IT BACK
2052 03631 5616          JMP I  DIGIT

```

```

2053 / STATEMENT NUMBER GETTER
2054 SNUM, 0 /GET A STATEMENT NUMBER
2055 DCA TEMP /SAVE DEFINED SWITCH
2056 JMS I QDIGIT /GET FIRST DIGIT
2057 JMP I SNUM /NO STATEMENT NUMBER
2058 DCA WORD2 /THIS WILL BE THE BUCKET
2059 TAD WORD2
2060 CLL RAL /TWO WORDS PER BUCKET
2061 TAD (SNUMS
2062 DCA BUCKET
2063 ISZ SNUM /OK, ITS A STMT NUMBER
2064 TAD (-4 /FIVE DIGITS MAX
2065 DCA TEMP2
2066 DCA WORD1 /CLEAR TOP WORD
2067 SNLOOP, JMS I QDIGIT /GET NEXT DIGIT
2068 JMP GOTSN /END OF NUMBER
2069 DCA WORD3 /SAVE IT
2070 TAD (-4 /SET SHIFT COUNT
2071 DCA ACO
2072 SHIFT, TAD WORD2 /SHIFT LEFT ONE BIT
2073 CLL RAL
2074 DCA WORD2
2075 TAD WORD1
2076 RAL
2077 DCA WORD1
2078 ISZ ACO /BUMP SHIFT COUNTER
2079 JMP SHIFT
2080 TAD WORD2 /PUT IN NEW DIGIT
2081 TAD WORD3
2082 DCA WORD2
2083 ISZ TEMP2 /BUMP DIGIT COUNT
2084 JMP SNLOOP
2085 GOTSN, JMS I QLUKUP2 /FIND STMT NUMBER
2086 BUCKET, 0
2087 -2
2088 JMP NEWSN /ITS A NEW STMT NUM
2089 CLL CML RAR /CHECK FOR MULTIPLY DEFINED
2090 AND SYMBOL
2091 AND TEMP
2092 SZA CLA
2093 JMP MDLABL /YES, IT IS
2094 TAD X10 /GET ADDR OF LABEL VALUE
2095 DCA TEMP2
2096 JMS SETFLD /GET TO FIELD OF ENTRY
2097 TAD TEMP /OR IN THESE BITS
2098 TAD SYMBOL
2099 DCA I TEMP2
2100 FINSN, CDF
2101 TAD LUFLD /GET FIELD BITS
2102 AND (70
2103 CLL RTL
2104 DCA TEMP /INTO A CONVIENIENT
2105 JMP I SNUM /PLACE
2106 NEWSN, JMS SETFLD /GET FIELD
2107 TAD TEMP /PUT IN BITS

```

2108	03720	3416		DCA I	NEXT	
2109	03721	1016		TAD	NEXT	/SAVE N3 ADDR
2110	03722	3022		DCA	TEMP2	
2111	03723	3416		DCA I	NEXT	/1 EXTRA WORD
2112	03724	5310		JMP	FINSN	
2113	03725	4473	MDLABL,	JMS I	QERMSG	/MULTIPLY DEFINED
2114	03726	1504		1504		/LABEL
2115	03727	5632		JMP I	SNUM	
2116	03730	0000	TTY,	0		/CONVERT TO ASCII AND PRINT
2117	03731	0375		AND	(77	/SIX BITS ONLY
2118	03732	1367		TAD	(-40	/WHAT SIDE OF FORTY ?
2119	03733	7510		SPA		
2120	03734	1222		TAD	07100	/LOW SIDE
2121	03735	1366		TAD	(240	/HIGH SIDE
2122	03736	4340		JMS	TTX	/PRINT CHAR
2123	03737	5730		JMP I	TTY	/RETURN
2124	03740	0000	TTX,	0		/PRINT CHAR ON TTY
2125	03741	7410		SKP		/((CONTROL 0 ZEROES THIS WORD)
2126	03742	5346		JMP	+.4	/((THUS KILLING ERROR REPORTING)
2127	03743	6041		TSF		
2128	03744	5343		JMP	.-1	
2129	03745	6046		TLS		
2130	03746	7200		CLA		
2131	03747	5740		JMP I	TTX	

```
2132      / CHAIN PROCESSOR
2133 03750 4474 CHAIN, JMS I QLODSN /OUTPUT STMT NUMBER
2134 03751 4505      JMS I QEXPR /GET CHAIN STRING
2135 03752 5510      JMP I QREMARK
2136 03753 4520      JMS I QLOAD /INTO SAC
2137 03754 1056      TAD TYPE1 /TYPE MUST BE STRING
2138 03755 7700      SMA CLA
2139 03756 4473      JMS I QERMSG /IT WASN'T
2140 03757 0616      0616 / (OK IF ERROR CODE IS NOP)
2141 03760 1365      TAD (CHN /OUTPUT CHAIN OPCODE
2142 03761 4500      JMS I QOUTWRD
2143 03762 5507      JMP I QNEWLIN
2144 03763 3000 XISUB, FISUB, AISUB
2145 03764 6400
```

```

2146      / SEVERAL SHORT UTILITY ROUTINES
2147      03765 7414
2148      03766 0240
2149      03767 7740
2150      03770 0070
2151      03771 7774
2152      03772 2532
2153      03773 0012
2154      03774 7706
2155      03775 0077
2156      03776 0033
2157      03777 7645
2158      40000
2159      04000 0000 BACK1, 0 /BACK UP ONE CHAR
2160      04001 7240 CLA CMA
2161      04002 1020 TAD NCHARS
2162      04003 3020 DCA NCHARS
2163      04004 7240 CLA CMA
2164      04005 1017 TAD CHRPTN
2165      04006 3017 DCA CHRPTN
2166      04007 5600 JMP I BACK1
2167      04010 0000 GETCWB, 0 /GET A CHARACTER (PRESERVE BLANKS)
2168      04011 2020 ISZ NCHARS
2169      04012 5216 JMP .+4
2170      04013 7240 CLA CMA
2171      04014 3020 DCA NCHARS /RESET NCHARS
2172      04015 5610 JMP I GETCWB
2173      04016 2210 ISZ GETCWB
2174      04017 1417 TAD I CHRPTN /GET THE CHAR
2175      04020 5610 JMP I GETCWB
2176      04021 0000 SAVECP, 0 /SAVE CHAR POSITION
2177      04022 1020 TAD NCHARS
2178      04023 3365 DCA NCSAVE
2179      04024 1017 TAD CHRPTN
2180      04025 3366 DCA CPSAVE
2181      04026 5621 JMP I SAVECP
2182      04027 0000 RESTCP, 0 /RESTORE CHAR POS
2183      04030 1366 TAD CPSAVE
2184      04031 3017 DCA CHRPTN
2185      04032 1365 TAD NCSAVE
2186      04033 3020 DCA NCHARS
2187      04034 5627 JMP I RESTCP
2188      04035 0000 GETC, 0 /GET A CHARACTER (IGNORING BLANKS)
2189      04036 2020 ISZ NCHARS
2190      04037 5243 JMP .+4
2191      04040 7240 CLA CMA
2192      04041 3020 DCA NCHARS
2193      04042 5635 JMP I GETC
2194      04043 1417 TAD I CHRPTN
2195      04044 1377 TAD (-40) /IS IT A BLANK
2196      04045 7450 SNA
2197      04046 5236 JMP GETC+1 /YES IGNORE IT
2198      04047 1376 TAD (-40) /FIX CHAR
2199      04050 2235 ISZ GETC
2200      04051 5635 JMP I GETC

```

```

2201 04052 0000 POP, 0 /GET TOP OF STACK
2202 04053 1015 TAD STACK
2203 04054 3262 DCA PUSH
2204 04055 7240 CLA CMA
2205 04056 1015 TAD STACK
2206 04057 3015 DCA STACK /DECREMENT STACK POINTER
2207 04060 1662 TAD I PUSH
2208 04061 5652 JMP I POP
2209 04062 0000 PUSH, 0 /PUT AC ONTO STACK
2210 04063 3415 DCA I STACK /STORE
2211 04064 1375 TAD (=STACK-STACKS+1
2212 04065 1015 TAD STACK /CHECK FOR OVERFLOW
2213 04066 7710 SPA CLA
2214 04067 5662 JMP I PUSH /OK, RETURN
2215 04070 4473 STKOV, JMS I QERMSG
2216 04071 2004 JMP I XABORT /ABORT COMPILATION
2217 04072 5406 PUSHO, 0 /PUSH OPERAND STACK
2218 04073 0000 DCA I OSTACK /PUSHIT
2219 04074 3414 TAD (=STACK-STOKS+1
2220 04075 1374 TAD OSTACK /CHECK FOR STACK OVERFLOW
2221 04076 1014 SPA CLA
2222 04077 7710 JMP I PUSHO
2223 04100 5673 JMP STKOV /TOO FULL
2224 04101 5270 COMAR, 0 /SKIP ON COMA OR RITE PAREN
2225 04102 0000 JMS I QGETC /GET CHAR
2226 04103 4502 JMP I COMARP
2227 04104 5702 TAD (=51
2228 04105 1373 SNA
2229 04106 7450 ISZ COMARP /RITE PAREN, SKIP 2
2230 04107 2302 SZA
2231 04110 7440 TAD (=51-54 /CHECK FOR ,
2232 04111 1372 SNA
2233 04112 7450 ISZ COMARP / , SKIP 1
2234 04113 2302 SZA CLA
2235 04114 7640 JMS I QBACK1 /NEITHER PUT BACK
2236 04115 4525 JMP I COMARP
2237 04116 5702 LOAD, 0 /LOAD SAC OR FAC
2238 04117 0000 JMS I QGETA1 /GET TOP OF STACK
2239 04120 4511 TAD TYPE1 /SET MODE
2240 04121 1056 JMS I QMODSET
2241 04122 4476 TAD TYPE1 /IS IT IN THE AC?
2242 04123 1056 AND Q400
2243 04124 0133 SZA CLA
2244 04125 7640 JMP I LOAD /YUP
2245 04126 5717 CLL CML RTR /SUBSCRIPTED ?
2246 04127 7132 AND TYPE1
2247 04130 0056 SNA CLA
2248 04131 7650 JMP ,+3 /NO
2249 04132 5335 JMS I QLOADSS /FILL SS REGS
2250 04133 4512 TAD (=AFLDA-FLDA
2251 04134 1371 TAD (FLDA /ARRAY OR SCALAR LOAD
2252 04135 1370 TAD SYMBL1 /PLUS SYMBOL NUMBER
2253 04136 1057 JMS I QOUTWRD
2254 04137 4500 JMP I LOAD
2255 04140 5717

```


2256	04141	5003	IFOPS,	JNEJ-7476	/<>
2257	04142	0302			
2258	04143	5003		JNEJ-7674	/><
2259	04144	0104			
2260	04145	5002		JGEJ-7576	/=>
2261	04146	0202			
2262	04147	5002		JGEJ-7675	/>=
2263	04150	0103			
2264	04151	5007		JLEJ-7574	/=<
2265	04152	0204			
2266	04153	5007		JLEJ-7475	/<=
2267	04154	0303			
2268	04155	0000		0	
2269	04156	5006		JEQJ-7500	/=
2270	04157	0300			
2271	04160	5004		JGTJ-7600	/>
2272	04161	0200			
2273	04162	5005		JLTJ-7400	/<
2274	04163	0400			
2275	04164	0000		0	
2276	04165	0000	NCSAVE,	0	
2277	04166	0000	CPSAVE,	0	

```

2278      / TEMP GENERATORS AND AC SAVING ROUTINES
2279      04170 2000
2280      04171 4640
2281      04172 7775
2282      04173 7727
2283      04174 6601
2284      04175 0601
2285      04176 0040
2286      04177 7740
2287      4200
2288      04200 0000  GENTMP, 0      PAGE      /GENERATE A TEMP
2289      04201 7640      SZA CLA
2290      04202 5213      JMP      STRTMP /ITS A STRING TEMP
2291      04203 1033      TAD      TMPCNT
2292      04204 2033      ISZ      TMPCNT /BUMP COUNT
2293      04205 3134      DCA      NAME1
2294      04206 4517      JMS I    QLUKUP2 /LOOK UP THIS TEMP
2295      04207 2556      TEMPS
2296      04210 7777      -1
2297      04211 4223      JMS      NEWVAR /NEW ONE ON ME
2298      04212 5600      JMP I    GENTMP
2299      04213 1035  STRTMP, TAD      STMPCT
2300      04214 2035      ISZ      STMPCT /BUMP COUNT
2301      04215 3134      DCA      NAME1
2302      04216 4517      JMS I    QLUKUP2 /LOOK UP THIS TEMP
2303      04217 2560      STEMS
2304      04220 7777      -1
2305      04221 4236      JMS      NWSVAR /NEW STRING TEMP
2306      04222 5600      JMP I    GENTMP
2307      04223 0000  NEWVAR, 0      /MAKE SYM NUM FOR VAR
2308      04224 1040      TAD      VARCNT /PUT SYM NUM
2309      04225 1377      TAD      (401
2310      04226 3027      DCA      SYMBOL /INTO SYMBOL
2311      04227 1027      TAD      SYMBOL /AND INTO ST ENTRY
2312      04230 4070      JMS      SETFLD
2313      04231 3416      DCA I    NEXT
2314      04232 6201      CDF
2315      04233 2040      ISZ      VARCNT /BUMP COUNT
2316      04234 5623      JMP I    NEWVAR /RETURN WITH SYM NUM
2317      04235 5776      JMP      STOVER /S.T. OVERFLOW
2318      04236 0000  NWSVAR, 0      /MAKE SYM NUM FOR VARS
2319      04237 1041      TAD      SVCNT /PUT SYM NUM
2320      04240 1377      TAD      (401
2321      04241 3027      DCA      SYMBOL
2322      04242 1027      TAD      SYMBOL /INTO SYMBOL AND
2323      04243 4070      JMS      SETFLD
2324      04244 3416      DCA I    NEXT /S.T. ENTRY
2325      04245 6201      CDF
2326      04246 2041      ISZ      SVCNT /OVERFLOW ?
2327      04247 5636      JMP I    NWSVAR /NO, WE'RE OK
2328      04250 5776      JMP      STOVER
2329      04251 0000  SAVAC, 0      /SAVE FAC (OR SAC) IF NECESSARY
2330      04252 1651      TAD I    SAVAC /GET ENTRY POINTER
2331      04253 1014      TAD      OSTACK
2332      04254 2251      ISZ      SAVAC

```

2333	04255	3300	DCA	SVTEMP	/ADDR OF TYPE WORD
2334	04256	1700	TAD I	SVTEMP	/LOOK AT IT
2335	04257	0133	AND	Q400	
2336	04260	7650	SNA CLA		
2337	04261	5651	JMP I	SAVAC	/NOT IN AC
2338	04262	7130	CLL CML	RAR	/SAVE STRING BIT ONLY
2339	04263	0700	AND I	SVTEMP	/OF TYPE WORD
2340	04264	3700	DCA I	SVTEMP	
2341	04265	1700	TAD I	SVTEMP	
2342	04266	4200	JMS	GENTMP	/GENERATE TEMP
2343	04267	1700	TAD I	SVTEMP	
2344	04270	4476	JMS I	QMODSET	/SET MODE
2345	04271	1301	TAD	XSTOR	
2346	04272	1027	TAD	SYMBOL	/GENERATE STORE
2347	04273	4500	JMS I	QOUTWRD	
2348	04274	1027	TAD	SYMBOL	/RETURN S.T. NUMBER
2349	04275	2300	ISZ	SVTEMP	/MOVE TO SYMBOL NUM WORD
2350	04276	3700	DCA I	SVTEMP	/SAVE THE TEMP NUM THERE
2351	04277	5651	JMP I	SAVAC	/RETURN WITH SAVE MADE
2352	04300	0000	SVTEMP,	0	
2353	04301	2400	XSTOR,	FSTA/AFSTA	
2354	04302	6700			

```

2355      / SUBSCRIPT REGISTER LOADING ROUTINE
2356 04303 0000 LOADSS, 0      /LOAD SS REGS
2357 04304 7144      CLL CMA RAL      /LOOK AT NUMBER OF SS
2358 04305 1021      TAD      TEMP
2359 04306 7650      SNA CLA
2360 04307 5314      JMP      LODSS2 /2 SS
2361 04310 7420      SNL
2362 04311 5320      JMP      TOOMNY /MORE THAN 2
2363 04312 4323      JMS      SSLOAD /LOAD SS REG 1
2364 04313 5703      JMP I      LOADSS
2365 04314 7201      LODSS2, CLA IAC
2366 04315 4323      JMS      SSLOAD /LOAD SS REG 2
2367 04316 4323      JMS      SSLOAD /NOW SS REG 1
2368 04317 5703      JMP I      LOADSS
2369
2370 04320 4473      SSTYPE,
2371 04321 2323      TOOMNY, JMS I      QERMSG /SUBSCRIPTING ERROR
2372 04322 5703      2323
2373 04323 0000      JMP I      LOADSS
2374 04324 3022      SSLOAD, 0      /LOAD A SS REG FROM TOP OF STACK
2375 04325 7144      DCA      TEMP2 /SS REG 1 OR 2 SWITCH
2376 04326 1014      CLL CMA RAL      /BACK UP ONE ENTRY
2377 04327 3014      TAD      OSTACK /ON THE OPERAND STACK
2378 04330 1014      DCA      OSTACK
2379 04331 3011      TAD      OSTACK
2380 04332 1411      DCA      X11      /USE X11 TO GET STUFF
2381 04333 7510      TAD I      X11      /GET TYPE WORD
2382 04334 5320      SPA
2383 04335 0133      JMP      SSTYPE /SS MUST BE A NUMBER
2384 04336 7640      AND      Q400      /GET AC BIT
2385 04337 5347      SZA CLA
2386 04340 1022      JMP      SSINAC /ITS IN THE AC
2387 04341 7640      TAD      TEMP2
2388 04342 1375      SZA CLA
2389 04343 1374      TAD      (LSS2-LSS1
2390 04344 1411      TAD      (LSS1 /LOAD REG 1 OR 2 ??
2391 04345 4500      TAD I      X11      /ANYHOW, THIS IS THE SOURCE
2392 04346 5723      JMS I      QOUTWRD /OUTPUT THE CODE
2393 04347 1022      JMP I      SSLOAD
2394 04350 1373      SSINAC, TAD      TEMP2
2395 04351 4500      TAD      (LSS1AC /NOTE: LSS2AC=LSS1AC+1
2396 04352 5723      JMS I      QOUTWRD /SO OUTPUT ONE OF THEM
                JMP I      SSLOAD

```

2397 /INPUT DEVICE HANDLER
2398 04373 7406
2399 04374 4000
2400 04375 0400
2401 04376 2337
2402 04377 0401
2403 4400 *INDEVH
2404 04400 0000 0

```

2405      /INITIALIZATION CODE FOR RUN CASE
2406      4600      PAGE
2407      04600    6212  RUNNED, CIF      10      /COME HERE IF .R BCOMP
2408      04601    4777      JMS I      (200    /CALL COMMAND DECODER
2409      04602    0005      5
2410      04603    0201      0201      /ASSUMED EXTENSION "BA"
2411      04604    6221      CDF      20
2412      04605    1413      TAD I      X13
2413      04606    6211      CDF      10      /SAVE LOCATIONS 27774 THRU 27777
2414      04607    3412      DCA I      X12      /THESE ARE THE BOS PARAMETERS
2415      04610    2021      ISZ      TEMP
2416      04611    5204      JMP      .-5
2417      04612    6211      CDF      10
2418      04613    3410      DCA I      X10      /ZERO EDITOR BLOCK NUMBER
2419      04614    6201      CDF
2420      04615    1411  FINDSV, TAD I      X11      /LOOKUP SOME SAVE FILES
2421      04616    7450      SNA
2422      04617    5236      JMP      LUBUF      /GO LOOK FOR BASIC,UF
2423      04620    3225      DCA      XXXXSV      /SAVE POINTER TO NAME
2424      04621    7201      CLA IAC      /THEY'RE ON SYS
2425      04622    6212      CIF      10
2426      04623    4777      JMS I      (200
2427      04624    0002      2
2428      04625    0000  XXXXSV, 0
2429      04626    0000      0
2430      04627    5776      JMP      NG      /ERROR
2431      04630    1225      TAD      XXXXSV      /GET STARTING BLOCK
2432      04631    7001      IAC      /PLUS 1
2433      04632    6211      CDF      10
2434      04633    3410      DCA I      X10      /INTO INFO AREA
2435      04634    6201      CDF
2436      04635    5215      JMP      FINDSV      /LOOP
2437      04636    7201  LUBUF, CLA IAC
2438      04637    6212      CIF      10
2439      04640    4777      JMS I      (200      /LOOKUP BASIC,UF
2440      04641    0002      2
2441      04642    7562      BUFN      / (USER DEFINED FUNCTIONS)
2442      04643    0000      0
2443      04644    5247      JMP      .+3      /OK IF NOT THERE
2444      04645    1242      TAD      .-3      /GET STARTING BLOCK +1
2445      04646    7001      IAC
2446      04647    6211      CDF      10
2447      04650    3410      DCA I      X10      /INTO INFO BLOCK
2448      04651    6201  STRT3, CDF
2449      04652    7201      CLA IAC      /ENTER TEMPORARY FILE
2450      04653    6212      CIF      10
2451      04654    4777      JMS I      (200
2452      04655    0003      3
2453      04656    7566  TMPBLK, TMPFIL
2454      04657    0000      0
2455      04660    5776      JMP      NG
2456      04661    1256      TAD      TMPBLK      /SAVE START OF TEMP FILE
2457      04662    3775      DCA      OUBLOK
2458      04663    1256      TAD      TMPBLK      /IN A COUPLE PLACES
2459      04664    3046      DCA      BLOCK

```

/09/8 BASIC COMPILER

PAL8-V8 10/30/72 PAGE 33-1

2460 04665 1257
2461 04666 3774'
2462 04667 5773'

TAD TMPBLK+1/ALSO THE SIZE
DCA OUSIZE
JMP GETDEV /GO FETCH DEVICE HANDLER

```

2463          / NUMERIC CONVERSION ROUTINE (PART ONE)
2464 04773 7032
2465 04774 6506
2466 04775 6475
2467 04776 7120
2468 04777 0200
2469 05000 5000
2470 05000 0000 NUMBER, 0 PAGE
2471 05001 3023 DCA DECPT /GENERAL NUMBER CONVERSION ROUTINE
2472 05002 3134 DCA WORD1 /ZERO DECIMAL POINT SWITCH
2473 05003 3135 DCA WORD2 /ZERO FAC
2474 05004 3136 DCA WORD3
2475 05005 3137 DCA ACO
2476 05006 3212 DCA SIGN /CLEAR SIGN SWITCH
2477 05007 4502 JMS I QGETC /GET A CHAR
2478 05010 5600 JMP I NUMBER /NO CHAR IS NO NUMBER
2479 05011 4777 JMS CHKSGN /CHECK FOR SIGN
2480 05012 0000 SIGN, 0 /THIS SWITCH GETS SET
2481 05013 3024 DCA NDIGIT /ZERO DIGIT COUNT
2482 05014 4531 CONVLP, JMS I QDIGIT /GET A DIGIT
2483 05015 5240 JMP TRYDEC /IS THERE A DECIMAL POINT ?
2484 05016 3355 DCA NXTDGT /SAVE THE DIGIT
2485 05017 2024 ISZ NDIGIT /INCR NUMBER OF DIGITS
2486 05020 1135 TAD WORD2 /PREPARE TO MULT BY 10
2487 05021 3141 DCA OP2
2488 05022 1136 TAD WORD3
2489 05023 3142 DCA OP3
2490 05024 1137 TAD ACO
2491 05025 3143 DCA OPO
2492 05026 4776 JMS I (AL1 /DOUBLE FAC
2493 05027 4776 JMS I (AL1 /DOUBLE AGAIN
2494 05030 4775 JMS I (OADD /TIMES FIVE
2495 05031 4776 JMS I (AL1 /ONE MORE DOUBLING IS TIMES 10
2496 05032 3141 DCA OP2
2497 05033 3142 DCA OP3 /PUT NEWEST DIGIT INTO OPERAND
2498 05034 1355 TAD NXTDGT
2499 05035 3143 DCA OPO
2500 05036 4775 JMS I (OADD /ADD IN NEWEST DIGIT
2501 05037 5214 JMP CONVLP
2502 05040 1023 TRYDEC, TAD DECPT /DECIMAL ALREADY ?
2503 05041 7640 SZL CLA
2504 05042 5256 JMP TRYE2 /YES, LOOK FOR EXPONENT
2505 05043 4502 JMS I QGETC /LOOK FOR .
2506 05044 5253 JMP DIGTST /SEE IF THERE WAS ANYTHING
2507 05045 1374 TAD (=56
2508 05046 7640 SZL CLA
2509 05047 5252 JMP TRYE1 /TRY FOR E
2510 05050 2023 ISZ DECPT /SET DECIMAL POINT SW
2511 05051 5213 JMP CONVLP-1 /LOOP FOR OTHER DIGITS
2512 05052 4525 TRYE1, JMS I QBACK1 /PUT BACK NON .
2513 05053 1024 DIGTST, TAD NDIGIT /ANY DIGITS YET ?
2514 05054 7650 SZL CLA
2515 05055 5600 JMP I NUMBER /NO, NO NUMBER
2516 05056 4502 TRYE2, JMS I QGETC /LOOK FOR E
2517 05057 5300 JMP NOEXP+1 /GO HANDLE EXPONENT

```


2518	05060	1360	TAD	WSTEP+2	/USE PART OF "STEP" LITERAL
2519	05061	7640	SZA	CLA	
2520	05062	5277	JMP	NOEXP	/NO EXPONENT
2521	05063	3267	GETEXP, DCA	ESIGN	/ZERO EXPONENT SIGN SWITCH
2522	05064	4502	JMS I	QGETC	/GET A CHAR
2523	05065	5277	JMP	NOEXP	/TREAT AS NO EXPONENT
2524	05066	4777	JMS	CHKSGN	/IS IT A SIGN
2525			FPRTNE,		
2526	05067	0000	ESIGN,	0	/THIS IS THE SWITCH TO SET
2527	05070	4773	JMS	SMLNUM	/GO GET THE EXPONENT
2528	05071	1267	FIXEXP, TAD	ESIGN	/CHECK EXPONENT SIGN
2529	05072	7650	SNA	CLA	
2530	05073	5301	JMP	NOEXP+2	
2531	05074	1025	TAD	EXPON	/COMPLEMENT EXPONENT
2532	05075	7041	CIA		
2533	05076	7410	SKP		
2534	05077	4525	NOEXP, JMS I	QBACK1	/PUT BACK NON E
2535	05100	3025	DCA	EXPON	/ZERO EXPONENT
2536	05101	1372	TAD	(43	/NORMALIZE THE NUMBER
2537	05102	3134	DCA	WORD1	
2538	05103	4771	JMS I	(ANORM	
2539	05104	1023	TAD	DECPT	/WAS THERE A DECIMAL POINT ?
2540	05105	7640	SZA	CLA	
2541	05106	1024	TAD	NDIGIT	/HOW MANY DIGITS TO THE RIGHT ?
2542	05107	7041	CIA		
2543	05110	1025	TAD	EXPON	/SUBTRACT THAT NUMBER FROM EXP
2544	05111	7500	SMA		
2545	05112	5317	JMP	POSEXP	/EXPONENT IS POSITIVE
2546	05113	7041	CIA		
2547	05114	3025	DCA	EXPON	/ONLY NEED ABS VALUE
2548	05115	1370	TAD	(FPDIV	/DO DIVIDES
2549	05116	5321	JMP	,+3	
2550	05117	3025	POSEXP, DCA	EXPON	
2551	05120	1367	TAD	(FPMUL	/DO MULTIPLIES
2552	05121	3267	DCA	FPRTNE	/MULTIPLY/DIVIDE ROUTINE
2553	05122	1366	TAD	(PETABL-1	
2554	05123	3011	DCA	X11	/POWERS OF TEN TABLE
2555	05124	1025	EXPMUL, TAD	EXPON	/LOOK AT THE EXPONENT
2556	05125	7450	SNA		
2557	05126	5350	JMP	DOSIGN	/IF 0 ITS THRU
2558	05127	7110	CLL	RAR	
2559	05130	3025	DCA	EXPON	/PUT LOWEST BIT INTO LINK
2560	05131	7420	SNL		
2561	05132	5345	JMP	SKPEXP	/THIS ONE DOESN'T COUNT
2562	05133	1411	TAD I	X11	/MOVE FACTOR INTO OPERAND
2563	05134	3140	DCA	OP1	
2564	05135	1411	TAD I	X11	
2565	05136	3141	DCA	OP2	
2566	05137	1411	TAD I	X11	
2567	05140	3142	DCA	OP3	
2568	05141	1411	TAD I	X11	
2569	05142	3143	DCA	OP0	
2570	05143	4667	JMS I	FPRTNE	/MULTIPLY OR DIVIDE BY THIS FACTOR
2571	05144	5324	JMP	EXPMUL	/CHECK NEXT BIT
2572	05145	1011	SKPEXP, TAD	X11	/SKIP OVER THIS FACTOR

2573	05146	1365		TAD	(4
2574	05147	5323		JMP	EXPMUL-1
2575	05150	1212	DOSIGN,	TAD	SIGN /CHECK THE SIGN
2576	05151	7640		SZA	CLA
2577	05152	4764		JMS I	(NEGFAC /NEGATE IF NEGATIVE
2578	05153	2200		ISZ	NUMBER /BUMP RETURN
2579	05154	5600		JMP I	NUMBER /RETURN
2580	05155	0000	NXTDGT,	0	
2581	05156	7655	WSTEP,	-123/-124/-105/-120	
2582	05157	7654			
2583	05160	7673			
2584	05161	7660			
2585	05162	0000	XADD,	FADD;AFADD	
2586	05163	6440			

```

2587      / NUMERIC CONVERSION ROUTINE (PART TWO)
2588      05164  5267
2589      05165  0004
2590      05166  5777
2591      05167  5200
2592      05170  5310
2593      05171  5236
2594      05172  0043
2595      05173  5400
2596      05174  7722
2597      05175  5353
2598      05176  5431
2599      05177  5444
2600      5200
2601      05200  0000      FPMUL,  0      PAGE 0      /FLOATING MULTIPLY ROUTINE
2602      05201  1134      TAD      WORD1      /COMPUTE NEW EXPONENT
2603      05202  1140      TAD      OP1
2604      05203  3140      DCA      OP1
2605      05204  1135      TAD      WORD2      /SAVE AC MANTISSA
2606      05205  3234      DCA      TW2
2607      05206  1136      TAD      WORD3
2608      05207  3235      DCA      TW3
2609      05210  1377      TAD      (-30      /SET ITERATION COUNTER
2610      05211  3371      DCA      ITRCNT
2611      05212  3135      DCA      WORD2      /ZERO FAC MANTISSA
2612      05213  3136      DCA      WORD3
2613      05214  3137      DCA      ACO
2614      05215  4776      MULLUP, JMS I  (AR1      /SHIFT FAC RIGHT ONE
2615      05216  1234      TAD      TW2      /SHIFT MULTIPLIER RIGHT
2616      05217  7110      CLL RAR
2617      05220  3234      DCA      TW2
2618      05221  1235      TAD      TW3
2619      05222  7010      RAR
2620      05223  3235      DCA      TW3
2621      05224  7430      SZL
2622      05225  4353      JMS      OADD      /ADD IF LINK IS ONE
2623      05226  2371      ISZ      ITRCNT      /BUMP COUNT
2624      05227  5215      JMP      MULLUP      /LOOP
2625      05230  1140      TAD      OP1      /PUT IN CORRECT EXPONENT
2626      05231  3134      DCA      WORD1
2627      05232  4236      JMS      ANORM      /NORMALIZE THE RESULT
2628      05233  5600      JMP I  FPMUL
2629
2630      05234  0000      D2,      0
2631      D3,
2632      05235  0000      TW2,      0
2633      TW3,      0
2634      05236  0000      NFCNT,
2635      05237  1135      ANORM,  0      /NORMALIZE FAC
2636      05240  7450      TAD      WORD2      /IS MANTISSA 0 ?
2637      05241  1136      SNA
2638      05242  7450      TAD      WORD3
2639      05243  1137      SNA
2640      05244  7650      TAD      ACO
2641      05245  5265      SNA CLA
2641      JMP      ZEXP      /YES, ZERO EXPONENT

```

2642	05246	7332	NORMLP,	CLA	CLL	CML	RTR	/IS HIGH ORDER MANTISSA = 6000
2643	05247	1135		TAD			WORD2	
2644	05250	7440		SZA				
2645	05251	5256		JMP		N06000		/NO, SKIP THIS CRAP
2646	05252	1136		TAD		WORD3		/YES, IS THE REST 0 ?
2647	05253	7450		SNA				
2648	05254	1137		TAD		ACO		
2649	05255	7640		SZA	CLA			/SKIP IF 600000 ... 0000
2650	05256	7710	N06000,	SPA	CLA			
2651	05257	5636		JMP	I	ANORM		/NORM IS DONE WHEN BITS DIFFER
2652	05260	4775		JMS	I	(AL1		/SHIFT LEFT ONE
2653	05261	7240		CLA	CMA			/DECREMENT EXPONENT
2654	05262	1134		TAD		WORD1		
2655	05263	3134		DCA		WORD1		
2656	05264	5246		JMP		NORMLP		/LOOP
2657	05265	3134	ZEXP,	DCA		WORD1		
2658	05266	5636		JMP	I	ANORM		
2659	05267	0000	NEGFAC,	0				/NEGATE FAC
2660	05270	1374		TAD		(ACO		/GET POINTER TO OPERAND
2661	05271	3307		DCA		NFPTR		
2662	05272	7146		CLL	CMA	RTL		/THREE WORD NEGATE
2663	05273	3236		DCA		NFCNT		
2664	05274	7100		CLL				
2665	05275	7004	NFLOOP,	RAL				
2666	05276	1707		TAD	I	NFPTR		/GET NEXT WORD
2667	05277	7161		CLL	CML	CIA		
2668	05300	3707		DCA	I	NFPTR		/RESTORE AFTER COMPLEMENTING
2669	05301	7260		CML	CLA	CMA		/LINK GETS COMPLEMENTED ONCE HERE
2670	05302	1307		TAD		NFPTR		/AND ONCE AGAIN HERE
2671	05303	3307		DCA		NFPTR		/RESTORE DECREMENTED POINTER
2672	05304	2236		ISZ		NFCNT		
2673	05305	5275		JMP		NFLOOP		
2674	05306	5667		JMP	I	NEGFAC		
2675	05307	0000	NFPTR,	0				
2676	05310	0000	FPDIV,	0				
2677	05311	4776		JMS	I	(AR1		/UNNORMALIZE AC BY ONE
2678	05312	1140		TAD		OP1		/COMPUTE FINAL EXPONENT
2679	05313	7041		CIA				
2680	05314	1134		TAD		WORD1		
2681	05315	3140		DCA		OP1		/AND SAVE IT
2682	05316	1377		TAD		(=30		/SET ITERATION COUNTER
2683	05317	3371		DCA		ITRCNT		
2684	05320	1135		TAD		WORD2		
2685	05321	7004		RAL				/INITIALIZE LINK
2686	05322	7210	FPDVLP,	CLA	RAR			/COMPARE SIGNS
2687	05323	1141		TAD		OP2		
2688	05324	7710		SPA	CLA			
2689	05325	5330		JMP		.+3		
2690	05326	1373		TAD		(OPO=ACO/NEGATE OPERAND		
2691	05327	4267		JMS		NEGFAC		
2692	05330	4353		JMS		OADD		/ADD OPERAND AND FAC
2693	05331	1235		TAD		D3		
2694	05332	7004		RAL				
2695	05333	3235		DCA		D3		
2696	05334	1234		TAD		D2		

2697	05335	7004	RAL		
2698	05336	3234	DCA	D2	
2699	05337	4775	JMS I	(AL1	/LEFT SHIFT FAC ONE
2700	05340	2371	ISZ	ITRCNT	/TEST ITERATION COUNT
2701	05341	5322	JMP	FPDVLP	
2702	05342	1140	TAD	OP1	/PUT QUOTIENT INTO FAC
2703	05343	3134	DCA	WORD1	
2704	05344	1234	TAD	D2	
2705	05345	3135	DCA	WORD2	
2706	05346	1235	TAD	D3	
2707	05347	3136	DCA	WORD3	
2708	05350	3137	DCA	ACO	
2709	05351	4236	JMS	ANORM	/NORMALIZE
2710	05352	5710	JMP I	FPDIV	
2711	05353	0000	OADD,	0	/ADD OPERAND TO FAC
2712	05354	7100	CLL		
2713	05355	1143	TAD	OP0	
2714	05356	1137	TAD	ACO	
2715	05357	3137	DCA	ACO	
2716	05360	7004	RAL		
2717	05361	1142	TAD	OP3	
2718	05362	1136	TAD	WORD3	
2719	05363	3136	DCA	WORD3	
2720	05364	7004	RAL		
2721	05365	1141	TAD	OP2	
2722	05366	1135	TAD	WORD2	
2723	05367	3135	DCA	WORD2	
2724	05370	5753	JMP I	OADD	
2725	05371	0000	ITRCNT,	0	

```

2726      / NUMERIC CONVERSION ROUTINE (FINALE)
2727      05373  0004
2728      05374  0137
2729      05375  5431
2730      05376  5414
2731      05377  7750
2732      5400
2733      05400  0000  SMLNUM, 0      /INPUT A NUMBER <= 4095
2734      05401  3025  EXPLUP, DCA     EXPON  /ZERO THE EXPONENT
2735      05402  4531      JMS I  QDIGIT  /GET THE NEXT DIGIT
2736      05403  5600      JMP I  SMLNUM  /NUMBER DONE
2737      05404  3143      DCA     OPO     /SAVE THE DIGIT
2738      05405  1025      TAD     EXPON   /MULT BY 10
2739      05406  7104      CLL RAL
2740      05407  7104      CLL RAL
2741      05410  1025      TAD     EXPON
2742      05411  7104      CLL RAL
2743      05412  1143      TAD     OPO     /ADD IN DIGIT
2744      05413  5201      JMP     EXPLUP  /STORE BACK INTO EXPONENT
2745      05414  0000  AR1,  0         /SHIFT FAC RIGHT 1 BIT
2746      05415  1135      TAD     WORD2
2747      05416  7110      CLL RAR
2748      05417  3135      DCA     WORD2
2749      05420  1136      TAD     WORD3
2750      05421  7010      RAR
2751      05422  3136      DCA     WORD3
2752      05423  1137      TAD     ACO
2753      05424  7010      RAR
2754      05425  3137      DCA     ACO
2755      05426  2134      ISZ     WORD1
2756      05427  5614      JMP I  AR1
2757      05430  5614      JMP I  AR1
2758      05431  0000  AL1,  0         /SHIFT FAC LEFT ONE
2759      05432  1137      TAD     ACO
2760      05433  7104      CLL RAL
2761      05434  3137      DCA     ACO
2762      05435  1136      TAD     WORD3
2763      05436  7004      RAL
2764      05437  3136      DCA     WORD3
2765      05440  1135      TAD     WORD2
2766      05441  7004      RAL
2767      05442  3135      DCA     WORD2
2768      05443  5631      JMP I  AL1
2769      05444  0000  CHKSGN, 0      /CHECK FOR SIGN
2770      05445  1377      TAD     (=55  /IS IT - ?
2771      05446  7450      SNA
2772      05447  2644      ISZ I  CHKSGN  /YES, SET SWITCH
2773      05450  7440      SZA
2774      05451  1376      TAD     (55=53 /IS IT + ?
2775      05452  7640      SZA CLA
2776      05453  4525      JMS I  QBACK1 /RETURN CHAR OTHERWISE
2777      05454  5644      JMP I  CHKSGN

```

```

2778      / STRING LITERAL SCANNER
2779      05455 0000  STRING, 0      /LOOK FOR A STRING
2780      05456 4513      JMS I    QCHECKC /LOOK FOR "
2781      05457 7736      M42,     -42
2782      05460 5655      JMP I    STRING /NONE MEANS NO STRING
2783      05461 2255      ISZ      STRING
2784      05462 3134      DCA      WORD1  /ZERO CHAR COUNT
2785      05463 1375      TAD      (WORD2 /SETUP POINTER
2786      05464 3021      DCA      TEMP
2787      05465 1374      TAD      (-44   /AND MAX SIZE
2788      05466 3022      DCA      TEMP2
2789      05467 4320      SLOOP,   JMS    GCS    /GET HIGH ORDER CHAR
2790      05470 5655      JMP I    STRING /END OF STRING
2791      05471 7106      CLL RTL
2792      05472 7006      RTL
2793      05473 7006      RTL
2794      05474 3421      DCA I    TEMP    /PUT INTO UPPER HALF OF WORD
2795      05475 4320      JMS      GCS     /GET LOWER CHAR
2796      05476 5314      JMP      PUT40  /FILL LAST WORD WITH BLANK
2797      05477 1421      TAD I    TEMP    /COMBINE THEM
2798      05500 3421      DCA I    TEMP
2799      05501 2021      ISZ      TEMP    /BUMP POINTER
2800      05502 2022      ISZ      TEMP2   /TOO BIG YET ?
2801      05503 5267      JMP      SLOOP   /NO, LOOP
2802      05504 4502      JMS I    QGETC   /MAX SIZE STRING, MUST FIND "
2803      05505 5311      JMP      STRGER  /BAD STRING LITERAL
2804      05506 1257      TAD      M42
2805      05507 7650      SNA CLA
2806      05510 5655      JMP I    STRING  /OK
2807      05511 4473      STRGER, JMS I    QERMSG /STRING ERROR
2808      05512 2123      2123
2809      05513 5655      JMP I    STRING
2810      05514 1421      PUT40, TAD I    TEMP    /GET LAST WORD
2811      05515 1373      TAD      (40     /PUT BLANK IN LOW CHAR
2812      05516 3421      DCA I    TEMP    /STORE NEW WORD
2813      05517 5655      JMP I    STRING  /RETURN
2814      05520 0000      GCS,     0        /GET A CHAR FOR STRING
2815      05521 4503      JMS I    QGETCWB /GET A CHAR (INCLUDE BLANKS)
2816      05522 5311      JMP      STRGER  /BAD
2817      05523 1257      TAD      M42     /IS IT "
2818      05524 7440      SZA
2819      05525 5335      JMP      NOTQOT  /NO
2820      05526 4503      JMS I    QGETCWB /IS IT ""
2821      05527 5720      JMP I    GCS     /NO, THAT WAS IT
2822      05530 1257      TAD      M42     /LOOK FOR SECOND "
2823      05531 7650      SNA CLA
2824      05532 5335      JMP      NOTQOT  /"" BECOMES "
2825      05533 4525      JMS I    QBACK1  /PUT IT BACK
2826      05534 5720      JMP I    GCS     /LITERAL IS DONE
2827      05535 1372      NOTQOT, TAD      (42   /RECREATE CHAR
2828      05536 0371      AND      (77   /ELIMINATE EXTRA BITS
2829      05537 2134      ISZ      WORD1  /BUMP STRING COUNT
2830      05540 2320      ISZ      GCS     /FIX RETURN
2831      05541 5720      JMP I    GCS
2832      05542 0000      MODSET, 0      /SET INTERPRETER MODE

```

2833	05543	1055	TAD	MODE	/SUM OF DESIRED AND CURRENT
2834	05544	7700	SMA	CLA	
2835	05545	5742	JMP	I	MODSET /THEY WERE THE SAME
2836	05546	1055	TAD	MODE	/OTHERWISE SWITCH MODES
2837	05547	7640	SZA	CLA	
2838	05550	1370	TAD	(NMODE-SMODE	
2839	05551	1367	TAD	(SMODE	/ENTER NMODE OR MAYBE SMODE
2840	05552	4500	JMS	I	QOUTWRD
2841	05553	7130	CLL	CML	RAR
2842	05554	1055	TAD	MODE	/CHANGE THE SWITCH
2843	05555	3055	DCA	MODE	
2844	05556	5742	JMP	I	MODSET /AND RETURN
2845	05557	3400	XIDIV,	FIDIV/AIDIV	
2846	05560	6740			
2847	05561	7660	WPNT,	-120/-116/-124/-50/0	
2848	05562	7662			
2849	05563	7654			
2850	05564	7730			
2851	05565	0000			


```

2852          / VARIABLE OR FUNCTION REFERENCE SCANNER
2853 05567 7561
2854 05570 7760
2855 05571 0077
2856 05572 0042
2857 05573 0040
2858 05574 7734
2859 05575 0135
2860 05576 0002
2861 05577 7723
2862          PAGE
2863 05600 0000 GETNAM, 0 /LOOK FOR VARIABLE OR FUNCT REFNC
2864 05601 3026 DCA TYPE /ZERO TYPE
2865 05602 4530 JMS I QLETTER /MUST START WITH LETTER
2866 05603 5600 JMP I GETNAM /NO NAME
2867 05604 3134 DCA NAME1
2868 05605 4531 JMS I QDIGIT /<LETTER><DIGIT> ?
2869 05606 5235 JMP TRYFUN /NO, LOOK FOR FUN REF
2870 05607 7001 IAC /INCREMENT DIGIT
2871 05610 3135 LFDOLR, DCA NAME2 /STORE AS NAME2
2872 05611 4502 JMS I QGETC /LOOK FOR $ (STRING)
2873 05612 5233 JMP GOTNAM+2/NOT THERE
2874 05613 1377 TAD (-44
2875 05614 7440 SZ
2876 05615 5224 JMP NOSTRG /NO $ MEANS NO STRING
2877 05616 7130 CLL CML RAR /SET STRING BIT
2878 05617 1026 TAD TYPE
2879 05620 3026 DCA TYPE
2880 05621 4502 JMS I QGETC /LOOK FOR ( (ARRAY)
2881 05622 5233 JMP GOTNAM+2/NAME FINI
2882 05623 1377 TAD (-44 /PRIME THE CHAR
2883 05624 1376 NOSTRG, TAD (44-50 /LOOK FOR ( (ARRAY)
2884 05625 7650 SNA CLA
2885 05626 7132 CLL CML RTR /YES, SET ARRAY BIT
2886 05627 7450 SNA
2887 05630 4525 JMS I QBACK1 /NO, BACKUP 1 CHAR
2888 05631 1026 GOTNAM, TAD TYPE /MODIFY TYPE
2889 05632 3026 DCA TYPE
2890 05633 2200 ISZ GETNAM /BUMP RETURN
2891 05634 5600 JMP I GETNAM
2892 05635 4501 TRYFUN, JMS I QSAVECP /SAVE CHAR POSITION
2893 05636 1134 TAD NAME1 /MOVE FIRST CHAR OVER
2894 05637 7106 CLL RTL
2895 05640 7006 RTL
2896 05641 7006 RTL
2897 05642 3135 DCA NAME2
2898 05643 4530 JMS I QLETTER /LOOK FOR SECOND LETTER
2899 05644 5210 JMP LFDOLR /NONE THERE, LOOK FOR $
2900 05645 1135 TAD NAME2 /COMBINE WITH FIRST LETTER
2901 05646 3135 DCA NAME2
2902 05647 4530 JMS I QLETTER /LOOK FOR THIRD LETTER
2903 05650 5302 JMP NOFNAM /NOT A FUNCTION NAME
2904 05651 3136 DCA NAME3 /PUT INTO NAME
2905 05652 1135 TAD NAME2 /IS IT A USER FUNCT ?
2906 05653 1375 TAD (-616 /FN

```

```

2907 05654 7650      SNA CLA
2908 05655 5304      JMP USRFUN /YES
2909 05656 1374      TAD (FUNS=1 /NO, CHECK VALIDITY OF NAME
2910 05657 3010      DCA X10
2911 05660 1410      FUNSRC, TAD I X10 /GET NEXT FUN NAME
2912 05661 7450      SNA
2913 05662 5302      JMP NOFNAM /END OF LIST, INVALID NAME
2914 05663 1135      TAD NAME2 /COMPARE FIRST 2 CHARS
2915 05664 7640      SZA CLA
2916 05665 5277      JMP NOMATC /THEY DON'T MATCH
2917 05666 1410      TAD I X10 /COMPARE 3RD CHAR
2918 05667 1136      TAD NAME3
2919 05670 7640      SZA CLA
2920 05671 5300      JMP NOMATC+1 /DON'T MATCH
2921 05672 1410      TAD I X10 /GET FUNCTION CODE
2922 05673 3027      FUNOK, DCA SYMBOL /SAVE IT AS SYMBOL VALU
2923 05674 1373      TAD (1000 /SET FUNCTION BIT
2924 05675 3026      DCA TYPE
2925 05676 5210      JMP LFDOLR /LOOK FOR QS) Q()
2926 05677 2010      NOMATC, ISZ X10 /SKIP THIRD CHAR
2927 05700 2010      ISZ X10 /SKIP FUNCTION NUMBER
2928 05701 5260      JMP FUNSRC /KEEP LOOKING
2929 05702 4504      NOFNAM, JMS I GRE8TCP /RESTORE CHAR POS
2930 05703 5210      JMP LFDOLR /LOOK FOR QS) Q()
2931 05704 1136      USRFUN, TAD NAME3 /GENERATE FUN NUMBER
2932 05705 5273      JMP FUNOK

```

```

2933      / ERROR MESSAGE REPORTER
2934 05706 0000 ERMSG, 0 /PRINT ERROR MESSAGE
2935 05707 7200 CLA
2936 05710 6201 CDF
2937 05711 1706 TAD I ERMSG /GET CODE
2938 05712 7112 CLL RTR /PRINT FIRST CHAR
2939 05713 7012 RTR
2940 05714 7012 RTR
2941 05715 4772' JMS TTY
2942 05716 1706 TAD I ERMSG /PRINT SECOND CHAR
2943 05717 4772' JMS TTY
2944 05720 2306 ISZ ERMSG /FIX RETURN ADDR
2945 05721 1350 TAD SPACE /PRINT SPACE
2946 05722 4772' JMS TTY
2947 05723 3772' DCA TTY /USE TTY AS A SWITCH
2948 05724 1053 TAD LINEH /PRINT HIGH ORDER
2949 05725 4335 JMS PSN
2950 05726 1054 TAD LINEL /THEN LOW ORDER
2951 05727 4335 JMS PSN /LINE NUMBER NATCH 1)
2952 05730 1371 TAD (215 /PRINT CARRIAGE RETURN
2953 05731 4770' JMS TTX
2954 05732 1367 TAD (212 /PRINT LINE FEED
2955 05733 4770' JMS TTX
2956 05734 5706 JMP I ERMSG /RETURN
2957 05735 0000 PSN, 0 /PRINT 3 DIGITS DECIMAL
2958 05736 3135 DCA WORD2
2959 05737 7146 CLL CMA RTL /-3
2960 05740 3021 DCA TEMP
2961 05741 1135 PRNTSN, TAD WORD2 /GET NEXT DIGIT
2962 05742 7106 CLL RTL /INTO THE LOW ORDER
2963 05743 7006 RTL /THREE BITS AND THE LINK
2964 05744 3135 DCA WORD2 /SAVE SHIFTED NUMBER
2965 05745 1135 TAD WORD2 /NOW DO LAST SHIFT
2966 05746 7004 RAL
2967 05747 0366 AND (17 /ONLY FOUR BITS
2968 05750 7440 SPACE, SZA
2969 05751 5355 JMP NOZERO /NOT A ZERO
2970 05752 1772' TAD TTY /ANY DIGITS YET ?
2971 05753 7650 SNA CLA
2972 05754 5357 JMP LEAD0 /NO, ITS A LEADING ZERO
2973 05755 1365 NOZERO, TAD (60 /MAKE IT ASCII
2974 05756 4772' JMS TTY /PRINT DIGIT
2975 05757 2021 LEAD0, ISZ TEMP /BUMP COUNT
2976 05760 5341 JMP PRNTSN /MORE DIGIT(S)
2977 05761 5735 JMP I PSN
2978 05762 1000 XMUL, FMPYJAFMPY
2979 05763 6540

```

2980 / EXPONENT TABLE

2981	05765	0060	
2982	05766	0017	
2983	05767	0212	
2984	05770	3740	
2985	05771	0215	
2986	05772	3730	
2987	05773	1000	
2988	05774	6070	
2989	05775	7162	
2990	05776	7774	
2991	05777	7734	
2992		6000	
2993	06000	0004	PETABL, PAGE 0004/2400/0000/0000
2994	06001	2400	
2995	06002	0000	
2996	06003	0000	
2997	06004	0007	0007/3100/0000/0000
2998	06005	3100	
2999	06006	0000	
3000	06007	0000	
3001	06010	0016	0016/2342/0000/0000
3002	06011	2342	
3003	06012	0000	
3004	06013	0000	
3005	06014	0033	0033/2765/7020/0000
3006	06015	2765	
3007	06016	7020	
3008	06017	0000	
3009	06020	0066	0066/2160/6744/6770
3010	06021	2160	
3011	06022	6744	
3012	06023	6770	
3013	06024	0153	0153/2356/1326/6501
3014	06025	2356	
3015	06026	1326	
3016	06027	6501	
3017	06030	0325	0325/3023/6017/5120
3018	06031	3023	
3019	06032	6017	
3020	06033	5120	
3021	06034	0652	0652/2235/6443/7114
3022	06035	2235	
3023	06036	6443	
3024	06037	7114	
3025	06040	1523	1523/2523/7565/7735
3026	06041	2523	
3027	06042	7565	
3028	06043	7735	
3029	06044	3245	3245/3430/6320/2565
3030	06045	3430	
3031	06046	6320	
3032	06047	2565	

3033			/ OPERATOR TABLE
3034	06050	2557	OPR8RS, PLUS/-53
3035	06051	7725	
3036	06052	2755	MINUS/-55
3037	06053	7723	
3038	06054	0356	STAR/-52
3039	06055	7726	
3040	06056	2761	SLASH/-57
3041	06057	7721	
3042	06060	0362	UPAROW/-136
3043	06061	7642	
3044	06062	1557	AMPSND/-46
3045	06063	7732	
3046	06064	0000	0
3047	06065	4000	SASIGN, 4000/XSTOR
3048	06066	4301	
3049	06067	0000	ASSIGN, 0/XSTOR
3050	06070	4301	

			/ FUNCTION NAME TABLE (INTERNAL FUNCTIONS)
3051			FUNS, -0102)-23)FUNC3
3052	06071	7676	
3053	06072	7755	
3054	06073	7400	
3055	06074	7655	-0123)-03)FUNC2
3056	06075	7775	
3057	06076	7417	
3058	06077	7654	-0124)-16)FUNC1
3059	06100	7762	
3060	06101	7416	
3061	06102	7470	-0310)-22)FUNC2+20
3062	06103	7756	
3063	06104	7437	
3064	06105	7461	-0317)-23)FUNC1+20
3065	06106	7755	
3066	06107	7436	
3067	06110	7377	-0401)-24)FUNC2+40
3068	06111	7754	
3069	06112	7457	
3070	06113	7250	-0530)-20)FUNC1+40
3071	06114	7760	
3072	06115	7456	
3073	06116	6662	-1116)-24)FUNC1+100
3074	06117	7754	
3075	06120	7516	
3076	06121	6373	-1405)-16)FUNC2+60
3077	06122	7762	
3078	06123	7477	
3079	06124	6361	-1417)-07)FUNC1+120
3080	06125	7771	
3081	06126	7536	
3082	06127	5761	-2017)-23)FUNC2+100
3083	06130	7755	
3084	06131	7517	
3085	06132	5562	-2216)-04)FUNC1+200
3086	06133	7774	
3087	06134	7616	
3088	06135	5473	-2305)-07)FUNC2+120
3089	06136	7771	
3090	06137	7537	
3091	06140	5471	-2307)-16)FUNC1+140
3092	06141	7762	
3093	06142	7556	
3094	06143	5467	-2311)-16)FUNC1+160
3095	06144	7762	
3096	06145	7576	
3097	06146	5457	-2321)-22)FUNC1+220
3098	06147	7756	
3099	06150	7636	
3100	06151	5454	-2324)-22)FUNC2+140
3101	06152	7756	
3102	06153	7557	
3103	06154	5177	-2601)-14)FUNC2+160
3104	06155	7764	
3105	06156	7577	

3106	06157	5356	-2422)-03)FUNC2+220
3107	06160	7775	
3108	06161	7637	
3109	06162	0000	ENDFNS, 0)0)FUNC4 /SPACE FOR NEW FUNCTIONS
3110	06163	0000	
3111	06164	7415	
3112	06165	0000	0)0)FUNC4+20
3113	06166	0000	
3114	06167	7435	
3115	06170	0000	0)0)FUNC4+40
3116	06171	0000	
3117	06172	7455	
3118	06173	0000	0)0)FUNC4+60
3119	06174	0000	
3120	06175	7475	
3121	06176	0000	0)0)FUNC4+100
3122	06177	0000	
3123	06200	7515	
3124	06201	0000	0)0)FUNC4+120
3125	06202	0000	
3126	06203	7535	
3127	06204	0000	0)0)FUNC4+140
3128	06205	0000	
3129	06206	7555	
3130	06207	0000	0)0)FUNC4+160
3131	06210	0000	
3132	06211	7575	
3133	06212	0000	0)0)FUNC4+200
3134	06213	0000	
3135	06214	7615	
3136	06215	0000	0)0)FUNC4+220
3137	06216	0000	
3138	06217	7635	
3139	06220	0000	0)0)FUNC4+240
3140	06221	0000	
3141	06222	7655	
3142	06223	0000	0)0)FUNC4+260
3143	06224	0000	
3144	06225	7675	
3145	06226	0000	0)0)FUNC4+300
3146	06227	0000	
3147	06230	7715	
3148	06231	0000	0)0)FUNC4+320
3149	06232	0000	
3150	06233	7735	
3151	06234	0000	0)0)FUNC4+340
3152	06235	0000	
3153	06236	7755	
3154	06237	0000	0)0)FUNC4+360 /SIXTEEN OF THEM
3155	06240	0000	
3156	06241	7775	
3157	06242	0000	0

3158			/ KEYWORD LIST
3159	06243	7664	KEYWRD, -114/-105/-124/LET
3160	06244	7673	
3161	06245	7654	
3162	06246	0316	
3163	06247	7667	-111/-106/-105/-116/-104/IFEND
3164	06250	7672	
3165	06251	7673	
3166	06252	7662	
3167	06253	7674	
3168	06254	2103	
3169	06255	7667	-111/-106/IF
3170	06256	7672	
3171	06257	2000	
3172	06260	7672	-106/-117/-122/FOR
3173	06261	7661	
3174	06262	7656	
3175	06263	1600	
3176	06264	7662	-116/-105/-130/-124/NEXTX
3177	06265	7673	
3178	06266	7650	
3179	06267	7654	
3180	06270	0704	
3181	06271	7671	WGOTO, -107/-117
3182	06272	7661	
3183	06273	7654	WTD, -124/-117/GOTO
3184	06274	7661	
3185	06275	2147	
3186	06276	7671	-107/-117/-123/-125/-102/GOSUB
3187	06277	7661	
3188	06300	7655	
3189	06301	7653	
3190	06302	7676	
3191	06303	2154	
3192	06304	7667	-111/-116/-120/-125/-124/INPUT
3193	06305	7662	
3194	06306	7660	
3195	06307	7653	
3196	06310	7654	
3197	06311	0400	
3198	06312	7660	-120/-122/-111/-116/-124/PRINT
3199	06313	7656	
3200	06314	7667	
3201	06315	7662	
3202	06316	7654	
3203	06317	0443	
3204	06320	7674	-104/-111/-115/DIM
3205	06321	7667	
3206	06322	7663	
3207	06323	0600	
3208	06324	7674	-104/-101/-124/-101/DATA
3209	06325	7677	
3210	06326	7654	
3211	06327	7677	
3212	06330	1431	

3213	06331	7674	-104/-105/-106/DEF
3214	06332	7673	
3215	06333	7672	
3216	06334	1200	
3217	06335	7672	-106/-111/-114/-105/FILE
3218	06336	7667	
3219	06337	7664	
3220	06340	7673	
3221	06341	2522	
3222	06342	7656	-122/-105/-101/-104/READY
3223	06343	7673	
3224	06344	7677	
3225	06345	7674	
3226	06346	1526	
3227	06347	7656	-122/-105/-115/REMARK
3228	06350	7673	
3229	06351	7663	
3230	06352	0205	
3231	06353	7656	-122/-105/-123/-124/-117/-122/-105/RESTOR
3232	06354	7673	
3233	06355	7655	
3234	06356	7654	
3235	06357	7661	
3236	06360	7656	
3237	06361	7673	
3238	06362	0542	
3239	06363	7656	-122/-105/-124/-125/-122/-116/RETURN
3240	06364	7673	
3241	06365	7654	
3242	06366	7653	
3243	06367	7656	
3244	06370	7662	
3245	06371	3561	
3246	06372	7655	-123/-124/-117/-120/STOPX
3247	06373	7654	
3248	06374	7661	
3249	06375	7660	
3250	06376	3565	
3251	06377	7656	-122/-101/-116/-104/-117/-115/-111/-132/-105/RANDOM
3252	06400	7677	
3253	06401	7662	
3254	06402	7674	
3255	06403	7661	
3256	06404	7663	
3257	06405	7667	
3258	06406	7646	
3259	06407	7673	
3260	06410	3564	
3261	06411	7675	-103/-114/-117/-123/-105/CLOSE
3262	06412	7664	
3263	06413	7661	
3264	06414	7655	
3265	06415	7673	
3266	06416	3342	
3267	06417	7675	-103/-110/-101/-111/-116/CHAIN

3268	06420	7670
3269	06421	7677
3270	06422	7667
3271	06423	7662
3272	06424	3750
3273	06425	7653
3274	06426	7674
3275	06427	7673
3276	06430	7672
3277	06431	1000
3278	06432	7653
3279	06433	7655
3280	06434	7673
3281	06435	1745
3282	06436	7673
3283	06437	7662
3284	06440	7674
3285	06441	0343
3286	06442	0000

-125/-104/-105/-106/UDEF

-125/-123/-105/USEX

-105/-116/-104/END

0

```

3287      / OS=8 OUTPUT ROUTINE
3288      06443 0000  OWTEMP, 0
3289      06444 0000  OUPTR, OUBUF
3290      06445 7377  OCOUNT, -401
3291      06446 0000  OUTWRD, 0
3292      06447 3243          DCA      OWTEMP  /OUTPUT ROUTINE
3293      06450 2045          ISZ      LOCTRL  /SAVE WORD
3294      06451 7410          SKP      LOCTRL  /INCREMENT PSEUDO CODE
3295      06452 2044          ISZ      LOCTRH  /LOCATION COUNTER
3296      06453 7000          NOP      LOCTRH  /BOTH HALVES
3297      06454 2245          ISZ      OCOUNT /IT'LL NEVER HAPPEN
3298      06455 5263          JMP      NOWRIT  /TEST FOR BUFFER FULL
3299      06456 4271          JMS      OUDUMP  /STILL SOME ROOM
3300      06457 1274          TAD      OUBLOK-1 /DUMP THE BUFFER
3301      06460 3244          DCA      OUPTR  /RESET BUFFER PARAMETERS
3302      06461 1377          TAD      (-400
3303      06462 3245          DCA      OCOUNT
3304      06463 1243  NOWRIT, TAD      OWTEMP  /PUT WORD
3305      06464 6211          CDF      10
3306      06465 3644          DCA I  OUPTR  /INTO BUFFER
3307      06466 6201          CDF
3308      06467 2244          ISZ      OUPTR  /MOVE POINTER
3309      06470 5646          JMP I  OUTWRD
3310      06471 0000  OUDUMP, 0
3311      06472 4776          JMS I  (7607  /DUMP OUT BUFFER
3312      06473 4210          4210  /CALL OUTPUT HANDLER
3313      06474 0000          OUBUF
3314      06475 0000  OUBLOK, 0
3315      06476 5302          JMP      QUERR
3316      06477 2275          ISZ      OUBLOK  /INCREMENT BLOCK NUMBER
3317      06500 2306          ISZ      OUSIZE  /CHECK FOR HOLE FULL
3318      06501 5671          JMP I  OUDUMP
3319      06502 4473  QUERR, JMS I  QERMSG  /OUTPUT FILE ERROR
3320      06503 1706          1706
3321      06504 5406          JMP I  XABORT  /ABORT COMPILATION
3322      06505 0000  ODEVH, 0
3323      06506 0000  OUSIZE, 0
3324      06507 4315  AMPRTN, JMS      LOD1ST  /LOAD OP1$
3325      06510 1561          AMPSND+2  /CONC OP2$
3326      06511 4313  SCRTN, JMS      LOD1ST  /LOAD OP1$
3327      06512 6570          SCOMPR+1  /CONC OP2$
3328      06513 0000  LOD1ST, 0
3329      06514 4524          JMS I  QSAVAC  /HANDLE ONE WAY INSTRUCTIONS
3330      06515 7777          -1  /STORE 2ND ARG IF IN SAC
3331      06516 7240          CLA CMA
3332      06517 1014          TAD      OSTACK  /GET TYPE OF 2ND ARG
3333      06520 3021          DCA      TEMP
3334      06521 7132          CLL CML RTR  /IS IT SUBSCRIPTED ?
3335      06522 0421          AND I  TEMP
3336      06523 7650          SNA CLA
3337      06524 5331          JMP      SKIP2  /NO, ENTRY IS ONLY 2 WORDS
3338      06525 1421          TAD I  TEMP  /GET NUMBER OF DIMS
3339      06526 0367          AND      SCOMPR /LITERAL 3
3340      06527 7104          CLL RAL  /DOUBLE IT
3341      06530 7041          CIA

```

```

3342 06531 1375 SKIP2, TAD      (=2      /FIND SIZE OF 2ND ARG
3343 06532 3354      DCA      OP2SIZ /AND SAVE IT
3344 06533 1014      TAD      OSTACK /BACK UP STACK
3345 06534 1354      TAD      OP2SIZ
3346 06535 3014      DCA      OSTACK
3347 06536 1014      TAD      OSTACK /AND SAVE THIS ADDR
3348 06537 3012      DCA      X12
3349 06540 4520      JMS I   QLOAD  /LOAD ARG 1
3350 06541 7130      CLL CML  RAR    /GET TYPE BIT
3351 06542 0056      AND     TYPE1 /PUT BACK ARG1
3352 06543 1133      TAD      Q400
3353 06544 3414      DCA I   OSTACK
3354 06545 3414      DCA I   OSTACK
3355 06546 1412      TAD I   X12    /PUT BACK ARG 2
3356 06547 3414      DCA I   OSTACK
3357 06550 2354      ISZ     OP2SIZ
3358 06551 5346      JMP     ,=3
3359 06552 1713      TAD I   LOD1ST /GET OPERATOR FINISH
3360 06553 5774      JMP     OUTOPR+1 /GO FINISH CODE
3361 06554 0000      OP2SIZ, 0      /SACRED COUNT WORD
3362 06555 0000      CHECKC, 0      /CHAR CHECKER
3363 06556 4502      JMS I   QGETC  /GET A CHARACTER
3364 06557 5365      JMP     ,+6    /FAILED
3365 06560 1755      TAD I   CHECKC /COMPARE
3366 06561 7450      SNA
3367 06562 2355      ISZ     CHECKC /MATCHES, SKIP TWO
3368 06563 7640      SZA CLA
3369 06564 4525      JMS I   QBACK1 /NO MATCH, REPLACE
3370 06565 2355      ISZ     CHECKC /ALWAYS SKIP AT LEAST 1
3371 06566 5755      JMP I   CHECKC
3372 06567 0003      SCOMPR, 3 /SCRTN=3 /4000 /XSCOMP /XSCOMP
3373 06570 6506
3374 06571 4000
3375 06572 6734
3376 06573 6734

```

```

3377      / OS-8 FILE INPUT ROUTINE
3378      06574 3401
3379      06575 7776
3380      06576 7607
3381      06577 7400
3382      6600
3383      06600 0000  ICHAR, 0      /READ CHAR FROM INPUT FILE
3384      06601 2255      ISZ      INJMP  /BUMP THREE WAY UNPACK SWITCH
3385      06602 2330      ISZ      INCHCT
3386      06603 5255      INJMPP, JMP      INJMP
3387      06604 1327      TAD      INEOF  /LAST READ YEILD END OF FILE ?
3388      06605 7640      SZA  CLA
3389      06606 5233      JMP      ENDFIL /YES
3390      06607 1332      INGBUF, TAD      INCTR  /BUMP RECORD COUNTER
3391      06610 7101      CLL  IAC
3392      06611 7420      SNL
3393      06612 3332      DCA      INCTR  /RESTORE IF IT HASN'T OVERFLOWED
3394      06613 7430      SZL
3395      06614 2327      ISZ      INEOF  /SET END OF FILE SWITCH
3396      06615 4731      JMS  I      INHNDL /OO THE READ
3397      06616 0200      0200      /ONE BLOCK TO FIELD 0
3398      06617 7200      INBUF, INBUF
3399      06620 0000      INREC, 0
3400      06621 5232      JMP      INERR  /HANDLER ERROR
3401      06622 2220      INBREC, ISZ      INREC  /BUMP RECORD NUMBER
3402      06623 1377      TAD      (-601  /SET CHAR COUNT
3403      06624 3330      DCA      INCHCT
3404      06625 1203      TAD      INJMPP /RESET THREE WAY JUMP SWITCH
3405      06626 3255      DCA      INJMP
3406      06627 1217      TAD      INBUF  /RESET BUFFER POINTER
3407      06630 3333      DCA      INPTR
3408      06631 5201      JMP      ICHAR+1 /GO AGAIN
3409      06632 7200      INERR, CLA
3410      06633 4473      ENDFIL, JMS  I      QERMSG /INPUT FILE ERROR
3411      06634 1505      1505
3412      06635 1376      ABORT, TAD      (4207 /RESTORE ^C LOCZTIONS
3413      06636 3775      DCA      7600
3414      06637 1374      TAD      (6213
3415      06640 3773      DCA      7605
3416      06641 6211      CDF      10
3417      06642 1772      TAD      INFO  /GET START OF BASIC.SV
3418      06643 6201      CDF
3419      06644 7450      SNA
3420      06645 5773      JMP      7605  /T^WERE RUNNED
3421      06646 3252      DCA      EDTBLK /SAVE MAGICAL BLOCK NUMBER
3422      06647 4771      JMS      7607  /USE SYS HANDLER
3423      06650 1600      EDTSIZ
3424      06651 0000      0      /INTO ZERO
3425      06652 0000      EDTBLK, 0      /FROM HERE
3426      06653 7402      HLT
3427      06654 5770      JMP      EDTBGN /GO RESTART EDITOR
3428      06655 7402      INJMP, HLT      /3 WAY CHAR UNPACK JUMP
3429      06656 5277      JMP      ICHAR1
3430      06657 5273      JMP      ICHAR2
3431      06660 1203      ICHAR3, TAD      INJMPP /RESET JUMP SWITCH

```

```

3432 06661 3255      DCA      INJMP
3433 06662 1733      TAD I     INPTR
3434 06663 0367      AND      (7400      /COMBINE THE HIGH ORDER BITS
3435 06664 7112      CLL RTR      /OF THE TWO WORDS
3436 06665 7012      RTR
3437 06666 1326      TAD      INTMP      /TO FORM THE THIRD CHAR
3438 06667 7012      RTR
3439 06670 7012      RTR
3440 06671 2333      ISZ      INPTR      /BUMP WORD POINTER
3441 06672 5300      JMP      ICHAR1+1 /DO SOME COMMON STUFF
3442 06673 1733      ICHAR2, TAD I     INPTR      /SAVE THE HIGH ORDER BITS
3443 06674 0367      AND      (7400
3444 06675 3326      DCA      INTMP      /FOR THE THIRD CHAR
3445 06676 2333      ISZ      INPTR      /GO TO THE SECOND WORD
3446 06677 1733      ICHAR1, TAD I     INPTR      /GET THE LOW 7 BITS
3447 06700 0366      AND      (177      /AND I MEAN ONLY 7 !!
3448 06701 1365      TAD      (-134      /CHECK FOR \ (STMT SEPARATOR)
3449 06702 7450      SNA
3450 06703 5600      JMP I      ICHAR      /TREAT LIKE CR
3451 06704 1364      TAD      (134-32 /IS IT *Z (END OF FILE)
3452 06705 7450      SNA
3453 06706 5233      JMP      ENDFIL      /YES, ITS END OF FILE
3454 06707 1363      TAD      (32-12
3455 06710 7450      SNA
3456 06711 5201      JMP      ICHAR+1 /IGNORE LINE FEEDS
3457 06712 7001      IAC      /TABS => BLANKS
3458 06713 7450      SNA
3459 06714 1362      TAD      (40-11
3460 06715 1361      TAD      (11-15
3461 06716 7450      SNA
3462 06717 5600      JMP I      ICHAR      /RETURN ON CARRIAGE RETURN
3463 06720 7001      IAC
3464 06721 7450      SNA
3465 06722 5201      JMP      ICHAR+1 /IGNORE FORM FEEDS
3466 06723 1360      TAD      (14      /FIX CHAR
3467 06724 2200      ISZ      ICHAR
3468 06725 5600      JMP I      ICHAR      /RETURN TO THE CALLING WORLD
3469 06726 0000      INTMP, 0
3470 06727 0000      INEOF, 0
3471 06730 7777      INCHCT, =1
3472 06731 0000      INHNDL, 0      /ENTRY ADDR GOES HERE
3473 06732 0000      INCTR, 0
3474 06733 0000      INPTR, 0
3475 06734 0400      XSCOMP, SCOMP;SACOMP
3476 06735 6440
3477 06736 0000      CHKWD, 0      /WORD CHECKER
3478 06737 1736      TAD I      CHKWD      /GET POINTER
3479 06740 2336      ISZ      CHKWD
3480 06741 3355      DCA      CWTEMP      /SAVE POINTER
3481 06742 1755      WLOOP, TAD I      CWTEMP      /GET NEXT CHAR
3482 06743 7500      SMA
3483 06744 2336      ISZ      CHKWD      /IF NON NEG, FIX RETURN
3484 06745 7710      SPA CLA
3485 06746 4902      JMS I      QGETC      /GET CHAR
3486 06747 5736      JMP I      CHKWD      /RETURN

```

3487	06750	1755	TAD I	CWTEMP	/COMPARE
3488	06751	2355	ISZ	CWTEMP	/INCR POINTER
3489	06752	7650	SNA CLA		
3490	06753	5342	JMP	WDLOOP	/MORE
3491	06754	5736	JMP I	CHKWD	/FAILED
3492	06755	0000	CWTEMP,	0	
3493	06756	1400	XDIV,	FDIV/AFDIV	
3494	06757	6600			

3495 / INITIALIZATION CODE

```

3496 06760 0014
3497 06761 7774
3498 06762 0027
3499 06763 0020
3500 06764 0102
3501 06765 7644
3502 06766 0177
3503 06767 7400
3504 06770 3012
3505 06771 7607
3506 06772 7604
3507 06773 7605
3508 06774 6213
3509 06775 7600
3510 06776 4207
3511 06777 7177
3512 7000
3513 07000 5777' *LINE
3514 07001 6211 START, JMP RUNNED /DO LOOKUPS, AND FIND TEMPFILE
3515 07002 1776 CHAINED, CDF 10
3516 07003 6201 TAD I (7644 /WAS IT A CHAIN FROM BRTS ?
3517 07004 0375 CDF
3518 07005 7650 AND (100
3519 07006 5213 SNA CLA
3520 07007 6212 JMP CHEDIT /NO, FROM THE EDITOR
3521 07010 4774 CIF 10 /CHAIN FROM BRTS, RESET
3522 07011 0013 JMS I (200 /TO FORGET DSK1 HANDLER
3523 07012 5773' 13
3524 07013 1372 CREDIT, TAD STRT3 /NOW GO OPEN TEMP FILE
3525 07014 3010 (INFO+7 /PICK UP SOME STUFF
3526 07015 6211 DCA X10
3527 07016 1410 CDF 10 /FROM THE INFO BLOCK
3528 07017 3046 TAD I X10 /START OF TEMP FILE
3529 07020 1410 DCA BLOCK
3530 07021 6201 TAD I X10 /SIZE OF HOLE
3531 07022 3771' CDF
3532 07023 1046 DCA OUSIZE
3533 07024 3770' TAD BLOCK
3534 07025 6211 DCA OUBLOK
3535 07026 1410 CDF 10
3536 07027 6201 TAD I X10 /ENTRY ADDR OF HANDLER
3537 07030 3767' CDF
3538 07031 5247 DCA INHNDL
3539 07032 6211 GETDEV, JMP STRT2
3540 07033 1766' CDF 10
3541 07034 6201 TAD 7617 /GET DEVICE NUM FOR INPUT FILE
3542 07035 6212 CIF 10
3543 07036 4774 JMS I (200 /GO FETCH THE DEVICE
3544 07037 0001 1
3545 07040 4401 INDEVH+1 /2 PAGE HANDLER IS OK
3546 07041 5320 JMP NG /ERROR
3547 07042 1240 TAD .#2 /GET HANDLER ADDRESS
3548 07043 3767' DCA INHNDL /SAVE IT
3549 07044 6212 CIF 10

```



```

3550 07045 4774 JMS I (200 /RESET SYSTEM TABLES
3551 07046 0013 13 /DELETING TENTATIVE FILES
3552 07047 6211 STRT2, CDF 10
3553 07050 1766' TAD 7617 /SET UP INPUT FILE PARAMS
3554 07051 6201 CDF
3555 07052 0365 AND (7760 /GET SIZE
3556 07053 1364 TAD (17
3557 07054 7132 CLL CML RTR
3558 07055 7012 RTR
3559 07056 3765' DCA INCTR
3560 07057 6211 CDF 10
3561 07060 1762' TAD 7620 /GET BLOCK NUMBER
3562 07061 6201 CDF
3563 07062 3761' DCA INREC
3564 07063 6211 CDF 10
3565 07064 1760' TAD INFO+3 /GET START OF BRTS.SV (+1)
3566 07065 3050 DCA BRTS
3567 07066 1757' TAD INFO /GET START OF BASIC.SV (+1)
3568 07067 3052 DCA ABORTX /BOTH FOR BLOAD
3569 07070 1756' TAD INFO+2 /GET START OF BLOAD.SV
3570 07071 6201 CDF
3571 07072 3755' DCA LDRBLK /FOR CHAIN TO BLOAD
3572 07073 6046 TLS /SET TTY FLAG
3573 07074 1354 INITST, TAD (VARST-1/INITIALIZE ST AREA
3574 07075 3012 DCA X12
3575 07076 1353 TAD (-436-436-436
3576 07077 3011 DCA X11 /SIZE OF NUM AND STRING TABLES
3577 07100 6211 CDF 10
3578 07101 7130 CLL CML RAR /SET TO 4000
3579 07102 3412 DCA I X12
3580 07103 2011 ISZ X11
3581 07104 5301 JMP .-3
3582 07105 1352 TAD (-440 /NOW ARRAY TABLES
3583 07106 3011 DCA X11 /AND BUCKETS
3584 07107 3412 DCA I X12
3585 07110 2011 ISZ X11 /SET THEM TO ZERO
3586 07111 5307 JMP .-2
3587 07112 6201 CDF
3588 07113 1325 TAD JABORT /MODIFY ^C LOCATIONS
3589 07114 3751' DCA 7600
3590 07115 1325 TAD JABORT
3591 07116 3750' DCA 7605
3592 07117 5747' JMP CORE /GET CORE SIZE
3593 07120 6046 NG, TLS
3594 07121 4473 JMS I QERMSG /SUPER ERROR
3595 07122 2331 2331
3596 07123 6041 TSF
3597 07124 5323 JMP .-1
3598 07125 5406 JABORT, JMP I XABORT /ABORT COMPILATION

```

3599	07147	7200			
3600	07150	7605			
3601	07151	7600			
3602	07152	7340			
3603	07153	6246			
3604	07154	0377			
3605	07155	0351			
3606	07156	7606			
3607	07157	7604			
3608	07160	7607			
3609	07161	6620			
3610	07162	7620			
3611	07163	6732			
3612	07164	0017			
3613	07165	7760			
3614	07166	7617			
3615	07167	6731			
3616	07170	6475			
3617	07171	6506			
3618	07172	7613			
3619	07173	4651			
3620	07174	0200			
3621	07175	0100			
3622	07176	7644			
3623	07177	4600			
3624		7200			
3625			CORE,	*INBUF	
3626					/CORE SIZE ROUTINE FROM
3627	07200	6201	COR0,	CDF	/OS8 SOFTWARE SUPPORT MANUAL
3628	07201	1325		TAD	CORSIZ
3629	07202	7006		RTL	
3630	07203	7004		RAL	
3631	07204	0215		AND	COR70
3632	07205	1230		TAD	COREX
3633	07206	3207		DCA	.*1
3634	07207	6201	COR1,	CDF	
3635	07210	1723		TAD I	CORLOC
3636	07211	7000	COR2,	NOP	
3637	07212	3207		DCA	COR1
3638	07213	1211		TAD	COR2
3639	07214	3723		DCA I	CORLOC
3640	07215	0070	COR70,	70	
3641	07216	1723		TAD I	CORLOC
3642	07217	7400	CORX,	7400	
3643	07220	1217		TAD	CORX
3644	07221	1324		TAD	CORV
3645	07222	7640		SZA CLA	
3646	07223	5230		JMP	COREX
3647	07224	1207		TAD	COR1
3648	07225	3723		DCA I	CORLOC
3649	07226	2325		ISZ	CORSIZ
3650	07227	5200		JMP	COR0
3651	07230	6201	COREX,	CDF	
3652	07231	7240		CLA CMA	
3653	07232	1325		TAD	CORSIZ

/HI FIELD IS #FIELDS-1

```

3654 07233 3047 DCA HIFLD
3655 07234 1047 TAD HIFLD
3656 07235 7041 CIA
3657 07236 3777' DCA NFLOS
3658 07237 7040 CMA /HOW MANY FIELDS ?
3659 07240 1047 TAD HIFLD /MUST THIS BASIC USE ?
3660 07241 7640 SZA CLA /((SOUNDS LIKE A LINE BY DYLAN))
3661 07242 5255 JMP GENER
3662 07243 1376 TAD (PATCH1+38177+5200 217
3663 07244 3775' DCA PATCH1 /ONLY 8K, DON'T USE CDF'S = 220
3664 07245 1374 TAD (PATCH2+118177+5200 001
3665 07246 3773' DCA PATCH2 = 2278
3666 07247 1372 TAD (PATCH3+48177+5200 001
3667 07250 3771' DCA PATCH3 = 1077
3668 07251 1370 TAD (PATCH4+38177+5200 001
3669 07252 3767' DCA PATCH4 = 1520
3670 07253 1366 TAD (7000 001
3671 07254 3765' DCA PATCH5 = 2222
3672 07255 4764' GENER, JMS GENTMP /GENERATE TEMP 0
3673 07256 4764' JMS GENTMP /GENERATE TEMP 1
3674 07257 4764' JMS GENTMP /GENERATE TEMP 2
3675 07260 7201 CLA IAC /GENERATE STRING TEMP 0
3676 07261 4764' JMS GENTMP
3677 07262 7201 CLA IAC
3678 07263 3134 DCA WORD1 /GENERATE LITERAL 1.0
3679 07264 7132 CLL CML RTR
3680 07265 3135 DCA WORD2
3681 07266 4517 JMS I QLUKUP2 /ENTER INTO ST
3682 07267 2562 LITRL
3683 07270 7775 =3
3684 07271 4763' JMS NEWVAR
3685 07272 1362 TAD (FNINIT /SET UP FUNCTIONS
3686 07273 3322 DCA FDPTR
3687 07274 1361 FOLLOOP, TAD (WORD1-1
3688 07275 3012 DCA X12
3689 07276 1722 TAD I FDPTR /GET FIRST WORD
3690 07277 2322 ISZ FDPTR
3691 07300 7450 SNA
3692 07301 5510 JMP I QREMARK /DONE, START COMPILER
3693 07302 3412 DCA I X12 /SAVE IN WORD1
3694 07303 7146 CLL CMA RTL /GET LOOKUP COUNT
3695 07304 1722 TAD I FDPTR
3696 07305 3320 DCA FUNSIZ
3697 07306 1320 TAD FUNSIZ /GET SIZE OF MOVE
3698 07307 7001 IAC
3699 07310 3021 DCA TEMP
3700 07311 1722 TAD I FDPTR /GET A WORD
3701 07312 2322 ISZ FDPTR
3702 07313 3412 DCA I X12 /PUT INTO WORDN
3703 07314 2021 ISZ TEMP
3704 07315 5311 JMP .-4
3705 07316 4517 JMS I QLUKUP2 /ENTER INTO S.T.
3706 07317 2570 FUNCTN
3707 07320 0000 FUNSIZ, 0
3708 07321 5274 JMP FOLLOOP /LOOP

```

3709	07322	0000	FDPTR,	0
3710	07323	7217	CORLOC,	CORX
3711	07324	1400	CORV,	1400
3712	07325	0001	COR8IZ,	1
3713	07326	7532	NAMLST,	BCOMP
3714	07327	7536		BLOADN
3715	07330	7542		BRT8N
3716	07331	7546		BAFN
3717	07332	7552		BSPN
3718	07333	7556		BFFN
3719	07334	0000		0

/SAVE FILE NAME=POINTER LIST

3720	07361	0133		
3721	07362	7400		
3722	07363	4223		
3723	07364	4200		
3724	07365	2222		
3725	07366	7000		
3726	07367	1520		
3727	07370	5323		
3728	07371	1477		
3729	07372	5303		
3730	07373	2270		
3731	07374	5301		
3732	07375	2214		
3733	07376	5217		
3734	07377	2345		
3735		7400	PAGE	
3736	07400	7400	FNINIT, FUNC3)-1;2000;0	/ABS
3737	07401	7777		
3738	07402	2000		
3739	07403	0000		
3740	07404	7416	FUNC1)-1;2000;0	/ATN
3741	07405	7777		
3742	07406	2000		
3743	07407	0000		
3744	07410	7417	FUNC2)-1;6000;0	/ASC
3745	07411	7777		
3746	07412	6000		
3747	07413	0000		
3748	07414	7436	FUNC1+20)-1;2000;0	/COS
3749	07415	7777		
3750	07416	2000		
3751	07417	0000		
3752	07420	7437	FUNC2+20)-1;2000;4000	/CHR
3753	07421	7777		
3754	07422	2000		
3755	07423	4000		
3756	07424	7456	FUNC1+40)-1;2000;0	/EXP
3757	07425	7777		
3758	07426	2000		
3759	07427	0000		
3760	07430	7457	FUNC2+40)-1;2000;4000	/DAT
3761	07431	7777		
3762	07432	2000		
3763	07433	4000		
3764	07434	7636	FUNC1+220)-1;2000;0	/SQR
3765	07435	7777		
3766	07436	2000		
3767	07437	0000		
3768	07440	7476	FUNC1+60)-2;0;2000;0	/EXP2
3769	07441	7776		
3770	07442	0000		
3771	07443	2000		
3772	07444	0000		
3773	07445	7477	FUNC2+60)-1;6000;0	/LEN
3774	07446	7777		

~~Not correct~~
~~V80 ODT here~~

3775	07447	6000		
3776	07450	0000		
3777	07451	7516	FUNC1+100/-1/2000/0	/INT
3778	07452	7777		
3779	07453	2000		
3780	07454	0000		
3781	07455	7517	FUNC2+100/-3/2000/4000/6000/0	/POS
3782	07456	7775		
3783	07457	2000		
3784	07460	4000		
3785	07461	6000		
3786	07462	0000		
3787	07463	7536	FUNC1+120/-1/2000/0	/LOG
3788	07464	7777		
3789	07465	2000		
3790	07466	0000		
3791	07467	7537	FUNC2+120/-3/0/2000/6000/4000	/SEG
3792	07470	7775		
3793	07471	0000		
3794	07472	2000		
3795	07473	6000		
3796	07474	4000		
3797	07475	7556	FUNC1+140/-1/2000/0	/SGN
3798	07476	7777		
3799	07477	2000		
3800	07500	0000		
3801	07501	7557	FUNC2+140/-1/2000/4000	/STR
3802	07502	7777		
3803	07503	2000		
3804	07504	4000		
3805	07505	7576	FUNC1+160/-1/2000/0	/SIN
3806	07506	7777		
3807	07507	2000		
3808	07510	0000		
3809	07511	7577	FUNC2+160/-1/6000/0	/VAL
3810	07512	7777		
3811	07513	6000		
3812	07514	0000		
3813	07515	7616	FUNC1+200/-1/2000/0	/RND
3814	07516	7777		
3815	07517	2000		
3816	07520	0000		
3817	07521	7637	FUNC2+220/-1/2000/0	/TRC
3818	07522	7777		
3819	07523	2000		
3820	07524	0000		
3821	07525	0000	0	
3822	07526	0201	BASICN, FILENAME BASIC.SV	/FILE NAMES
3823	07527	2311		
3824	07530	0300		
3825	07531	2326		
3826	07532	0203	BCOMP, FILENAME BCOMP.SV	/FOR LOOKUPS
3827	07533	1715		
3828	07534	2000		
3829	07535	2326		

3830	07536	0214	BLOADN, FILENAME BLOAD.SV
3831	07537	1701	
3832	07540	0400	
3833	07541	2326	
3834	07542	0222	BRTSN, FILENAME BRTS.SV
3835	07543	2423	
3836	07544	0000	
3837	07545	2326	
3838	07546	0201	BAFN, FILENAME BASIC.AF
3839	07547	2311	
3840	07550	0300	
3841	07551	0106	
3842	07552	0201	BSFN, FILENAME BASIC.SF
3843	07553	2311	
3844	07554	0300	
3845	07555	2306	
3846	07556	0201	BFFN, FILENAME BASIC.FF
3847	07557	2311	
3848	07560	0300	
3849	07561	0606	
3850	07562	0201	BUFN, FILENAME BASIC.UF
3851	07563	2311	
3852	07564	0300	
3853	07565	2506	
3854	07566	0201	TMPFIL, FILENAME BASIC.TM
3855	07567	2311	
3856	07570	0300	
3857	07571	2415	
3858			

\$

ABORT	6635	CHEKST	2441	EDTSIZ	1600	FSUB1	3314
ABORTX	0052	CHGARG	1320	END	0343	FSUB2	3326
ACH	3506	CHKDIM	0632	ENDEXP	2741	FUNCTN	2570
ACNT	0042	CHKSDM	0667	ENDFIL	6633	FUNC1	7416
ACO	0137	CHKSGN	5444	ENDFNS	6162	FUNC2	7417
AFADD	6440	CHKVAL	2230	ENTERV	2307	FUNC3	7400
AFDIV	6600	CHKWD	6736	EOST	7570	FUNC4	7415
AFLDA	6640	CHN	7414	ERMSG	5706	FUNFLD	1336
AFMPY	6540	CHRPTR	0017	ERMSG2	1712	FUNNAM	1246
AFSTA	6700	CLOSE	3342	ESIGN	5067	FUNNUM	3336
AFSUB	6500	CLOSEF	7434	EXPLUP	5401	FUNOK	5673
AIDIV	6740	CLRFN	7501	EXPMUL	5124	FUNPTR	1127
AISUB	6400	COLON	0062	EXPON	0025	FUNS	6071
AL1	5431	COMARP	4102	EXPR	2600	FUNSIZ	7320
AMPRTN	6507	CONVLP	5014	EXPTN	3544	FUNSRC	5660
AMPSND	1557	CORE	7200	FADD	0000	FUNSS	3000
ANORM	5236	COREX	7230	FDIV	1400	GCS	5520
ARRAYST	2132	CORLOC	7323	FDLOOP	7274	GENER	7255
ARGCNT	0061	CORSIZ	7325	FDPTR	7322	GENTMP	4200
ARGLUP	1216	CORV	7324	FESC	7410	GETA1	3517
AR1	5414	CORX	7217	FFLD2	2311	GETC	4035
ASSIGN	6067	COR0	7200	FIDIV	3400	GETCWB	4010
A1OK	3465	COR1	7207	FILE	2522	GETDEV	7032
A2OK	3445	COR2	7211	FILENO	7402	GETEXP	5063
BACK1	4000	COR70	7215	FILESW	2556	GETFN	2111
BADDEF	1253	CPSAVE	4166	PINCAL	3271	GETIFN	2045
BADEXP	3122	CWTEMP	6755	FINDA	2470	GETLIN	0222
BADFN	2131	DACNT	0062	FINDEX	0067	GETNAM	5600
BADFOR	3323	DALOOP	1036	FINDIM	0657	GOSUB	2154
BADFS8	3115	DAPUSH	1060	FINDSV	4615	GOTARY	2504
BADGO2	2166	DAPUT	1111	FINDTO	1621	GOTCR	0235
BADIF	2062	DATA	1431	FININP	0423	GOTKW	0314
BADLET	0340	DATFLD	1473	FINSN	3710	GOTNAM	5631
BADNXT	0762	DATLST	2566	FISUB	3000	GOTO	2147
BADPF	0504	DATPTR	1525	FIXEXP	5071	GOTOPR	2715
BAFN	7546	DECPT	0023	FLDA	2000	GOTREL	2027
BASICN	7526	DEF	1200	FLIMIT	1733	GOTSN	3671
BCOMPN	7532	DEFBAD	1051	FMPY	1000	GOTSYM	2453
BFFN	7556	DELOOP	1507	FNEG	7403	HIFLD	0047
BLOADN	7536	DENTRY	1455	FNINIT	7400	HOOKIN	2257
BLOCK	0046	DIGIT	3616	POPENS	2552	ICHAR	6600
BOSINF	7774	DIGTST	5053	FOR	1600	ICHAR1	6677
BRTS	0050	DIM	0600	FORCP	1736	ICHAR2	6673
BRTSN	7542	DIMERR	0701	FORFLD	1660	ICHAR3	6660
B8FN	7552	DIM1	0742	FORJMP	0062	IF	2000
BUCKET	3672	DLSIZE	0051	FORNC	1735	IFEND	2103
BUFN	7562	DOCALL	3200	FPODIV	5310	IFLUP1	2017
CALLUF	3303	DOLU	2432	FPOVLP	5322	IFLUP2	2072
CASEAM	3556	DOSIGN	5150	FPMUL	5200	IFNREG	0063
CASEMA	3554	DSSIZE	1453	FPRNE	5067	IFOPS	4141
CASEMM	3555	DSTRNG	1441	FREE	2572	INBREC	6622
CHAIN	3750	D2	5234	FREFLD	2266	INBUF	7200
CHaine	7001	D3	5235	FSTA	2400	INBUFP	6617
CHECKC	6555	EDTBGN	3012	FSTEP	1734	INCHCT	6730
CHEDIT	7013	EDTBLK	6652	FSUB	0400	INCTR	6732

INDEVH 4400	LSS2 4400	NOFNAM 5702	OUSIZE 6506
INEOF 6727	LSS2AC 7407	NOGOOD 0304	OUTCAL 3210
INERR 6632	LTEMP 2344	NOLETR 3614	OUTOLD 2733
INFO 7604	LUARAY 2463	NOMATC 5677	OUTOPR 3400
INGBUF 6607	LUBUF 4636	NOREGS 3336	OUTWRD 6446
INHNDL 6731	LUFLD 0071	NORMLP 5246	OWTEMP 6443
INITST 7074	LUKUP2 2200	NOSN 0062	Q7100 3622
INJMP 6655	LUPFLD 0742	NOSNUM 0267	PATCH1 2214
INJMPP 6603	LURETN 2451	NOSTAR 2701	PATCH2 2270
INPTR 6733	LUSARY 2516	NOSTRG 5624	PATCH3 1477
INPUT 0400	LUSTRG 2456	NOTHEN 2055	PATCH4 1520
INPUTL 0402	LUVAR 2427	NOTMIN 2627	PATCH5 2222
INREC 6620	L6201 2242	NOTNEW 3100	PCRLF 0062
INTMP 6726	MAC 3456	NOTNUM 3150	PETABL 6000
ITRCNT 5371	MDLABL 3725	NOTQOT 5535	PEXP 0522
JABORT 7125	MINUS 2755	NOTSAM 2250	PFCALL 0475
JARFLD 1413	MISARG 2752	NOTSSD 3042	PLUS 2557
JAROND 0062	MIXED 3511	NOTVAR 3136	POP 4052
JEOP 5400	MM 3452	NOWRIT 6463	POSEXP 5117
JEQ 5006	MODE 0055	NOZERO 5755	POSTX 0350
JFOR 5010	MODSET 5542	NO6000 5256	PRFUN 0547
JGE 5002	MORDAT 1435	NRDL 7521	PRINT 0443
JGT 5004	MOVARG 3227	NUMBER 5000	PRINTL 0446
JLE 5007	MULLUP 5215	NUMCMP 2041	PRNTSN 5741
JLT 5005	M42 5457	NWSVAR 4236	PRTEND 0534
JNE 5003	M5 1131	NXTDGT 5155	PSN 5735
JSUB 5000	NAME1 0134	N3SIZE 2346	PUSH 4062
JTABLE 2416	NAME2 0135	OADD 5353	PUSH0 4073
JUMP 5001	NAME3 0136	OCOUNT 6445	PUSH2 2727
KBOCHK 2346	NAMLST 7326	ODEVH 6505	PUT1 0532
KEYWRD 6243	NCHARS 0020	OKFUN 1256	PUT2 0540
KWLOOP 0273	NCSAVE 4165	OKINAC 3253	PUT40 5514
LDRBLK 0351	NDIGIT 0024	OLON3 2342	QBACK1 0125
LEAD0 5757	NDUK 3104	OLOOP 0031	QCHECK 0113
LEFT 0030	NEGFAC 5267	OLOSTK 0060	QCHKWD 0075
LET 0316	NEWARY 2511	ONSTAK 3107	QCOMAR 0115
LETTER 3600	NEWFLD 2233	OPENAF 7454	QDIGIT 0131
LFDOLR 5610	NEWLIN 0201	OPENAV 7474	QERMSG 0073
LINE 7000	NEWN3 2343	OPENNF 7514	QEXPR 0105
LINEH 0053	NEWOP 0032	OPENNV 7534	QGETA1 0111
LINEL 0054	NEWSN 3716	OP0 0143	QGETC 0102
LINMAX 0120	NEWVAR 4223	OPRAND 2643	QGETCW 0103
LITRL 2562	NEXT 0016	OPRLUP 2705	QGETNA 0114
LOAD 4117	NEXTL 0713	OPR8R 2663	QLETTE 0130
LOADSN 6000	NEXTX 0704	OPR8RS 6050	QLOAD 0120
LOADSS 4303	NFCNT 5236	OP1 0140	QLOADS 0112
LOCTRH 0044	NFLOS 2345	OP2 0141	QLODSN 0074
LOCTRL 0045	NFLOOP 5275	OP2SIZ 6554	QLOOKU 0116
LODSN 0547	NFPTR 5307	OP3 0142	QLUKUP 0117
LODS82 4314	NFUNS 1133	OSTACK 0014	QMOOSE 0076
LOD1ST 6513	NG 7120	QUBLOK 6475	QNEWLI 0107
LOOKUP 2400	NMODE 7541	QUBUF 0000	QNOREG 0132
LOOK2 2212	NOCR 0461	QUOUMP 6471	QNUMBE 0126
LSS1 4000	NOUIGT 3630	QUERR 6502	QOUTOP 0106
LSS1AC 7406	NOEXP 5077	QUPTR 6444	QOUTWR 0100

QPOP 0122	SNUM8 2532	TRYE1 5052	X13 0013
QPU8H 0121	SPACE 5750	TRYE2 5056	ZEXP 5265
QPUSHO 0123	SPCIAL 3514	TRYFUN 5635	
QREMAR 0110	SRDL 7461	TRYPNT 0514	
QRESTC 0104	SREAD 1000	TTX 3740	
QSAVAC 0124	SSINAC 4347	TTY 3730	
QSAVEC 0101	SSLOAD 4323	TTYFIL 2140	
Q8NUM 0077	SSLOOP 3014	TWODIM 0624	
QSTRIN 0127	SSREG1 0064	TW2 5234	
Q10 1132	SSREG2 0065	TW3 5235	
Q2 1130	SSTORE 2400	TYPCHK 3467	
Q400 0133	SSTYPE 4320	TYPE 0026	
RANDOM 3564	STACK 0015	TYPE1 0056	
READ 7411	STACKA 7120	TYPE2 3557	
READX 1526	STACKO 1134	UDEF 1000	
RELOPR 2111	STAKSZ 0060	UMOPR 0765	
REMARK 0205	STAR 0356	UMRTNE 3535	
RESARG 1400	START 7000	UNOPR 2614	
REST 7405	STCHEK 2317	UNREFD 0647	
RESTCP 4027	STEMPS 2560	UPAROW 0362	
RESTOR 0542	STEP1 1727	USE 7540	
RET 7404	STKLVL 0066	USEERR 1754	
RETURN 3561	STKQVR 4070	USEX 1745	
RND0 7421	STMPCT 0035	USRFUN 5704	
RSTRNG 0431	STMPLV 0036	VARCNT 0040	
RUNNED 4600	STOKSZ 0044	VARST 0400	
SACNT 0043	STOP 7441	VCPTR 2415	
SACOMP 6440	STOPX 3565	VERS 0007	
SACON 6400	STOVER 2337	VERSION 0100	
SALOAD 6640	STPTR 0037	WDLOOP 6742	
SAREAD 6500	STRARG 1054	WGOTO 6271	
SARY8T 2332	STRCMP 2065	WORD1 0134	
SASIGN 6065	STRGER 5511	WORD2 0135	
SASTOR 6700	STRING 5455	WORD3 0136	
SAVAC 4251	STRTMP 4213	WPNT 5561	
SAVECP 4021	STRT2 7047	WRITE 7412	
SAVEJF 1643	STRT3 4651	WSTEP 5156	
SCOMP 0400	SVARST 1036	WTAB 3132	
SCOMPR 6567	SVCNT 0041	WTHEN 1737	
SCON 0000	SVTEMP 4300	WTO 6273	
SCRTN 6511	SWRITE 7413	XABORT 0006	
SETFLD 0070	SWTARG 1325	XADD 5162	
SETJF 7401	SYMBL1 0057	XDIV 6756	
SHIFT 3654	SYMBL2 3560	XIDIV 5557	
SIGN 5012	SYMBOL 0027	XISUB 3763	
SKIP2 6531	TABPNT 0464	XLOAD 0770	
SKPEXP 5145	TEMP 0021	XMUL 5762	
SLASH 2761	TEMPS 2556	XSCOMP 6734	
SLITRL 2564	TEMP2 0022	XSTCHE 1126	
SLOAD 2000	TMPBLK 4656	XSTOR 4301	
SLOOP 5467	TMPCNT 0033	XSUB 1743	
SMLNUM 5400	TMPFIL 7566	XXXXSV 4625	
SMODE 7561	TMPLVL 0034	X10 0010	
SNLOOP 3647	TOOMNY 4320	X11 0011	
SNUM 3632	TRYDEC 5040	X12 0012	

ERRORS DETECTED: 0
3859
LINKS GENERATED: 99
3860

ABORT	19	3412#								
ABORTX	56#	3568								
ACM	1938	1967#								
ACNT	48#	1440								
ACO	117#	2071	2070	2475	2490	2613	2639	2648	2660	2690
	2708	2714	2715	2752	2754	2759	2761			
AF	3838									
AFADD	199#	214	2586							
AFDIV	202#	3494								
AFLOA	203#	216	635	1677	2251					
AFMPY	201#	2979								
AFSTA	204#	217	398	975	2354					
AFSUB	200#	215	1100							
AIDIV	205#	2846								
AISUB	198#	213	2145							
ALI	2492	2493	2495	2652	2699	2758#	2768			
AMPRTN	985	3324#								
AMP8ND	983#	3044	3325							
ANORM	2538	2627	2634#	2651	2658	2709				
ARRAYST	160#	161	1442							
ARGCNT	63#	745	767	773	789	812	815	817	818	877
ARGLUP	754#	770								
ARI	2614	2677	2745#	2756	2757					
ASSIGN	344	3049#								
A1OK	1947	1950#								
A2OK	1930	1934#								
BACK1	104	2159#	2166							
BADDEF	742	750	755	759	769	783#	853			
BADEXP	1547	1554	1628	1732#	1734	1784				
BADFN	1217	1221#								
BADFOR	1002	1005	1011	1019	1038	1056	1061#			
BADF83	1666	1687	1717	1727#						
BADGO2	1169	1237	1243	1251#						
BADIF	1135	1141	1181#	1191						
BADLET	335	347#								
BADNXT	582	586	591	628#						
BADPF	449#	454								
BAFN	3716	3838#								
BASIC	3822	3838	3842	3846	3850	3854				
BASICN	3822#									
BCOMP	3826									
BCOMP	3713	3826#								
BFFN	3718	3846#								
BLOAD	3830									
BLOADN	3714	3830#								
BLOCK	52#	2459	3528	3532						
BOSINF	24	140#								
BRTS	54#	3566	3834							
BRTSN	3715	3834#								
BSFN	3717	3842#								
BUCKET	2062	2086#								
BUFN	2441	3850#								
CALLUF	1834	1845#								
CASEAM	1908	1933	1967	2007#						
CASEMA	1906	1949	1950	2005#						
CASEMM	1904	1932	1939	2006#						
CHAIN	2133#	3272								
CHaine	3514#									
CHECKC	94	3362#	3365	3367	3370	3371				

CHEEDIT	3519	3524#								
CHEKST	1422#	1466								
CHGARG	820#	837								
CHKDIM	526	530#								
CHKSDM	525	536	567#	570	571	576				
CHKSGN	2479	2524	2769#	2772	2777					
CHKVAL	1287#	1295								
CHKWD	80	3477#	3478	3479	3483	3486	3491			
CHN	224#	2141								
CHRPTR	28#	284	1012	1042	2164	2165	2174	2179	2184	2194
CLOSE	1881#	3266								
CLOSEF	226#	1884								
CLRFN	231#	1231								
COLON	68#	1206	1212							
COMARP	96	2225#	2227	2230	2234	2237				
CONVLP	2482#	2501	2511							
CORE	3592	3625#								
COREX	3632	3646	3651#							
CORLOC	3635	3639	3641	3648	3710#					
CORSIZ	3628	3649	3653	3712#						
CORV	3644	3711#								
CORX	3642#	3643	3710							
COR0	3627#	3650								
COR1	3634#	3637	3647							
COR2	3636#	3638								
COR70	3631	3640#								
CPSAVE	2180	2183	2277#							
CWTEMP	3480	3481	3487	3488	3492#					
DACNT	65#	705	714	721						
DALoop	674#	700								
DAPUSH	684	690	692#							
DAPUT	717#	722								
DATA	896#	903	3212							
DATFLD	930#	952								
DATLST	167#	168	956							
DATPTR	933	934	936	950	956#					
DECP	32#	2471	2502	2510	2539					
DEF	740#	3216								
DEFBAD	646	652	658	673	675	679	685#	691	699	
DELOOP	942#	947								
DENTRY	898	913	916#	917	919	955				
DIGIT	108	2041#	2043	2049	2050	2052				
DIGTST	2506	2513#								
DIM	512#	566	3207							
DIMERR	513	518	523	535	549	577#				
DIM1	521	557	611#							
DLSIZE	55#	921	922							
DOCALL	1700	1777#								
DOLU	1415#	1439								
DOSIGN	2557	2575#								
DSSIZE	909	914#								
OSTRNG	897	904#								
D2	2629#	2696	2698	2704						
D3	2631#	2693	2695	2706						
EDTBGN	142#	3427								
EDTBLK	3421	3425#								
EDTSIZ	141#	3423								
END	350#	3285								
ENDEXP	1574	1591	1617#	1619						
ENDFIL	3389	3410#	3453							

[illegible]

	3744	3752	3760	3773	3781	3791	3801	3809	3817	
FUNC3	246#	428	441	461	476	3054	3736			
FUNC4	247#	3111	3114	3117	3120	3123	3126	3129	3132	3135
	3138	3141	3144	3147	3150	3153	3156			
FUNFLD	800	819	834#							
FUNNAM	753	771	776	778#						
FUNNUM	1788	1791	1872#							
FUNOK	2922#	2932								
FUNPTR	655	656	660	661	706	707	731#			
FUNS	2909	3052#								
FUNSI2	3696	3697	3707#							
FUNSRC	2911#	2928								
FUNSS	1566	1650#								
GCS	2789	2795	2814#	2821	2826	2830	2831			
GENER	3661	3672#								
GENTMP	824	1866	2288#	2298	2306	2342	3672	3673	3674	3676
GETA1	92	1934	1943	1976#	1989					
GETC	85	2188#	2193	2197	2199	2200				
GETCWB	86	2167#	2172	2173	2175					
GETDEV	2462	3539#								
GETEXP	2521#									
GETFN	382	417	481	1200	1205#	1227	1230	1234	1484	1883
GETIFN	1168#	1180								
GETLIN	268#	272								
GETNAM	95	2863#	2866	2890	2891					
GOSUB	1241#	3191								
GOTARY	1452	1457#								
GOTCR	269	276	279#							
GOTKW	311	326#								
GOTNAM	2873	2881	2888#							
GOTO	1236#	3185								
GOTOPR	1597#	1613								
GOTREL	1154#	1196								
GOTSN	2068	2085#								
GOTSYM	1420	1432#								
HIFLD	53#	3654	3655	3659						
HOOKIN	1280	1304	1310#							
ICHAR	268	275	3383#	3408	3450	3456	3462	3465	3467	3468
ICHAR1	3429	3441	3446#							
ICHAR2	3430	3442#								
ICHAR3	3431#									
IF	1131#	3171								
IFEND	1198#	3168								
IFLUP1	1146#	1153								
IFLUP2	1148	1189#	1197							
IFNREG	70#	1224	1228	1233	1875					
IFOPS	1144	2256#								
INBREC	3401#									
INBUF	151#	3398	3624							
INBUFP	3398#	3406								
INCHCT	3385	3403	3471#							
INCTR	3390	3393	3473#	3559						
INDEVH	145#	2403	3545							
INEOF	3387	3395	3470#							
INERR	3400	3409#								
INFO	21	23	122#	3417	3524	3565	3567	3569		
INGBUF	3390#									
INHNDL	3396	3472#	3537	3548						
INITST	3573#									
INJ	3384	3386	3405	3428#	3432					

[illegible]

[illegible]

PFCALL	442#	462									
PLUS	1501#	3034									
POP	101	2201#	2208								
POSEXP	2545	2550#									
POSTX	355#	360									
PRFUN	442	455	484#								
PRINT	416#	3203									
PRINTL	419#	432	472								
PRNTSN	2961#	2976									
PRTEND	421	473#									
PSN	2949	2951	2957#	2977							
PUSH	100	2203	2207	2209#	2214						
PUSHO	102	2218#	2223								
PUSH2	1599	1607#									
PUT1	456	471#									
PUT2	477#	483									
PUT40	2796	2810#									
QBACK1	104#	436	1188	1555	1584	1614	2039	2051	2236	2512	
	2534	2776	2825	2887	3369						
QCHECK	94#	333	402	452	533	563	666	671	726	851	
	900	979	1009	1207	1215	1475	1479	1548	2780		
QCHKWD	80#	438	458	1022	1027	1165	1177				
QCOMAR	96#	522	698	768	1686						
QDIGIT	108#	2056	2067	2482	2735	2868					
QERMSG	78#	253	273	347	358	449	577	628	685	783	
	1109	1181	1221	1251	1358	1491	1552	1626	1730	1782	
	1861	1970	2113	2139	2215	2370	2807	3319	3410	3594	
QEXPR	88#	331	336	384	418	430	433	443	464	854	
	960	1132	1154	1210	1485	1546	1665	1855	2134		
QGETA1	92#	386	962	1667	2239						
QGETC	85#	251	313	420	1018	1036	1134	1140	1532	1573	
	1578	2028	2042	2226	2477	2505	2516	2522	2802	2872	
	2880	3363	3485								
QGETCW	86#	2815	2820								
QGETNA	95#	512	581	674	741	754	1001	1104	1556		
QLETTE	107#	645	651	657	2865	2898	2902				
QLOAD	99#	445	466	856	1218	1487	1801	1857	1992	2136	
	3349										
QLOADS	93#	397	412	974	1672	1931	1948	2250			
QLODSN	79#	329	381	416	479	958	1000	1131	1198	1241	
	1472	1881	2010	2133							
QLOOKU	97#	529	540	583	760	1006	1113	1558			
QLUKUP	98#	711	779	1749	1760	1793	2085	2294	2302	3681	
	3705										
QMODSE	81#	291	390	407	587	751	964	1186	1238	1244	
	1676	1816	1831	1910	2011	2241	2344				
QNEWLI	90#	404	435	475	478	565	627	728	894	902	
	905	981	1175	1250	1495	1886	2016	2143			
QNOREG	109#	292	622	740	893	1083	1845				
QNUMBE	105#	896	1747								
QOUTOP	89#	345	1163	1185	1612	1616					
QOUTWR	83#	351	392	401	429	471	477	491	493	598	
	602	607	610	807	809	880	969	978	1033	1059	
	1064	1066	1068	1071	1103	1115	1172	1174	1226	1232	
	1247	1249	1494	1679	1820	1836	1850	1869	1885	1941	
	1952	2015	2142	2254	2347	2391	2395	2840			
QPOP	101#	592	594	596	599	718	821	825	827	1597	
	1617	1620	1623	1662	1689	1691	1693	1695	1732	1735	
	1737										
QPUSH	100#	697	762	764	766	1073	1075	1077	1079	1523	

	1527	1528	1541	1607	1609	1656	1658	1660	1664	
QPU SHO	102#	1568	1570	1721	1723					
QREMAR	91#	385	444	451	465	579	630	687	785	855
	961	1086	1116	1133	1155	1183	1211	1253	1486	1863
	2135	3692								
QRESTC	87#	318	457	463	1017	1176	2929			
QSAVAC	103#	1561	1653	1673	1724	1789	1990	3329		
QSAVEC	84#	308	437	1021	1164	2892				
QSNUM	82#	288	1168	1236	1242					
QSTRIN	106#	904	1754							
Q10	703	734#								
Q2	692	693	732#							
Q400	110#	1680	1837	1924	1936	1962	2243	2335	2383	3352
RANDOM	2013#	3260								
READ	239#	391								
READX	958#	982	3226							
RELOPR	1149	1171	1192	1202	1204#					
REMARK	91	252	255#	332	337	349	3230			
RESARG	859	870#	878							
REST	235#	482								
RESTCP	87	2182#	2187							
RESTOR	479#	3238								
RET	234#	879	2012							
RETURN	2010#	3245								
RNDO	221#	2012	2013							
RSTRNG	389	406#								
RUNNED	2407#	3513								
SACNT	49#	1467								
SACOMP	214#	3476								
SACON	213#									
SALOAD	216#									
SAREAD	215#	413								
SARYST	161#	162	1469							
SASIGN	3047#									
SASTOR	217#									
SAVAC	103	2329#	2330	2332	2337	2351				
SAVECP	84	2176#	2181							
SAVEJF	1035#	1090								
SCOMP	209#	3475								
SCOMPR	1184	3327	3339	3372#						
SCON	208#									
SCR TN	3326#	3373								
SETFLD	75#	77	297	715	719	792	937	944	1272	1321
	1802	1823	2096	2106	2312	2323				
	220#	1032								
SETJF	3842									
SF	2072#	2079								
SHIFT	2476	2480#	2575							
SIGN	3337	3342#								
SKIP2	2561	2572#								
SKPEXP	1633#	3040								
SLASH	166#	167	1761							
SLITRL	211#									
SLOAD	2789#	2801								
SLOOP	519	532	2527	2733#	2736					
SMLNUM	242#	2038	2839							
SMODE	2067#	2084								
SNLOOP	82	2054#	2057	2063	2105	2115				
SNUM	162#	163								

SPCIAL	1901	1973#								
SRDL	223#	967								
SREAD	210#	413	414							
SSINAC	2385	2393#								
SSLOAD	2363	2366	2367	2373#	2392	2396				
SSLOOP	1662#	1688								
SSREG1	71#									
SSREG2	72#									
SSTORE	212#									
SSTYPE	2369#	2382								
STACK	26#	263	589	620	810	847	850	858	1084	1728
	1729	2202	2205	2206	2210	2212				
STACKA	26	73	148#	588	2211					
STACKO	25	260	736#	737	2220					
STAKSZ	149#	2211								
STAR	361#	3038								
START	3513#									
STCHEK	730	777	926	1044	1314	1342#	1346	1347	1349	1357
STEMPS	164#	165	2303							
STEP1	1029	1087#								
STKLVL	73#	262	621	1085						
STKQVR	2215#	2224								
STMPCT	42#	259	891	2299	2300					
STMPLV	43#	258	892							
STOKSZ	737#	2220								
STOP	222#	350	2013	2014						
STOPX	2014#	3250								
STOVER	1358#	1431	2317	2328						
STPTR	44#	542	554	555	556	558	559	561	761	828
	831	871	874	1416	1418	1427	1443	1445	1450	1453
	1454	1455	1458	1463	1464	1659	1692	1706	1711	1712
STRARG	681	688#								
STRCMP	1161	1184#								
STRGER	2803	2807#	2816							
STRING	106	2779#	2782	2783	2790	2806	2809	2813		
STRTMP	2290	2299#								
STRT2	3538	3552#								
STRT3	2448#	3523								
SV	3822	3826	3830	3834						
SVARST	159#	160	1437							
SVCNT	47#	1435	2319	2326						
SVTEMP	2333	2334	2339	2340	2341	2343	2349	2350	2352#	
SWRITE	241#									
SWTARG	825#									
SYMBL1	61#	400	977	1678	1951	1985	2253			
SYMBL2	1919	1940	1968	2009#						
SYMBOL	36#	623	752	763	830	833	1007	1114	1298	1425
	1426	1432	1459	1559	1694	1722	1787	1797	1821	1867
	1870	2001	2090	2098	2310	2311	2321	2322	2346	2348
	2922									
TABPNT	427	433#								
TEMP	30#	267	271	312	315	326	327	340	341	539
	547	550	552	593	609	615	616	618	663	682
	683	702	814	836	918	920	924	938	946	1139
	1142	1158	1159	1170	1194	1246	1521	1524	1586	1594
	1622	1625	1690	1701	1720	1736	1739	1798	1805	1808
	1812	1818	1922	1974	1975	1988	1999	2055	2091	2097
	2104	2107	2358	2415	2786	2794	2797	2798	2799	2810
	2812	2960	2975	3333	3335	3338	3699	3703		
TEMP	163#	164	2295							

TEMP2	31#	298	300	301	303	665	688	689	701	744
	839	1143	1151	1173	1248	1560	1569	1657	1704	1710
	1715	2065	2003	2095	2099	2110	2374	2386	2393	2788
	2800									
TM	3854									
TMPBLK	2453#	2456	2458	2460						
TMPCNT	40#	257	889	1081	2291	2292				
TMPFIL	2453	3854#								
TMPLVL	41#	256	890	1082						
TOOMNY	2362	2370#								
TRYDEC	2483	2502#								
TRYE1	2509	2512#								
TRYE2	2504	2516#								
TRYFUN	2869	2892#								
TRYPNT	440	457#								
TTX	1377	2122	2124#	2131	2953	2955				
TTY	2116#	2123	2941	2943	2946	2947	2970	2974		
TTYFIL	1209	1228#								
TWODIM	524	532#								
TW2	2606	2615	2617	2630#						
TW3	2608	2618	2620	2632#						
TYPCHK	1952#	1969	1996							
TYPE	35#	514	528	568	584	676	696	743	746	757
	765	822	823	832	1003	1106	1112	1397	1563	1567
	1655	1696	1697	1719	1827	1902	1909	1958	1963	1993
	1997	2864	2878	2879	2888	2889	2924			
TYPE1	60#	387	394	409	446	468	670	723	963	965
	971	1219	1488	1669	1675	1807	1815	1825	1828	1838
	1842	1858	1935	1945	1954	1959	1983	1986	2137	2240
	2242	2247	3351							
TYPE2	1917	1920	1923	1928	1955	1994	2008#			
UDEF	644#	729	737	3277						
UF	3850									
UMOPR	631#	1540								
UMRTNE	633	1990#								
UNOPR	1532#	1536	1542	1610						
UNREFD	545	551#								
UPAROW	365#	3042								
USE	248#	1102								
USEERR	1105	1109#								
USEX	1102#	3281								
USRFUN	2908	2931#								
VARCNT	45#	1412	2308	2315						
VARST	158#	159	1414	3573						
VCPTTR	1402#	1413	1423	1429	1436	1441	1468			
VERS	20#									
VERSION	17#	20								
WDL00P	3481#	3490								
WGOTO	1178	3181#								
WORD1	112#	250	293	489	708	772	906	910	912	940
	1283	1332	1756	1792	1832	2066	2075	2077	2472	2537
	2602	2626	2654	2655	2657	2680	2703	2755	2784	2829
	3678	3687								
WORD2	114#	295	492	2058	2059	2072	2074	2080	2082	2473
	2486	2605								

[illegible]

AL1571	968	
AL1572	967	
AL1573	940	
AL1576	923	
AL1577	879	
AL1764	1102	
AL1765	1089	
AL1766	1087	
AL1767	1070	
AL1770	1061	
AL1774	1034	
AL1775	1032	
AL2171	1245	
AL2172	1231	
AL2173	1225	
AL2174	1201	
AL2175	1184	
AL2176	1162	
AL2177	1144	
AL2371	1374	
AL2372	1371	
AL2373	1370	
AL2374	1351	
AL2375	1312	
AL2376	1283	1332
AL2377	1270	
AL2564	1473	
AL2565	1469	
AL2566	1467	
AL2567	1457	
AL2570	1442	
AL2571	1440	
AL2572	1437	
AL2573	1435	1465
AL2575	1421	
AL2576	1414	
AL2577	1412	
AL2765	1587	
AL2766	1585	
AL2767	1582	
AL2770	1575	1580
AL2771	1564	
AL2774	1543	
AL2775	1540	
AL2776	1537	
AL2777	1534	
AL3172	1727	
AL3174	1707	
AL3176	1677	
AL3177	1650	1698
AL3370	1884	
AL3373	1849	
AL3374	1847	
AL3375	1819	1868
AL3570	2014	
AL3571	2013	
AL3572	2012	
AL3574	2000	
AL3575	1995	
AL3576	1921	1987

±L3577	1903			
±L3765	2141			
±L3766	2121			
±L3767	2118			
±L3770	2102			
±L3771	2064	2070		
±L3772	2061			
±L3773	2046			
±L3774	2044			
±L3775	2036	2117		
±L3776	2033			
±L3777	2030			
±L4170	2252			
±L4171	2251			
±L4172	2232			
±L4173	2228			
±L4174	2220			
±L4175	2211			
±L4176	2198			
±L4177	2195			
±L4373	2394			
±L4374	2389			
±L4375	2388			
±L4377	2309	2320		
±L4777	2408	2426	2439	2451
±L5164	2577			
±L5165	2573			
±L5166	2553			
±L5167	2551			
±L5170	2548			
±L5171	2538			
±L5172	2536			
±L5174	2507			
±L5175	2494	2500		
±L5176	2492	2493	2495	
±L5373	2690			
±L5374	2660			
±L5375	2652	2699		
±L5376	2614	2677		
±L5377	2609	2682		
±L5567	2839			
±L5570	2838			
±L5571	2828			
±L5572	2827			
±L5573	2811			
±L5574	2787			
±L5575	2785			
±L5576	2774			
±L5577	2770			
±L5765	2973			
±L5766	2967			
±L5767	2954			
±L5771	2952			
±L5773	2923			
±L5774	2909			
±L5775	2906			
±L5776	2883			
±L5777	2874	2882		
±L6575	3342			
±L6576	3311			

±L6577	3302		
±L6760	3466		
±L6761	3460		
±L6762	3459		
±L6763	3454		
±L6764	3451		
±L6765	3448		
±L6766	3447		
±L6767	3434	3443	
±L6774	3414		
±L6776	3412		
±L6777	3402		
±L7152	3582		
±L7153	3575		
±L7154	3573		
±L7164	3556		
±L7165	3555		
±L7172	3524		
±L7174	3521	3543	3550
±L7175	3517		
±L7176	3515		
±L7361	3687		
±L7362	3685		
±L7366	3670		
±L7370	3668		
±L7372	3666		
±L7374	3664		
±L7376	3662		

```
1      /OS/8 BASIC LOADER
2      /
3      /DEC-S8-LBASA-A-LA
4      /
5      /COPYRIGHT,1972
6      /
7      /DIGITAL EQUIPMENT CORPORATION
8      /MAYNARD,MASSACHUSETTS 01754
9      /
10     /AUGUST 19, 1972
11     /
12     /HANK MAURER
13     /
14     /
15     /
16     /
17     0100      VERSION=100
```

```
18      / OS8 BASIC COMPILER POST PROCESSOR
19      0010      X10=10
20      0011      X11=11
21      0013      X13=13
22      0015      STACK=15
23      0020      STCDF=20      /KEY INTERPRETER LOCATIONS
24      0021      NSTADR=STCDF+1
25      0022      NASTAD=NSTADR+1
26      0023      SSTADR=NASTAD+1
27      0024      SASTAD=SSTADR+1
28      0025      CODCDF=SASTAD+1
29      0026      CODBGN=CODCDF+1
30      0027      DATTOP=CODBGN+1
31      0030      DATPTR=DATTOP+1
32      0031      SWPINF=DATPTR+1
33      0040      VARCNT=40      /LOCATIONS DEFINED BY COMPILER
34      0041      SVCNT=VARCNT+1
35      0042      ACNT=SVCNT+1
36      0043      SACNT=ACNT+1
37      0044      LOCTRH=SACNT+1
38      0045      LOCTRL=LOCTRH+1
39      0046      BLOCK=LOCTRL+1
40      0047      HIFLD=BLOCK+1
41      0050      BRTS=HIFLD+1
42      0051      DLSIZE=BRTS+1
43      0052      ABORTX=DLSIZE+1
44      0053      FREEHI=ABORTX+1      /LOCATIONS USED BY RELOCATION CODE
45      0054      FREELO=FREEHI+1
46      0055      TEMP=FREELO+1
47      0056      TEMP2=TEMP+1
48      0057      TEMP3=TEMP2+1
49      0060      WORD1=TEMP3+1
50      0061      WORD2=WORD1+1
51      0062      WORD3=WORD2+1
52      0063      NCHARS=WORD3+1
53      0064      SUBHI=NCHARS+1
54      0065      SUBLO=SUBHI+1
55      0066      CODSZ1=SUBLO+1
56      0067      CODSZ2=CODSZ1+1
57      0070      LOCHI=CODSZ2+1
58      0071      LOCLO=LOCHI+1
59      0072      CODB=LOCLO+1
60      0073      CODF=CODB+1
61      0074      ICOUNT=CODF+1
62      0075      OCOUNT=ICOUNT+1
63      0076      AC1=OCOUNT+1
64      0077      AC2=AC1+1
65      0100      AC3=AC2+1
66      0101      SC=AC3+1
67      0102      LINEH=SC+1
68      0103      LINEL=LINEH+1
69      0104      XLABEL=LINEL+1
70      0105      CLRFLD=XLABEL+1
71      0106      CLREND=CLRFLD+1
72      0107      RESADR=CLREND+1
```

/OS/8 BASIC LOADER

PAL8-V8 10/30/72 PAGE 2-1

73	1036
74	2132
75	2332
76	2560
77	2562
78	2564
79	2566
80	7120
81	3012
82	1600
83	0200
84	3400
85	7760

SVARST=1036	/MORE COMPILER DEFINITIONS
ARAYST=2132	
SARYST=2332	
STEMPS=2560	
LITRL=STEMPS+2	
SLITRL=LITRL+2	
DATLST=SLITRL+2	
STACKA=7120	/MAIN STACK OF COMPILER
EDTBGN=3012	/START OF EDITOR
EDTSIZ=1600	/SIZE OF EDITOR
BRTBGN=200	/START OF BRYS
BRTSIZ=3400	/SIZE OF BRYS
DCB=7760	

```

86      0400      0400      *400
87      00400     1047  LOADER, TAD      HIFLO  /SET UP FREE LOCATION (8K = 1)
88      00401     3053      DCA      FREEHI
89      00402     1377      TAD      (7577
90      00403     3054      DCA      FREELO
91      00404     3031      DCA      SWPINF /CLEAR SWAPPER WORD
92      00405     3102      DCA      LINEH  /CLEAR LINE NUMBER
93      00406     3103      DCA      LINEL
94      00407     1015      TAD      STACK  /ANY UNCLOS'D FOR'S ?
95      00410     7041      CIA
96      00411     1376      TAD      (STACKA-1
97      00412     7650      SNA CLA
98      00413     5216      JMP      .+3      /NO
99      00414     4775      JMS      ERMMSG /YES
100     00415     2506      2506
101     00416     7240      CLA CMA -1
102     00417     1047      TAD      HIFLO  /NO CDF'S IF ONLY 8K
103     00420     7640      SZA CLA
104     00421     5233      JMP      NOPATCH /NO PATCHES
105     00422     1374      TAD      (PATLST-1
106     00423     3010      DCA      X10
107     00424     1410  PATLUP, TAD I    X10
108     00425     7450      SNA
109     00426     5773      JMP      STSTUF
110     00427     3055      DCA      TEMP
111     00430     1372      TAD      (7410 /ALWAYS TWO WORDS SKIP
112     00431     3455      DCA I    TEMP
113     00432     5224      JMP      PATLUP
114     00433     6211  NOPATCH, CDF  10
115     00434     1771      TAD I    (DCB      /CHECK FOR TD8E SYSTEM
116     00435     0370      AND      (770      /ED FRIEDMAN GAVE ME THIS CODE
117     00436     1367      TAD      (-210 /AND I'M TAKING IT ON FAITH
118     00437     6201      CDF
119     00440     7650      SNA CLA
120     00441     1766      TAD      7642 /IS IT A ROM SYSTEM ?
121     00442     1365      TAD      (-6223
122     00443     7640      SZA CLA
123     00444     5250      JMP      NOTD8E /NO TD/8E OR ROM TD/8E
124     00445     1364      TAD      (7377 /TD8E SYS WASTES 400 WORDS
125     00446     3054      DCA      FREELO
126     00447     7130      CLL CML RAR /SET SWAP INFO
127     00450     7001  NOTD8E, IAC
128     00451     3031      DCA      SWPINF
129     00452     4763      JMS      FREEF /GET CDF TO HIGHEST FIELD
130     00453     3336      DCA      SWPF1 /INTO 2 PLACES
131     00454     1336      TAD      SWPF1
132     00455     3345      DCA      SWPF2
133     00456     3314      DCA      SWPFLAG /CLEAR THE SWAP FLAG
134     00457     4261      JMS      SWAP  /MOVE OS8 OUT
135     00460     5773      JMP      STSTUF /DO SYMBOL TABLE STUFF
136     00461     0000  SWAP, 0 /SWAP OS8 RESIDENT
137     00462     7130      CLL CML RAR /4000
138     00463     0031      AND      SWPINF /IS IT A TD8E SYS ?
139     00464     7640      SZA CLA
140     00465     5272      JMP      TD8ESYS /YES

```

141	00466	4315	JMS	SWPSUB	/SWAP 17600 TO/FROM N7600
142	00467	6211	CDF	10	
143	00470	7600	7600		
144	00471	5661	JMP I	SWAP	
145	00472	4315	T08ESYS, JMS	SWPSUB	/SWAP 17600 TO/FROM N7400
146	00473	6211	CDF	10	
147	00474	7400	7400		
148	00475	4315	JMS	SWPSUB	/SWAP 27600 TO/FROM N7600
149	00476	6221	L6221, CDF	20	
150	00477	7600	L7600, 7600		
151	00500	7126	CLL CML	RTL	/FIX UP 07600 STUFF TO MATCH
152	00501	1336	TAD	SWPF1	/CIF CDF N0
153	00502	3766	DCA	7642	
154	00503	1336	TAD	SWPF1	
155	00504	7001	IAC		/CIF N0
156	00505	3762	DCA	7721	
157	00506	1762	TAD	7721	
158	00507	3761	DCA	7727	
159	00510	5661	JMP I	SWAP	
160	00511	7200	SWPRET, CLA		
161	00512	6201	CDF		/RETURN IF 8K
162	00513	5661	JMP I	SWAP	
163	00514	0000	SWPFLAG, 0		
164	00515	0000	SWPSUB, 0		/SWAPPER
165	00516	1715	TAD I	SWPSUB	/GET FIELD
166	00517	3334	DCA	SWP1	/TWICE
167	00520	1334	TAD	SWP1	
168	00521	3347	DCA	SWP2	/ONCE FOR EACH DIRECTION
169	00522	2315	ISZ	SWPSUB	
170	00523	1715	TAD I	SWPSUB	/GET HI FIELD ADDR
171	00524	3055	DCA	TEMP	
172	00525	2315	ISZ	SWPSUB	
173	00526	1277	TAD	L7600	/GET COUNT/POITER
174	00527	3056	DCA	TEMP2	
175	00530	1314	TAD	SWPFLAG	/WHICH WAY ?
176	00531	7640	SZA CLA		
177	00532	5345	JMP	SWPF2	/PUT OS8 BACK
178	00533	2314	ISZ	SWPFLAG	/MOVE OS8 OUT
179	00534	7402	SWP1, HLT		
180	00535	1456	TAD I	TEMP2	/GET PART OF RESIDENT
181	00536	5311	SWPF1, JMP	SWPRET	/RETURN IF 8K ONLY
182	00537	3455	DCA I	TEMP	/INTO HI FIELD
183	00540	2055	ISZ	TEMP	/BUMP POINTER
184	00541	2056	ISZ	TEMP2	/AND PTR/CTR
185	00542	5334	JMP	SWP1	/LOOP
186	00543	6201	CDF		
187	00544	5715	JMP I	SWPSUB	
188	00545	5661	SWPF2, JMP I	SWAP	/IF 8K JUST RETURN
189	00546	1455	TAD I	TEMP	/GET WORD OF HI FIELD
190	00547	7402	SWP2, HLT		
191	00550	3456	DCA I	TEMP2	/BACK WHERE IT BELONGS
192	00551	2055	ISZ	TEMP	
193	00552	2056	ISZ	TEMP2	
194	00553	5345	JMP	SWPF2	
195	00554	6201	CDF		

/OS/8 BASIC LOADER

PAL8-V8 10/30/72 PAGE 3-2

196 00555 1276
197 00556 3336
198 00557 5715

TAD L6221 /SET UP TO FIX FIELD 0 CDFS
DCA SWPF1
JMP I SWPSUB

199	00561	7727		
200	00562	7721		
201	00563	2130		
202	00564	7377		
203	00565	1555		
204	00566	7642		
205	00567	7570		
206	00570	0770		
207	00571	7760		
208	00572	7410		
209	00573	1000		
210	00574	1343		
211	00575	1712		
212	00576	7117		
213	00577	7577		
214		0600	PAGE	
215	00600	6201	NODATA, CDF	
216	00601	4777	JMS	FREEF /SAVE FIELD
217	00602	7041	CIA	
218	00603	3105	DCA	CLRFLD /FOR ARRAY CLEARING
219	00604	1054	TAD	FREELO /SAVE THIS ADDR
220	00605	7041	CIA	
221	00606	3106	DCA	CLREND /FOR END OF ARRAY CLEAR
222	00607	2054	ISZ	FREELO /MAKE IT NEXT FREE + 1
223	00610	1376	TAD	(SVARST-1
224	00611	3010	DCA	X10 /ALLOCATE STRING VARS
225	00612	1375	TAD	(-436
226	00613	3055	DCA	TEMP
227	00614	6211	ASVLUP, CDF	10
228	00615	1410	TAD I	X10 /LOOK FOR DEFINED STRING VAR
229	00616	3056	DCA	TEMP2 /SAVE SYMBOL NUMBER
230	00617	1410	TAD I	X10 /GET SIZE
231	00620	7510	SPA	
232	00621	1374	TAD	(4010 /IF UNDEF USE 16 CHARS
233	00622	3057	DCA	TEMP3
234	00623	1056	TAD	TEMP2 /IS IT DEFINED ?
235	00624	6201	CDF	
236	00625	7700	SMA CLA	
237	00626	4260	JMS	SVSTOR /YES, CREATE ENTRY
238	00627	2055	ISZ	TEMP /BUMP COUNT
239	00630	5214	JMP	ASVLUP /LOOP
240	00631	6211	CDF	10 /ALLOCATE STRING TEMPS
241	00632	1773	P6, TAD I	(STEMPS+1
242	00633	3243	DCA	STEMPF /INIT FIELD
243	00634	1772	TAD I	(STEMPS /AND POINTER
244	00635	7410	SKP	
245	00636	1055	STMLUP, TAD	TEMP /LOOK AT NEXT ENTRY
246	00637	7450	SNA	
247	00640	5771	JMP	ALLOCA /DONE GO ALLOCATE ARRAYS
248	00641	1370	TAD	(-1
249	00642	3010	DCA	X10 /GET POINTER
250	00643	6211	STEMPF, CDF	10
251	00644	1410	TAD I	X10 /GET ADDR OF NEXT ENTRY
252	00645	3055	DCA	TEMP /SAVE IT
253	00646	1410	P7, TAD I	X10 /AND ITS FIELD

254	00647	3243	DCA	STEMPF	
255	00650	2010	ISZ	X10	/SKIP TEMP NUMBER
256	00651	1410	TAD I	X10	/GET SYM NUMBER
257	00652	3056	DCA	TEMP2	
258	00653	6201	CDF		
259	00654	1367	TAD	(110	/GIVE IT MAX SIZE
260	00655	3057	DCA	TEMP3	
261	00656	4260	JMS	SVSTOR	/ALLOOcate IT
262	00657	5236	JMP	STMLUP	/LOOP
263	00660	0000	SVSTOR,	0	/MAKE ST ENTRY FOR STRING VAR
264	00661	1056	TAD	TEMP2	/FIND ST ADDR
265	00662	7104	CLL RAL		
266	00663	1056	TAD	TEMP2	
267	00664	1023	TAD	SSTADR	
268	00665	3011	DCA	X11	
269	00666	1057	TAD	TEMP3	/NUMBER OF CHARS
270	00667	1366	TAD	(3	
271	00670	7110	CLL RAR		
272	00671	3065	DCA	SUBLO	/NUMBER OF WORDS
273	00672	3064	DCA	SUBHI	
274	00673	4331	JMS	SUB	/FREEHI,LO=FREEHI,LO=SUBHI,LO
275	00674	1054	TAD	FREEL0	/SAVE ADDR
276	00675	3411	DCA I	X11	
277	00676	4777	JMS	FREEF	/AND FIELD
278	00677	3411	DCA I	X11	
279	00700	1057	TAD	TEMP3	/PUT IN MAX LENGTH
280	00701	7041	CIA		/(NEGATIVE)
281	00702	3411	DCA I	X11	
282	00703	5660	JMP I	SVSTOR	
283	00704	0000	PSN,	0	/PRINT 3 DIGITS DECIMAL
284	00705	3061	DCA	WORD2	
285	00706	7146	CLL CMA	RTL	/-3
286	00707	3104	DCA	XLABEL	
287	00710	1061	PRNTSN,	TAD	WORD2
288	00711	7106	CLL RTL		/GET NEXT DIGIT
289	00712	7006	RTL		/INTO THE LOW ORDER
290	00713	3061	DCA	WORD2	/THREE BITS AND THE LINK
291	00714	1061	TAD	WORD2	/SAVE SHIFTED NUMBER
292	00715	7004	RAL		/NOW DO LAST SHIFT
293	00716	0365	AND	(17	/ONLY FOUR BITS
294	00717	7440	SPACE,	SZA	
295	00720	5324	JMP	NOZERO	/NOT A ZERO
296	00721	1764	TAD	TTY	/ANY DIGITS YET ?
297	00722	7650	SNA	CLA	
298	00723	5326	JMP	LEAD0	/NO, ITS A LEADING ZERO
299	00724	1363	NOZERO,	TAD	(60
300	00725	4764	JMS	TTY	/MAKE IT ASCII
301	00726	2104	LEAD0,	ISZ	/PRINT DIGIT
302	00727	5310	JMP	PRNTSN	/BUMP COUNT
303	00730	5704	JMP I	PSN	/MORE DIGIT(S)
304	00731	0000	SUB,	0	
305	00732	1065	TAD	SUBLO	/DOUBLE SUBTRACT
306	00733	7161	CLL CML	CIA	/SUBTRACT LOWER
307	00734	1054	TAD	FREEL0	
308	00735	3054	DCA	FREEL0	

309	00736	7004		RAL		/GET BORROW
310	00737	1064		TAD	SUBHI	
311	00740	7041		CIA		
312	00741	1053		TAD	FREHI	/SUBTRACT UPPER
313	00742	3053		DCA	FREHI	/SAVE NEW UPPER
314	00743	1053		TAD	FREHI	/DID IT FIT ?
315	00744	7740		SMA SZA	CLA	
316	00745	5731		JMP I	SUB	/YUP
317	00746	3102	TOOBIG,	DCA	LINEH	/CLEAR LINE NUMBER
318	00747	3103		DCA	LINEL	
319	00750	4762'		JMS	ERMSG	/WRITE MESSAGE
320	00751	2402		2402		/TOO BIG
321	00752	5761'		JMP	ABORTL	/ABORT RUN
322	00753	0000	TTX,	0		/PRINT CHAR ON TTY
323	00754	6041		TSF		/WAIT FOR PREVIOUS CHAR
324	00755	5354		JMP	.-1	
325	00756	6046		TLS		/PRINT THIS ONE
326	00757	7200		CLA		
327	00760	5753		JMP I	TTX	

```

328      / CAUTION !!!
329      / THIS PAGE AND THE NEXT ONE ARE
330      / OVERLAYED BY THE INPUT BUFFER
331      / AS SOON AS THE ROUTINE "INWORD"
332      / IS CALLED. THIS FIRST HAPPENS
333      / AFTER THE TAG "RELCIT" .
334      00761 2136
335      00762 1712
336      00763 0060
337      00764 1740
338      00765 0017
339      00766 0003
340      00767 0110
341      00770 7777
342      00771 1400
343      00772 2560
344      00773 2561
345      00774 4010
346      00775 7342
347      00776 1035
348      00777 2130
349      1000
350      01000 1054 STSTUF, TAD FREELO /SAVE START OF RESIDENT -1
351      01001 7041 CIA /NEGATED
352      01002 3107 DCA RESADR /USED TO COMPUTE AMOUNT OF MOVE
353      01003 1040 TAD VARCNT /GET NUMBER OF
354      01004 1377 TAD (401 /VARIABLES
355      01005 7041 CIA
356      01006 3040 DCA VARCNT
357      01007 1041 TAD SVCNT /STRING VARIABLES
358      01010 1377 TAD (401
359      01011 7041 CIA
360      01012 3041 DCA SVCNT
361      01013 1042 TAD ACNT /ARRAYS
362      01014 1376 TAD (41
363      01015 7041 CIA
364      01016 3042 DCA ACNT
365      01017 1043 TAD SACNT /AND STRING ARRAYS
366      01020 1376 TAD (41
367      01021 7041 CIA
368      01022 3043 DCA SACNT
369      01023 4775 JMS FREEF /SAVE HIGH FIELD
370      01024 3020 DCA STCDF
371      01025 1040 TAD VARCNT /SUBTRACT SPACE FOR
372      01026 7104 CLL RAL /SCALAR TABLE (3 WORDS A PIECE)
373      01027 1040 TAD VARCNT
374      01030 1054 TAD FREELO /DON'T BOTHER WITH A
375      01031 3054 DCA FREELO /DOUBLE PREC. SUBTRACTION
376      01032 1054 TAD FREELO /SAVE START OF SCALAR TABLE
377      01033 7001 IAC /FOR INTERPRETER
378      01034 3021 DCA NSTADR
379      01035 1054 TAD FREELO /CLEAR ALL VARIABLES
380      01036 3010 DCA X10 /IN THE
381      01037 3410 DCA I X10 /SCALAR TABLE
382      01040 3410 DCA I X10

```

383	01041	3410	DCA I	X10	
384	01042	2040	ISZ	VARCNT	
385	01043	5237	JMP	-4	/JUST TO BE NICE
386	01044	6211	CDF	10	/PREPARE TO MOVE
387	01045	1774	P1, TAD I	(LITRL+1	/THE NUMERIC LITERALS <i>SKP</i>
388	01046	3257	DCA	LFLD	/INTO THE SCALAR TABLE
389	01047	1773	TAD I	(LITRL	
390	01050	6201	CDF		
391	01051	7410	SKP		
392	01052	1055	NLLOOP, TAD	TEMP	/ADDR OF NEXT LITERAL
393	01053	7450	SNA		
394	01054	5311	JMP	NONL	/NO MORE NUMERIC LITERALS
395	01055	1372	TAD	(-1	
396	01056	3010	DCA	X10	
397	01057	6211	LFLD, CDF	10	
398	01060	1410	TAD I	X10	/GET ADDR OF NEXT LITERAL
399	01061	3055	DCA	TEMP	
400	01062	1410	P2, TAD I	X10	/ALSO ITS FIELD <i>SKP</i>
401	01063	3257	DCA	LFLD	
402	01064	1410	TAD I	X10	/NOW ITS VALUE
403	01065	3060	DCA	WORD1	
404	01066	1410	TAD I	X10	
405	01067	3061	DCA	WORD2	
406	01070	1410	TAD I	X10	
407	01071	3062	DCA	WORD3	
408	01072	1410	TAD I	X10	/NOW THE SYMBOL NUMBER
409	01073	3056	DCA	TEMP2	
410	01074	1056	TAD	TEMP2	/TIMES THREE
411	01075	7104	CLL RAL		
412	01076	1056	TAD	TEMP2	
413	01077	1054	TAD	FREELO	/PLUS START
414	01100	3011	DCA	X11	/GIVES STORE ADDR
415	01101	6201	CDF		
416	01102	1060	TAD	WORD1	/NOW PUT LITERAL INTO TABLE
417	01103	3411	DCA I	X11	
418	01104	1061	TAD	WORD2	
419	01105	3411	DCA I	X11	
420	01106	1062	TAD	WORD3	
421	01107	3411	DCA I	X11	
422	01110	5252	JMP	NLLOOP	/DO NEXT LITERAL
423	01111	1042	NONL, TAD	ACNT	/ALLOCATE ARRAY TABLE
424	01112	7104	CLL RAL		
425	01113	7104	CLL RAL		/FOUR WORDS PER
426	01114	1054	TAD	FREELO	/SUBTRACT FROM LOWER END
427	01115	3054	DCA	FREELO	
428	01116	1054	TAD	FREELO	/SAVE THIS
429	01117	3022	DCA	NASTAD	/START OF ARRAY TABLE
430	01120	1041	TAD	SVCNT	/ALLOCATE
431	01121	7104	CLL RAL		/STRING VAR TABLE
432	01122	1041	TAD	SVCNT	
433	01123	1054	TAD	FREELO	/3 WORDS EACH
434	01124	3054	DCA	FREELO	
435	01125	1054	TAD	FREELO	/AND SAVE IT FOR THE INT
436	01126	3023	DCA	SSTADR	
437	01127	1043	TAD	SACNT	/NOW SPACE FOR STRING

438	01130	7104	CLL RAL	/ARRAY
439	01131	7104	CLL RAL	
440	01132	1054	TAD	FRELO /TABLE
441	01133	3054	DCA	FRELO
442	01134	1054	TAD	FRELO /SAVE FOR INT
443	01135	3024	DCA	SASTAD
444	01136	6211	COF	10 /PREPARE TO MOVE
445	01137	1771	P3, TAD I	(SLITRL+1 <i>SLP</i>)
446	01140	3362	DCA	SLFLD /STRING LITERALS
447	01141	1770	TAD I	(SLITRL
448	01142	6201	COF	
449	01143	7410	SKP	
450	01144	1055	SLLOOP, TAD	TEMP /IS NEXT LIT THERE ?
451	01145	7450	SNA	
452	01146	5767	JMP	NOSL /NO, END OF THE LINE
453	01147	1372	TAD	(=1
454	01150	3010	DCA	X10
455	01151	4361	JMS	SFLD /SET THE FIELD
456	01152	1410	TAD I	X10 /GET ADDR OF NEXT
457	01153	3055	DCA	TEMP
458	01154	1410	P4, TAD I	X10 /ALSO FIELD <i>SLP</i>
459	01155	3056	DCA	TEMP2
460	01156	1410	TAD I	X10 /THEN CHAR COUNT
461	01157	3063	DCA	NCHARS
462	01160	5766	JMP	SLIT2 /DO REST OF STRING LIT
463	01161	0000	SFLD, 0	
464	01162	6211	SLFLD, COF	10
465	01163	5761	JMP I	SFLD

466	01166	1200	
467	01167	1257	
468	01170	2564	
469	01171	2565	
470	01172	7777	
471	01173	2562	
472	01174	2563	
473	01175	2130	
474	01176	0041	
475	01177	0401	
476		1200	
477	01200	1063	SLIT2, PAGE
478	01201	1377	TAD NCHARS /COMPUTE WORD COUNT
479	01202	7110	TAD (3
480	01203	1010	CLL RAR
481	01204	3057	TAD X10 /TO GET ADDR OF SYMBOL NUMBER
482	01205	1457	DCA TEMP3
483	01206	7104	TAD I TEMP3
484	01207	1457	CLL RAL /SYM NUMBER TIMES 3
485	01210	1023	TAD I TEMP3
486	01211	3011	TAD SSTADR /PLUS BASE
487	01212	1063	DCA X11 /GIVES ST ADDR
488	01213	7001	TAD NCHARS /ALLOCATE SPACE FOR IT
489	01214	7170	IAC
490	01215	3057	CLL CML CMA RAR
491	01216	1057	DCA TEMP3 /(SAVE NUMBER OF WORDS)
492	01217	7100	TAD TEMP3
493	01220	1054	CLL
494	01221	3054	TAD FREELO
495	01222	7420	DCA FREELO /BELOW THE SYMBOL TABLES
496	01223	5337	SNL
497	01224	1054	JMP TMSLIT /TOO MUCH STRING LITERALS
498	01225	1376	TAD FREELO
499	01226	7630	TAD (-END-10
500	01227	5337	SZL CLA
501	01230	1054	JMP TMSLIT /DITTO
502	01231	7001	TAD FREELO /STICK THE ADDR
503	01232	6201	IAC
504	01233	3411	CDF
505	01234	4775	DCA I X11 /INTO THE ST ENTRY
506	01235	3411	JMS FREEF /ALSO THE FIELD
507	01236	1063	DCA I X11
508	01237	7041	TAD NCHARS /ALSO THE SIZE
509	01240	3411	CIA
510	01241	1054	DCA I X11
511	01242	3011	TAD FREELO /THIS IS WHERE IT GOES
512	01243	1063	DCA X11
513	01244	7041	TAD NCHARS /PUT IN THE LENGTH TOO
514	01245	5251	CIA /(-NEGATIVE)
515	01246	4774	JMP .+4
516	01247	1410	JMS SFLD
517	01250	6201	TAD I X10
518	01251	3411	CDF
519	01252	2057	DCA I X11 /MOVE THE LITERAL TEXT
520	01253	5246	ISZ TEMP3
			JMP MOVSL

```

521 01254 1056 P5, TAD TEMP2 /PUT THE FIELD OF THE NEXT SKP
522 01255 3773 DCA SLFLD /ENTRY WHERE IT DOES THE MOST GOOD
523 01256 5772 JMP SLLOOP /DO THE NEXT LITERAL
524 01257 1054 NOSL, TAO FREELO /SAVE TOP OF DATA LIST
525 01260 3027 DCA DATTOP
526 01261 1027 TAD DATTOP /IF EMPTY MAKE TOP=BOTTOM
527 01262 3030 DCA DATPTR
528 01263 1051 TAD DLSIZE
529 01264 7450 SNA /IS ANY DATA ?
530 01265 5771 JMP NODATA /NO
531 01266 7100 CLL
532 01267 1054 TAD FREELO /GET START OF DATA
533 01270 3054 DCA FREELO
534 01271 7420 SNL
535 01272 5332 JMP TMDATA /TOO MUCH DATA
536 01273 1054 TAD FREELO
537 01274 1376 TAD (=END-10
538 01275 7630 SZL CLA
539 01276 5332 JMP TMDATA /DITTO
540 01277 1054 TAD FREELO /SAVE IT
541 01300 3030 DCA DATPTR
542 01301 1054 TAD FREELO /USE X13 TO FILL LIST
543 01302 3013 DCA X13
544 01303 1370 TAD (DATLST-1
545 01304 3010 DCA X10
546 01305 6211 CDF 10
547 01306 1410 DATLUP, TAD I X10 /ANY MORE DATA ELEMENTS ?
548 01307 7450 SNA
549 01310 5771 JMP NODATA
550 01311 3055 DCA TEMP /SAVE ADDR
551 01312 1410 P8, TAD I X10 /GET NEW FIELD SKP
552 01313 3320 DCA DATAF1
553 01314 1320 P9, TAD DATAF1 /TWICE SKP
554 01315 3326 DCA DATAF2
555 01316 1055 TAD TEMP /START WITH NEW ELEMENT
556 01317 3010 DCA X10
557 01320 6211 DATAF1, CDF 10
558 01321 1455 TAD I TEMP /GET COUNT
559 01322 3055 DCA TEMP
560 01323 1410 DATMOV, TAD I X10 /GET NEXT WORD
561 01324 6201 CDF
562 01325 3413 DCA I X13 /MOVE INTO DATA AREA
563 01326 6211 DATAF2, CDF 10
564 01327 2055 ISZ TEMP
565 01330 5323 JMP DATMOV
566 01331 5306 JMP DATLUP /DO NEXT ELEMENT
567 01332 3103 TMDATA, DCA LINEL /ZERO LINE NUMBER
568 01333 3102 DCA LINEH
569 01334 4767 JMS ERMSG /PRINT ERROR MESSAGE
570 01335 2404 2404
571 01336 5766 JMP ABORTL
572 01337 3102 TMSLIT, DCA LINEH /CLEAR THE LINE NUMBER
573 01340 3103 DCA LINEL
574 01341 4767 JMS ERMSG /PRINT MESSAGE
575 01342 2423 2423

```


576	01343	5766'	JMP	ABORTL
577	01344	1045	PATLST, P1/P2/P3/P4/P5/P6/P7/P8/P9/0	
578	01345	1062		
579	01346	1137		
580	01347	1154		
581	01350	1254		
582	01351	0632		
583	01352	0646		
584	01353	1312		
585	01354	1314		
586	01355	0000		

587	01366	2136			
588	01367	1712			
589	01370	2565			
590	01371	0600			
591	01372	1144			
592	01373	1162			
593	01374	1161			
594	01375	2130			
595	01376	5370			
596	01377	0003			
597		1400			
598	01400	1042	ALLOCA,	PAGE	
599	01401	7650		TAD	ACNT /ANY ARRAYS ?
600	01402	5260		SNA CLA	
601	01403	1377		JMP	ALLOCS /NO
602	01404	3010		TAD	(ARAYST /ALLOCATE ARRAYS
603	01405	1022		DCA	X10
604	01406	3011		TAD	NASTAD
605	01407	6211	DOARRAY,	DCA	X11
606	01410	1410		CDP	10
607	01411	3055		TAD I	X10 /GET NEXT ARRAY
608	01412	1410		DCA	TEMP
609	01413	7450		TAD I	X10 /GET FIRST DIM
610	01414	1376		SNA	
611	01415	7001		TAD	(12 /USE 10 IF NONE
612	01416	3056		IAC	/ALLOCATE 0TH ELEMENT
613	01417	1410		DCA	TEMP2
614	01420	7450		TAD I	X10 /GET SECOND DIM
615	01421	1376		SNA	
616	01422	7001		TAD	(12
617	01423	3057		IAC	
618	01424	1057		DCA	TEMP3
619	01425	3065		TAD	TEMP3 /GET READY TO SUBTRACT
620	01426	3064		DCA	SUBLO
621	01427	6201		DCA	SUBHI
622	01430	7132		CDP	
623	01431	0055		CLL CML	RTR
624	01432	7650		AND	TEMP /HOW MANY DIMS ?
625	01433	5237		SNA CLA	
626	01434	1056		JMP	ONLY1 /ONE
627	01435	4775		TAD	TEMP2 /PRODUCT OF DIMS
628	01436	5242		JMS	MUL12
629	01437	3057	ONLY1,	JMP	TIMES3 /MULT BY 3
630	01440	1056		DCA	TEMP3 /ZERO SECOND DIMENSION
631	01441	3065		TAD	TEMP2
632	01442	1374		DCA	SUBLO
633	01443	4775	TIMES3,	TAD	(3 /MULT SIZE BY 3
634	01444	4773		JMS	MUL12
635	01445	1054		JMS	SUB /SUBTRACT FROM FREE
636	01446	3411		TAD	FREELO
637	01447	4772		DCA I	X11 /SAVE ADDR IN S.T.
638	01450	3411		JMS	FREEF
639	01451	1056		DCA I	X11
640	01452	3411		TAD	TEMP2 /ALSO DIMS
641	01453	1057		DCA I	X11
				TAD	TEMP3

642	01454	3411	DCA I	X11	
643	01455	2010	ISZ	X10	/SKIP SYMBOL NUMBER
644	01456	2042	ISZ	ACNT	
645	01457	5207	JMP	DOSARY	
646	01460	1043	ALLOCS, TAD	SACNT	/ANY STRING ARRAYS
647	01461	7650	SNA CLA		
648	01462	5771	JMP	RELCIT	/NO
649	01463	1370	TAD	(SARYST+1	
650	01464	3010	DCA	X10	/ALLOCATE STRING ARRAYS
651	01465	1024	TAD	SASTAD	
652	01466	3011	DCA	X11	
653	01467	6211	DOSARY, CDF	10	
654	01470	1410	TAD I	X10	
655	01471	7450	SNA		
656	01472	1376	TAD	(12	/USE 10 FOR DIM
657	01473	7001	IAC		
658	01474	3057	DCA	TEMP3	
659	01475	1410	TAD I	X10	/GET DIM
660	01476	7450	SNA		
661	01477	1367	TAD	(10	/USE 16 IF NO SIZE SPEC
662	01500	3056	DCA	TEMP2	
663	01501	1057	TAD	TEMP3	
664	01502	3065	DCA	SUBLO	/PREPARE FOR MULT
665	01503	3064	DCA	SUBHI	
666	01504	6201	CDF		
667	01505	1056	TAD	TEMP2	/GET NUM WORDS PER STRING
668	01506	1374	TAD	(3	
669	01507	7110	CLL RAR		
670	01510	4775	JMS	MUL12	/GET ARRAY SIZE
671	01511	4773	JMS	SUB	/DO SUBTRACTION
672	01512	1054	TAD	FREELO	/SAVE ADDR
673	01513	3411	DCA I	X11	
674	01514	4772	JMS	FREEF	
675	01515	3411	DCA I	X11	
676	01516	1056	TAD	TEMP2	/AND STRING SIZE
677	01517	7041	CIA		/(SIZES ARE NEG)
678	01520	3411	DCA I	X11	
679	01521	1057	TAD	TEMP3	/AND NUMBER OF STRINGS
680	01522	3411	DCA I	X11	
681	01523	2010	ISZ	X10	/SKIP NEXT NAME
682	01524	2010	ISZ	X10	/AND NEXT SYM NUMBER
683	01525	2043	ISZ	SACNT	
684	01526	5267	JMP	DOSARY	
685	01527	5771	JMP	RELCIT	
686	01530	0000	INWORD, 0		/READ FROM CODE FILE
687	01531	2074	ISZ	ICOUNT	/ANYTHING IN BUFFER
688	01532	5345	JMP	NOREAD	/YASSUMI
689	01533	4766	JMS I	(7607	/READ NEXT BLOCK
690	01534	0200		200	
691	01535	1000		1000	/NOTE: THIS OVERLAYS USED CODE
692	01536	0000	INBLOK, 0		
693	01537	5765	JMP	IOERR	
694	01540	2336	ISZ	INBLOK	/BUMP BLOCK COUNTER
695	01541	1335	TAD	INBLOK-1	/RESET BUFFER POINTER
696	01542	3350	DCA	INPTR	

/OS/8 BASIC LOADER

PAL8-V8 10/30/72 PAGE 7-2

697	01543	1364		TAD	(-400	/AND COUNTER
698	01544	3074		DCA	ICOUNT	
699	01545	1750	NOREAD,	TAD I	INPTR	/GET WORD
700	01546	2350		ISZ	INPTR	/BUMP POINTER
701	01547	5730		JMP I	INWORD	
702	01550	0000	INPTR,	0		

703	01564	7400			
704	01565	2312			
705	01566	7607			
706	01567	0010			
707	01570	2333			
708	01571	1600			
709	01572	2130			
710	01573	0731			
711	01574	0003			
712	01575	2254			
713	01576	0012			
714	01577	2132			
715		1600			
716	01600	1045	RELCIT,	PAGE TAD	LOCTRL /FIND START OF CODE
717	01601	7101		CLL IAC	
718	01602	3065		DCA	SUBLO /BY SUBTRACTING
719	01603	7004		RAL	
720	01604	1044		TAD	LOCTRH /AMOUNT FROM FREE
721	01605	3064		DCA	SUBHI
722	01606	4777		JMS	SUB
723	01607	1054		TAD	FREELO /THIS IS THE START OF THE CODE
724	01610	3026		DCA	CODBGH /MINUS ONE
725	01611	1053		TAD	FREEHI /THIS IS THE FIELD NUMBER
726	01612	3025		DCA	CODCDF
727	01613	1045		TAD	LOCTRL /SET UP PROG SIZE COUNT
728	01614	7161		CLL CML	CIA
729	01615	3066		DCA	CODSZ1 /LOWER COUNT
730	01616	7004		RAL	
731	01617	1044		TAD	LOCTRH
732	01620	7041		CIA	
733	01621	3067		DCA	CODSZ2 /UPPER COUNT
734	01622	1046		TAD	BLOCK /SET UP FOR READ AND WRITE
735	01623	3776		DCA	OUBLOK
736	01624	1046		TAD	BLOCK
737	01625	3775		DCA	INBLOK
738	01626	1374		TAD	(-401
739	01627	3075		DCA	OCOUNT
740	01630	7240		CLA CMA	
741	01631	3074		DCA	ICOUNT
742	01632	4773	RELOOP,	JMS	INWORD /GET A WORD OF CODE
743	01633	3055		DCA	TEMP
744	01634	1372		TAD	(3000
745	01635	1055		TAD	TEMP
746	01636	0371		AND	(7000 /CHECK FOR OPCODE 5000 (GOTO)
747	01637	7640		SZA CLA	
748	01640	5303		JMP	NORELC /NO JUMP
749	01641	1055		TAD	TEMP
750	01642	0370		AND	(340 /REMOVE FIELD BITS
751	01643	7112		CLL RTR	
752	01644	1264		TAD	CDF0
753	01645	3254		DCA	LBLFLO /FIELD OF LABEL ENTRY
754	01646	1055		TAD	TEMP
755	01647	0367		AND	(7437 /ZERO FIELD BITS
756	01650	3055		DCA	TEMP
757	01651	4773		JMS	INWORD /GET REST OF ADDR

758	01652	3056		DCA	TEMP2	
759	01653	4766'		JMS	CHKLBL	/CHECK FOR UNDEFINED LABEL
760	01654	7402	LBLFLD,	HLT		
761	01655	1456		TAD I	TEMP2	
762	01656	0365		AND	(7	/GET ADDR TO BE RELOCATED
763	01657	3070		DCA	LOCHI	
764	01660	2056		ISZ	TEMP2	
765	01661	1456		TAD I	TEMP2	
766	01662	7100		CLL		
767	01663	1026		TAD	CODBGN	/ADD BASE ADDR
768	01664	6201	CDF0,	CDF		
769	01665	3071		DCA	LOCLO	/SAVE LOW PART OF JUMP
770	01666	7004		RAL		
771	01667	1025		TAD	CODCDF	/GET HIGH PART
772	01670	1070		TAD	LOCHI	
773	01671	7106		CLL RTL		/PUT IT INTO CORRECT PLACE
774	01672	7006		RTL		
775	01673	7004		RAL		
776	01674	1055		TAD	TEMP	/PLUS INSTRUCTION
777	01675	4764'		JMS	OUTWRD	
778	01676	2066		ISZ	CODSZ1	/BUMP COUNTER
779	01677	7410		SKP		
780	01700	2067		ISZ	CODSZ2	/CAN'T BE LAST WORD
781	01701	1071		TAD	LOCLO	/OUTPUT LOW ORDER ADDR
782	01702	7410		SKP		
783	01703	1055	NORELC,	TAD	TEMP	/JUST OUTPUT IT
784	01704	4764'	RELOUT,	JMS	OUTWRD	
785	01705	2066		ISZ	CODSZ1	/DOUBLE WORD ISZ BUMP
786	01706	5232		JMP	RELOOP	
787	01707	2067		ISZ	CODSZ2	
788	01710	5232		JMP	RELOOP	
789	01711	5763'		JMP	LOADIT	/DONE RELOCATING, GO LOAD
790	01712	0000	ERMSG,	0		/PRINT ERROR MESSAGE
791	01713	6201		CDF		
792	01714	1712		TAD I	ERMSG	/GET CODE
793	01715	7112		CLL RTR		/PRINT FIRST CHAR
794	01716	7012		RTR		
795	01717	7012		RTR		
796	01720	4340		JMS	TTY	
797	01721	1712		TAD I	ERMSG	/PRINT SECOND CHAR
798	01722	4340		JMS	TTY	
799	01723	2312		ISZ	ERMSG	/FIX RETURN ADDR
800	01724	1762'		TAD	SPACE	/PRINT SPACE
801	01725	4340		JMS	TTY	
802	01726	3340		DCA	TTY	/USE TTY AS A SWITCH
803	01727	1102		TAD	LINEH	/PRINT HIGH ORDER
804	01730	4761'		JMS	PSN	
805	01731	1103		TAD	LINEL	/THEN LOW ORDER
806	01732	4761'		JMS	PSN	/((LINE NUMBER NATCH 1)
807	01733	1360		TAD	(215	/PRINT CARRIAGE RETURN
808	01734	4757'		JMS	TTX	
809	01735	1356		TAD	(212	/PRINT LINE FEED
810	01736	4757'		JMS	TTX	
811	01737	5712		JMP I	ERMSG	/RETURN
812	01740	0000	TTY,	0		/CONVERT TO ASCII AND PRINT

/OS/8 BASIC LOADER

PAL8-V8 10/30/72 PAGE 8-2

813 01741 0355
814 01742 1354
815 01743 7510
816 01744 1353
817 01745 1352
818 01746 4757'
819 01747 5740

AND (77 /SIX BITS ONLY
TAD (-40 /WHAT SIDE OF FORTY ?
SPA
TAD (100 /LOW SIDE
TAD (240 /HIGH SIDE
JMS TTX /PRINT CHAR
JMP I TTY /RETURN

820	01752	0240			
821	01753	0100			
822	01754	7740			
823	01755	0077			
824	01756	0212			
825	01757	0753			
826	01760	0215			
827	01761	0704			
828	01762	0717			
829	01763	2000			
830	01764	2336			
831	01765	0007			
832	01766	2105			
833	01767	7437			
834	01770	0340			
835	01771	7000			
836	01772	3000			
837	01773	1530			
838	01774	7377			
839	01775	1536			
840	01776	2363			
841	01777	0731			
842		2000			
843	02000	4777	LOADIT,	JMS	OUTDUMP /DUMP LAST BLOCK
844	02001	1045		TAD	LOCTRL /SET UP COUNTER
845	02002	7161		CIA CLL	CML
846	02003	3066		DCA	CODSZ1
847	02004	7004		RAL	
848	02005	1044		TAD	LOCTRH
849	02006	7041		CIA	
850	02007	3067		DCA	CODSZ2
851	02010	1026		TAD	CODBGN
852	02011	3055		DCA	TEMP /CODE BEGIN -1
853	02012	1046		TAD	BLOCK /SET UP BLOCK NUMBER
854	02013	3776		DCA	INBLOK
855	02014	7240		CLA CMA	
856	02015	3074		DCA	ICOUNT
857	02016	1025		TAD	CODCDF /SET UP CODE CDF
858	02017	7106		CLL RTL	
859	02020	7004		RAL	
860	02021	1375		TAD	(6201
861	02022	3025		DCA	CODCDF
862	02023	1025		TAD	CODCDF
863	02024	3233		DCA	CF
864	02025	2055	LODLUP,	ISZ	TEMP /BUMP POINTER
865	02026	5232		JMP	NOFJMP /FIELD IS OK
866	02027	1233		TAD	CF /BUMP THE FIELD
867	02030	1374		TAD	(10
868	02031	3233		DCA	CF
869	02032	4773	NOFJMP,	JMS	INWORD /GET NEXT WORD
870	02033	7402	CF,	HLT	
871	02034	3455		DCA I	TEMP /SAVE THE WORD
872	02035	6201	CDFZER,	CDF	
873	02036	2066		ISZ	CODSZ1 /MORE CODE ?
874	02037	5225		JMP	LODLUP /YES

875	02040	2067	ISZ	CODSZ2	
876	02041	5225	JMP	LODLUP	/YES
877	02042	1233	TAD	CF	/GET THE FIELD
878	02043	3261	DCA	CLEARF	/AND SAVE IT
879	02044	1106	CLRLUP, TAD	CLREND	/IS THIS THE END OF CLEAR ?
880	02045	1055	TAD	TEMP	
881	02046	7640	SZA	CLA	
882	02047	5254	JMP	MORCLR	/NO, KEEP GOING
883	02050	1105	TAD	CLRFLD	/DO FIELDS MATCH ?
884	02051	1261	TAD	CLEARF	
885	02052	7650	SNA	CLA	
886	02053	5264	JMP	DONCLR	/YES, ARRAYS ARE CLEARED
887	02054	2055	MORCLR, ISZ	TEMP	/BUMP POINTER
888	02055	5261	JMP	CLEARF	/DON'T BUMP FIELD
889	02056	1261	TAD	CLEARF	/DO BUMP FIELD
890	02057	1374	TAD	(10	
891	02060	3261	DCA	CLEARF	
892	02061	7402	CLEARF, HLT		
893	02062	3455	DCA I	TEMP	/CLEAR THE WORD
894	02063	5244	JMP	CLRLUP	/DO MORE
895	02064	1261	DONCLR, TAD	CLEARF	/COPY THE FIELD
896	02065	3277	DCA	STFLDM	
897	02066	1055	TAD	TEMP	/GET THE COUNT
898	02067	1107	TAD	RESADR	/OF HOW MUCH SYMBOL TABLE
899	02070	3056	DCA	TEMP2	/TO MOVE
900	02071	1055	TAD	TEMP	/PUT IT INTO AUTO XR'S
901	02072	3013	DCA	X13	
902	02073	1013	TAD	X13	
903	02074	3011	DCA	X11	
904	02075	6201	MOVSTL, CDF		
905	02076	1411	TAD I	X11	/GET NEXT WORD OF ST
906	02077	7402	STFLDM, HLT		
907	02100	3413	DCA I	X13	/STORE IT
908	02101	2056	ISZ	TEMP2	
909	02102	5275	JMP	MOVSTL	
910	02103	4340	JMS	MOVFIN	/MOVE FINI PAGE INTO 7000-7177
911	02104	5772	JMP	7000	/GO READ BRIS,SV
912	02105	0000	CHKLBL, 0		/CHECK LABEL FOR UNDEF
913	02106	1705	TAD I	CHKLBL	/GET FIELD
914	02107	3310	DCA	.+1	
915	02110	7402	HLT		
916	02111	1456	TAD I	TEMP2	/GET FIRST WORD OF LABEL
917	02112	7710	SPA	CLA	
918	02113	5705	JMP I	CHKLBL	/SIGN BIT IS DEFINED
919	02114	7144	CLL	CMA	RAL
920	02115	1056	TAD	TEMP2	/GET ADDR OF LINE NUM
921	02116	3104	DCA	XLABEL	
922	02117	1504	TAD I	XLABEL	/GET HIGH ORDER LINE
923	02120	3102	DCA	LINEM	
924	02121	2104	ISZ	XLABEL	
925	02122	1504	TAD I	XLABEL	/GET LOW ORDER
926	02123	3103	DCA	LINEL	
927	02124	6201	CDF		
928	02125	4771	JMS	ERMSG	/PRINT MESSAGE
929	02126	2523			

930	02127	5705	JMP I	CHKLBL	/RETURN
931	02130	0000	FREEF,	0	/MAKE A CDF FROM FREEHI
932	02131	1053	TAD	FREEHI	
933	02132	7106	CLL	RTL	
934	02133	7004	RAL		
935	02134	1235	TAD	CDFZER	
936	02135	5730	JMP I	FREEF	
937	02136	4340	ABORTL,	JMS	MOVFIN /PUT FINI PAGE INTO 7000-7177
938					/AND ABORT THE RUN
939	02137	5770	JMP I	(ABORT-FINI+7000	
940	02140	0000	MOVFIN,	0	/FINI PAGE MOVER
941	02141	6201	CDF		
942	02142	1367	TAD	(FINI-1	/MOVE INT READING CODE
943	02143	3010	DCA	X10	
944	02144	1366	TAD	(6777	/INTO 7000
945	02145	3011	DCA	X11	
946	02146	1365	TAD	(-200	
947	02147	3055	DCA	TEMP	/PUT CORRECT COUNT HERE
948	02150	1410	TAD I	X10	
949	02151	3411	DCA I	X11	/MOVE CODE
950	02152	2055	ISZ	TEMP	
951	02153	5350	JMP	.-3	
952	02154	5740	JMP I	MOVFIN	

953	02165	7600		
954	02166	6777		
955	02167	2177		
956	02170	7115		
957	02171	1712		
958	02172	7000		
959	02173	1530		
960	02174	0010		
961	02175	6201		
962	02176	1536		
963	02177	2357		
964		2200		
965	02200	1651	FINI,	PAGE
966	02201	7640		TAD I XERMSG /ANY ERRORS ?
967	02202	5315		SZA CLA
968	02203	1250		JMP ABORT /YES, DON'T RUN IT
969	02204	3301		TAD XINT /MOVE INT STUFF
970	02205	1247		DCA FTEMP
971	02206	3303		TAD M12 /10 KEY LOCATIONS
972	02207	1253		DCA FCNT
973	02210	3302		TAD XSAVE /INTO A SAFE PLACE
974	02211	1701		DCA FTEMP2
975	02212	2301		TAD I FTEMP
976	02213	3702		ISZ FTEMP
977	02214	2302		DCA I FTEMP2
978	02215	2303		ISZ FTEMP2
979	02216	5211		ISZ FCNT
980	02217	1050		JMP .-5 /MOVE LOOP
981	02220	3224		TAD BRTS /READ IN BRTS
982	02221	4652		DCA BRTSB
983	02222	3400		JMS I X7607
984	02223	0000		BRTSIZ
985	02224	0000	BRTSB,	0
986	02225	5312		JMP IOERR
987	02226	1253		TAD XSAVE
988	02227	3301		DCA FTEMP
989	02230	1250		TAD XINT /MOVE STUFF BACK
990	02231	3302		DCA FTEMP2
991	02232	1247		TAD M12
992	02233	3303		DCA FCNT
993	02234	1701		TAD I FTEMP
994	02235	2301		ISZ FTEMP
995	02236	3702		DCA I FTEMP2
996	02237	2302		ISZ FTEMP2
997	02240	2303		ISZ FCNT
998	02241	5234		JMP .-5
999	02242	1377		TAD (5561 /PATCH "C LOCATIONS
1000	02243	3776'		DCA 7600
1001	02244	1377		TAD (5561
1002	02245	3775'		DCA 7605
1003	02246	5774'		JMP BRTBGN /GO START BRTS
1004	02247	7766	M12,	-12
1005	02250	0020	XINT,	20
1006	02251	1712	XERMSG,	ERMSG
1007	02252	7607	X7607,	7607

1008	02253	7054	XSAVE,	7001+XSAVE-FINI		
1009	02254	0000	MUL12,	0	/MULTIPLY 12BITS AND 24 BITS	
1010	02255	3100		DCA AC3	/SAVE 12 BIT THING	
1011	02256	3077		DCA AC2	/CLEAR REST OF AC	
1012	02257	3076		DCA AC1		
1013	02260	1373		TAD (-15	/ONLY TEST 12 BITS	
1014	02261	3101		DCA SC		
1015	02262	5300		JMP MULBGN		
1016	02263	7420	MULLUP,	SNL	/WAS BIT ON ?	
1017	02264	5272		JMP NOADD	/NO, DON'T ADD	
1018	02265	1065		TAD SUBLO	/ADD TO HIGH ORDER 2/3'S OF AC	
1019	02266	1077		TAD AC2		
1020	02267	3077		DCA AC2		
1021	02270	7024		CML RAL		
1022	02271	1064		TAD SUBHI		
1023	02272	1076	NOADD,	TAD AC1	/SHIFT AC RIGHT	
1024	02273	7110		CLL RAR		
1025	02274	3076		DCA AC1		
1026	02275	1077		TAD AC2		
1027	02276	7010		RAR		
1028	02277	3077		DCA AC2		
1029	02300	1100	MULBGN,	TAD AC3		
1030	02301	7010	FTEMP,	RAR		
1031	02302	3100	FTEMP2,	DCA AC3		
1032	02303	2101	FCNT,	ISZ SC	/BUMP SHIFT COUNTER	
1033	02304	5263		JMP MULLUP		
1034	02305	1077		TAD AC2	/ANSWER IS LOWER 2/3'S OF AC	
1035	02306	3064		DCA SUBHI		
1036	02307	1100		TAD AC3		
1037	02310	3065		DCA SUBLO		
1038	02311	5654		JMP I MUL12		
1039	02312	3103	IOERR,	DCA LINEL	/ZERO LINE NUMBER	
1040	02313	4651		JMS I XERMSG	/PRINT MESSAGE	
1041	02314	1117		1117		
1042	02315	4772	ABORT,	JMS SWAP	/SWAP OS8 BACK	
1043	02316	6041		TSF	/WAIT FOR TTY DONE	
1044	02317	5316		JMP .-1	/TO PREVENT OS8 , FROM BEING RUINED	
1045	02320	1371		TAD (4207	/RESTORE "C LOCATIONS	
1046	02321	3776		DCA 7600		
1047	02322	1370		TAD (6213		
1048	02323	3775		DCA 7605		
1049	02324	1052		TAD ABORTX	/CALLED VIA CHAIN ?(FROM EDIT)	
1050	02325	7450		SNA		
1051	02326	5775		JMP 7605	/NO, RETURN TO OS8	
1052	02327	3333		DCA EDTBLK	/YES, SAVE EDITOR START	
1053	02330	4652		JMS I X7607	/READ IN EDITOR	
1054	02331	1600		EDTSIZ	/THIS MUCH	
1055	02332	0000		0		
1056			OWTEMP,			
1057	02333	0000	EDTBLK,	0		
1058	02334	5775		JMP 7605	/ERROR	
1059	02335	5767		JMP EDTBGN	/GO START EDITOR	
1060	02336	0000	OUTWRD,	0	/OUTPUT WORD TO TEMP FILE	
1061	02337	2075		ISZ OCOUNT	/ANY ROOM ?	
1062	02340	5351		JMP NOWRIT	/YES	

1063	02341	3333	DCA	OWTEMP	/SAVE WORD
1064	02342	4357	JMS	ODDUMP	/WRITE BLOCK
1065	02343	2363	ISZ	OUBLOK	/BUMP BLOCK NUMBER
1066	02344	1362	TAD	OUBLOK-1	/RESET BUFFET POINTER
1067	02345	3356	DCA	OUPTR	
1068	02346	1366	TAD	(-400	
1069	02347	3075	DCA	OCOUNT	/AND COUNT
1070	02350	1333	TAD	OWTEMP	/RESTORE AC
1071	02351	6211	NOWRIT, CDF	10	
1072	02352	3756	DCA I	OUPTR	/INTO BUFFER
1073	-02353	6201	CDF		
1074	02354	2356	ISZ	OUPTR	
1075	02355	5736	JMP I	OUTWRD	
1076	02356	0000	OUPTR, 0		
1077	02357	0000	ODDUMP, 0		/WRITE BLOCK
1078	02360	4652	JMS I	X7607	/WRITE BLOCK
1079	02361	4210		4210	
1080	02362	0000		0	
1081	02363	0000	OUBLOK, 0		
1082	02364	5312	JMP	IOERR	
1083	02365	5757	JMP I	ODDUMP	
1084		2400		END=FINI+200	
1085	02366	7400			
1086	02367	3012			
1087	02370	6213			
1088	02371	4207			
1089	02372	0461			
1090	02373	7763			
1091	02374	0200			
1092	02375	7605			
1093	02376	7600			
1094	02377	5561			
1095					

S

ABORT	2315	ICOUNT	0074	P2	1062	TTY	1740
ABORTL	2136	INBLOK	1536	P3	1137	VARCNT	0040
ABORTX	0052	INPTR	1550	P4	1154	VERSION	0100
ACNT	0042	INWORD	1530	P5	1254	WORD1	0060
AC1	0076	IOERR	2312	P6	0632	WORD2	0061
AC2	0077	LBLFLD	1654	P7	0646	WORD3	0062
AC3	0100	LEAD0	0726	P8	1312	XERMSG	2251
ALLOCA	1400	LFLD	1057	P9	1314	XINT	2250
ALLOCS	1460	LINEH	0102	RELCIT	1600	XLABEL	0104
ARRAYST	2132	LINEL	0103	RELOOP	1632	XSAVE	2253
ASVLUP	0614	LITRL	2562	RELOUT	1704	X10	0010
BLOCK	0046	LOADER	0400	RESADR	0107	X11	0011
BRTBGN	0200	LOADIT	2000	SACNT	0043	X13	0013
BRTS	0050	LOCHI	0070	SARYST	2332	X7607	2252
BRTSB	2224	LOCLO	0071	SASTAD	0024		
BRTSIZ	3400	LOCTRH	0044	SC	0101		
CDPZER	2035	LOCTRL	0045	SFLD	1161		
CDP0	1664	LODLUP	2025	SLFLD	1162		
CF	2033	L6221	0476	SLITRL	2564		
CHKLBL	2105	L7600	0477	SLIT2	1200		
CLEARF	2061	MORCLR	2054	SLLOOP	1144		
CLREND	0106	MOVFIN	2140	SPACE	0717		
CLRFLO	0105	MOVSL	1246	SSTADR	0023		
CLRLUP	2044	MOVSTL	2075	STACK	0015		
COOB	0072	MULBGN	2300	STACKA	7120		
CODBGN	0026	MULLUP	2263	STCDF	0020		
CODCDF	0025	MUL12	2254	STEMPF	0643		
CODF	0073	M12	2247	STEMPS	2560		
CODSZ1	0066	NASTAD	0022	STFLOM	2077		
CODSZ2	0067	NCHARS	0063	STMLUP	0636		
DATAF1	1320	NLLOOP	1052	STSTUF	1000		
DATAF2	1326	NOADD	2272	SUB	0731		
DATLST	2566	NODATA	0600	SUBHI	0064		
DATLUP	1306	NOFJMP	2032	SUBLO	0065		
DATMOV	1323	NONL	1111	SVARST	1036		
DATPTR	0030	NOPATC	0433	SVCNT	0041		
DATTOP	0027	NOREAD	1545	SVSTOR	0660		
DCB	7760	NORELC	1703	SWAP	0461		
DL\$IZE	0051	NOSL	1257	SWPFLA	0514		
DOARRAY	1407	NOTD8E	0450	SWPF1	0536		
DONCLR	2064	NOWRIT	2351	SWPF2	0545		
DOSARY	1467	NOZERO	0724	SWPINF	0031		
EDTBGN	3012	NSTADR	0021	SWPRET	0511		
EDTBLK	2333	OCOUNT	0075	SWPSUB	0515		
EDTSIZ	1600	ONLY1	1437	SWP1	0534		
END	2400	OUBLOK	2363	SWP2	0547		
ERMSG	1712	QUDUMP	2357	TD8ESY	0472		
FCNT	2303	QUPTR	2356	TEMP	0055		
FINI	2200	OUTWRD	2336	TEMP2	0056		
FREF	2130	OWTEMP	2333	TEMP3	0057		
FREHI	0053	PATLST	1344	TIMES3	1442		
FRELO	0054	PATLUP	0424	TMDATA	1332		
FTEMP	2301	PRNTSN	0710	TMSLIT	1337		
FTEMP2	2302	PSN	0704	TOOBIG	0746		
HIFLD	0047	P1	1045	TTX	0753		

ERRORS DETECTED: 0
1096
LINKS GENERATED: 68
1097

ABORT	939	967	1042#							
ABORTL	321	571	576	937#						
ABORTX	43#	44	1049							
ACNT	35#	36	361	364	423	598	644			
AC1	63#	64	1012	1023	1025					
AC2	64#	65	1011	1019	1020	1026	1028	1034		
AC3	65#	66	1010	1029	1031	1036				
ALLOCA	247	598#								
ALLOCS	600	646#								
ARAYST	74#	601								
ASVLUP	227#	239								
BLOCK	39#	40	734	736	853					
BRTBGN	83#	1003								
BRTS	41#	42	980							
BRTSB	981	985#								
BRTSIZ	84#	983								
CDFZER	872#	935								
CDF0	752	768#								
CF	863	866	868	870#	877					
CHKLBL	759	912#	913	918	930					
CLEARF	878	884	888	889	891	892#	895			
CLREND	71#	72	221	879						
CLRFLO	70#	71	218	883						
CLRLUP	879#	894								
COOB	59#	60								
COOBGN	29#	30	724	767	851					
CODCDF	28#	29	726	771	857	861	862			
CODF	60#	61								
CODSZ1	55#	56	729	778	785	846	873			
CODSZ2	56#	57	733	780	787	850	875			
DATAF1	552	553	557#							
DATAF2	554	563#								
DATLST	79#	544								
DATLUP	547#	566								
DATMOV	560#	565								
DATPTR	31#	32	527	541						
DATTOP	30#	31	525	526						
DCB	85#	115								
DLSIZE	42#	43	528							
DOARAY	605#	645								
DONCLR	886	895#								
DOSARY	653#	684								
EDTBGN	81#	1059								
EDTBLK	1052	1057#								
EDTSIZ	82#	1054								
END	498	537	1084#							
ERMSG	99	319	569	574	790#	792	797	799	811	928
	1006									
FCNT	971	978	992	997	1032#					
FINI	939	942	965#	1008	1084					
FREEF	129	216	277	369	505	637	674	931#	936	
FREEHI	44#	45	88	312	313	314	725	932		
FREELO	45#	46	90	125	219	222	275	307	308	350
	374	375	376	379	413	426	427	428	433	434
	435	440	441	442	493	494	497	501	510	524
	532	533	536	540	542	635	672	723		
FTEMP	969	974	975	988	993	994	1030#			
FTEMP2	973	976	977	990	995	996	1031#			
HIFLD	40#	41	87	102						

ICOUNT	61#	62	687	698	741	856			
INBLOK	692#	694	695	737	854				
INPTR	696	699	700	702#					
INWORD	686#	701	742	757	869				
IOERR	693	986	1039#	1082					
LBLFLD	753	760#							
LEAD0	298	501#							
LFLD	388	397#	401						
LINEH	67#	68	92	317	568	572	803	923	
LINEL	68#	69	93	318	567	573	805	926	1039
LITRL	77#	78	387	389					
LOADER	87#								
LOADIT	789	843#							
LOCHI	57#	58	763	772					
LOCLO	58#	59	769	781					
LOCTRH	37#	38	720	731	848				
LOCTRL	38#	39	716	727	844				
LODLUP	864#	874	876						
L6221	149#	196							
L7600	150#	173							
MORCLR	882	887#							
MOVFIN	910	937	940#	952					
MOVSL	515#	520							
MOVSTL	904#	909							
MULBGN	1015	1029#							
MULLUP	1016#	1033							
MUL12	627	633	670	1009#	1038				
M12	970	991	1004#						
NASTAD	25#	26	429	603					
NCHARS	52#	53	461	477	487	507	512		
NLLOOP	392#	422							
NOADD	1017	1023#							
NODATA	215#	530	549						
NOFJMP	865	869#							
NONL	394	423#							
NOPATC	104	114#							
NOREAD	688	699#							
NORELC	748	783#							
NOSL	452	524#							
NOTD8E	123	127#							
NOWRIT	1062	1071#							
NOZERO	295	299#							
NSTADR	24#	25	378						
OCOUNT	62#	63	739	1061	1069				
ONLY1	625	629#							
OUBLOK	735	1065	1066	1081#					
OUOUMP	843	1064	1077#	1083					
OUPTR	1067	1072	1074	1076#					
OUTWRD	777	784	1060#	1075					
OWTEMP	1056#	1063	1070						
PATLST	105	577#							
PATLUP	107#	113							
PRNTSN	287#	302							
PSN	283#	303	804	806					
P1	387#	577							
P2	400#	578							
P3	445#	579							
P4	458#	580							
P5	521#	581							
P6	241#	582							

P7	253#	583								
P8	551#	584								
P9	553#	585								
RELCIT	648	685	716#							
RELOOP	742#	786	788							
RELOUT	784#									
RESAOR	72#	352	898							
SACNT	36#	37	365	368	437	646	683			
SARYST	75#	649								
SASTAD	27#	28	443	651						
SC	66#	67	1014	1032						
SFLD	455	463#	465	515						
SLFLD	446	464#	522							
SLITRL	78#	79	445	447						
SLIT2	462	477#								
SLLOOP	450#	523								
SPACE	294#	800								
SSTAOR	26#	27	267	436	485					
STACK	22#	94								
STACKA	80#	96								
STCDF	23#	24	370							
STEMPF	242	250#	254							
STEMPS	76#	77	241	243						
STFLDM	896	906#								
STMLUP	245#	262								
STSTUF	109	135	350#							
SUB	274	304#	316	634	671	722				
SUBHI	53#	54	273	310	620	665	721	1022	1035	
SUBLO	54#	55	272	305	619	631	664	718	1018	1037
SVARST	73#	223								
SVCNT	34#	35	357	360	430	432				
SVSTOR	237	261	263#	282						
SWAP	134	136#	144	159	162	188	1042			
SWPFLA	133	163#	175	178						
SWPF1	130	131	152	154	181#	197				
SWPF2	132	177	188#	194						
SWPINF	32#	91	128	138						
SWPRET	160#	181								
SWPSUB	141	145	148	164#	165	169	170	172	187	198
SWP1	166	167	179#	185						
SWP2	168	190#								
TD8ESY	140	145#								
TEMP	46#	47	110	112	171	182	183	189	192	226
	238	245	252	392	399	450	457	550	555	558
	559	564	607	623	743	745	749	754	756	776
	783	852	864	871	880	887	893	897	900	947
	950									
TEMP2	47#	48	174	180	184	191	193	229	234	257
	264	266	409	410	412	459	521	612	626	630
	639	662	667	676	758	761	764	765	899	908
	916	920								
TEMP3	48#	49	233	260	269	279	481	482	484	490
	491	519	617	618	629	641	658	663	679	
TIMES3	628	632#								
TMDATA	535	539	567#							
TMSLIT	496	500	572#							
TOOBIG	317#									
TTX	322#	327	808	810	818					
TTY	296	300	796	798	801	802	812#	819		
VARCNT	33#	34	353	356	371	373	384			

[illegible]

±L1765	762	
±L1767	755	
±L1770	750	
±L1771	746	
±L1772	744	
±L1774	738	
±L2165	946	
±L2166	944	
±L2167	942	
±L2170	939	
±L2174	867	890
±L2175	860	
±L2366	1068	
±L2370	1047	
±L2371	1045	
±L2373	1013	
±L2377	999	1001

/OS/8 BASIC RUNTIME SYSTEM
/
/DEC-S8-LBASA-A-LA
/
/COPYRIGHT,1972
/
/DIGITAL EQUIPMENT CORPORATION
/MAYNARD,MASSACHUSETTS 01754
/
/AUGUST 19, 1972
/
/R.G. BEAN

3400 /ADDRESS OF START OF 5 PAGE OVERLAY BUFFER
OVERLAY=3400

/ASSEMBLY INSTRUCTIONS

/WHEN ASSEMBLED AND LOADED VIA THE ABS. LOADER,THE
/CORE LAYOUT IS AS FOLLOWS:

/BRTS IS AT 0-6777
/OVERLAY BASIC,AF IS AT 3400-4577 *0V1*
/OVERLAY BASIC,SF IA AT 12000-13177 *0V2*
/OVERLAY BASIC,FF IS AT 13400-14577 *0V2*

/TO CREATE SAVE IMAGE FILES PRIOR TO RUNNING BASIC,
/ASSEMBLE THIS SOURCE IN A 12K OR MORE MACHINE,THEN
/PERFORM THE FOLLOWING SEQUENCE OF OS/8 COMMANDS

/.R ABSLDR
/*BRTSS (*BRTS,EAEQVRS IF YOU WISH TO USE ON EAE MACHINE)
/.SAVE SYS:BRTS 0-6777 ✓
/
/.SAVE SYS:BASIC,AF 3400-4577 ✓
/
/.SAVE SYS:BASIC,SF 12000-13177 ✓
/
/.SAVE SYS:BASIC,FF 13400-14577 ✓

/THE BASIC RUN-TIME SYSTEM IS CONDITIONALIZED TO TAKE ADVANTAGE
/OF THE PDP-8/E KE8/E EAE OPTION.
/NORMALLY,THE SYSTEM IS ASSEMBLED SUCH THAT IT WILL RUN ON ANY
/PDP-8 OR PDP-12. TO TAKE ADVANTAGE OF THE ADDITIONAL HARDWARE,SET
/THE SWITCH EAE=1 IF THE SYSTEM INCLUDES A KE8/E EAE.
/THE RESULTING BINARY IS THEN LOADED OVER THE NORMAL SYSTEM
/BINARY AS AN OVERLAY USING THE ABS LOADER,AND THE MODIFIED SYSTEM
/IS SAVED. IN OTHER WORDS,TO CREATE A NON-EAE SYSTEM,ASSEMBLE THIS
/SOURCE ONCE,WITH EAE=0, AND PERFORM THE SAVE OPERATIONS ABOVE ON THE
/BINARY THAT RESULTS. TO CREATE AN EAE SYSTEM,ASSEMBLE THIS SOURCE
/TWICE,ONCE WITH EAE=0 AND ONCE WITH EAE=1. USE THE ABSOLUTE LOADER
/TO LOAD BOTH RESULTING BINARIES (THE EAE BINARY MUST BE LOADED

/AFTER THE NORMAL BINARY), THEN PERFORM THE SAVE
/OPERATIONS ON THE RESULT,

/EAE#0
/EAE#1

/USE STANDARD FLOATING POINT PACKAGE
/ASSEMBLE EAE OVERLAY

IFNZRD EAE <
NOPUNCH
>

/PAGE 0 LOCATIONS

	0003		*3	
00003	0000	TEMP14,	0	
00004	0000	TEMP15,	0	/TEMPS USED BY CHARACTER UNPACKING ROUTINES
00005	6361	NUMCOM,	NUMCOL	/LINK TO 12 BIT COMPARE ROUTINE
00006	0000	USECON,	0	/USE CONSTANT GENERATED BY "USE" STATEMENT
00007	0000	TEMP2,	0	
	0010		*10	
00010	0000	XR0,	0	
00011	0000	XR1,	0	
00012	0000	XR2,	0	
00013	0000	XR3,	0	
00014	0000	XR4,	0	/INDEX REGISTERS
		XR5,		
00015	0000	BABS,	0	/USED BY ABSOLUTE VALUE COMPARING ROUTINE
00016	0000	DLPTR,	0	/POINTER FOR IN-CORE DATA LIST
00017	2713	SPINNR,	2713	/AT RUNTIME, THIS LOCATION IS SPUN FOR RND SEED
	0020		*20	
		/COMPILER=INTERPRETER CONTROL BLOCK. LOCATIONS MARKED BY		
		/A ** ARE EXPECTED TO CONTAIN VALUES SUPPLIED BY THE COMPILER PRIOR		
		/TO THE BRTS LOAD		
00020	6211	CDFIU,	6211	/** /CDF FOR I/O TABLE AND SYMBOL TABLES
00021	0000	SCSTRT,	0	/** /POINTER TO START OF SCALAR SYMBOL TABLE
00022	0000	ARSTRT,	0	/** /POINTER TO START OF ARRAY SYMBOL TABLE-1
00023	0000	STSTRT,	0	/** /POINTER TO START OF STRING SYMBOL TABLE-1
00024	0000	SASTRT,	0	/** /POINTER TO START OF STRING ARRAY TABLE-1
00025	0000	CDFPS,	0	/** /CDF FOR START OF PSEUDO-CODE
00026	0000	PSSTRT,	0	/** /POINTER TO START OF PSEUDO CODE-1
00027	0000	DLSTOP,	0	/** /POINTER TO TOP OF DATA LIST
00030	0000	DLSTRT,	0	/** /POINTER TO BOTTOM OF INCORE DATA LIST-1

/SYSTEM REGISTERS

00031	0000	PSFLAG, 0	/IF BIT 0 ON, T08/E PG2 MOVED
			/IF BIT 11 ON, PG 17600 HAS BEEN MOVED
00032	0000	STRLEN, 0	/LENGTH OF STRING IN SAC
00033	0000	S1, 0	/SUBSCRIPT 1 (MUST BE FOLLOWED BY S21)
00034	0000	S2, 0	/SUBSCRIPT 2 (MUST BE PRECEDED BY S11)
00035	0000	DMAP, 0	/MAP OF DRIVER PAGES
00036	0000	BMAP, 0	/MAP OF FILE BUFFERS
0037		*37	
		/FLOATING POINT PACKAGE LOCATIONS. THE FOLLOWING 21 LOCATIONS ARE USED	
		/FOR VARIOUS PURPOSES BY THE FLOATING POINT PACKAGE. THOSE WITH DOUBLE	
		/LABELS ARE USED BY BRTS AS TEMPORARIES WHEN NOT CALLING THE PACKAGE.	
		/THE SECOND TAG IS THE ONE USED BY THE FLOATING POINT PACKAGE, THE FIRST	
		/IS USED BY BRTS.	
00037	0000	FF, 0	/SPECIAL MODE FLIP-FLOP
		TEMP1,	
00040	0000	AC0, 0	
00041	0000	AC1, 0	
		TEMP3,	
00042	0000	AC2, 0	
		TM,	
00043	6201	TEMP4, 6201	
		EXP,	
00044	0000	ACX, 0	/FAC-EXPONENT
		HORD,	
00045	0000	ACH, 0	/FAC-HIGH ORDER MANTISSA
		LORD,	
		ACLO,	
00046	0000	ACL, 0	/FAC-MANTISSA LOW
		TEMP5,	
00047	0000	OPX, 0	
		TEMP6,	
00050	0000	OPH, 0	
		TEMP7,	
00051	0000	OPL, 0	
00052	0000	DSWIT, 0	/SWITCH USED BY INPUT ROUTINE
00053	0215	CHAR, 215	/TERMINATOR OF LAST INPUT
		K215,	
00054	0215	SWIT1, 215	/=0 FOR NO LF AFTER CR ON INPUT
		M215,	
00055	7563	SWIT2, -215	/=0 FOR NO CR/LF AFTER OUTPUT
00056	7777	EFLG, 7777	/O=E FORMAT
00057	0024	FLOW, 24	/FIELD WIDTH OF OUTPUT
00060	0012	DADP, 12	/#OF PLACES AFTER DEC. PT
00061	0000	TEMP10, 0	/LOC NEEDED BY FPP
00062	0000	TEMP11, 0	/LOC NEEDED BY FPP

/SYSTEM REGISTERS USED OFTEN BY INTERPRETER CODE

00063	0000	MODESW, 0	/0 FOR ARITHMETIC MODE, 1 FOR STRING MODE
00064	0000	INSAV, 0	/CURRENT PSEUDO-INSTRUCTION BEING EXECUTED
00065	0000	LINEHI, 0	/HI ORDER BITS OF LINE # CURRENTLY BEING EXECUTED
00066	0000	LINELO, 0	/LOW ORDER BITS OF CURRENT LINE NUMBER
00067	0452	GSP, GSTCK-1	/POINTER INTO GOSUB STACK
00070	0000	STRMAX, 0	/MAXIMUM # OF CHARS ALLOWED IN CURRENT STRING
00071	0000	STRCNT, 0	/- # OF CHARACTERS IN CURRENT STRING
00072	0000	STRPTR, 0	/POINTER TO CURRENT OPERAND STRING

/OFT USED CONSTANTS

00073	0010	K0010, 0010
00074	0017	K0017, 0017
00075	0077	K0077, 0077
00076	0100	K0100, 100
		USR,
00077	0200	K0200, 0200
	0077	K200=K0200
00100	0340	K0340, 0340
00101	0377	K0377, 0377
00102	0400	K0400, 0400
00103	7400	K7400, 7400
00104	7700	K7700, 7700
00105	7477	K7477, 7477
00106	7740	KM40, -40
00107	7764	M14, -14

/OFT USED LINKS

00110	1034	PRINT, XPRINT	/LINK FOR TTY DRIVER HOOKS
00111	0320	SACPTR, SAC-1	/POINTER TO STRING ACCUMULATOR
00112	3253	PUTCHL, PUTCH	/LINK TO FILE BUFFER STUFFING ROUTINE
00113	0212	ILOOP, ILOOP	/POINTER TO START OF ILOOP
00114	1615	INTL, UNSFIX	/LINK TO UNSIGNED 12-BIT INTEGER FIX
00115	0206	CDFPSL, CDFPSU	/POINTER TO PSEUDO-CODE CDF
00116	1460	ERROR, ERRDIS	/ERROR ROUTINE DISPATCH
00117	6525	FBITS, FBITGT	/ROUTINE TO ISOLATE FUNCTION BITS FROM INST
00120	0200	PWFECCL, PWFECCH	/ROUTINE TO GET NEXT WORD FROM PSEUDO-CODE STREAM
00121	2260	MPYLNK, MPY	/LINK TO 12 BY 12 BIT MULTIPLY
00122	1000	XPUT, XPUTCH	/ROUTINE TO PUT CHAR IN TTY RING BUFFER
00123	2304	FIDLE, IDLE	/LINK TO FILE IDLE CHECK ROUTINE
00124	0537	DEVCAL, DRCALL	/LINK TO DEVICE DRIVER CALLING ROUTINE
00125	3023	WRITFW, WRITFL	/ROUTINE TO WRITE 1 WORD IN FILE BUFFER
00126	2636	STHINL, STHINI	/LINK TO STH INITIALIZER
00127	2676	LDHINL, LDHINI	/LINK TO LDH INITIALIZE
00130	2600	STH, STHL	/STORE HALF ROUTINE
00131	2646	LDH, LDHL	/LOAD HALF ROUTINE
00132	3371	FACSAV, FACSAV	/ROUTINE TO SAVE FAC IN TEMPORARY
00133	2373	FACREL, FACRES	/ROUTINE TO RESTORE FAC FROM TEMPORARY
00134	6241	FGETL, FGET	/LINK TO FPP GET ROUTINE
00135	6256	FPUTL, FPUT	/LINK TO FPP PUT ROUTINE
00136	6215	FNORL, FPNOR	/LINK TO FPP NORMALIZE ROUTINE
00137	0365	FCLR, FACCLR	/ROUTINE TO ZERO FAC
00140	6135	FNEGL, FFNEG	/LINK TO FPP NEGATE ROUTINE
00141	4533	FLOATL, FFLOAT	/LINK TO FPP FLOAT ROUTINE
00142	3126	GETCHL, GETCH	/LINK FOR ASCII CHAR GET ROUTINE
00143	2252	EOFSEL, EOFSET	/ROUTINE TO SET EOF BIT
00144	2240	BSWL, BSWP	/LINK FOR BYTE SWAP ROUTINE
00145	2735	PACKL, PACKCH	/ROUTINE TO PACK ASCII,3 FOR 2
00146	3016	CNOCLL, CNOCLR	/ROUTINE TO INITAILIZE CHAR # TO 1
00147	2706	BUFCHL, BUFCHK	/CHECK STATUS OF BUFFER POINTER
00150	6566	FTYPL, FTYPE	/ROUTINE TO DETERMINE FILE TYPE
00151	6661	CHRNOL, CHARNOL	/ROUTINE TO DETERMINE CHARACTER NUMBER
00152	3302	NEXREL, NEXREC	/ROUTINE TO FILL BUFFER WITH NEXT RECORD
00153	2347	CRLF, CRLFR	/ROUTINE TO PRINT CR,LF
00154	3503	VALLK, VALGET	/ROUTINE USED BY FINPUT TO FETCH CHARS DURING VALS FUNCTION
00155	5364	PATCHP, PATCHF	/LINK TO FPP SPECIAL MODE PATCH
00156	1230	P1SWAP, PSWAP	/ROUTINE TO SWAP HI CORE AND PAGE 17600
00157	0567	LDHRST, LRESET	/ROUTINE TO RESET LDH TO FIELD 0
00160	0771	STHRST, SRESET	/ROUTINE TO RESET STH TO FIELD 0
00161	0563	FSTOP1, FSTOPI	/LINK FOR ^C HOOKS IN DRIVERS

/***** THE ABOVE LINK MUST BE AT 161 *****/

/I/O TABLE POINTER AREA-THIS BLOCK HOLDS POINTERS TO THE I/O TABLE
/ENTRY FOR THE CURRENT FILE.THE POINTERS ARE CHANGED EVERY TIME AN
/SFN IS EXECUTED. A TAD I OFF ONE OF THE POINTERS WILL GET THE INFORMATION
/NOTED IN THE COMMENT FOR THE CURRENT I/O DEVICE
/THIS BLOCK IS INITIALIZED FOR TTY

86
US

00162	0000	ENTNO,	0	/ENTRY NUMBER NOW IN AREA
00163	6677	WORD0,	TTYF	/HEADER WORD
00164	6700	WORD1,	TTYF+1	/BUFFER ADDRESS
00165	6701	WORD2,	TTYF+2	/CURRENT BLOCK IN BUFFER
00166	6702	WORD3,	TTYF+3	/READ\WRITE POINTER
00167	6703	WORD4,	TTYF+4	/HANDLER ENTRY POINT
00170	6704	WORD5,	TTYF+5	/FILE STARTING BLOCK #
00171	6705	WORD6,	TTYF+6	/ACTUAL FILE LENGTH
00172	6706	WORD7,	TTYF+7	/
00173	6707	WORD10,	TTYF+10	DEVICE / (FILE MAXIMUM LENGTH)
00174	6710	WORD11,	TTYF+11	NAME / (POSITION OF PRINT HEAD)
00175	6711	WORD12,	TTYF+12	/
00176	6712	WORD13,	TTYF+13	FILE
00177	6713	WORD14,	TTYF+14	NAME

/BRTS MAINLINE-THIS IS THE INTERPRETER INSTRUCTION LOOP. IT IS IN THIS
/LOOP THAT THE NEXT INSTRUCTION IS FETCHED, DECODED, AND USED AS A DISPATCH
/TO THE PROPER EXECUTION ROUTINES FOR THAT INSTRUCTION.

0200

*200

/SUBROUTINE PWFECH-RETURNS WITH NEXT WORD FROM PSEUDO-CODE STREAM IN AC

00200	5606	PWFECH, JMP I CDFPSU	/START ONCE ONLY CODE IN TTY BUFFER
00201	2302	ISZ INTPC	/BUMP PSEUDO-CODE PROGRAM COUNTER
00202	5206	JMP CDFPSU	/NO-SKIP; JUST GET NEXT PSEUDO-CODE WORD
00203	1206	TAD CDFPSU	/SKIP MEANS WE HAVE TO INCREMENT PS-CODE FIELD
00204	1073	TAD K0010	
00205	3206	DCA CDFPSU	
00206	0321	CDFPSU, START1	/SET DF TO FIELD OF PSEUDO-CODE
00207	1702	TAD I INTPC	/GET NEXT WORD OF CODE
00210	6201	CDF 0	/SET DATA FIELD BACK TO INTERPRETER FIELD
00211	5600	JMP I PWFECH	/RETURN

/*****
/BRTS I-LOOP
/*****

00212	7300	ILOOP, CLA CLL	/FLUSH
00213	3037	DCA FF	/PUT FPP IN SI MODE
00214	4200	JMS PWFECH	/GET NEXT PSEUDO-INSTRUCTION
00215	3064	DCA <u>INSAV</u>	/SAVE FOR LATER
00216	4510	JMS I PRINT	/CALL TO TTY DRIVER
00217	7000	NOP	
00220	1064	TAD INSAV	
00221	0103	AND K7400	/STRIP TO OPCODE BITS
00222	7106	CLL RTL	
00223	7006	RTL	
00224	7004	RAL	/OPCODE NOW IN BITS 8-11
00225	1250	TAD KM10	/SUBTRACT 10
00226	7500	SMA	/IS OPCODE <10?
00227	5243	JMP SCASE	/CALL TO INSTRUCTION COMMON TO SMODE AND AMODE
00230	3040	DCA TEMP1	/YES-SAVE THE OFFSET
00231	1063	TAD MODESW	/WHICH MODE?
00232	7640	SZA CLA	
00233	5271	JMP SMODE	/STRING MODE
00234	1040	TAD TEMP1	/ARITHMETIC MODE-GET OFFSET
00235	1246	TAD JMSI	/MAKE JMS TO FP PACKAGE ROUTINE
00236	3240	DCA .+2	/PUT IN LINE
00237	4307	JMS ARGPRE	/SET UP ARGUMENT FROM SYMBOL TABLE
00240	0240	ILOOPF, .	/JMS TO THE FLOATING POINT PACKAGE ROUTINE
00241	7000	NOP	/FPP SOMETIMES RETURNS TO CALL+2
00242	5212	JMP ILOOP	/DONE
00243	1247	SCASE, TAD JMPI	/JUST DISPATCH TO ROUTINE CALLED FOR
00244	3245	DCA .+1	
00245	0245	.	/JUMP TO APPROPRIATE ROUTINE
00246	4661	JMSI, JMS I SEP1	/JMS USED FOR CALLS TO FPP BY AMODE INST
00247	5661	JMPI, JMP I SEP1	/JMP USED TO CALL ROUTINES COMMON TO AMODE AND SMODE

/OS/8 BASIC RUNTIME SYSTEM

PAL8-V7 10/24/72 PAGE 7-1

00250 7770 KM10, -10

/JUMP TABLE FOR AMODE INSTRUCTIONS

00251	6000	FFADD	/FAC_C(A)+FAC	OPCODE 0
00252	6117	FFSUB	/FAC_C(A)-FAC	OPCODE 1
00253	5600	FFMPY	/FAC_C(A)*FAC	OPCODE 2
00254	5722	FFDIV	/FAC_C(A)/FAC	OPCODE 3
00255	6241	FFGET	/FAC_C(A)	OPCODE 4
00256	6256	FFPUT	/C(A)_FAC	OPCODE 5
00257	5400	FFSUB1	/FAC_C(A)-FAC	OPCODE 6
00260	5412	FFDIV1	/FAC_C(A)/FAC	OPCODE 7
/ALL INSTRUCTIONS BEYOND THIS POINT ARE COMMON TO AMODE AND SMODE				
00261	1400	SEPI, LS1I	/S1_C(A)	OPCODE 10
00262	1410	LS2I	/S2_C(A)	OPCODE 11
00263	0400	FJOCI	/IF TRUE, PC_C(PC, PC+1)	OPCODE 12
00264	0445	JEOFI	/IF EOF, PC_C(PC, PC+1)	OPCODE 13
00265	1140	LINEI	/LINE NUMBER	OPCODE 14
00266	0600	ARRAYI	/ARRAY INST	OPCODE 15
00267	0113	ILOOP	/NOP	OPCODE 16
00270	1200	OPERI	/OPERATE INST	OPCODE 17

00271	1040	SMODE, TAD TEMP1	/INST OFFSET
00272	1306	TAD JMSSI	/BUILD JMP OFF STRING TABLE
00273	3276	DCA SDIS	/PUT IN LINE
00274	7100	CLL	/STRING SCALAR TABLE
00275	4705	JMS I STFIND	/SET UP ARGUMENT ADDRESS
00276	0276	SDIS, .	/CALL STRING ROUTINE REQUESTED

/JUMP TABLE FOR SMODE INSTRUCTIONS

/ A "/" IN THE COMMENT MEANS THAT THAT OPCODE IS NOT USED, SO WE
/USE THE SLOT FOR REGULAR STORAGE

00277	2200	SCON1	/SAC_SAC&C(AS)
00300	2061	SCOMP	/IF SAC_NE, C(AS), PC_PC+2
00301	2417	SREAD	/C(AS)_DEVICE
00302	0302	INTPC, .	/* INTERPRETER PC
00303	3150	SLOAD	/SAC_C(AS)
00304	0477	SSTORE	/C(AS)_SAC
00305	1666	STFIND, STFIND	/* LINK TO STRING FINDING ROUTINE
00306	5707	JMSSI, JMP I ,+1	/* DISPATCH JUMP FOR SMODE INSTRUCTIONS
/*****			
/END OF I-LOOP			
/*****			

/ARGPRE-ROUTINE TO TRANSLATE OPERAND FIELD INTO 12 BIT POINTER
/INTO SCALAR TABLE FOR USE IN FPP CALLS.

```

00307 0000  ARGPRE, 0
00310 1064          TAD INSAV      /GET INSTRUCTION
00311 0101          AND K0377      /STRIP TO OPERAND FIELD
00312 3040          DCA TEMP1      /SAVE
00313 1040          TAD TEMP1
00314 7104          CLL RAL        /*2
00315 1040          TAD TEMP1      /PTR*3
00316 1021          TAD SCSTR1     /MAKE 12 BIT ADDR
00317 1000  SCALDF, 1000          /DF TO SCALAR FIELD (CDF INITIALIZED BY LOADER)
00320 5707          JMP I ARGPRE   /RETURN

```

////////////////////////////////////
 ////////////////////////////////// STRING ACCUMULATOR //////////////////////////////////
 //////////////////////////////////

/36 LOCATIONS USED TO HOLD STRING OPERANDS AND RESULTS FOR STRING
 /OPERATIONS. AT LOAD TIME, IT IS FULL OF ONCE-ONLY STARTUP CODE

```

START1,
00321 7404  SAC,      OSR
00322 7640          SZA CLA
00323 7000          NOP

00324 6046          TLS
00325 2017          ISZ SPINNR
00326 7000          NOP
00327 6041          TSF
00330 5325          JMP .-3
00331 1020          TAD CDF10
00332 3763          DCA I PS1L
00333 1020          TAD CDF10
00334 3764          DCA I PS2L
00335 4556          JMS I P1SWAP
00336 1317          TAD SCALDF
00337 3757          DCA I L7746
00340 1361          TAD PINFO
00341 3011          DCA XR1
00342 1362          TAD POVTAB
00343 3012          DCA XR2
00344 1360          TAD MINUS4
00345 3040          DCA TEMP1
00346 6211  OVML,    CDF 10
00347 1411          TAD I XR1
00350 6201          CDF
00351 3412          DCA I XR2
00352 2040          ISZ TEMP1
00353 5346          JMP OVML

/A HLT PLACED HERE WILL ALLOW YOU TO STOP
/MACHINE BEFORE RUNTIME SYSTEM STARTS BY
/SETTING SWITCH REGISTER
/SET TTY FLAG
/SPIN RANDOM NUMBER SEED
/WHILE WAITING FOR INITIALIZING TLS
/FLAG UP YET?
/NO
=6211 in 8K SYS CDF 10
/SET UP CDFS IN PSWAP
/RESTORE PAGE 17600
/SET PROG NOT RESTARTABLE BIT
/TELL USR TO SAVE 1000-1777
/POINTER TO INFO TABLE IN 17600 - 7647
/POINTER TO BLOCK TABLE IN OVERLAY DRIVER 1530
/WE HAVE TO GET 4 BLOCK NUMBERS
/GET BLOCK NUMBER FOR THIS OVERLAY FROM INFO AREA
/PUT IN TABLE IN OVERLAY DRIVER
/DONE?
/NO

```

```
00354 4556      JMS I PISWAP      /YES-FLUSH PAGE 17600
00355 5756      JMP I .+1
00356 1152      START3           /CONTINUE THE INITIALIZING CODE IN INTERMEDIATE BUFFER
00357 7746      L7746, 7746
00360 7774      MINUS4, -4
00361 7607      PINFU, 7607
00362 1530      POVTAB, ARITHA-1
00363 1240      PS1L, P1CDF
00364 1245      PS2L, P1CDF1
```

```
////////////////////////////////////
////////////////////////////////////
```

/ROUTINE TO ZERO FAC

```
00365 0000      FACCLR, 0
00366 7600      L7600, 7600      /CLA
00367 3044      DCA EXP          /ZERO EXPONENT
00370 3046      DCA LORD         /ZERO LOW MANTISSA
00371 3045      DCA HORD         /ZERO HIGH MANTISSA
00372 5765      JMP I FACCLR
```

/CARRIAGE RETURN FUNCTION (KNOWN ONLY TO COMPILER FOR TERMINATING
/PRINT STATEMENTS)

```
00373 0000      CRFUNC, 0
00374 4550      JMS I FTYPL      /IS FILE NUMERIC?
00375 5513      JMP I ILOOP      /YES-WE DON'T WANT TO OUTPUT CRLF
00376 4553      JMS I CRLF       /DO AS WE ARE TOLD
00377 5513      JMP I ILOOP      /NEXT INST
```


0400

PAGE

/JUMP ON CONDITION

```

00400 1064 FJOCI, TAD INSAV /GET JUMP INSTRUCTION
00401 0074 AND K0017 /MASK OFF JUMP CONDITION
00402 7450 SNA /IS IT GOSUB?
00403 5215 JMP GOSUB /YES-PUSH PC ON STACK THEN JUMP
00404 1363 TAD FSTOPI /BASE TAD FOR BUILD OF TAD INSTRUCTION
00405 3206 DCA ,+1 /PUT IN LINE
00406 0406 . /GET PROPER SKIP
00407 3211 DCA ,+2 /PUT IN LINE
00410 1045 TAD MORD /GET HIGH ORDER FAC
00411 0411 . /SKIP INSTRUCTION
00412 5221 JMP SUCJMP /CONDITION TRUE-JUMP
00413 4520 JFAIL, JMS I PWFECL /CONDITION FALSE-DON'T JUMP,BUT BUMP PC
00414 5513 JMP I ILOOPL /DONE

```

```

00415 1515 GOSUB, TAD I CDFPSL /GET CURRENT PC DATA FIELD
00416 4642 JMS I PUSHGL /PUSH ON GOSUB STACK
00417 1644 TAD I INTPCL /GET CURRENT PC
00420 4642 JMS I PUSHGL /PUSH ON GOSUB STACK
/ FALLS INTO UNCONDITIONAL JUMP BECAUSE A
/ GOSUB IS MERELY A PUSH FOLLOWED BY A JUMP

```

```

00421 4520 SUCJMP, JMS I PWFECL /GET WORD FOLLOWING JUMP INS.
00422 3644 DCA I INTPCL /STORE AS NEW PC
00423 1064 TAD INSAV /GET JUMP INSTRUCTION
00424 0100 AND K0340 /MASK OFF DESTINATION FIELD
00425 7112 CLL RTR /SLIDE OVER
00426 1341 TAD CDFINL /MAKE A CDF INSTRUCTION
00427 3515 DCA I CDFPSL /AND SET NEW PC INSTRUCTION FIELD
00430 5513 JMP I ILOOPL /NEXT INSTRUCTION

```

```

00431 7554 K7554, 7554 /*****THIS CONST CAN NOT BE MOVED, THERE
/MUST BE A CONSTANT BEFORE THE SKIP TABLE,AND
/OTHER MUST BE A TAD OF THAT CONSTANT ON THIS PAGE

```

/SKIP TABLE USED TO HOLD TESTS FOR VARIOUS CONDITIONS

```

00432 7600 K7600, 7600 /UNCONDITIONAL (CLA)
00433 7700 SMA CLA /JPA
00434 7640 SZA CLA /JNA
00435 7740 SMA SZA CLA /JPA JNA
00436 7710 SPA CLA /JMA
00437 7650 SNA CLA /JZA
00440 7750 SPA SNA CLA /JMA JZA
00441 5643 JMP I JFORL /FORLOOP JUMP ROUTINE

```

```

00442 2326 PUSHGL, PUSHG
00443 2042 JFORL, JFOR
00444 0302 INTPCL, INTPC

```

/JUMP ON END OF FILE

00445	4523	JEOF1,	JMS I FIDLE	/SEE IF FILE OPEN
00446	1563		TAD I WORD0	/1ST WORD OF I/O TABLE ENTRY
00447	7112		CLL RTR	/GET EOF BIT IN LINK
00450	7620		SNL CLA	/EOF?
00451	5213		JMP JFAIL	/NO-DON'T JUMP
00452	5221		JMP SUCJMP	/JUMP

00453	0000	////////// GOSUB STACK//////////
00454	0000	GSSTCK, 0 /START OF GOSUB STACK
00455	0000	0
00456	0000	0
00457	0000	0
00460	0000	0
00461	0000	0
00462	0000	0
00463	0000	0
00464	0000	0
00465	0000	0
00466	0000	0
00467	0000	0
00470	0000	0
00471	0000	0
00472	0000	0
00473	0000	0
00474	0000	0
00475	0000	0
00476	0000	GSSTOP, 0 /TOP OF GOSUB STACK

/STRING ACCUMULATOR STORE

```
00477 3736 SSTORE, DCA I STHCDP /STORE CDF FOR OPERAND IN STH
00500 1111      TAD SACPTR
00501 7101      CLL IAC /SET AC TO ADDR OF SAC
00502 4527      JMS I LDHINL /INITIALIZE LHM TO PULL CHARS FROM SAC
00503 4557      JMS I LDHRST /SAC IS IN FIELD 0
00504 1072      TAD STRPTR /POINTER INTO OPERAND
00505 7101      CLL IAC /AC POINTS TO OPERAND
00506 4526      JMS I STHINL /INITIALIZE STH TO STORE IN OPERAND
00507 3071      DCA STRCNT /ZERO COUNT
00510 1032      TAD STRLEN /STRING LENGTH
00511 7450      SNA /IS IT NULL STRING?
00512 5327      JMP SSTEX /YES-WE DON'T HAVE TO STORE ANYTHING-JUST ZERO COUNT
00513 3040      DCA TEMP1 /SERVES AS CHARACTER COUNTER
00514 4531 SSLOOP, JMS I LDH /GET CHAR FROM SAC
00515 4530      JMS I STH /STORE IN OPERAND STRING
00516 2071      ISZ STRCNT /BUMP OPERAND COUNT
00517 2040      ISZ TEMP1 /SAC ALL MOVED YET?
00520 7410      SKP /NO-CHECK IF THERE'S ROOM FOR THE REST
00521 5327      JMP SSTEX /YES-DONE
00522 1071      TAD STRCNT /# OF CHARS IN STRING SO FAR
00523 1070      TAD STRMAX /COMPARE TO MAXIMUM SIZE
00524 7740      SMA SZA CLA /MAXIMUM SIZE REACHED YET?
00525 4516 SL, JMS I ERROR /YES-STRING TOO LONG OR UNDEFINED
00526 5314      JMP SSLOOP /NO-MOVE NEXT CHAR

00527 1736 SSTEX, TAD I STHCDP /DF FOR STRING
00530 3331      DCA ,+1 /PUT IN LINE
00531 0531 TEMP24, . /DF TO STRING FIELD
00532 1071      TAD STRCNT /DONE-GET # OF CHARS MOVED
00533 7041      CIA /NEGATE (ALL COUNTS ARE NEGATIVE
00534 3472      DCA I STRPTR /AND STORE AS COUNT WORD FOR OPERAND STRING
00535 5513      JMP I ILOOP /THAT'S ALL, FOLKS!

00536 2603 STHCDP, STHDF
```

/CALL TO DEVICE DRIVER FOR FILE I/O. ASSUMES ARGS HAVE BEEN SET UP

```
00537 0000 DRCALL, 0
00540 3351 DCA DRARG1 /FUNCTION WORD INTO DRIVER CALL
00541 6201 CDFINL, CDF /DF TO CURRENT FIELD
00542 1564 TAD I WORD1 /GET BUFFER ADDRE FROM I/O TABLE ENTRY
00543 3352 DCA DRARG2 /PUT IN DRIVER CALL
00544 1565 TAD I WORD2 /GET BLOCK NUMBER FROM I/O TABLE
00545 3353 DCA DRARG3 /PUT IN DRIVER CALL
00546 1567 TAD I WORD4 /GET DRIVER ENTRY
00547 3331 DCA TEMP24 /SAVE
00550 4731 JMS I TEMP24 /CALL DRIVER
00551 0000 DRARG1, 0 /FUNCTION CONTROL WORD
00552 0000 DRARG2, 0 /BUFFER ADDRESS
00553 0000 DRARG3, 0 /BLOCK #
00554 5356 JMP DRERR /DEVICE ERROR
00555 5737 JMP I DRCALL /ALLS WELL

00556 7710 DRERR, SPA CLA /DETERMINE ERROR TYPE
00557 4516 DE, JMS I ERROR /FATAL
00560 4516 EF, JMS I ERROR /LOGICAL EOF
```

/CALL TO INTERPRETER EXITING ROUTINE

```
00561 4510 FSTOPN, JMS I PRINT /ON NORMAL EXITS, WE MUST EMPTY RING BUFFER
00562 5361 JMP , -1 /FIRST
00563 1231 FSTOPI, TAD K7554
00564 3064 DCA INSAV /FAKE A CALL TO BASIC, FF FUNCTION 6
00565 5766 JMP I , +1 /CALL OVERLAY
00566 1471 FUNC5I
```

/ROUTINE TO RESET LDH FIELD TO 0

```
00567 0000 LRESET, 0
00570 1341 TAD CDFINL
00571 3773 DCA I LDHDCCK /CHANGE TO CDF 0
00572 5767 JMP I LRESET
00573 2647 LDHDCCK, LDHDF
```

/USE FUNCTION-TAKES WORD FOLLOWING CALL AND STUFFS IT IN USECON FOR
/USE A BUFFER POINTER FOR USER SUBROUTINE

```
00574 0000 USE, 0
00575 4520 JMS I PWFECCL /GET NEXT WORD FROM PSEUDO-CODE STREAM
00576 3006 DCA USECON /STORE IN PAGE 0 SLOT
00577 5774 JMP I USE /RETURN
```

0600

PAGE

/ARRAY INSTRUCTIONS

/ARRAY INSTRUCTIONS WORK BY FINDING THE ADDRESS OF THE ARGUMENT FROM THE ARRAY S
/TABLE, THEN CALLING THE APPROPRIATE FLOATING POIN PACKAGE ROUTINE.

00600	1063	ARRAYI, TAD MODESW	/WHICH MODE?
00601	7640	SZA CLA	
00602	5324	JMP SARRAY	/SMODE
00603	1064	TAD INSAV	/GET ARRAY INSTRUCTION
00604	0342	AND K0037	/MASK OFF ARRAY OPERAND
00605	7106	CLL RTL	/MULTIPLY BY 4 (ENTRY LENGTH)
00606	1022	TAD ARSTR	/MAKE POINTER INTO ARRAY TABLE
00607	3011	DCA XR1	/POINTS TO ARRAY FOR THIS OPERATION
00610	0610	ATABUF, .	/CHANGE OF TO ARRAY TABLE FIELD (SET BY START)
00611	1411	TAD I XR1	/GET POINTER TO FIRST ARRAY ELEMENT
00612	3007	DCA TEMP2	/SAVE FOR LATER
00613	1411	TAD I XR1	/GET DF FOR VARIABLE
00614	3307	DCA ADFC	/PUT IN LINE AT END OF ROUTINE
00615	1411	TAD I XR1	/GET ARRAY DIMENSION 1
00616	3042	DCA TEMP3	/SAVE
00617	1033	TAD S1	/GET SUBSCRIPT 1
00620	3015	DCA BABS	/SET UP 12 BIT COMPARE
00621	1042	TAD TEMP3	/DIMENSION 1 +1
00622	4405	JMS I NUMCOM	/S1 TOO BIG?
00623	4516	SU, JMS I ERROR	/YES-SUBSCRIPT OUT OF BOUNDS ERROR
00624	3050	DCA TEMP6	/CLEAR TEMPORARY
00625	1411	TAD I XR1	/GET DIMENSION 2
00626	7450	SNA	/IS SECOND DIMENSION 0?(ARRAY UNIDIMENSIONAL)
00627	5240	JMP ADCALC	/YES=DON'T CHECK S2 FOR OUT OF BOUNDS
00630	3371	DCA TEMP30	/SAVE DIM2+1
00631	1034	TAD S2	/GET SUBSCRIPT 2
00632	3015	DCA BABS	/SAVE 12 BIT COMPARE
00633	1371	TAD TEMP30	
00634	4405	JMS I NUMCOM	/S2 BIGGER THAN DIM2?
00635	5223	JMP SU	/YES
00636	1034	TAD S2	/MULTIPLY DIM1+1 BY S2
00637	4521	JMS I MPYLNK	/12 BY 12 MULTIPLY ROUTINE
00640	7100	ADCALC, CLL	
00641	1033	TAD S1	/LORD OF S1+(DIM1+1)*S2
00642	3047	DCA TEMP5	/SAVE
00643	7004	RAL	/CARRY TO BIT 11
00644	1050	TAD TEMP6	/WORD OF S1+(DIM1+1)*S2
00645	3050	DCA TEMP6	/SAVE
00646	1047	TAD TEMP5	/LORD OF S1+(DIM1+1)*S2
00647	7104	CLL RAL	/*2
00650	3051	DCA TEMP7	/LORD OF [S1+(DIM1+1)*S2]*2
00651	1050	TAD TEMP6	/WORD OF S1+(DIM1+1)*S2
00652	7004	RAL	/*2
00653	3042	DCA TEMP3	/WORD OF [S1+(DIM1+1)*S2]*2
00654	7100	CLL	
00655	1047	TAD TEMP5	/LORD OF S1+(DIM1+1)
00656	1051	TAD TEMP7	/LORD OF [S1+(DIM1+1)*S2]

00657	3051	DCA TEMP7	/LORD OF 3*[S1+(DIM1+1)*S2]
00660	7004	RAL	/CARRY TO BIT 11
00661	1050	TAD TEMP6	/HORD OF [S1+(DIM1+1)*S2]*2
00662	1042	TAD TEMP3	/HORD OF S1+(DIM1+1)*S2
00663	3050	DCA TEMP6	/HORD OF 3*[S1+(DIM1+1)*S2]
00664	7100	CLL	
00665	1051	TAD TEMP7	/INDEX TO ELEMENT
00666	1007	TAD TEMP2	/AC POINTS TO CORRECT ARRAY ELEMENT
00667	3011	DCA XR1	/SAVE POINTER
00670	7004	RAL	/CARRY TO BIT 11
00671	1050	TAD TEMP6	/COMBINE TO MAKE TOTAL # OF FIELD OVERLAPS
00672	7106	CLL RTL	
00673	7004	RAL	/SLIDE OVERLAPS TO FIELD BITS (6-8)
00674	1307	TAD ADFC	/ADD ANY CHANGE IN DATA FIELD TO CDF
00675	3307	DCA ADFC	/PUT ABSOLUTE CDF IN LINE
00676	1064	TAD INSAV	/GET ARRAY INSTRUCTION AGAIN
00677	0100	AND K0340	/MASK OFF ARRAY OPCODE
00700	7112	CLL RTR	
00701	7012	RTR	
00702	7010	RAR	/SLIDE TO BITS 9-11
00703	1346	TAD JMP12	/AND USE AS INDEX INTO JUMP TABLE
00704	3311	DCA ARJMP	/PUT JUMP IN LINE OF CODE
00705	7001	IAC	
00706	3037	DCA FF	/PUT FPP IN "SPECIAL MODE"
00707	0707	ADFC, .	/CHANGE DF TO DF OF ARRAY ELEMNT
00710	1011	TAD XR1	/AC POINTS TO ARRAY ELEMENT
00711	0711	ARJMP, .	/PERFORM THE REQUIRED OPERATION
00712	7000	NOP	/FPP SOMETIMES RETURNS TO CALL+2
00713	5513	JMP I ILOOPL	/DONE

/ARRAY JUMP TABLE

00714	5400	AJT, FFSUB1	/FAC=A(S1,S2)=FAC	OPCODE 0
00715	6000	FFADD	/FAC=FAC+A(S1,S2)	OPCODE 1
00716	6117	FFSUB	/FAC=FAC-A(S1,S2)	OPCODE 2
00717	5600	FFMPY	/FAC=FAC*A(S1,S2)	OPCODE 3
00720	5722	FFDIV	/FAC=FAC/A(S1,S2)	OPCODE 4
00721	6241	FFGET	/FAC=C(A(S1,S2)	OPCODE 5
00722	6256	FPUTLL, FFPUT	/C(A(S1,S2)=FAC	OPCODE 6
00723	5412	FFDIV1	/FAC=A(S1,S2)/FAC	OPCODE 7

/STRING ARRAY DISPATCH

00724	1064	SARRAY, TAD INSAV	/GET INSTRUCTION
00725	0100	AND K0340	/ISOLATE ARRAY OPCODE
00726	7112	CLL RTR	
00727	7012	RTR	/AND SLIDE IT OVER FOR AN OFFSET
00730	7010	RAR	
00731	1336	TAD JMPISA	/BUILD A JUMP TO STRING INSTRUCTION
00732	3335	DCA SAD	/AND PUT IN LINE
00733	7120	STL	/TELL SFIND TO USE ARRAY TABLE
00734	4743	JMS I STFILK	/SET UP ARGUMENT ADDRESS
00735	0735	SAD, .	/EXECUTE INSTRUCTION

/STRING ARRAY JUMP TABLE

/USED WHEN ARRAYI CALLED IN SMODE

/ A "/*" IN THE COMMENT MEANS THAT OPCODE IS UNDEFINED AND THE SLOT

/IN THE TABLE IS USED FOR NORMAL STORAGE

00736	5737	JMPISA, JMP I .+1	/DISPATCH JUMP FOR STRING ARRAY INSTRUCTIONS
00737	2200	SCON1	/SAC_SAC&C(AS(S1))
00740	2061	SCOMP	/SKIP IF SAC=C(AS(S1))
00741	2417	SREAD	/AS(S1)_DEVICE
00742	0037	K0037, 37	/*
00743	1666	STFILK, STFIND	/* LINK TO STRING FINDING ROUTINE
00744	3150	SLOAD	/SAC_C(AS(S1))
00745	0477	SSTORE	/C(AS(S1))_SAC
00746	4714	JMPI2, JMS I AJT	/* DISPATCH JUMP FOR ARRAY INST

/ROUTINE TO PUT ONE WORD IN FILE BUFFER IN FIELD 1

```
00747 0000 BCPUT, 0
00750 3050 DCA TEMP6 /SAVE AC
00751 4523 JMS I FIDLE /CHECK IF FILE OPEN
00752 1566 TAD I WORD3 /GET READ/WRITE POINTER
00753 3051 DCA TEMP7 /SAVE
00754 1162 TAD ENTNO /GET FILE #
00755 7640 SZA CLA /IF TTY,BUFFER FIELD IS 0
00756 6211 CDF 10
00757 1050 TAD TEMP6 /GET WORD TO STORE AGAIN
00760 3451 DCA I TEMP7 /STORE IT IN BUFFER
00761 6201 CDF0, CDF
00762 1563 TAD I WORD0 /HEADER WORD
00763 0370 AND K7737 /TURN OFF BLOCK WRITTEN BIT
00764 1367 TAD K40 /TURN IT ON AGAIN
00765 3563 DCA I WORD0
00766 5747 JMP I BCPUT /RETURN
```

```
00767 0040 K40, 40
00770 7737 K7737, 7737
```

/ROUTINE TO SET STM DF TO 0

```
TEMP30,
00771 0000 SRESET, 0
00772 1361 TAD CDF0
00773 3775 DCA I STHDKK
00774 5771 JMP I SRESET
00775 2603 STHDKK, STHDF
```


1000

PAGE

/TELETYPE DRIVING ROUTINE

/2 ENTRY POINTS-XPUTCH PUTS A CHARACTER IN THE RING BUFFER

/ XPRINT TYPES A CHARACTER IF POSSIBLE

/ AND RETURNS TO CALL+1 IF THERE

/ ARE MORE CHARCTERS IN THE BUFFER,CALL+2

/ IF THE BUFFER IS EMPTY

/THE IDEA IS THE PLACE CALLS TO XPRINT AT VARIOUS POINTS IN THE INTER-

/PRETER AND THUS KEEP THE TTY BUSY WITHOUT WASTING THE TIME WAITING FOR

/THE TTY FLAG, THE SUCCESS OF THIS SCHEME DEPENDS HEAVILY ON THE NUMBER

/AND PLACEMENT OF THE CALLS TO XPRINT.

```

01000 0000 XPUTCH, 0
01001 3040          DCA CHRSAV      /SAVE THE CHARACTER
01002 2017 XPUT1,  ISZ SPINNR      /SPIN RANDOM # SEED
01003 4234          JMS XPRINT     /START A CHAR IF POSSIBLE
01004 7000          NOP
01005 1230          TAD BCNT        /GET THE NUMBER OF AVAILABLE SLOTS
01006 7650          SNA CLA        /ARE THERE ANY?
01007 5202          JMP XPUT1      /NO-TRY TO RPINT 1 AND FREE UP A SPACE
01010 1040 PUTCHR, TAD CHRSAV      /GET CHARACTER AGAIN
01011 3625          DCA I BUFIN    /PUT CHARACTER IN RING BUFFER
01012 2225          ISZ BUFIN      /BUMP BUFEER POINTER OF INPUT
01013 7340          CLA CLL CMA    /=1 IN AC
01014 1230          TAD BCNT        /DECREMENT AVAILABLE SLOT COUNT
01015 3230          DCA BCNT
01016 1225          TAD BUFIN      /GET BUFFER INPUT POINTER
01017 1231          TAD MBEND      /SUBTRACT ADDR OF END OF BUFFER
01020 7750          SPA SNA CLA    /PAST EDN OF BUFFER?
01021 5600          JMP I XPUTCH   /NO-RETURN
01022 1227          TAD BSTRTA     /YES-RESET INPUT POINTER TO BEGINNING OF BUFFER
01023 3225          DCA BUFIN
01024 5600          JMP I XPUTCH   /RETURN

01025 1070 BUFIN,  BSTRT          /POINTER TO NEXT SLOT FOR BUFFER INPUT
01026 1070 BUFOUT, BSTRT         /POINTER TO NEXT CHARACTER TO BE PRINTED
01027 1070 BSTRTA, BSTRT         /ADDR OF START OF TTY BUFFER
01030 0050 BCNT,  50             /# OF AVAILABLE SLOTS IN BUFFER (40 INITIALLY)
01031 0040 CHRSAV=TEMP1
01031 6641 MBEND,  -BEND          /-ADDR OF END OF RING BUFFER
01032 7575 MCTRLC, -203
01033 7730 M50,  -50

01034 0000 XPRINT, 0
01035 6031          KSF            /IS KEYBOARD FLAG UP?
01036 5244          JMP NOCC       /NO-NO CHANCE FOR A CTRL/C
01037 1077          TAD K0200      /FORCE PARAITY BIT
01040 6034          KRS            /YES-GET THE CHAR IN KEYBOARD BUFFER
01041 1232          TAD MCTRLC     /IS IT CTRL/C
01042 7650          SNA CLA
01043 5561          JMP I FSTOP1   /YES-ABORT TO EDITOR
01044 1230 NOCC,   TAD BCNT        /# OF AVAILABLE SLOTS IN BUFFER

```

```

01045 1233      TAD M50      /IS BUFFER EMPTY?
01046 7650      SNA CLA
01047 5266      JMP RECP2    /YES=RETURN TO CALL+2
01050 6041      TSF         /NO-TTY FLAG UP YET?
01051 5634      JMP I XPRINT /NO-GO ABOUT YOUR BUSINESS
01052 1626      TAD I BUFOUT /GET NEXT CHARACTER

/*****
/N.B. BECAUSE OF THE ABOVE INSTRUCTION,THE DF MUST BE SET TO THE
/INTERPRETER FIELD WHENEVER XPRINT IS CALLED. WATCH YOUR HOOK PLACEMENT!
*****/

01053 6046      TLS         /TYPE IT
01054 7300      CLA CLL
01055 2226      ISZ BUFOUT   /BUMP BUFFER OUTPUT POINTER
01056 1226      TAD BUFOUT   /GET OUTPUT POINTER
01057 1231      TAD MBEND    /SUBTRACT END OF BUFFER
01060 7750      SPA SNA CLA  /IS OUTPUT POINTER PAST END?
01061 5264      JMP BOUTRS   /NO=FREE UP A SPOT
01062 1227      TAD BSTRTA   /YES=RESET POINTER TO BEGINNING
01063 3226      DCA BUFOUT
01064 2230      BOUTRS, ISZ BCNT /INCREMENT # OF FREE SLOTS (WE JUST PRINTED ONE)
01065 5634      JMP I XPRINT /RETURN

01066 2234      RECP2, ISZ XPRINT /BUMP RETURN
01067 5634      JMP I XPRINT /RETURN TO CALL+2 FOR EMPTY BUFFER

```

/TELETYPE RING BUFFER

```

01070 0000      BSTR, 0      /START OF BUFFER
01071 0000      0
01072 0000      0
01073 0000      0
01074 0000      0
01075 0000      0
01076 0000      0
01077 0000      0
01100 0000      0
01101 0000      0
01102 0000      0
01103 0000      0
01104 0000      0
01105 0000      0
01106 0000      0
01107 0000      0
01110 0000      0
01111 0000      0
01112 0000      0      /40 CHARACTERS LONG
01113 0000      0
01114 0000      0
01115 0000      0
01116 0000      0
01117 0000      0
01120 0000      0
01121 0000      0
01122 0000      0

```

01123	0000	0
01124	0000	0
01125	0000	0
01126	0000	0
01127	0000	0
01130	0000	0
01131	0000	0
01132	0000	0
01133	0000	0
01134	0000	0
01135	0000	0
01136	0000	0
01137	0000	0

BEND,

/END OF TTY BUFFER

/LINE NUMBERS

```

01140 1064 LINE1, TAD INSAV /GET INSTRUCTION
01141 3065 DCA LINEHI /SAVE
01142 4520 JMS I PWFECL /GET WORD FOLLOWING LINE # INST
01143 3066 DCA LINELO /SAVE AS LOW ORDER LINE #
01144 5513 TRHOOK, JMP I ILOOP /RETURN TO I-LOOP
01145 1351 TAD KC240 /IF TRACE IS ON,FAKE CALL
01146 3064 DCA INSAV /TO FUNC2,#12
01147 5750 JMP I .+1
01150 1472 FUNC2I /DISPATCH TO TRACE FUNCTION

```

```

////////////////////
////////// INTERMEDIATE TELETYPE BUFFER //////////
////////////////////
/USED TO BUFFER OUTPUT FROM FPP BEFORE WE PUT IT
/IN BASIC FORMAT FOR TRANSPORTATION TO THE TTY RING
/BUFFER FILLED WITH INITIALIZATION CODE WHEN ENTERED

```

```

01151 0240 KC240, 240 /STOPPER TO MARK BEGINNING OF BUFFER
INTERB,
START3, TAD COFPS /CDF FOR PSEUDO-CODE 6211 for 8K Sys
01152 1025 DCA I COFPSL /PUT IN-LINE TO ILOOP
01153 3515 TAD PSSTRY /START OF PSEUDO-CODE
01154 1026 DCA I INTPCK /PUT INTO PC
01155 3775 JMS I FCLR /ZERO FAC
01156 4537 TAD CDFIO /CDF FOR SYMBOL TABLE FIELD 6211 for 8K Sys
01157 1020 DCA I STDFL /PUT IN LINE FOR STRING FUNCTIONS
01160 3774 FPPTM5, TAD CDFIO /CDF FOR SYMBOL TABLES
01161 1020 DCA I ATABDL /PUT IN LINE FOR ARRAY CALCULATIONS
01162 3773 TAD CDFIO /CDF FOR SCALAR TABLE
01163 1020 FPPTM4, DCA I SCALDL /PUT IN LINE FOR ARGPRE
01164 3777 TAD CDFIO
01165 1020 DCA I DLCDL /DATA FIELD FOR DATA LIST
01166 3776 FPPTM3, TAD DLSTRY
01167 1030 DCA DLPTR /DO A RESTORE IN INCORE DATA LIST
01170 3016 JMP I .+1 /CONTINUE INITIALIZATION CODE IN TTY INPUT BUFFER
01171 5772 FPPTM2, START4
01172 6600 ATABDL, ATABDF
01173 0610 STDFL, STDF
01174 1701 FPPTM1, /FLOATING POINT TEMPORARY
01175 0302 INTPCK, INTPC
01176 2317 DLCDL, DLCDL
01177 0317 SCALDL, SCALDF

```

```

////////////////////

```

1200

PAGE

/OPERATE CLASS INSTRUCTIONS

01200	1064	OPERI,	TAD INSAV	/GET OPERATE INSTRUCTION
01201	0074		AND K0017	/MASK OFF OPERATE OPCODE
01202	1205		TAD JMP13	/BUILD JUMP OFF OPERATE JUMPTABLE
01203	3204		DCA .+1	/STORE THE JUMP IN LINE
01204	1204		,	/DISPATCH TO PROPER OPERATE ROUTINE
01205	5606	JMP13,	JMP I .+1	/JUMP TO OPERATE ROUTINE CALLED FOR

/OPERATE JUMP TABLE

01206	1542		FUNC3I	/CALL RESIDENT FUNCTION	OPCODE 0
01207	1600		SPFUNC	/SPECIAL FUNCTIONS	OPCODE 1
01210	2000		SFN	/SET FILE NUMBER	OPCODE 2
01211	1226		FNEGI	/NEGATE FAC	OPCODE 3
01212	6550		RETRNI	/GOSUB RETURN	OPCODE 4
01213	2572		RESTOR	/RESTORE DEVICE	OPCODE 5
01214	1403		LSUB1I	/LOAD S1 FROM FAC	OPCODE 6
01215	1413		LSUB2I	/LOAD S2 FROM FAC	OPCODE 7
01216	0020	MSPACE,	20	/THIS OPCODE NOT DEFINED,SO WE PUT A CONST HERE	
01217	3105		READI	/READ DEVICE	OPCODE 11
01220	3200		WRITEI	/WRITE DEVICE	OPCODE 12
01221	2460		SWRITE	/STRING WRITE	OPCODE 13
01222	1471		FUNC5I	/CALL FILE FUNCTION	OPCODE 14
01223	1466		FUNC4I	/CALL USER FUNCTION	OPCODE 15
01224	1473		FUNC1I	/CALL FUNCTIONS 1	OPCODE 16
01225	1472		FUNC2I	/CALL FUNCTIONS 2	OPCODE 17

/

/FLOATING NEGATE

01226	4540	FNEGI,	JMS I FNEGL	/CALL NEGATE ROUTINE
01227	5513		JMP I ILOOP	/RETURN TO ILOOP

/ROUTINE TO SWAP PG 17600 WITH N7400 OR N7600 (WHICHEVER THE CASE MAY BE)
/WHERE N IS THE HIGH CORE FIELD

40
1262 →

01230	0000	PSWAP,	0	
01231	1253	TAD KK7600		/POINTER TO 17600 AND COUNTER Patch for 8K 5256
01232	3040	DCA TEMP1		
01233	1031	TAD PSFLAG		/GET RESIDENT STATUS FLAG Initial 0
01234	7700	SMA CLA		/WHICH HI-CORE PAGE IS IT IN?
01235	1077	TAD K200		/7600
01236	1103	TAD K7400		/7400
01237	3007	DCA TEMP2		/POINTER TO HIGH CORE
01240	7402	P1CDF, HLT	CDF 10, 8K*	/DF TO HI CORE 6221 12K CDF 20
01241	1407	TAD I TEMP2		/GET WORD FROM HI CORE
01242	3043	DCA TEMP4		/SAVE IT
01243	6211	P2CDF, CDF	10	
01244	1440	TAD I TEMP1		/GET WORD FROM 17600
01245	7402	P1CDF1, HLT	CDF 10, 8K*	/DF TO HI CORE AGAIN 6221 12K CDF 20
01246	3407	DCA I TEMP2		/PUT 17600 WORD IN HI CORE
01247	6211	P2CDF1, CDF	10	
01250	1043	TAD TEMP4		/GET SAVED HI CORE WORD
01251	3440	DCA I TEMP1		/AND PUT IN 17600
01252	2007	ISZ TEMP2		/BUMP HI CORE POINTER
01253	7600	KK7600, 7600		/CLA
01254	2040	ISZ TEMP1		/BUMP 17600 POINTER AND CHECK FOR DONE
01255	5240	JMP P1CDF		/NO DONE-MOVE NEXT WORD
01256	6201	CDF		
01257	5630	JMP I PSWAP		/DONE-RETURN

* Set by Init code starting @ 321

/SUBROUTINE ASCOUT

/ROUTINE CALLED BY WRITE WITH THE NUMBER TO BE WRITTEN IN FAC.

/CALLS THE FPP TO OUTPUT THE DIGITS TO AN INTERMEDIATE BUFFER, THEN

/MESSAGES THAT BUFFER TO PUT OUTPUT IN BASIC FORMAT.

```

01260 0000  ASCOUT, 0
01261 4532      JMS I FACSAL      /SAVE THE FAC
01262 1045      TAD HORD          /GET HI MANTISSA
01263 7650      SNA CLA          /IS NUMBER 0?
01264 5302      JMP FFORMT       /YES=USE F FORMAT
01265 4776      JMS I ABSVLL      /ABS(X)
01266 4771      JMS I FSUBLK      /ABS(X)=999999
01267 2514      A999
01270 1045      TAD HORD          /GET HI MANTISSA OF RESULT
01271 7740      SMA SZA CLA      /IS ABS(X)>999999?
01272 5307      JMP E20P10       /YES=USE E FORMAT FOR OUTPUT
01273 4533      JMS I FACREL      /GET X AGAIN
01274 4776      JMS I ABSVLL      /ABS(X)
01275 4771      JMS I FSUBLK      /ABS(X)=.000001
01276 2175      AP0001
01277 1045      TAD HORD
01300 7710      SPA CLA          /IS ABS(X)>.000001?
01301 5307      JMP E20P10       /NO=USE E FORMAT
01302 1073  FFORMT, TAD K0010
01303 3060      DCA DADP          /8 PLACES AFTER DEC PT
01304 1216      TAD MSPACE
01305 3057      DCA FLDW          /16 COLUMNS IN FIELD WIDTH
01306 7001      IAC              /SET FLAG FOR F FORMAT
01307 3056  E20P10, DCA EFLG      /SET FORMAT FLAG
01310 4533      JMS I FACREL      /GET X BACK IN FAC
01311 1372      TAD INTRB        /ADDR OF INTERMEDIATE BUFFER-1
01312 3013      DCA XR3          /XR3 POINTS TO INTERMEDIATE BUFFER
01313 4773      JMS I FFOUTL      /USE FPP TO PUT ASCII NUMBER IN INTERMEDIATE BUFFER
01314 7240      CLA CMA          /-1 IN AC
01315 1013      TAD XR3          /ADDR OF LF IN INTER BUFFER-1
01316 3061      DCA TEMP10       /TEMP10 POINTS TO CR IN BUFFER
01317 3007      DCA TEMP2        /CLEAR CHARACTER COUNT
01320 3042      DCA TEMP3        /CLEAR ZERO REPLACE FLAG
01321 3043      DCA TEMP4        /CLEAR DECIMAL POINT SEEN FLAG
01322 7240  CFETCH, CLA CMA      /-1 IN AC
01323 1061      TAD TEMP10
01324 3061      DCA TEMP10       /BACK UP POINTER TO NEXT CHAR
01325 1461      TAD I TEMP10      /GET CHAR FROM BUFFER
01326 1374      TAD M260         /="0"
01327 7450      SNA              /IS IT "0"?
01330 5342      JMP ZR           /YES-REPLACE WITH CR IF ZERO FLAG NOT SET
                                /OR ALTMODE IF IN E FORMAT AND DECPT HAS BEEN SEEN.
01331 1216      TAD MSPACE       /IS IT " "?
01332 7450      SNA
01333 5660      JMP I ASCOUT      /YES-DONE-PREPARE THE NUMBER FOR TYPING
01334 2007  ZROFF, ISZ TEMP2      /NO-BUMP CHAR COUNT
01335 1375      TAD MDECPT       /IS IT "."?
01336 7650      SNA CLA

```

01337	5363		JMP COUNCK	/YES-IF COUNT=0,REPLACE WITH CR
01340	2042		ISZ TEMP3	/NO-TURN OF ZERO REPLACE
01341	5322		JMP CFETCH	/NEXT
01342	1056	ZR,	TAD EFLG	/YES-GET FORMAT FLAG
01343	7640		SZA CLA	/ARE WE IN E FORMAT?
01344	5352		JMP ZRCONT	/NO-PROCEED TO CHECK ZERO REPLACE FLAG
01345	1043		TAD TEMP4	
01346	7650		SNA CLA	/HAS DECIMAL POINT BEEN SEEN YET?
01347	5334		JMP ZROFF	/NO-THIS ZERO STAYS,SO COUNT IT
01350	1101		TAD K0377	/YES-THIS IS THE ZERO BEFORE THE POINT
01351	5361		JMP CRREP+1	/SO REPLACE IT WITH AN ALTMODE
01352	1043	ZRCONT,	TAD TEMP4	/HAS A PERIOD BEEN SEEN YET?
01353	7640		SZA CLA	
01354	5334		JMP ZROFF	/YES-THIS ZERO STAYS
01355	1042		TAD TEMP3	/GET ZERO REPLACE FLAG
01356	7640		SZA CLA	/IS IT ON?
01357	5334		JMP ZROFF	/YES-DON'T REPLACE ZEROES
01360	1054	CRREP,	TAD K215	/NO-REPLACE THIS ZERO WITH A CR
01361	3461		DCA I TEMP10	/YES-REPLACE 0 WITH CR
01362	5322		JMP CFETCH	/NEXT CHAR
01363	2043	COUNCK,	ISZ TEMP4	/SET DECIMAL POINT SEEN FLAG
01364	7240		CLA CMA	/-1 IN AC
01365	1007		TAD TEMP2	/GET CHAR COUNT
01366	7640		SZA CLA	/IS IT 1 (. WAS FIRST COUNTED CHAR)?
01367	5322		JMP CFETCH	/NO-DON'T REPLACE , WITH CR
01370	5360		JMP CRREP	/YES-REPLACE , WITH CR
01371	6117	FSUBLK,	FFSUB	
01372	1151	INTRB,	INTERB-1	
01373	4600	FFOUTL,	FFOUT	
01374	7520	M260,	=260	
01375	7762	MDECPT,	=16	
01376	2366	ABSVLL,	ABSVAL	

1400

PAGE

/LOAD SUBSCRIPT 1

01400	4532	LS1I,	JMS I FACSAL	/PRESERVE FAC
01401	4620		JMS I ARGPRL	/GET ARG POINTER INTO AC
01402	4534		JMS I FGETL	/LOAD ARG INTO FAC (SKIPS NEXT INST ON RETURN)
01403	4532	LSUB1I,	JMS I FACSAL	/SAVE THE FAC
01404	4514		JMS I INTL	/GET INT(FAC)
01405	3033		DCA S1	/SET RESULT AS SUBSCRIPT 1
01406	4533		JMS I FACREL	/RESTORE FAC
01407	5513		JMP I ILOOP	/NEXT INSTRUCTION

/LOAD SUBSCRIPT 2

01410	4532	LS2I,	JMS I FACSAL	/PRESERVE FAC
01411	4620		JMS I ARGPRL	/GET ARG POINTER INTO AC
01412	4534		JMS I FGETL	/LOAD ARG INTO FAC (SKIPS NEXT INST ON RETURN)
01413	4532	LSUB2I,	JMS I FACSAL	/SAVE THE FAC
01414	4514		JMS I INTL	/GET INT(FAC)
01415	3034		DCA S2	/SET RESULT AS SUBSCRIPT 2
01416	4533		JMS I FACREL	/RESTORE THE FAC
01417	5513		JMP I ILOOP	/BACK TO ILOOP
01420	0307	ARGPRL,	ARGPRE	

/JMP DISPATCH FOR FUNC1 CALLS

01421 4622 JMSI4, JMS I .+1 /CALL FOR CANNED FUNCTION SET 1

/JUMP TABLE FOR FUNCTION CALL 1

01422	4200	FFATN	/FUNCTION BITS= 0
01423	4053	FFCOS	/ 1
01424	4120	FFEXP	/ 2
01425	3477	EXPON	/ 3
01426	3400	INT	/ 4
01427	4263	FFLOG	/ 5
01430	3632	SGN	/ 6
01431	4000	FFSIN	/ 7
01432	4543	RND	/ 10
01433	3646	FROOT	/ 11

/JUMP FOR FUNC2 DISPATCH

01434 4635 JMSI5, JMS I .+1 /JMS OFF THE SET 2 TABLE

/JUMP TABLE FOR FUNCTION SET 2

01435	3407	ASC	/FUNCTION BITS= 0
01436	3400	CHR	/ 1
01437	3600	DATE	/ 2
01440	3414	LEN	/ 3
01441	4400	POS	/ 4
01442	4266	SEG	/ 5
01443	3422	STR	/ 6
01444	3462	VAL	/ 7
01445	4013	ERRRR	/ 10
/ERRRR MUST BE FUNCTION #10, ELSE "ERROPC" MUST CHANGE			
01446	4000	TRACE	/ 11
01447	3677	TPRINT	/ 12
/TPRINT MUST BE #12 OR TRHOOK+1 MUST CHANGE			

/DISPATCH FOR FUNC5 CALLS

01450 5651 JMPFIL, JMP I .+1 /CALL FORR FILE MANIPULATING FUNCTIONS

/JUMP TABLE FOR FILE FUNCTIONS

5A

1450	01451	3600 ✓	CHAIN	/FUNCTION BITS= 0
1450	01452	3405-6	CLOSE	/ 1
1451	01453	4001-	OPENAF	/ 2
1452	01454	4000-	OPENAV	/ 3
3	01455	4004-	OPENNF	/ 4
4	01456	4003-	OPENNV	/ 5
5	01457	3704-3737	FSTOP	/INT. EXIT 6

/ROUTINE TO CALL ERROR ROUTINE BY FAKING A FUNC2 CALL TO FUNCTION #10

01460	0000	ERRDIS, 0	
01461	7300	CLACLL	/FLUSH

/OS/8 BASIC RUNTIME SYSTEM

PAL8-V7 10/24/72 PAGE 23-1

01462 1327
01463 3064
01464 5272

TAD LK607
DCA INSAV
JMP FUNC27

/FAKE A FUNC CALL TO FUNC2 #10

/ERROR CALL FOR USER FUNCTIONS-USER FUNCTION SHOULD "JMS I IAL"

01465 4260 IA, JMS ERDIS

156
USA

/FUNCTION OVERLAY DRIVER

```

01466 4510 FUNC4I, JMS I PRINT      /PURGE TTY RING BUFFER
01467 5266          JMP .-1        /BEFORE CALLING USER FUNCTION
01470 7001          IAC            /LOOK FOR OVERLAY FLAG=3
01471 7001 FUNC5I, IAC            /LOOK FOR OVERLAY FLAG=2
01472 7001 FUNC2I, IAC            /LOOK FOR OVERLAY FLAG=1
01473 3040 FUNC1I, DCA TEMP1      /LOOK FOR OVERLAY FLAG=0
01474 6201          CDF            /OF TO THIS FIELD
01475 1040          TAD TEMP1      /GET OVERLAY # AGAIN
01476 7041          CIA            /NEGATE
01477 1330          TAD OVRLAY     /COMPARE AGAINST OVERLAY FLAG
01500 7650          SNA CLA        /IS THE ONE WE WANT ALREADY RESIDENT?
01501 5316          JMP OVDNE     /YES-JUST JUMP TO FUNCTION
01502 1040          TAD TEMP1      /NO-GET NUMBER OF OVERLAY DESIRED
01503 1326          TAD DATADI     /USE AS OFFSET TO BUILD STARTING BLOCK TAD
01504 3007          DCA TEMP2      /POINTS TO PROPER STARTING BLOCK #
01505 1407          TAD I TEMP2    /GET STARTING BLOCK FOR THIS OVERLAY
01506 3312          DCA OVADD      /PUT IN DRIVER CALL
01507 4727          JMS I L7607    /CALL SYSTEM HANDLER
01510 0500          0500          /OVERLAY 3400-4600
01511 3400          3400
01512 1512 OVADD, .              /STARTING BLOCK # OF OVERLAY
01513 4516 OE, JMS I ERROR       /I/O ERROR
01514 1040          TAD TEMP1
01515 3330          DCA OVRLAY     /CHANGE RESIDENT FLAG
01516 1040 OVDNE, TAD TEMP1      /FUNCTION #
01517 1335          TAD JMSTAD     /BUILD A TAD OF THE PROPER DISPATCH JMS
01520 3322          DCA .+2        /PUT IN LINE
01521 4517          JMS I FBITS    /GET # OF FUNCTION DESIRED
01522 1522          .              /BUILD JUMP OFF JUMP TABLE
01523 3324 FUJUMP, DCA .+1        /PUT JUMP IN LINE
01524 1524          .              /GO TO DESIRED FUNCTION
01525 5513          JMP I ILOOPL   /DONE

01526 1531 DATADI, ARITHA
01527 7607 L7607, 7607
01530 0000 OVRLAY, 0             /# OF CURRENTLY RESIDENT OVERLAY
                                   /0=ARITHMETIC,1=STRING,2=FILE,3=USER

/OVERLAY TABLE=CONTAINS STARTING BLOCK # OF SYSTEM OVERLAYS
/INITIALIZED BY LOADER Init code starting @ 321F

01531 1531 ARITHA, .              /STARTING BLOCK OF ARITHMETIC OVERLAY
01532 1532 STRNGA, .              /STARTING BLOCK OF STRING OVERLAY
01533 1533 FILEFA, .              /STARTING BLOCK OF FILE OVERLAY
01534 1534 USRA, .                /STARTING BLOCK OF USER FUNCTIONS

01535 1736 JMSTAD, TAD I TADTAB

01536 1421 TADTAB, JMSI4
01537 1434          JMSI5
01540 1450          JMPFIL

```

/OS/8 BASIC RUNTIME SYSTEM

PAL8-V7 10/24/72 PAGE 24-1

01541 1557

JMSUSR

/CALL FOR RESIDENT FUNCTION

```
01542 4517 FUNC3I, JMS I FBITS  /ISOLATE FUNCTION #
01543 1345      TAD JMSI7      /MAKE A JUMP OFF JUMP TABLE
01544 5323      JMP FUJUMP     /PUT THE JUMP IN LINE AND EXECUTE IT
```

```
01545 4746 JMSI7, JMS I .+1
```

/JUMP TABLE FOR RESIDENT FUNCTIONS

01546	2366	ABSVAL	/FUNCTION BITS= 0
01547	2517	COMMA	/ 1
01550	0373	CRFUNC	/ 2
01551	0240	ILOOPF	/ 3
01552	1757	TAB	/ 4
01553	1770	PNT	/ 5
01554	0574	USE	/ 6

```
1557      *1557  /*****N,8,****
           /THIS TABLE CANNOT BE MOVED!!!!
```

/JUMP DISPATCH FOR USER ROUTINES

```
01557 4760 JMSUSR, JMS I .+1
```

/JUMP TABLE FOR USER FUNCTIONS

01560	0240	ILOOPF	/USER FUNCTION 1
01561	0240	ILOOPF	/ 2
01562	0240	ILOOPF	/ 3
01563	0240	ILOOPF	/ 4
01564	0240	ILOOPF	/ 5
01565	0240	ILOOPF	/ 6
01566	0240	ILOOPF	/ 7
01567	0240	ILOOPF	/ 8
01570	0240	ILOOPF	/ 9
01571	0240	ILOOPF	/ 10
01572	0240	ILOOPF	/ 11
01573	0240	ILOOPF	/ 12
01574	0240	ILOOPF	/ 13
01575	0240	ILOOPF	/ 14
01576	0240	ILOOPF	/ 15
01577	0240	ILOOPF	/ 16

1600

PAGE

/SPECIAL FUNCTIONS

01600	4517	SPFUNC, JMS I FBITS	/ISOLATE FUNCTION BITS
01601	1204	TAD JMP16	/MAKE A JUMP OFF SPECIAL FUNCTION TABLE
01602	3203	DCA .+1	/PUT IN LINE
01603	1603	.	

01604	5605	JMP16, JMP I .+1	/JUMP TO SPECIAL FUNCTION ROUTINE
-------	------	------------------	-----------------------------------

/SPECIAL FUNCTION JUMP TABLE

01605	2054	SETF	/SET FSWITCH	0
01606	2342	FRANDM	/RANDOMIZE	1
01607	0561	FSTOPN	/LEAVE INTERPRETER	2
01610	2400	SRLIST	/STRING READ FROM DATA LIST	3
01611	2001	CSFN	/SET FILE # TO TTY	4
01612	6556	RDLIST	/READ DATA LIST	5
01613	2324	AMODE	/SWITCH TO A MODE	6
01614	2323	SSMODE	/SWITCH TO S MODE	7

/SUBROUTINE UNSFIX-UNSIGNED INTEGER FIX ROUTINE. FIXES A POSITIVE 12 BIT
/NUMBER OUT OF FAC MANTISSA AND LEAVES RESULT IN AC.RESULT IS AN UNSIGNED,
/12 BIT INTEGER

V3
V4
V5A

01615	0000	UNSFIX, 0	
01616	6201	CDF 0	
01617	1046	TAD LORD	/LOW MANTISSA
01620	7104	CLL RAL	/HI BIT OF LO MANTISSA TO LINK
01621	7200	CLA	
01622	1045	TAD HORD	/HIGH MANTISSA
01623	7510	SPA	/IS NUMBER POSITIVE?
01624	4516	FM, JMS I ERROR	/NO-BOO!!!
01625	7004	RAL	/SHIFT THE SIGN BIT OUT AND THE MANTISSA OVER,
01626	3045	DCA HORD	/MAKING 12 BITS OF MANTISSA AND BINARY POINT BEFORE BIT 0 <i>OK</i>
01627	1044	TAD EXP	/GET EXPONENT
01630	7750	SPA SNA CLA	/IS X>1?
01631	5615	JMP I UNSFIX	/NO-FIX IT TO 0
01632	1044	TAD EXP	/YES-GET EXPONENT
01633	1107 <i>52</i>	TAD M14	/SET BINARY POINT AT 12
01634	7450	SNA	/DONE ALREADY?
01635	5246	JMP UNSOUT	/YES
01636	7500	SMA	/NO-IS # TOO BIG?
01637	4516 <i>64</i>	FM, JMS I ERROR	/YES
01640	3044	DCA EXP	/NO-STORE COUNT
01641	1045	TAD HORD	/HI MANTISSA
01642	7110	UNSLP, CLL RAR	/SCALE RIGHT
01643	2044	ISZ EXP	/DONE?
01644	5242	JMP UNSLP	/NO
01645	5615	JMP I UNSFIX	/YES-RETURN
01646	1045	UNSLP, TAD HORD	/ANSWER IN AC
01647	5615	JMP I UNSFIX	<i>FREE</i>

/ Patch goes here

/RESTORE

01650	0000	RESTI, 0	
01651	4664	JMS I WRBLKL	/NO-WRITE CURRENT BUFFER
01652	7240	CLA CMA	/-1
01653	1570	TAD I WORD5	/STARTING BLOCK-1
01654	3565	DCA I WORD2	/SET CURRENT BLOCK #
01655	1564	TAD I WORD1	/GET BUFFER ADDRESS
01656	3566	DCA I WORD3	/USE IT TO RESET READ\WRITE POINTER
01657	1563	TAD I WORD0	/GET HEADER WORD
01660	0265	AND K7435	/CLEAR EOF BIT,BUFFER WRITTEN BIT,AND CHAR #
01661	3563	DCA I WORD0	
01662	4552	JMS I NEXREL	/READ FIRST BLOCK INTO BUFFER
01663	5650	JMP I RESTI	/DONE

01664 3350 WRBLKL, WRBLK
01665 7435 K7435, 7435

/SUBROUTINE STFIND=WHEN CALLED, IF LINK=1 STRING ARRAY TABLE IS
 /USED, IF LINK=0 STRING SYMBOL TABLE IS USED, RETURNS WITH AC SET
 /TO CDF OF OPERAND STRING, STRPTR POINTING TO THE FIRST WORD
 /IN THE STRING, AND THE MAX LENGTH OF THE STRING IS IN STRMAX. ALSO,
 /THE ACTUAL LENGTH OF THE STRING IS IN STRCNT

01666	0000	STFIND, 0	
01667	7430	SZL	/IS THIS AN ARRAY INST?
01670	5347	JMP SAFIND	/YES=POINTER IS INTO ARRAY TABLE
01671	1064	TAD INSAV	/GET INST AGAIN
01672	0101	AND K0377	/ISOLATE OPERAND POINTER
01673	3040	DCA TEMP1	/NO-SAVE OPERAND POINTER
01674	1040	TAD TEMP1	/N
01675	7104	CLL RAL	/2N
01676	1040	TAD TEMP1	/3N (3 WORDS/ENTRY)
01677	1023	TAD STSTR	/ADD BASE ADR OF STRING TABLE
01700	3012	STCOM, DCA XR2	/POINTER TO THIS ENTRY IN STRING TABLE
01701	1701	STOF, .	/DF TO THAT OF SYMBOL TABLES (SET BY START)
01702	1412	TAD I XR2	/GET POINTER TO STRING
01703	3072	DCA STRPTR	
01704	1412	TAD I XR2	/GET CDF FOR OPERAND STRING
01705	3062	DCA TEMP11	/SAVE
01706	1412	TAD I XR2	/GET MAX LENGTH OF STRING
01707	3070	DCA STRMAX	/SAVE
01710	7420	SNL	/ARRAY ELEMENT?
01711	5337	JMP SCDF	/NO-SKIP THIS SUBSCRIPT CALCULATION
01712	1033	TAD S1	/GET SUBSCRIPT
01713	3015	DCA BABS	/SET UP 12 BIT COMPARE
01714	1412	TAD I XR2	/GET DIMENSION
01715	4405	JMS I NUMCOM	/IS S1>DIMENSION?
01716	5756	JMP I SUBERL	/YES
01717	1070	TAD STRMAX	/NO-GET ELEMENT LENGTH
01720	7041	CIA	/MAKE POSITIVE
01721	7101	CLL IAC	/ROUND OFF TO NEAREST MULTIPLE OF 2
01722	7110	CLL RAR	/ DIVIDE BY TWO (COUNT/2=WORD COUNT)
01723	7101	CLL IAC	/ADD A WORD FOR HEADER
01724	3042	DCA TEMP3	/# OF WORDS IN EACH ARRAY ELEMENT
01725	1033	TAD S1	/GET SUBSCRIPT
01726	4521	JMS I MPYLNK	/S1*ELEMENT LENGTH (ASSUMES LINK UNCHANGED ON RETURN)
01727	1072	TAD STRPTR	/ARRAY OFFSET+POINTER TO A(0)
01730	3072	DCA STRPTR	/FINAL STRING POINTER
01731	7004	RAL	/CARRY TO BIT 11
01732	1050	TAD TEMP6	/ADD TO ACCUMULATED OVERLAPS FROM MULTIPLY
01733	7106	CLL RTL	
01734	7004	RAL	/PUT OVERLAP # INTO BITS 6-8
01735	1062	TAD TEMP11	/ADD TO CDF IF NECESSARY
01736	3062	DCA TEMP11	/SAVE AGAIN
01737	1062	SCDF, TAD TEMP11	/GET DF OF STRING
01740	3341	DCA .+1	/PUT IN LINE
01741	1741	.	/DF TO STRING FIELD
01742	1472	TAD I STRPTR	/GET STRING LENGTH
01743	3071	DCA STRCNT	/ACTUAL LENGTH OF STRING OPERAND
01744	1062	TAD TEMP11	/CDF TO OPERAND IN AC
01745	6201	CDF	

```
01746 5666      JMP I STFIND    /RETURN

01747 1064      SAFIND, TAD INSAV    /GET INST
01750 0355      AND K0037L    /ISOLATE OPERAND POINTER
01751 7106      CLL RTL    /4N (4 WORDS/ENTRY)
01752 1024      TAD SASRT    /USE STRING ARRAY TABLE
01753 7120      STL    /SET LINK FOR ARRAY INST
01754 5300      JMP STCOM    /RETURN TO SUBROUTINE MAINLINE

01755 0037      K0037L, 0037
01756 0623      SUBERL, SU
```

/TAB FUNCTION

```
01757 0000      TAB, 0
01760 4514      JMS I INTL    /FIX X TO INTEGER
01761 7041      CIA    /NEGATE
01762 1573      TAD I WORD10    /COMPARE DESIRED COLUMN TO REAL COLUMN
01763 7001      IAC    /BUMP BY 1 (WORD 7=COL #-1)
01764 7500      SMA    /IS X>=CURRENT COLUMN?
01765 5513      JMP I ILOOPL    /YES-THEN DO NOTHING
01766 5767      JMP I SLOVEL    /NO-AC CONTAINS # OF SPACES NEEDED TO REACH DESIRED COLUMN
```

```
01767 2547      SLOVEL, SLOVER
```

/PNT FUNCTION
/VALUE OF X SENT TO TTY

```
01770 0000      PNT, 0
01771 4514      JMS I INTL    /FIX X
01772 0376      AND K0177    /STRIP TO 7 ASCII BITS
01773 1077      TAD K0200    /FORCE CHANNEL 8
01774 4512      JMS I PUTCHL    /PUT IN FILE BUFFER
01775 5513      JMP I ILOOPL    /DONE

01776 0177      K0177, 177
```

2000

PAGE

/ROUTINE SFN-ROUTINE TO RESET POINTERS IN PAGE ZERO FILE POINTER
/AREA TO REFLECT A CHANGE IN THE CURRENT FILE NUMBER

02000	4514	SFN,	JMS I INTL	/FIX FAC TO GET FILE #
02001	3044	CSFN,	DCA EXP	/IF ENTRY IS HERE, FILE # = 0 (TTY)
02002	1044		TAD EXP	/GET NUMBER AGAIN
02003	1231		TAD KM4	/IS RESULT A LEGAL FILE #?
02004	7740		SMA SZA CLA	
02005	4516	FN,	JMS I ERROR	/NO-ERROR
02006	1044		TAD EXP	/YES-GET FILE #
02007	3162		DCA ENTNO	/SAVE AS CURRENT FILE #
02010	1044		TAD EXP	
02011	1234		TAD IOTPTR	/USE AS INDEX INTO TABLE OF MASTER POINTERS
02012	3007		DCA TEMP2	/POINTS TO FIRST WORD OF EACH I/O TABLE ENTRY
02013	1407		TAD I TEMP2	/GET POINTER TO FIRST WORD OF I/O TABLE ENTRY WE WANT
02014	3163		DCA WORD0	/PUT IN WORK AREA
02015	1107		TAD M14	/WE HAVE TO CHANGE 12 POINTERS
02016	3007		DCA TEMP2	
02017	1232		TAD WORD0A	/POINTER TO LAST ENTRY MADE
02020	3011		DCA XR1	
02021	1233		TAD WORD1A	/POINTER TO NEXT ENTRY TO BE BUILT
02022	3012		DCA XR2	
02023	1411	SFNLP,	TAD I XR1	/EACH ENTRY IS BUILT
02024	7001		IAC	/BY ADDING 1 TO THE PREVIOUS ENTRY
02025	3412		DCA I XR2	
02026	2007		ISZ TEMP2	/DONE?
02027	5223		JMP SFNLP	/NO
02030	5513		JMP I ILOOPL	/YES-NEW TABLE IS NOW BUILT
02031	7774	KM4,	-4	
02032	0162	WORD0A,	WORD0-1	
02033	0163	WORD1A,	WORD1-1	
02034	2035	IOTPTR,	IOTAB	
02035	6677	IOTAB,	TTYF	/POINTERS TO THE FIRST WORD IN EACH OF THE FIVE
02036	6714		FILE1	/I/O TABLE ENTRIES
02037	6731		FILE2	
02040	6746		FILE3	
02041	6763		FILE4	

/FOR-LOOP JUMP ROUTINE

02042	7300	JFOR,	CLA CLL	
02043	1045		TAD HORD	/GET HIGH MANTISSA
02044	7450		SNA	/IS FAC=0?
02045	5653		JMP I JFAILL	/YES-DO NOT JUMP
02046	1260		TAD FSWITC	/ADD FSWITCH
02047	7710		SPA CLA	/ARE SIGN BIT=FSWITCH?
02050	5653		JMP I JFAILL	/NO-DO NOT JUMP
02051	5652		JMP I SUCJML	/YES-DO JUMP
02052	0421	SUCJML,	SUCJMP	
02053	0413	JFAILL,	JFAIL	
/ROUTINE TO INITIALIZE FSWITCH				
02054	7130	SETF,	CLL CML RAR	/4000 IN AC
02055	0045		AND HORD	/ISOLATE SIGN OF MANTISSA
02056	3260		DCA FSWITC	/STORE IN FSWITCH
02057	5513		JMP I ILOOP	/DONE
02060	0000	FSWITC,	0	

/STRING COMPARE

02061	3346	SCOMP,	DCA OCDF	/OF TO OPERNAD IN LINE
02062	3063		DCA MODESW	/RETURN IN AMODE
02063	4537		JMS I FCLR	/INITIALIZE FAC TO 0
02064	1032		TAD STRLEN	/LENGTH OF STRING IN SAC
02065	1071		TAD STRCNT	/LENGTH OF OPERAND
02066	7650		SNA CLA	/ARE THEY BOTH ZERO?
02067	5513		JMP I ILOOPL	/YES-THEY ARE EQUAL,SO RETURN WITH FAC=0
02070	7100		CLL	
02071	1032		TAD STRLEN	/NO-LENGTH OF SAC
02072	7650		SNA CLA	/IS IT ZERO?
02073	5326		JMP SNEQ=1	/YES-THEN THEY ARE NOT EQUAL
02074	1071		TAD STRCNT	/LENGTH OF OPERAND
02075	7650		SNA CLA	/IS IT EMPTY
02076	5327		JMP SNEQ	/YES-THEY ARE NOT EQUAL
02077	1111		TAD SACPTR	/POINTER INTO SAC
02100	7101		CLL IAC	
02101	4527		JMS I LDHINL	/INIT LDH
02102	4557		JMS I LDHRST	/TO LOAD FROM SAC
02103	4366		JMS PTRBMP	/ISZ STRPTR OVER COUNT WORD
02104	3042		DCA SWITCC	/INIT LDHPST
02105	4343	LDHC,	JMS LDHPST	/HALF LOAD
02106	3007		DCA TEMP2	/AND SAVE
02107	4531		JMS I LDH	/GET CHAR FROM SAC
02110	7141		CIA CLL	/NEGATE IT
02111	1007		TAD TEMP2	/AND COMPARE TO OPERAND CHARACTER
02112	7640		SZA CLA	/ARE THEY EQUAL?
02113	5327		JMP SNEQ	/NO-RETURN WITH FAC SIGN SET APPROPRIATELY
02114	2071		ISZ STRCNT	/MORE OPERAND CHARS?
02115	5332		JMP SACCHK	/YES-SEE IF SAC EMPTY
02116	2032	SAC40C,	ISZ STRLEN	/MORE CHARS IN SAC?
02117	7410		SKP	/YES
02120	5513		JMP I ILOOPL	/STRINGS ARE EQUAL-RETURN WITH 0 FAC
02121	4531		JMS I LDH	/GET CHAR FROM SAC
02122	7100		CLL	
02123	1106		TAD KM40	/COMPARE TO SPACE
02124	7650		SNA CLA	/IS IT A SPACE?
02125	5316		JMP SAC40C	/YES-CHECK NEXT CHAR
02126	7020	SNEQ1,	CML	
02127	7270	SNEQ,	CLA CMA CML RAR	
02130	3045		DCA HORD	/SET SIGN BIT OF MANTISSA TO REFLECT RESULTS OF COMPARE
02131	5513		JMP I ILOOPL	
02132	2032	SACCHK,	ISZ STRLEN	/SAC EMPTY?
02133	5305		JMP LDHC	/NO-COMPARE NEXT TWO CHARS
02134	4343	STC40C,	JMS LDHPST	/YES-GET CHAR FROM OPERAND
02135	1106		TAD KM40	/COMPARE TO SPACE
02136	7640		SZA CLA	/IS IT A SPACE?
02137	5326		JMP SNEQ1	/NO-STRINGS AREN'T EQUAL
02140	2071		ISZ STRCNT	/YES-MORE CHARS?
02141	5334		JMP STC40C	/YES-CHECK THEM
02142	5513		JMP I ILOOPL	/NO-STRINGS ARE EQUAL-RETURN WITH FAC=0

/ROUTINE TO GRAB 1 CHAR AT A TIME FROM OPERAND STRING

```
02143 0000 LDHPST, 0
02144 1042      TAD SWITCC      /GET HALF SWITCH
02145 7110      CLL RAR        /PUT IN LINK
02146 2146      OCDF, .        /DF TO OPERAND
02147 1472      TAD I STRPTR   /GET TWO CHARS FROM STRING
02150 6201      CDF
02151 7420      SNL            /RIGHT HALF?
02152 4544      JMS I BSWL     /NO-SWAP BYTES
02153 0075      AND K0077     /ISOLATE RIGHT CHAR
02154 3007      DCA TEMP2     /SAVE
02155 1042      TAD SWITCC
02156 7110      CLL RAR        /HALFWORD SWITCH TO LINK
02157 7430      SZL            /RIGHT HALF?
02160 4366      JMS PTRBMP     /BUMP STRING POINTER
02161 7620      SNL CLA        /FLIP HALFWORD SWITCH
02162 7121      CLL CML IAC    /((LEAVE LINK=1)
02163 3042      DCA SWITCC
02164 1007      TAD TEMP2     /GET CHAR AGAIN
02165 5743      JMP I LDHPST
```

0042 SWITCC=TEMP3

/SUBROUTINE TO BUMP STRPTR AND WATCH FOR FIELD OVERLAP

```
02166 0000 PTRBMP, 0
02167 2072      ISZ STRPTR
02170 5766      JMP I PTRBMP   /NO-SKIP/RETURN
02171 1346      TAD OCDF       /SKIP MEANS WE MUST INCREMENT FIELD
02172 1073      TAD K0010
02173 3346      DCA OCDF
02174 5766      JMP I PTRBMP
```

/FLOATING POINT CONSTANT USED BY ASCOUT

```
02175 7755 AP0001, 7755      /,000001
02176 2061      2061
02177 5734      5734
```

2200

PAGE

/STRING CONCATENATE

```

02200 3637 SCON1, DCA I LDHCF /OF FOR LDH
02201 1071 TAD STRCNT /OPERAND=0?
02202 7650 SNA CLA
02203 5513 JMP I ILOOPL /YES-THEN THERE IS NOTHING TO DO
02204 1072 TAD STRPTR /ADDR OF OPERAND
02205 7101 CLL IAC /ADDR OF OPERAND 1ST CHARACTER
02206 4527 JMS I LDHINL /INITIALIZE LDH TO PULL FROM OPERAND
02207 1032 TAD STRLEN /# OF CHARS IN AC
02210 7450 SNA /SAC EMPTY?
02211 5216 JMP SACEM /YES-CONCAT ESSENTIALLY IS A LOAD
02212 7130 CLL CML RAR /DIVIDE BY TWO
02213 7440 SZA
02214 7041 CIA /POSITIVE WORD COUNT
02215 7420 SNL
02216 7001 SACEM, IAC
02217 1111 TAD SACPTR /USE AS DISPLACEMENT OFF START OF SAC
02220 4526 JMS I STHINL /INITIALIZE STH TO SAC+STRLEN/2
02221 4560 JMS I STHRST /SAC IS IN FLD 0
02222 4531 SEGCOM, JMS I LDH /GET CHAR FROM OPERAND
02223 4530 JMS I STH /PUT CHAR IN SAC
02224 7240 CLA CMA /-1
02225 1032 TAD STRLEN /"BUMP" STRING COUNT FOR SAC
02226 3032 DCA STRLEN
02227 1032 TAD STRLEN
02230 1236 TAD K110 /IS SAC FULL YET?
02231 7710 SPA CLA
02232 4516 SC, JMS I ERROR /YES-TRUNCATION ERROR
02233 2071 ISZ STRCNT /NO-MORE CHARS LEFT IN OPERAND?
02234 5222 JMP SEGCOM /YES-GO GETTEM
02235 5513 JMP I ILOOPL /NO-DONE
02236 0110 K110, 110
02237 2647 LDHCF, LDHDF

```

/ROUTINE TO SIMULATE HARDWARE BYTE SWAP

V3
V4

```

02240 0000 BSWP, 0
02241 7112 CLL RTR
02242 7012 RTR
02243 7012 RTR /LEFT HALF NOW IN RIGHT HALF
02244 3311 DCA TEMP12 /SAVE
02245 1311 TAD TEMP12
02246 0075 AND K0077 /ISOLATE LEFT HALF
02247 1311 TAD TEMP12 /DO A PARTIAL SHIFT OF BITS 6-11 LEFT ONE
02250 7010 RAR /MOVE INTO POSITION
02251 5640 JMP I BSWP

```

Free

```

/ROUTINE TO SET EOF BIT IN I/O ENTRY
02252 1563 EOFSET, TAD I WORD0 /HEADER
02253 7112 CLL RTR /EOF BIT TO LINK
02254 7120 CLL CML /SET LINK
02255 7006 RTL /PUT LINK IN EOF BIT
02256 3563 DCA I WORD0 /STORE IN I/O TABLE ENTRY
02257 5513 JMP I ILOOP /EOF BIT SET-ABORT TO ILOOP

```

```

/SUBROUTINE MPY= 12 BIT BY 12 BIT MULTIPLY, MULTIPLIES THE CONTENTS
/OF TEMP3 BY THE CONTENTS OF THE AC, LEAVING THE HI RESULT IN TEMP6
/AND THE LOW RESULT IN THE AC

```

V3
V4
USA

```

-----
022603 0000 MPY, 0
022604 3061 54 DCA TEMP10
022605 3050 DCA TEMP6
022606 1107 52 TAD M14
022607 3047 DCA TEMP5
022608 1042 MP12LP, TAD TEMP3
022609 7010 RAR
022610 23042 DCA TEMP3
022611 1050 TAD TEMP6
022612 7420 SNL
022613 5273 300 JMP ,+3 /12 BIT MULTIPLY USED TO FIND (DIM1+1)*S2
022614 7100 CLL
022615 1064 54 TAD TEMP10
022616 7010 RAR
022617 3050 DCA TEMP6
022618 2047 ISZ TEMP5
022619 5263 70 JMP MP12LP
022620 1042 TAD TEMP3 /LORD OF (DIM1+1)*S2 IN AC
022621 7010 RAR /WORD OF (DIM1+1)*S2 IN TEMP6
022622 5660 3 JMP I MPY /RETURN
-----

```

Patch

EAE

2260 TEMP13=MPY

```

/ROUTINE TO CHECK IF FILE IDLE

```

```

02304 0000 IDLE, 0
02305 1567 TAD I WORD4 /GET HANDLER ENTRY
02306 7650 SNA CLA /IS IT EMPTY?
02307 4516 FI, JMS I ERROR /YES-USER TRIED TO DO SOMETHING TO AN UNOPEN FILE
02310 5704 JMP I IDLE /NO-RETURN

```


/ROUTINE TO READ NEXT WORD IN DATALIST INTO AC

```
TEMP12,
DLREAD, 0
02311 0000
02312 1016 TAD DLPTR /DATA LIST POINTER
02313 3015 DCA BABS /SET UP 12 BIT COMPARE
02314 1027 TAD DLSTOP /ADDR OF END OF DATA LIST
02315 4405 JMS I NUMCOM /POINTER AT END OF LIST?
02316 4516 DA, JMS I ERROR /YES
02317 2317 DLCDF, /NO-OF TO DATA LIST
02320 1416 TAD I DLPTR /FETCH WORD FROM DATA LIST
02321 6201 CDF
02322 5711 JMP I DLREAD /DONE
```

/ROUTINES TO SWITCH INTERPRETER MODE

```
02323 7001 SSMODE, IAC /SET SWITCH TO SMODE
02324 3063 AMODE, DCA MODESW /SET SWITCH TO A MODE
02325 5513 JMP I ILOOPL /DONE
```

/SUBROUTINE PUSHG

/ROUTINE TO PUSH AC ON TOP OF GOSUB STACK

```
02326 0000 PUSHG, 0
02327 3040 DCA TEMP1 /SAVE ELEMENT TO BE PUSHED
02330 2067 ISZ GSP /BUMP GOSUB STACK POINTER
02331 1067 TAD GSP /GET STACK POINTER
02332 7041 CIA /NEGATE
02333 1341 TAD GSTCKT /ADD ADR OF TOP OF STACK
02334 7710 SPA CLA /STACK OVERFLOW?
02335 4516 GS, JMS I ERROR /YES-TOO MANY NESTED GOSUBS
02336 1040 TAD TEMP1 /NO-GET ELEMENT TO BE STACKED
02337 3467 DCA I GSP /STACK IT
02340 5726 JMP I PUSHG /RETURN

02341 0476 GSTCKT, GSSTOP /ADDR OF TOP OF STACK
```

/ROUTINE TO RANDOMIZE RND(X)

```
02342 1017 FRANDM, TAD SPINNR      /USE SPINNR FOR NEW SEED FOR RND(X)
02343 7124      CLL CML RAL      /MAKE SURE SEED IS ODD
02344 3346      DCA RSEED
02345 5513      JMP I ILOOPL      /DONE
02346 2713 RSEED, 2713
```

/SUBROUTINE CR,LF

```
02347 0000 CRLFR, 0
02350 1054      TAD K215
02351 4512      JMS I PUTCHL
02352 1356      TAD K212L
02353 4512      JMS I PUTCHL      /PRINT A CR, AND LF
02354 3573      DCA I WORD10      /ZERO COLUMN POINTER
02355 5747      JMP I CRLFR
02356 0212 K212L, 212
```

/SUBROUTINE FOTYPE

/RETURNS TO CALL+1 IF FILE FIXED LENGTH, CALL+2 IF VARIABLE

```
02357 0000 FOTYPE, 0
02360 1563      TAD I WORD0      /GET HEADER
02361 0365      AND K0004      /ISOLATE TYPE BIT
02362 7640      SZA CLA      /IS IT FIXED LENGTH?
02363 2357      ISZ FOTYPE      /NO-BUMP RETURN
02364 5757      JMP I FOTYPE      /RETURN
02365 0004 K0004, 4
```

/SUBROUTINE TO REPLACE FAC WITH ABS(FAC)

```
02366 0000 ABSVAL, 0
02367 1045      TAD HORD
02370 7710      SPA CLA      /IS FAC<0?
02371 4540      JMS I FNEGL      /YES-NEGATE IT
02372 5766      JMP I ABSVAL      /RETURN
```

/ROUTINE TO RESTORE THE FAC FROM FP TEMP

```
02373 0000 FACRES, 0
02374 4534      JMS I FGETL      /GET FAC
02375 1152      INTERB
02376 5773      JMP I FACRES      /RETURN
```

2400

PAGE

/STRING DATA LIST READ

02400	4616	SRLIST, JMS I DLREAL	/READ COUNT FROM DATA LIST
02401	3032	DCA STRLEN	/SAVE AS NEW COUNT FOR FAC
02402	1032	TAD STRLEN	/COUNT FOR SAC STRING
02403	7450	SNA	/NULL STRING?
02404	5513	JMP I ILOOPL	/YES-NO OPERATION TO PERFORM
02405	7130	CLL CML RAR	/AND DIVIDE BY TWO FOR WORD COUNT
02406	3071	DCA STRCNT	/SAVE AS MOVE COUNTER
02407	1111	TAD SACPTR	
02410	3012	DCA XR2	/POINTS INTO SAC
02411	4616	SRLOOP, JMS I DLREAL	/READ 2 CHARS FROM DATA LIST
02412	3412	DCA I XR2	/AND PUT THEM IN SAC
02413	2071	ISZ STRCNT	/BUMP STRING COUNT
02414	5211	JMP SRLOOP	/NEXT 2
02415	5513	JMP I ILOOPL	/DONE
02416	2311	DLREAL, DLREAD	

/STRING READ ROUTINE

02417	3657	SREAD,	DCA I STHCDL	/DF FOR STH
02420	3071		DCA STRCNT	/0 STRING COUNT
02421	7101		CLL IAC	/LEAVE FIELD AS IS
02422	1072		TAD STRPTR	/ADDR OF OPERAND
02423	4526		JMS I STHINL	/INIT STORE HALF TO STORE IN OPERAND
02424	4542	FTCOM,	JMS I GETCHL	/GET CHAR FROM FILE OR TTY
02425	1053		TAD CHAR	
02426	1055		TAD M215	/IS IT CR?
02427	7450		SNA	
02430	5247		JMP SRFIN	/YES-STRING IS FINISHED
02431	1256		TAD MLF	/IS IT LF?
02432	7650		SNA CLA	
02433	5224		JMP FTCOM	/YES-IGNORE IT
02434	1071		TAD STRCNT	/NO-GET LENGTH OF STRING SO FAR
02435	1070		TAD STRMAX	/COMPARE AGAINST UPPER LIMIT OF DESTINATION
02436	7700		SMA CLA	/ANY MORE ROOM?
02437	5244		JMP ST	/NO-TRUNCATION ERROR
02440	1053		TAD CHAR	/YES
02441	4530		JMS I STH	/STORE CHAR IN STRING
02442	2071		ISZ STRCNT	/BUMP COUNT
02443	5224		JMP FTCOM	/GET NEXT CHAR
02444	4516	ST,	JMS I ERROR	/YES-TRUNCATION ERROR
02445	1054		TAD K215	/SET CHAR TO 215
02446	3053		DCA CHAR	/SO TTY BUFFER CLEARED BEFORE NEXT INPUT
02447	4453	SRFIN,	TAD STHCDL	/GET DF OF STRING
02450	3251		DCA .+1	/PUT IN LINE
02451	2451		.	/DF TO THAT OF STRING
02452	1071		TAD STRCNT	/STRING DONE-GET LENGTH
02453	7041		CIA	/NEGATE
02454	3472		DCA I STRPTR	/STORE AS COUNT FOR STRING
02455	5513		JMP I ILOOPL	/DONE
02456	0003	MLF,	3	
02457	2603	STHCDL,	STHDF	

/STRING WRITE ROUTINE

02460 3357 SWRITE, DCA COMMAS /CLEAR COMMA SWITCH
 02461 1032 TAD STRLEN /# OF CHARS IN STRING
 02462 7450 SNA /NULL STRING?
 02463 5513 JMP I ILOOPL /YES=NOTHING TO WRITE
 02464 7041 CIA /MAKE A POSITIVE NUMBER
 02465 1573 TAD I WORD10 /ADD TO COLUMN NUMBER
 02466 1313 TAD MM110 /COMPARE AGAINST END OF LINE
 02467 7740 SMA SZA CLA /WILL STRING FIT ON LINE?
 02470 4553 JMS I CRLF /NO=ISSUE A CRLF FIRST
 02471 1111 TAD SACPTR
 02472 7101 CLL IAC /AC POINTS TO LEFT SAC CHAR 1
 02473 4527 JMS I LDHINL /INITIALIZE LOAD HALF ROUTINE
 02474 4557 JMS I LDHRST /LOAD FROM SAC IN FLD 0
 02475 1032 TAD STRLEN /# OF CHARS
 02476 3071 DCA STCNT /USE AS COUNTER
 02477 4531 SWCLP, JMS I LDH /LOAD HALF CHAR FROM STRING
 02500 3040 DCA TEMP1 /SAVE
 02501 1040 TAD TEMP1
 02502 1106 TAD KM40 /SUBTRACT 40
 02503 7710 SPA CLA /IS CHAR <40?
 02504 1076 TAD K0100 /NO=MAKE IT 300 SERIES
 02505 1077 TAD K0200 /MAKE IT 200 SERIES
 02506 1040 TAD TEMP1
 02507 4512 JMS I PUTCHL /PUT CHAR IN FILE OR ON TTY
 02510 2071 ISZ STCNT /DONE?
 02511 5277 JMP SWCLP /NO=NEXT CHAR
 02512 5513 JMP I ILOOPL /YES
 02513 7670 MM110, -110

7B60

/FLOATING POINT CONSTANT USED BY ASCOUT FOR FORMAT CONVERSION

```

02514 0024 A999, 24 /999999
02515 3641 3641
02516 0770 0770

```

/COMMA FUNCTION (KNOWN ONLY TO COMPILER FOR FORMATTING PRINT STATEMENTS)

```

02517 0000 COMMA, 0
02520 4550 JMS I FTYPL /IS FILE NUMERIC?
02521 5513 JMP I ILOOPL /YES-COMMA FUNCTION IS A NOP
02522 1357 TAD COMMAS /GET COMMA SWITCH
02523 7650 SNA CLA /WAS LAST THING PRINTED A COMMA?
02524 5327 JMP ,+3 /NO-WE ARE OK
02525 1365 TAD C240 /YES-PRINT A SPACE BEFORE DOING COMMA CALCULATION
02526 4512 JMS I PUTCHL
02527 7001 IAC
02530 3357 DCA COMMAS /SET COMMA SWITCH
02531 1366 TAD M4
02532 3007 DCA TEMP2 /ONLY 4 COLUMNS TO CHECK
02533 1360 TAD POSPTA
02534 3014 DCA XR4 /POINTS TO POSITION #'S OF COLUMNS
02535 1573 COMLOP, TAD I WORD10 /GET CURRENT PRINT HEAD POSITION
02536 1414 TAD I XR4 /COMPARE AGAINST COLUMN MARKER
02537 7510 SPA /PAST THIS ONE?
02540 5347 JMP SLOVER /YES-SLIDE PRINT HEAD TO START OF NEXT
02541 7650 SNA CLA /EXACTLY ON A COLUMN?
02542 5513 JMP I ILOOPL /YES-DONE
02543 2007 ISZ TEMP2 /ALL MARKERS CHECKED YET?
02544 5335 JMP COMLOP /NO-DO NEXT
02545 4553 JMS I CRLF /YES-NEXT COLUMN IS 0
02546 5513 JMP I ILOOPL /DONE

02547 3777' SLOVER, DCA TEMP19 /-# OF COLUMNS TO NEXT MARKER
02550 4550 JMS I FTYPL /IS FILE NUMERIC?
02551 5513 JMP I ILOOPL /YES-THIS IS A NOP
02552 1365 TAD C240 /GET SPACE
02553 4512 JMS I PUTCHL /PRINT IT
02554 2777' ISZ TEMP19 /THERE YET?
02555 5350 JMP SLOVER+1 /NO-TYPE ANOTHER SPACE
02556 5513 JMP I ILOOPL /YES-DONE

```

3371 TEMP19=FACSAV

```

02557 0001 COMMAS, 1 /SET TO 1 IF LAST PRINT WAS A COMMA MOVE
02560 2560 POSPTA, POSTP-1
02561 7762 POSTP, -16 /COLUMN MARKERS
02562 7744 -34 /MINUS TTY COLUMN NUMBER THAT MARKS BEGINNING
02563 7726 -52 /OF ONE OF THE BASIC COLUMNS
02564 7710 -70
02565 0240 C240, 240
02566 7774 M4, -4

```

/RESTORE FOR IN-CORE DATA LIST

02567	1030	RESCLS, TAD DLSTRT	/ADDRESS OF START OF INCORE DATA LIST
02570	3016	DCA CLPTR	/USE IT TO RESET DATA LIST POINTER
02571	5513	JMP I ILOOP	/THATS ALL!

/RESTORE ROUTINE

02572	1162	RESTOR, TAD ENTNO	/GET CURRENT FILE #
02573	7650	SNA CLA	/IS IT 0?
02574	5367	JMP RESCLS	/YES=RESTORE DATA LIST
02575	4777	JMS I RESTIL	/NO=RESTORE A FILE
02576	5513	JMP I ILOOP	/DONE
PE			
02577	1650	RESTIL, RESTI	

2600

PAGE

/SUBROUTINE STH-SIMULATES AN AUTO-INDEXING STORE HALF INSTRUCTION.
 /STORES THE RIGHT HALF OF THE AC IN THE HALFWORD FOLLOWING THE
 /LAST HALFWORD STORED. TO CHANGE THE STORAGE ADDRESS, CALL STHINI

V3
V4

02600	0000	STHL,	0		
02601	0075		AND K0077	/STRIP TO 6 BITS	
02602	0062	STHR	DCA TEMP11	/SAVE	
02603	2603	STHDF,	.	/DF TO STORE FIELD	
02604	1244		TAD STHSWT	/GET SWITCH FOR HALF TO STORE IN	
02605	7640		SZA CLA	/WHICH HALF?	
02606	5233		JMP RIGHTS	/STORE IN RIGHT HALF	
02607	1062	STHR	TAD TEMP11	/STORE IN LEFT HALF	
02610	4544		JMS I BSWL	/SWAP BYTES	
02611	3062	STHR	DCA TEMP11	/SAVE AGAIN	
02612	1645		TAD I STHR	/GET CURRENT VALUES	
02613	0075		AND K0077	/PRESERVE RIGHT HALF	
02614	1062	STHR	TAD TEMP11	/COMBINE WITH NEW LEFT HALF	
02615	3645		DCA I STHR	/AND STORE IT	
02616	1244		TAD STHSWT	/GET HALF SWITCH	
02617	7650		SNA CLA	/WAS THIS RIGHT HALF?	
02620	5227		JMP JSL	/NO-JUST FLIP SWITCH	
02621	2245		ISZ STHR	/BUMP POINTER	
02622	5230		JMP JSL+1	/POINTER IS BUMPED-SET HALF SWITCH TO LEFT	
02623	1203		TAD STHDF	/SKIP MEANS WE HAVE TO BUMP STH CDF	
02624	1073		TAD K0010		
02625	3203		DCA STHDF		
02626	7410		SRP	/SET HALF SWITCH TO 0	
02627	7040	JSL,	CMA	/FLIP HALF SWITCH	
02630	3244		DCA STHSWT		
02631	6201		CDF		
02632	5600		JMP I STHL	/DONE	
02633	1645	RIGHTS,	TAD I STHR	/GET LEFT HALF	
02634	0104		AND K7700	/CLEAR ANY GARBAGE THAT MIGHT BE IN RIGHT HALF	
02635	5214		JMP SLRCOM	/FLIP SWITCH AND RETURN	

Left
Right

/SUBROUTINE STHINI-USED TO SET THE HALFWORD ADDRESS STORED INTO BY STH.
 /ON CALL, WORD ADDR IS IN AC, LINK SET TO 0 FOR LEFT HALF, 1 FOR RIGHT HALF.

Write

02636	0000	STHINI,	0		
02637	3245		DCA STHR	/STORE ADDRESS	
02640	7630		SZL CLA	/WHICH HALF TO START	
02641	7040		CMA	/RIGHT-SET STHSWT	
02642	3244		DCA STHSWT	/LEFT-CLEAR STHSWT	
02643	5636		JMP I STHINI	/DONE	
02644	0000	STHSWT,	0	/STORE HALFWORD SWITCH	
02645	0000	STHR,	0	/HALFWORD POINTER FOR STH	

Now

EAK

18 Cyl

was

29 Cyl

22 Cyl

R Sthr 31 + BSWL

/SUBROUTINE LDH-SIMULATES AN AUTO-INDEXING LOAD HALF INSTRUCTION. WHEN /CALLED, IT LOADS THE NEXT HALFWORD INTO AC. TO CHANGE ADDRESS FROM /WHICH IT LOADS, CALL LDHINI. DF MUST BE SET TO DF OF SOURCE ON CALL.

V3
V4
V5A
Rewrite
Free

02646	0000	LDHL, 0	
02647	2647	LDHDF, .	/DF FROM WHICH TO GET WORDS
02650	1304	TAD LDHSWT	/WHICH HALF TO LOAD?
02651	7640	SZA CLA	
02652	5265	JMP RIGHTL	/RIGHT HALF
02653	1705	TAD I LDHR	/LEFT HALF-GET BOTH
02654	4544	JMS I BSWL	/SWAP BYTES
02655	0075	LRSCOM, AND K0077	/ISOLATE CHAR
02656	3062	DCA TEMP11	/SAVE
02657	1304	TAD LDHSWT	
02660	7040	CMA	/FLIP LDHSWT
02661	3304	DCA LDHSWT	
02662	1062	TAD TEMP11	
02663	6201	CDF	
02664	5646	JMP I LDHL	/RETURN
02665	1705	RIGHTL, TAD I LDHR	/GET WORD
02666	2305	ISZ LDHR	/BUMP POINTER TO NEXT WORD
02667	5255	JMP LRSCOM	/NO SKIP SO JUST CONTINUE
02670	3236	DCA TEMP21	/SKIP MEANS WE HAVE TO BUMP LDH DF
02671	1247	TAD LDHDF	
02672	1073	TAD K0010	
02673	3247	DCA LDHDF	
02674	1236	TAD TEMP21	/GET WORD AGAIN
02675	5255	JMP LRSCOM	/FLIP SWITCH AND RETURN

/SUBROUTINE LDHINI-USED TO SET HALFWORD ADDRESS LDH DRAWS FROM. ON CALL, /AC=FULL WORD ADDRESS, AND LINK=0 FOR LEFT HALF, 1 FOR RIGHT.

02676	0000	LDHINI, 0	
02677	3305	DCA LDHR	/SAVE LDH POINTER
02700	7630	SZL CLA	/WHICH HALF?
02701	7040	CMA	/RIGHT-LDHSWT=7777
02702	3304	DCA LDHSWT	/LEFT-LDHSWT=0
02703	5676	JMP I LDHINI	
02704	0000	LDHSWT, 0	/LOAD HALFWORD SWITCH
02705	0000	LDHR, 0	/HALFWORD POINTER FOR LDH

2636 TEMP21=STHINI

/SUBROUTINE BUFCHK-CHECKS THE POSITION OF THE BUFFER POINTER FOR
 /THE DEVICE WHOSE I/O TABLE ENTRY IS IN WORKING AREA. RETURNS TO CALL+1
 /IF THE POINTER IS AT THE END AND CHAR NUMBER IS 1 (LAST
 /AVAILABLE CHAR 3 HAS BEEN USED),CALL+2 IF THE POINTER IS AT THE
 /END BUT THE CHAR # IS NOT 1 (THERE IS 1 CHAR 3 LEFT), CALL+3
 /IF THERE IS 1 WORD LEFT IN BUFFER,CALL+4 IF MORE THAN 1 LEFT.

2677
 V5A

02706	0000	BUFCHK, 0	
02707	1162	TAD ENTNO	/GET DEVICE #
02710	7650	SNA CLA	/IS IT TTY?
02711	1327	TAD MK61	/YES-CHECK FOR A BUFFER 60 WORDS LONG
02712	1102	TAD K0400	/NO-CHECK FOR A BUFFER 400 WORDS LONG
02713	1564	TAD I WORD1	/ADD LENGTH TO BUFFER ADDRESS
02714	7041	CIA	/-ADDR OF END OF BUFFER
02715	1566	TAD I WORD3	/CHECK AGAINST CURRENT POINTER
02716	7450	SNA	/IS POINTER AT END OF BUFFER?
02717	5330	JMP EBC	/AT END-CHECK THE CHAR #
02720	2306	ISZ BUFCHK	
02721	2306	ISZ BUFCHK	/NO=BUMP RETURN
02722	7001	IAC	
02723	7650	SNA CLA	/WAS POINTER AT LAST WORD?
02724	5706	JMP I BUFCHK	/YES-RETURN TO CALL+3
02725	2306	ISZ BUFCHK	/NO
02726	5706	JMP I BUFCHK	/RETURN TO CALL+4
02727	7461	MK61, 7461	
02730	4551	EBC, JMS I CHRNL	/GET CHAR #
02731	5706	JMP I BUFCHK	/IT WAS 1-RETURN TO CALL+1
02732	7000	NOP	/IT WAS 3-RETURN TO CALL+2
02733	2306	ISZ BUFCHK	/IT WAS 2-RETURN TO CALL+2
02734	5706	JMP I BUFCHK	

/SUBROUTINE PACKCH=PACKS ASCII CHARS,3 FOR 2, INTO BUFFER FOR THE
/DEVICE IN WORK AREA, CALL WITH THE CHARACTER IN THE AC

```

02735 0000 PACKCH, 0
02736 3040      DCA TEMP1      /SAVE
02737 4551      JMS I CHRNL    /DETERMINE CHARACTER NUMBER
02740 7410      SKP            /1
02741 5346      JMP CHAR3P     /3
02742 1040      TAD TEMP1      /1 OR 2-GET CHAR AGAIN
02743 4525      JMS I WRITFW    /STORE IN BUFFER
02744 4765      JMS I CNOBMK    /BUMP CHARACTER NUMBER
02745 5735      JMP I PACKCH   /DONE

02746 7344 CHAR3P, CLA CLL CMA RAL /-2 IN AC
02747 1566      TAD I WORD3     /BACK BUFFER POINTER UP TO POINT TO CHAR 1
02750 3566      DCA I WORD3
02751 1040      TAD TEMP1      /CHAR
02752 7106      CLL RTL        /SLIDE LEFT HALF INTO BITS 0-3
02753 7006      RTL
02754 3040      DCA TEMP1      /SAVE
02755 1040      TAD TEMP1
02756 4366      JMS COMBNE     /ISOLATE LEFT HALF, COMBINE WITH CHAR1, AND PUT IN FILE
02757 1040      TAD TEMP1      /CHAR AGAIN
02760 7106      CLL RTL
02761 7006      RTL           /SLIDE RIGHT HALF INTO BITS 0-3
02762 4366      JMS COMBNE     /ISOLATE RIGHT HALF, COMBINE WITH CHAR 2, AND PUT IN FILE
02763 4546      JMS I CNOCLL   /CLEAR THE CHARACTER NUMBER (RESET IT TO 1)
02764 5735      JMP I PACKCH   /DONE

02765 5573 CNOBMK, CNOBML

02766 0000 COMBNE, 0
02767 0103      AND K7400      /ISOLATE HALF IN QUESTION
02770 3007      DCA TEMP2      /SAVE
02771 4776      JMS I BCGETL    /GET A WORD FROM FILE BUFFER IN FIELD 1
02772 0101      AND K0377      /FLUSH ANY SLUSH IN BITS 0-3
02773 1007      TAD TEMP2      /COMBINE
02774 4525      JMS I WRITFW    /PUT IN BUFFER
02775 5766      JMP I COMBNE    /RETURN

02776 3035 BCGETL, BCGET

```

```
3000          PAGE

/ROUTINE TO READ WORD FROM FILE BUFFER AND BUMP POINTER

03000 0000 READFL, 0
03001 4615      JMS I FTYL      /IS FILE VARIABLE LENGTH
03002 7410      SKP
03003 4516 VR,   JMS I ERROR    /YES-IT IS AN ERROR TO TRY AND READ IT
03004 1563      TAD I WORD0     /CHECK IF MORE THERE
03005 7112      CLL RTR        /EOF BIT TO LINK
03006 7620      SNL CLA         /EOF?
03007 5212      JMP .+3         /NO-CONTINUE
03010 4516 RE,   JMS I ERROR    /YES-ATTEMPT TO READ BEYOND EOF
03011 5513      JMP I ILOOP     /NOT FATAL-RETURN TO I LOOP
03012 4235      JMS BCGET       /GET WORD FROM FILE BUFFER
03013 2566      ISZ I WORD3     /BUMP POINTER
03014 5600      JMP I READFL    /DONE
03015 2357      FTYL, FOTYPE

/ROUTINE TO RESET CHARACTER NUMBER TO 1

03016 0000 CNOCLR, 0
03017 1563      TAD I WORD0
03020 0105      AND K7477      /SET CHAR BITS TO 0
03021 3563      DCA I WORD0
03022 5616      JMP I CNOCLR    /RETURN

/ROUTINE TO WRITE AC IN FILE BUFFER AND INCREMENT POINTER

03023 0000 WRITFL, 0
03024 4634      JMS I BCPUTL    /STORE AC IN FILE BUFFER
03025 2566      ISZ I WORD3     /BUMP POINTER
03026 1563      TAD I WORD0     /GET FILE HEADER WORD
03027 7112      CLL RTR        /EOF BIT TO LINK
03030 7620      SNL CLA         /WAS FILE PAST END?
03031 5623      JMP I WRITFL    /NO-RETURN
03032 4516 WE,   JMS I ERROR    /YES-ATTEMPT TO WRITE PAST END OF FILE
03033 5513      JMP I ILOOP     /NON-FATAL RETURN TO ILOOP

03034 0747 BCPUTL, BCPUT

/ROUTINE TO GET ONE WORD FROM FILE BUFFER IN FIELD 1

03035 0000 BCGET, 0
03036 4523      JMS I FIDLE     /CHECK IF FILE OPEN
03037 1566      TAD I WORD3     /GET READ WRITE POINTER
03040 3223      DCA TEMP17      /SAVE
03041 1162      TAD ENTNO       /GET FILE #
03042 7640      SZA CLA         /IF ITY,BUFFER FIELD IS 0
03043 6211      CDF 10         /OF TO BUFFER FIELD
03044 1623      TAD I TEMP17    /GET WORD FROM BUFFER
03045 6201      CDF
03046 5635      JMP I BCGET     /RETURN
```

3023 TEMP17=WRITFL

/SUBROUTINE UNPACK-UNPACKS ASCII, 3 FOR 2 ,FROM THE FILE IN THE I/O
/WORKING AREA, RETURNS WITH THE CHAR IN CHAR.

```

03047 0000 UNPACK, 0
03050 4551      JMS I CHRNOL      /GET CHAR #
03051 7410      SKP              /1
03052 5264      JMP CHAR3U       /3
03053 4704      JMS I CNOBMP      /BUMP CHAR NUMBER
03054 4747      JMS I READFW      /GET CHAR AGAIN
03055 0101      AND K0377         /STRIP TO EIGHT BITS
03056 3053      U123C, DCA CHAR    /SAVE
03057 1053      TAD CHAR          /GET CHAR AGIAN
03060 1303      TAD MCTRLZ        /IS IT CTRL/Z?
03061 7650      SNA CLA
03062 5543      JMP I EOFSEL      /YES-SET EOF BIT
03063 5647      JMP I UNPACK      /RETURN

03064 4216      CHAR3U, JMS CNOCLR /RESET CHAR # TO 1
03065 7344      CLA CLL CMA RAL   /-2 IN AC
03066 1566      TAD I WORD3
03067 3566      DCA I WORD3       /BACK BUFFER POINTER UP 2
03070 4747      JMS I READFW      /GET LEFT HALF OF CHAR
03071 0103      AND K7400
03072 3216      DCA TEMP18        /SAVE
03073 4747      JMS I READFW      /GET NEXT WORD WITH RIGHT HALF
03074 0103      AND K7400         /ISOLATE RIGHT HALF
03075 7112      CLL RTR
03076 7012      RTR              /SLIDE RIGHT HALF OVER
03077 1216      TAD TEMP18        /COMBINE WITH LEFT HALF
03100 7112      CLL RTR
03101 7012      RTR              /MOVE TO BITS 4-11
03102 5256      JMP U123C         /REJOIN MAINLINE
03103 7546      MCTRLZ, =232
03104 5573      CNOBMP, CNOBML
          3016      TEMP18=CNOCLR

```

/READ FUNCTION-GETS NUMBERS INTO VARIABLES

```
03105 4550 READI, JMS I FTYPL /IS FILE NUMERIC?
03106 7410 SKP /YES-WRITE DATA
03107 5323 JMP ASCHR /NO-WRITE ASCII
03110 4547 JMS I BUFCHL /YES-CHECK BUFFER POINTER
03111 7000 NOP /PAST END-NEXT RECORD
03112 7000 NOP /AT END-NEXT RECORD
03113 4552 JMS I NEXREL /ONLY 1 WORD LEFT-IT IS UNUSED IN NUMERIC FMT
03114 4747 JMS I READFW /GET WORD FROM FILE
03115 3044 DCA EXP /STORE AS EXPONENT
03116 4747 JMS I READFW /GET WORD FROM FILE
03117 3045 DCA HORD /STORE AS HIGH MANTISSA
03120 4747 JMS I READFW /GET WORD FROM FILE
03121 3046 DCA LORD /STORE AS LOW MANTISSA
03122 5513 JMP I ILOOPL /DONE

03123 4725 ASCHR, JMS I FFINL /USE FPP INPUT TO GET NUMBER
03124 5513 JMP I ILOOPL /DONE
03125 5200 FFINL, FFIN

/Routine to fetch ASCII characters from file buffer
03126 0000 GETCH, 0
03127 4550 JMS I FTYPL /IS FILE ASCII?
03130 4516 SR, JMS I ERROR /NO-ERROR
03131 1053 TAD CHAR /GET LAST CHAR
03132 1055 TAD M215 /WAS IT CR?
03133 7650 SNA CLA
03134 5343 JMP TTEST /YES-IF TTY, WE MUST REFILL BUFFER
03135 4547 NTTY, JMS I BUFCHL /NO-CHECK STATUS OF BUFFER
03136 4552 JMS I NEXREL /LAST CHAR READ-NEXT RECORD
03137 7000 NOP /CHAR 3 NOT USED YET
03140 7000 NOP /CHAR 2 AND 3 LEFT
03141 4247 JMS UNPACK /UNPACK CHAR FROM BUFFER
03142 5726 JMP I GETCH /RETURN

03143 1162 TTEST, TAD ENTNO /ENTRY NUMBER
03144 7650 SNA CLA /IS IT TTY?
03145 4524 JMS I DEVCAL /YES-FILL TTY BUFFER
03146 5335 JMP NTTY /RESUME

03147 3000 READFW, READFL
```

/STRING ACCUMULATOR LOAD

03150	3365	SLOAD,	DCA LOADDF	/PUT DF FOR OPERAND FIELD IN LINE
03151	1111		TAD SACPTR	/POINTER TO START OF SAC
03152	3012		DCA XR2	/POINTS INTO SAC
03153	1071		TAD STRCNT	/GET LENGTH OF THIS STRING
03154	3032		DCA STRLEN	/SET THAT LENGTH AS LENGTH OF STRING IN SAC
03155	1032		TAD STRLEN	/GET LENGTH OF NEW STRING
03156	7650		SNA CLA	/IS IT A NULL STRING?
03157	5513		JMP I ILOOP	/YES-WE DON'T HAVE TO MOVE ANYTHING
03160	2072	SSLP,	ISZ STRPTR	/POINT TO FIRST PAIR OF CHARACTERS
03161	5365		JMP LOADDF	
03162	1365		TAD LOADDF	/SKIP MEANS WE HAVE TO BUMP DF
03163	1073		TAD K0010	
03164	3365		DCA LOADDF	
03165	3165	LOADDF,	.	/DF TO OPERAND FIELD
03166	1472		TAD I STRPTR	/GET 2 CHARS FROM STRING
03167	6201		CDF	/DF TO SAC FIELD
03170	3412		DCA I XR2	/PUT IN SAC
03171	2071		ISZ STRCNT	/DONE?
03172	7410		SKP	/NO-TWO CHARS/WORD
03173	5513		JMP I ILOOP	/YES-NEXT INST
03174	2071		ISZ STRCNT	/DOES SECOND CHAR MAKE COUNT 0?
03175	5360		JMP SSLP	/NO-LOOP
03176	5513		JMP I ILOOP	/YES-NEXT INST

3200

PAGE

/WRITE FUNCTION=PUTS NUMBERS IN FILE BUFFERS

03200	4550	WRITEI,	JMS I FTYPL	/GET FILE TYPE
03201	7410		SKP	/NUMERIC-WRITE DATA
03202	5217		JMP PDNE	/ASCII
03203	4547		JMS I BUFCHL	/FILE IS NUMERIC-CHECK BUFFER STATUS
03204	0240	K240,	240	/PAST END-NEW RECORD (AND INST SERVES AS NOP)
03205	0210	K0210,	0210	/AT END-NEW RECORD (AND SERVES AS NOP)
03206	4552		JMS I NEXREL	/ONE WORD LEFT-DON'T USE IT
03207	1044		TAD EXP	/EXPONENT
03210	4525		JMS I WRITFW	/WRITE IN BUFFER
03211	1045		TAD HORD	/HIGH MANTISSA
03212	4525		JMS I WRITFW	/WRITE IN BUFFER
03213	1046		TAD LORD	/LOW MANTISSA
03214	4525		JMS I WRITFW	/WRITE IN BUFFER
03215	5246		JMP WDOONE	/DONE

03216	1260	ASCOUL,	ASCOUT	/LINK TO FPP CALLER AND FORMATTER
-------	------	---------	--------	-----------------------------------

/PDNE=CALLS ASCOUT TO GET NUMBER INTO INTERMEDIATE
/BUFFER, THEN TYPES IT ON DEVICE

03217	4616	PDNE,	JMS I ASCOUL	/GET # INTO INTER BUFFER
03220	2061		ISZ TEMP10	/MOVE POINTER PAST SPACE THAT SENT US HERE
03221	1461		TAD I TEMP10	/GET SIGN
03222	1251		TAD MPLUS	
03223	7640		SZA CLA	/IS IT PLUS?
03224	5227		JMP MONE	/NO-ITS MINUS
03225	1204		TAD K240	/SPACE
03226	3461		DCA I TEMP10	/REPLACE "+" WITH SPACE
03227	1007	MONE,	TAD TEMP2	/GET COUNT OF CHARS TO BE PRINTED
03230	1573		TAD I WORD10	/ADD TO PRINT HEAD POSITION
03231	1252		TAD M110	/COMPARE AGAINST "72"
03232	7740		SMA SZA CLA	/WILL THE NUMBER FIT ON THIS LINE?
03233	4553		JMS I CRLF	/NO-ISSUE A CR,LF
03234	1461	CPLOOP,	TAD I TEMP10	/GET CHAR FROM INTERMEDIATE BUFFER
03235	1055		TAD M215	/IS IT CR?
03236	7650		SNA CLA	
03237	5244		JMP ASCNDE	/YES=NUMBER ALL OUTPUTTED
03240	1461		TAD I TEMP10	/NO-GET CHAR AGAIN
03241	4253		JMS PUTCH	/PUT ON DEVICE
03242	2061		ISZ TEMP10	/BUMP POINTER
03243	5234		JMP CPLOOP	/NEXT

03244	1204	ASCNDE,	TAD K240	
03245	4253		JMS PUTCH	/FOLLOW THE NUMBER WITH A SPACE
03246	3650	WDOONE,	DCA I COMMAP	/CLEAR COMMA SWITCH
03247	5513		JMP I ILOOP	/WRITE IS DONE

03250	2557	COMMAP,	COMMAS
03251	7525	MPLUS,	-253
03252	7670	M110,	-110

V4A

/ROUTINE TO PUT ASCII CHARS IN FILE BUFFER. IGNORES RUBOUTS.

```
03253 0000  PUTCH, 0
03254 3040          DCA TEMP1      /SAVE CHAR
03255 1040          TAD TEMP1      /GET CHAR AGAIN
03256 1301          TAD MRUBOT
03257 7650          SNA CLA        /IS IT A RUBOUT?
03260 5653          JMP I PUTCH    /YES=RETURN
03261 4550          JMS I FTYPL     /IS FILE NUMERIC?
03262 4516  SW,     JMS I ERROR     /YES=ERROR
03263 2573          ISZ I WORD10    /BUMP COULMN NUMBER
03264 1162          TAD ENTNO       /GET ENTRY #
03265 7650          SNA CLA        /IS IT TTY?
03266 5276          JMP TOUT        /YES=JUST PUT CHARS IN RING BUFFER
03267 4547          JMS I BUFCHL    /NO=IS BUFFER FULL?
03270 4552          JMS I NEXREL    /YES=NEXT RECORD
03271 0040  KK40,   40             /THERE IS A CHAR 3 LEFT (AND IS A NOP)
03272 0020  K20,    20             /THERE IS A CHAR 2 AND 3 LEFT (AND IS A NOP)
03273 1040          TAD TEMP1      /GET CHAR AGAIN
03274 4545          JMS I PACKL     /PUT IN BUFFER
03275 5653          JMP I PUTCH    /RETURN

03276 1040  TOUT,   TAD TEMP1      /GET CHAR
03277 4522          JMS I XPUT      /PUTCH CHAR IN OUTPUT BUFFER FOR TTY
03300 5653          JMP I PUTCH    /RETURN

03301 7401  MRUBOT, -377
```

/SUBROUTINE NEXREC=WRITES THIS BUFFER IN FILE, THEN READS IN NEXT BUFFER
 /IF POSSIBLE, ELSE SETS EOF BIT. IF DEVICE IS READ OR WRITE ONLY
 /IT JUST READS OR WRITES A BLOCK, WHICHEVER IS APPROPRIATE

03302	0000	NEXREC, 0	
03303	1563	TAD I WORD0	/GET HEADER
03304	0272	AND K20	/GET READ/WRITE ONLY BIT
03305	7650	SNA CLA	/IS IT ON?
03306	5316	JMP FILSTR	/NO-DEVICE IS FILE STRUCTURED
03307	4743	JMS I FOTYPL	/YES-IS IT INPUT OR OUTPUT FILE?
03310	5314	JMP RONLY	/INPUT-DEVICE IS READ ONLY
03311	4350	JMS WRBLK	/DEVICE IS WRITE ONLY-WRITE THIS BLOCK IF USED
03312	4361	RWONG, JMS BLINIT	/INIT FILE TABLE ENTRIES
03313	5702	JMP I NEXREC	/DONE
03314	4344	RONLY, JMS BLREAD	/READ BLOCK INTO CORE
03315	5312	JMP RWONG	/INIT POINTERS
03316	4350	FILSTR, JMS WRBLK	/WRITE THE CURRENT BLOCK IF IT HAS BEEN CHANGED
03317	4361	JMS BLINIT	/INIT FILE TABLE ENTRIES
03320	2565	ISZ I WORD2	/BUMP BLOCK #
03321	1570	TAD I WORD5	/STARTING BLOCK
03322	7041	CIA	/NEGATE
03323	1565	TAD I WORD2	/SUBTRACT FROM CURRENT BLOCK FOR FILE LENGTH
03324	3015	DCA BABS	/SET UP CURRENT FILE LENGTH FOR 12 BIT COMPARE
03325	1571	TAD I WORD6	/COMPARE TO ACTUAL LENGTH
03326	4405	JMS I NUMCOM	/IS IT > CURRENT LENGTH?
03327	5332	JMP LASTB	/YES-EXTEND THE FILE IF IT IS OUTPUT
03330	4344	JMS BLREAD	/READ IN THE NEXT RECORD
03331	5702	JMP I NEXREC	/RETURN
03332	4743	LASTB, JMS I FOTYPL	/IS FILE FIXED LENGTH?
03333	5543	JMP I EOFSEL	/YES-SET EOF FLAG
03334	1571	TAD I WORD6	/NO-GET ACTUAL LENGTH
03335	3015	DCA BABS	
03336	1572	TAD I WORD7	/MAXIMUM LENGTH
03337	4405	JMS I NUMCOM	/IS ACTUAL LENGTH >= MAXIMUM LENGTH?
03340	5543	JMP I EOFSEL	/YES-SET EOF BITS
03341	2571	ISZ I WORD6	/NO-BUMP ACTUAL LENGTH
03342	5702	JMP I NEXREC	/RETURN WITHOUT READING NEXT RECORD
03343	2357	FOTYPL, FOTYPE	

/ROUTINE TO READ 2 PAGES FROM DEVICE

```
03344 0000 BLREAD, 0
03345 1205      TAD K0210      /"READ 2 PAGES"
03346 4524      JMS I DEVCAL   /HANDLER CALL
03347 5744      JMP I BLREAD
```

/ROUTINE TO WRITE 2 PAGES ONTO DEVICE

```
03350 0000 WRBLK, 0
03351 1563      TAD I WORD0     /GET FILE HEADER
03352 0271      AND KK40        /GET FILE WRITTEN BIT
03353 7650      SNA CLA         /HAS THIS BLOCK BEEN CHANGED?
03354 5750      JMP I WRBLK     /NO-RETURN
03355 1360      TAD K4210      /"WRITE 2 PAGES"
03356 4524      JMS I DEVCAL   /CALL TO DEVICE HANDLER
03357 5750      JMP I WRBLK
03360 4210 K4210, 4210
```

/ROUTINE TO INITIALIZE I/O TABLE ENTRIES AFTER READ OR WRITE

```
03361 0000 BLINIT, 0
03362 1564      TAD I WORD1
03363 3566      DCA I WORD3     /INIT READ/WRITE POINTER
03364 1563      TAD I WORD0
03365 0370      AND K7437      /SET CHAR # TO 1 AND CLEAR BLOCK WRITTEN BIT
03366 3563      DCA I WORD0
03367 5761      JMP I BLINIT
03370 7437 K7437, 7437
```

/ROUTINE TO SAVE THE FAC IN FP TEMP

```
03371 0000 FACSAV, 0
03372 4535      JMS I FPUTL     /STORE FAC
03373 1152      INTERB         /USE INTERMEDIATE BUFFER FOR TEMP STORAGE
03374 5771      JMP I FACSAV    /RETURN
```

*Go to Page 92
4600

```
////////////////////////////////////  
////////////////////////////////////  
//////////////// OVERLAY BUFFER 3400-4600 //////////////////  
//////////////// CONTAINS FUNCTION OVERLAYS //////////////////  
//////////////// AT RUN TIME ///////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////
```

```

////////////////////////////////////
////////////////////////////////////
//////////////// OVERLAY 1-ARITHMETIC FUNCTIONS //////////////////
////////////////////////////////////
////////////////////////////////////

```

3400

*OVERLAY

```

/INTEGER FUNCTION
/RANGE=ALL X

```

```

03400 0000 INT,      0
03401 4535          JMS I FPUTL      /SAVE X
03402 1175          FPPTM1
03403 1044          TAD EXP          /GET EXPONENT
03404 7740          SMA SZA CLA      /IS EXP<0?
03405 5213          JMP INSC        /NO-GO ON
03406 1045          TAD HORD        /YES
03407 7710          SPA CLA          /IS X<0?
03410 5270          JMP MIR         /YES-INT=-1
03411 4537          JMS I FCLR      /YES=RETURN A 0
03412 5600          JMP I INT
03413 1045 INSC,    TAD HORD        /GET HI MANTISSA
03414 7700          SMA CLA          /IS IT <0?
03415 5220          JMP INTPOS      /NO-USE FAC AS IS
03416 4540          JMS I FNEGL     /YES-NEGATE FAC (MAKE IT POS)
03417 7001          IAC             /AND SET FLAG
03420 3042 INTPOS,  DCA TEMP3       /FLAG FOR NEGATIVE
03421 3047          DCA TEMP5       /ZERO LORD MASK
03422 7130          CLL CML RAR
03423 3043          DCA TEMP4
03424 1044          TAD EXP
03425 7041          CIA             /= COUNT
03426 3007          DCA TEMP2
03427 1043 MASKL,  TAD TEMP4
03430 7130          CLL CML RAR      /ROTATE 1'S THROUGH 3 WORD MASK
03431 3043          DCA TEMP4        /
03432 1047          TAD TEMP5        /UNTIL THERE IS A COUNT OF ZERO
03433 7010          RAR
03434 3047          DCA TEMP5
03435 2007          ISZ TEMP2        /DONE?
03436 5227          JMP MASKL       /NO
03437 1045          TAD HORD        /YES=MASK HORD
03440 0043          AND TEMP4
03441 3045          DCA HORD
03442 1046          TAD LORD        /MASK LORD
03443 0047          AND TEMP5
03444 3046          DCA LORD
03445 1042          TAD TEMP3        /NEG FLAG
03446 7650          SNA CLA          /WAS ORIGINAL NUMBER <0?
03447 5600          JMP I INT       /NO-DONE
03450 4535          JMS I FPUTL     /SAVE INT(X)
03451 1172          FPPTM2

```

03452	4673	JMS I FADDLK	/-INT(X)+(X)
03453	1175	FPPTM1	
03454	1045	TAD HORD	/SAVE HORD
03455	3042	DCA TEMP3	
03456	4537	JMS I FCLR	/FLUSH FAC
03457	1042	TAD TEMP3	/WAS INT(X)=X?
03460	7650	SNA CLA	
03461	5264	JMP JUSNEG	/YES=JUST NEGATE INT(X)
03462	4673	JMS I FADDLK	/NO=ADD 1
03463	3474	ONE	
03464	4673	JUSNEG, JMS I FADDLK	/GET INT(X)
03465	1172	FPPTM2	
03466	4540	JNEG, JMS I FNEGL	/AND NEGATE (INT(S.3)=-6)
03467	5600	JMP I INT	/DONE
03470	4534	MIR, JMS I FGETL	/LOAD FAC WITH 1
03471	3474	ONE	
03472	5266	JMP JNEG	/JUST NEGATE AND RETURN
03473	6000	FADDLK, FFADD	
03474	0001	ONE, 1	
03475	2000	2000	
03476	0000	0	

/EXPONENTIATION FUNCTION

/IF B=0, A*B=1

/IF A=0 AND B>0, A*B=0

/IF A=0 AND B<0, DIVIDE BY ZERO ERROR MESSAGE RESULTS AND A*B=0

/IF B=INTEGER > 0, A*B=A*A*A*.....*A

/IF B=INTEGER < 0, A*B=1/A*A*A*.....*A

/IF B=REAL AND A>0, A*B=EXP(B*LOG(A))

/IF B=REAL AND A<0, A FATAL ERROR RESULTS

03477	0000	EXPON, 0	
03500	4535	JMS I FPUTL	/SAVE A
03501	1161	FPPTM5	
03502	4535	JMS I FPUTL	/SET UP RUNNING PRODUCT IN CASE OF
03503	1164	FPPTM4	/MULTIPLIES
03504	1045	TAD HORD	/HI ORDER OF A
03505	3277	DCA EXPON	/SAVE IT
03506	3064	DCA INSAV	/POINTER TO B IN SYMBOL TABLE
03507	4711	JMS I ARGPLL	/FIND B
03510	4534	JMS I FGETL	/GET B
03511	0307	ARGPLL, ARGPRE	/LOC SKIPPED BY FPP, SO WE USE IT FOR CONSTANT
03512	6201	CDF	
03513	1045	TAD HORD	/HI ORDER OF B
03514	7450	SNA	/IS B=0?
03515	5771	JMP I RETRNO	/YES A*B=1
03516	7700	SMA CLA	/IS B<0?
03517	5323	JMP .+4	/NO
03520	1277	TAD EXPON	/YES=GET HI ORDER A
03521	7650	SNA CLA	/IS A=0?
03522	5774	JMP I DVTRAP	/YES=DIVIDE BY ZERO ERROR
03523	1277	TAD EXPON	/B>0, IS A=0?
03524	7650	SNA CLA	
03525	5365	JMP RET0	/YES A*B=0
03526	4535	JMS I FPUTL	/SAVE B
03527	1167	FPPTM3	
03530	4200	JMS INT	/GET INT(B)
03531	4773	JMS I FSUBLL	/INT(B)-B
03532	1167	FPPTM3	
03533	1045	TAD HORD	/IS INT(B)-B=0?
03534	7640	SZA CLA	
03535	5767	JMP I USELOL	/NO=USE LOGS
03536	4534	JMS I FGETL	/YES=USE REPETITIVE MULTIPLY
03537	1167	FPPTM3	/GET B AGAIN
03540	1045	TAD HORD	
03541	3277	DCA EXPON	/SAVE SIGN OF B
03542	4775	JMS I ABSV	/ B
03543	4535	JMS I FPUTL	/USE ABS(B) AS MULTIPLY COUNT
03544	1167	FPPTM3	
03545	4534	EMLOOP, JMS I FGETL	/GET B
03546	1167	FPPTM3	
03547	4773	JMS I FSUBLL	/B-1
03550	3474	ONE	
03551	4535	JMS I FPUTL	/SAVE NEW COUNT

03552	1167		FPPTM3	
03553	1045		TAD HORD	
03554	7650		SNA CLA	/IS COUNT ZERO YET
03555	5770		JMP I EMDONL	/YES-MULTIPLIES ARE DONE
03556	4534		JMS I FGETL	/NO-GET RUNNING PRODUCT
03557	1164		FPPTM4	
03560	4772		JMS I FMPYL	/MULTIPLY BY A
03561	1161		FPPTM5	
03562	4535		JMS I FPUTL	/SAVE NEW RUNNING PRODUCT
03563	1164		FPPTM4	
03564	5345		JMP EMLOOP	
03565	4537	RET0,	JMS I FCLR	/RETURN WITH 0 IN FAC
03566	5513		JMP I ILOOP	
03567	3613	USELOL,	USELOG	
03570	3600	EMDONL,	EMDONE	
03571	3610	RETRNO,	RETRN1	
03572	5600	FMPYL,	FFMPY	
03573	6117	FSUBLL,	FFSUB	
03574	6355	DVTRAP,	DV	
03575	2366	ABSV,	ABSVAL	

```

3600      PAGE
03600 4534 EMDONE, JMS I FGETL    /GET RUNNING PRODUCT
03601 1164      FPPTM4
03602 1630      TAD I EXPONK    /GET SIGN OF B
03603 7700      SMA CLA        /WAS IT -?
03604 5513      JMP I ILOOPL    /NO-A*B=A*A*A*...*A
03605 4631      JMS I FIDVP     /YES-INVERT
03606 3474      ONE
03607 5513      JMP I ILOOPL    /A*B=1/A:A*A*...*A

03610 4534 RETRN1, JMS I FGETL
03611 3474      ONE            /SET FAC TO 1
03612 5513      JMP I ILOOPL

03613 1630 USELOG, TAD I EXPONK  /SIGN OF A
03614 7710      SPA CLA        /A<0?
03615 4516 EM,    JMS I ERROR    /YES-PRINT A MESSAGE
03616 4534      JMS I FGETL    /LOAD A
03617 1161      FPPTM5
03620 4626      JMS I FFLOGL    /LOG(A)
03621 4627      JMS I FMPYLV    /B*LOG(A)
03622 1167      FPPTM3
03623 4625      JMS I FFEXPL    /EXP(B*LOG(A))
03624 5513      JMP I ILOOPL    /DONE

03625 4120 FFEXPL, FFEXP
03626 4263 FFLOGL, FFLOG
03627 5600 FMPYLV, FFMPY
03630 3477 EXPONK, EXPON
03631 5412 FIDVP, FFDIV1

      /SGN FUNCTION

03632 0000 SGN,    0
03633 1045      TAD HORD        /GET HIGH MANTISSA
03634 7450      SNA            /IS X=ZERO?
03635 5513      JMP I ILOOPL    /YES-THEN LEAVE IT ALONE
03636 7710      SPA CLA        /IS X>0?
03637 5242      JMP ,+3        /NO
03640 7001      IAC            /YES-SET FAC=1
03641 7410      SKP
03642 7040      CMA            /NO-SET FAC=-1
03643 3044      DCA EXP        /SET UP FLOAT
03644 4541      JMS I FLOATL    /FLOAT VALUE OF SGN FUNCTION
03645 5513      JMP I ILOOPL    /DONE

```

IFZERO EAE <
 /FLOATING SQUARE ROOT
 /USES A HARDWARE TYPE ALGORITHM FOR BINARY SQUARE ROOTS
 /REF: THE LOGIC OF COMPUTER ARITHMETIC-IVAN FLORES-P 409
 /

```

03646 0000  FROOT, 0
03647 7332  CLA CLL CML RTR /SET RESULT TO 2000;0000
03650 3375  DCA AN1
03651 3376  DCA AN2
03652 6201  CDF /DF TO PACKAGE FIELD
03653 1377  TAD KM22 /SET COUNTER FOR DEVELOPING 22 BITS OF RESULT
03654 3042  DCA AC2 /ALREADY HAVE 1
03655 1045  TAD ACH
03656 7450  SNA
03657 5646  JMP I FROOT /ZERO FAC-NORMALIZED!-RETN, SAME
03660 7710  SPA CLA
03661 4540  JMS I FNEGL /TAKE ROOT OF ABSOL VALUE
03662 1044  TAD ACX /GET EXPONENT OF FAC
03663 7510  SPA /IF NEGATIVE-MUST PROPAGATE SIGN
03664 7020  CML
03665 7010  RAR /DIVIDE EXP, BY 2
03666 3044  DCA ACX /STORE IT BACK
03667 7430  SZL /INCREMENT EXP. IF ORIGINAL EXP
03670 2044  ISZ ACX /WAS ODD
03671 7000  NOP
03672 7420  SNL /DO A PRE-SHIFT FOR EVEN EXPONENTS
03673 4774  JMS I AL1K /SO FIRST BIT PAIR IS 10 NOT 01
03674 7344  CLA CLL CMA RAL /SET COUNTER FOR DETECTING A
03675 3373  DCA ZCNT /ZERO REMAINDER
03676 7332  CLA CLL CML RTR /SET UP POSITION OF TRIAL BIT
03677 7012  RTR /FOR FIRST PASS THRU LOOP
03700 3050  DCA OPH
03701 3051  DCA OPL
03702 1372  TAD K6000 /GET A FAST FIRST BIT-WE KNOW
03703 1045  TAD ACH /THIS WILL WORK SINCE # IS NORMALIZED
03704 3045  DCA ACH /IF # IS A POWER OF TWO, AND A PERFECT
03705 1045  TAD ACH /SQUARE-WE ARE DONE HERE!
03706 7450  SNA /WELL IS IT?
03707 1046  TAD ACLO /COULD BE-CHECK LOW ORDER
03710 7650  SNA CLA
03711 5365  JMP DONE /WHOOPEE-WE WIN BIG.
03712 5322  JMP LOP01 /NOPE-LOOP DON'T SHIFT FIRST TIME
03713 1050  SLOOP, TAD OPH /SHIFT TRIAL BIT 1 PLACE
03714 7110  CLL RAR /TO THE RIGHT
03715 3050  DCA OPH /AND STORE BACK
03716 1051  TAD OPL
03717 7010  RAR
03720 3051  DCA OPL
03721 4774  JMS I AL1K /SHIFT FAC LEFT 1 PLACE
03722 1051  LOP01, TAD OPL /ADD TRIAL BIT TO ANSWER
03723 1376  TAD AN2 /SO FAR
03724 7141  CLL CMA IAC /NEGATE IT
03725 1046  TAD ACLO /AND ADD TO FAC (REMAINDER SO FAR)
03726 7450  SNA /IS RESULT ZERO?
03727 2373  ISZ ZCNT /YES-INCREMENT COUNTER

```

/OS/8 BASIC RUNTIME SYSTEM

PAL8-V7 10/24/72 PAGE 56-1

03730 3043

DCA

TM

/STORE RESULT IN TEMPORARY

```

03731 7024      CML      RAL      /ADD CARRY TO HIGH ORDER FOR SUBTRACT
03732 1050      TAD      OPH      /ADD TRIAL BIT
03733 1375      TAD      AN1      /ADD RESULT SO FAR (HI ORDER)
03734 7141      CLL CMA IAC      /AND SUBTRACT FROM HI ORDER FAC
03735 1045      TAD      ACH
03736 7420      SNL
03737 5361      JMP      GON      /RESULT NEGATIVE?
03740 7440      SZ      /YES-NEXT RESULT BIT IS 0
03741 5346      JMP      LOP02    /NO-IS HI ORDER RESULT=0?
03742 2373      ISZ      ZCNT      /NO-GO ON
03743 5346      JMP      .+3      /YES-WAS LOW ORDER =0?
03744 7040      CMA      /NO-GO ON
03745 3042      DCA      AC2      /YES-REM.=0-SET COUNTER SO
03746 3045      DCA      ACH      /LOOKS LIKE WE'RE DONE
03747 1043      TAD      TM      /STORE HIGH ORDER REM, IN FAC
03750 3046      DCA      ACLO     /STORE LO ORDER REM, IN FAC
03751 1051      TAD      OPL
03752 7104      CLL      RAL      /TRIAL BIT SHIFTED LEFT 1 IS
03753 1376      TAD      AN2      /RESULT BIT-ADD IT TO ROOT DEVELOPED
03754 3376      DCA      AN2      /SO FAR
03755 1050      TAD      OPH
03756 7004      RAL
03757 1375      TAD      AN1
03760 3375      DCA      AN1
03761 7344      GON,      CLA CLL CMA RAL /RESET COUNTER FOR ZERO REM.
03762 3373      DCA      ZCNT
03763 2042      ISZ      AC2      /DONE ALL 23 RESULT BITS?
03764 5313      JMP      SLOOP    /NO-GO ON
03765 1375      DONE,    TAD      AN1 /YES-STORE ANSWER IN FAC
03766 3045      DCA      ACH      /ITS NORMALIZED ALREADY
03767 1376      TAD      AN2
03770 3046      DCA      ACLO
03771 5646      JMP I      FROOT   /AND RETURN

03772 6000      K6000,    6000
03773 0000      ZCNT,     0
03774 6057      AL1K,     0
03775 0000      AN1,      0
03776 0000      AN2,      0
03777 7752      KM22,     -26
>

```

IFNZRO EAE <
ENPUNCH

```

/
/FLOATING SQUARE ROOT
/USES MODIFIED HARDWARE ALGORITHM FOR BINARY SQUARE ROOTS
/REF: THE LOGIC OF COMPUTER ARITHMETIC-IVAN FLORES; P-409
*SGN+14
FROOT, 0
CLA CLL CML RTR /SET RESLT TO 2000,0000
DCA OPL
DCA OPH
SWAB /MODE B OF EAE-ALSO DOES MQL
CDF
DCA RBCNT /CLR. SHIFT COUNTER
TAD KM22
DCA AC2 /SET COUNTER FOR 23 BITS OF RESULT
TAD ACX /GET EXPONENT OF FAC
ASR /DIVIDE BY 2
1
DCA ACX /STORE IT BACK
DPSZ /INCREMENT EXP. IF ORIG. EXP
ISZ ACX /WAS ODD
NOP
MOA /DETERMINE WHETHER TO DO A
CLL RAL /PRE-SHIFT FOR EVEN EXPONENTS.
CML RAL
DCA RKNT /STORE BIT-0 OR 1 SHIFT CNT
CLL CML RTR /SET UP FIRST TRIAL BIT
RTR
DCA AC1
DCA AC0 /STORE AWAY
DCA ACNT /ZERO COUNTER
DLD /GET THE FAC
ACH
SWP /GET IN RIGHT ORDER
SNA /IS IT ZERO? (HI ORD=0)
JMP I FROOT /YES-ROOT = 0
SPA /NEGATIVE?
DCM /YES-TAKE ABSOL. VALUE
SHL /SHIFT # 1 BIT IF EXP WAS EVEN
RKNT, 0 /SO FIRST BIT PAIR IS 10 NOT 01
TAD K6000 /SUBTRACT 2000-KNOW FIRST BIT
DPSZ /IS 1(NORMALIZED)-DONE??
JMP LOP1 /NO-WE MUST LOOP
JMP DONE /YES-AN EASY ONE!!!
LOOP, DLD /GET THE FAC
ACH
SHL /SHIFT FAC APPROPRIATELY
1
LOP1, DST /MUST STOR BACK IN CASE RESLT
ACH /BIT IS 0
DLD /GET TRIAL BIT
AC0

```

	ASR		/SHIFT THE BIT APPROPRIATELY
ACNT,	0		
	ISZ	ACNT	/SHIFT 1 MORE NEXT TIME
	DAD		/ADD IN RESULT SO FAR
	OPH		
	DCM		/NEGATE IT
	ISZ	RBCNT	/BUMP COUNTER FOR RESLT BIT
	DAD		/DO THE SUBTRACT
	ACH		
	SNL		/RESULT NEGATIVE?
	JMP	GON	/YES-NEXT RESULT BIT = 0
	DPSZ		/NO-DID WE GET A ZERO REMAINDER?
	JMP	NOTZRO	/NOPE
ZREM,	CMA		/YES-SET SO LOOKS LIKE WE'RE DONE
	DCA	AC2	
NOTZRO,	DST		/GOOD SUBTR.-MODIFY FAC
	ACH		/ITS NOT CHANGED BY BAD SUBTRACT
	CAM		/CLEAR EVERYTHING
	RTR		
	ASR		/SHIFT RESLT BIT TO RIGHT PLACE
RBCNT,	0		
	DAD		/ADD IT TO THE RESULT SO FAR
	OPH		/WE APPEND IT TO RIGHT OF LAST
	DST		/BIT
	OPH		/STORE IT BACK
GON,	ISZ	AC2	/DONE 23 BITS?
	JMP	LOOP	/NO-GO ON
DONE,	DLD		/YES-GET RESULT-ITS NORMALIZED
	OPH		
	DCA	ACH	/STORE HIGH ORDER BACK
	SWP		
	DCA	ACLO	/STORE LOW ORDER BACK
	JMP I	FROOT	/RETURN
KM22,	-26		
K6000,	6000		
	NOPUNCH		
	>		

/23-BIT EXTENDED FUNCTIONS

/1-31-72 R BEAN

4000

*4000

/*****SINE*****/

```

04000 0000 SIN, 0
04001 4303 JMS NHNDLE /IF X<0,NEGATE X AND SET NFLAG
04002 4661 JMS I FMPYLK /X*2/PI
04003 4160 TOVPI
04004 4271 JMS FRACT /SAVE X IN TEMP1,THE INTEGER PART OF X IN NUM,AND GET FRACTIONAL PART
04005 1313 TAD NUM /GET INTEGER PART OF (2/PI)*X
04006 0270 AND C3 /ISOLATE BITS 10,11
04007 1212 TAD JMPISN
04010 3211 DCA .+1 /MAKE JUMP TO ARGUMENT REDUCING ROUTINE
04011 5211 JMP , /AND ADJUST ARG ACCORDING TO QUADRANT OF X
04012 5613 JMPISN, JMP I .+1
04013 4026 POLYSN /X IN QUAD1,SIN(X)=SIN(X)
04014 4017 QUAD2 /X IN QUAD2,SIN(X)=SIN(1-X)
04015 4022 QUAD3 /X IN QUAD3,SIN(X)=SIN(-X)
04016 4024 QUAD4 /X IN QUAD4,SIN(X)=SIN(X-1)

04017 4663 QUAD2, JMS I FSUBL /1-X
04020 3474 ONE
04021 5226 JMS POLYSN /CALCULATE SIN(1-X)
04022 4540 QUAD3, JMS I FNEGL /=X
04023 5226 JMS POLYSN /CALCULATE SIN(-X)
04024 4664 QUAD4, JMS I FSUBL /X-1
04025 3474 ONE
04026 4535 POLYSN, JMS I FPUTL /SAVE X
04027 1175 FPPTM1
04030 4665 JMS I FSQRL /U=X**2
04031 4535 JMS I FPUTL /SAVE U
04032 1172 FPPTM2
04033 4661 JMS I FMPYLK /A7*U
04034 4377 SINA7
04035 4660 JMS I FADDL /A5+A7*U
04036 4374 SINA5
04037 4661 JMS I FMPYLK /A5*U+A7*U**2
04040 1172 FPPTM2
04041 4660 JMS I FADDL /A3+A5(U)+A7(U**2)
04042 4371 SINA3
04043 4661 JMS I FMPYLK /A3(U)+A5(U**2)+A7(U**3)
04044 1172 FPPTM2
04045 4660 JMS I FADDL /A1+A3(U)+A5(U**2)+A7(U**3)
04046 4366 SINA1
04047 4661 JMS I FMPYLK /A1(X)+A3(X**3)+A5(X**5)+A7(X**7)
04050 1175 FPPTM1
04051 4313 JMS NCHK /IF NFLAG IS SET,SET SIN(X)=-SIN(X)
04052 5600 JMP I SIN /FAC=SIN(X)

```


/*****COSINE*****/

/USES SIN ROUTINE TO CALCULATE COS(X)

```
04053 0000 COS,      0
04054 4660          JMS I FADDL    /COS(X)=SIN(PI/2+X)
04055 4402          PIOV2
04056 4200          JMS SIN
04057 5653          JMP I COS      /RETURN
```

```
04060 6000 FADDL,    FFADD
04061 5600 FMPYLK,   FFMPY
04062 5722 FDIVL,    FFDIV
04063 5400 FSUB1L,   FFSUB1
04064 6117 FSUBL,    FFSUB
04065 6347 FSQRL,    FFSQ
04066 4500 FIXL,     FFIX
04067 5412 FDIV1L,   FFDIV1
04070 0003 C3,      3
```

/ROUTINE TO SEPERATE THE INTEGER AND FRACTIONAL PARTS OF FAC
/ORIGINAL FAC IS SAVED IN TEMP1, THE INTEGER PORTION OF FAC IS
/SAVED AT NUM, AND THE FRACTIONAL FORTION OF THE FAC IS LEFT IN THE FAC

```
04071 0000  FRACT, 0
04072 4535      JMS I FPUTL      /SAVE X
04073 1175      FPPTM1
04074 4666      JMS I FIXL      /INTEGER PORTION OF X
04075 1044      TAD EXP
04076 3313      DCA NUM          /SAVE FIXED FORTION OF X
04077 4541      JMS I FLOATL    /FAC=FLOAT(FIX(X))
04100 4663      JMS I FSUB1L    /FAC=X-INT(X)=FRACTION (X)
04101 1175      FPPTM1
04102 5671      JMP I FRACT     /RETURN
```

/ROUTINE TO CHECK IF FAC<0; IF IT IS, FAC IS NEGATED AND NFLAG IS
/SET TO 1

```
04103 0000  NHNDLE, 0
04104 1045      TAD HORD        /FETCH HIGH ORDER MANTISSA
04105 7700      SMA CLA        /IS IT <0?
04106 5311      JMP NFLGST      /NO-CLEAR NFLAG
04107 4540      JMS I FNEGL     /YES-NEGATE FAC
04110 7001      IAC            /AND SET NFLAG
04111 3320  NFLGST, DCA NFLAG
04112 5703      JMP I NHNDLE
```

/ROUTINE TO NEGATE FAC IF NFLAG IS NOT =0

```
04113 0000  NCHK, 0          /LOC ALSO USED FOR TEMP STORAGE
04114 1320      TAD NFLAG
04115 7640      SZA CLA        /IS NFLAG=0?
04116 4540      JMS I FNEGL     /NO-NEGATE FAC
04117 5713      JMP I NCHK      /YES-RETURN

4113          NUM=NCHK
```

/*****EXPONENTIAL*****/

```

04120 0000 EXPON1, 0      /LOC USED FOR TEMP STORAGE BY SIN, ARCTAN
04121 4661      JMS I FMPYLK /Y=XLOG2(E)
04122 4405      LOG2E
04123 4271      JMS FRAC   /GET FRACTIONAL PART OF Y
04124 4661      JMS I FMPYLK /(FRACTION(Y))*(LN2/2)
04125 4410      LN2OV2
04126 4535      JMS I FPUTL /SAVE Y
04127 1175      FPPTM1
04130 4665      JMS I FSQRL  /Y**2
04131 4660      JMS I FADDL  /B1+Y**2
04132 4413      EXPB1
04133 4667      JMS I FDIVL  /A1/(B1+Y**2)
04134 4416      EXPA1
04135 4660      JMS I FADDL  /A0+A1/(B1+Y**2)
04136 4421      EXPA0
04137 4664      JMS I FSUBL  /A0-Y+A1/(B1+Y**2)
04140 1175      FPPTM1
04141 4535      JMS I FPUTL  /SAVE
04142 1172      FPPTM2
04143 4534      JMS I FGETL  /GET Y
04144 1175      FPPTM1
04145 2044      ISZ EXP     /MULT. BY 2=2Y
04146 7000      NOP
04147 4662      JMS I FDIVL  /2Y/(A0-Y+A1/(B1+Y**2))
04150 1172      FPPTM2
04151 4660      JMS I FADDL  /1+2Y/(A0-Y+A1/(B1+Y**2))
04152 3474      ONE
04153 4665      JMS I FSQRL  /[(1+2Y/(A0-Y+A1/(B1+Y**2)))]**2=EXP(Y)
04154 1313      TAD NUM
04155 1044      TAD EXP     /EXP(X)=(2**N)(EXPY)
04156 3044      DCA EXP
04157 5720      JMP I EXPON1 /FAC=EXPON(X)

      4120      NFLAG=EXPON1

```

```

/CONSTANT THAT WOULDN'T FIT ELSEWHERE
04160 0000 TOVPI, 0      /,.6366198
04161 2427      2427
04162 6302      6302

```

4200

*4200

/*****ARC TANGENT*****/

```

04200 0000 ATAN, 0
04201 4661 JMS I NHNDLL /IF X<0,SET NFLAG AND NEGATE
04202 4535 JMS I FPUTM /SAVE X
04203 1175 FPPTM1
04204 4762 JMS I FSUBM /X-1
04205 3474 ONE
04206 1045 TAD HORD /GET HI MANTISSA
04207 7710 SPA CLA /WAS X>1?
04210 5220 JMP ARGPOL /NO-CLEAR GT1FLG
04211 4534 JMS I FGETM /YES-ATAN(X)=PI/2-ATAN(1/X)
04212 3474 ONE
04213 4760 JMS I FDIVM /1/X
04214 1175 FPPTM1
04215 4535 JMS I FPUTM
04216 1175 FPPTM1
04217 7001 IAC /SET GT1FLG
04220 3263 ARGPOL, DCA GT1FLG
04221 4534 JMS I FGETM /GET X OR 1/X
04222 1175 FPPTM1
04223 4764 JMS I FSQRM /Y**2
04224 4535 JMS I FPUTM /SAVE
04225 1172 FPPTM2
04226 4757 JMS I FADDM /Y**2+B3
04227 4446 ATANB3
04230 4761 JMS I FDIV1M /A3/(Y**2+B3)
04231 4443 ATANA3
04232 4757 JMS I FADDM /B2+A3/(Y**2+B3)
04233 4440 ATANB2
04234 4757 JMS I FADDM /Y**2+B2+A3/(Y**2+B3)
04235 1172 FPPTM2
04236 4761 JMS I FDIV1M /A2/(Y**2+B2+A3/(Y**2+B3))
04237 4435 ATANA2
04240 4757 JMS I FADDM /B1+A2/(Y**2+B2+A3/(Y**2+B3))
04241 4432 ATANB1
04242 4757 JMS I FADDM /Y**2+B1+A2/(Y**2+B2+A3/(Y**2+B3))
04243 1172 FPPTM2
04244 4761 JMS I FDIV1M /A1/(Y**2+B1+A2/(Y**2+B2+A3/(Y**2+B3)))
04245 4427 ATANA1
04246 4757 JMS I FADDM /B0+A1/(Y**2+B1+A2/(Y**2+B2+A3/(Y**2+B3)))
04247 4424 ATANB0
04250 4756 JMS I FMPYM /ATAN(Y)=X*(B0+A1/(Y**2+B1+A2/(Y**2+B2+A3/(Y**2+B3))))
04251 1175 FPPTM1
04252 1263 TAD GT1FLG /WAS X>1?
04253 7650 SNA CLA
04254 5257 JMP NGT /NO-TEST IF X<0?
04255 4763 JMS I FSUB1M /ATAN(X)=PI/2-ATAN(1/X)
04256 4402 PIOV2
04257 4662 NGT, JMS I NCHKL /IF NFLAG SET,NEGATE FAC
04260 5600 JMP I ATAN /FAC=ATAN(X)
04261 4103 NHNDLL, NHNDLE
04262 4113 NCHKL, NCHK

```

/*****NAPERIAN LOGARITHM*****/

```

      4200      GTFLG=ATAN
04263 0000 LOG, 0
04264 1045 TAD HORD
04265 7550 SPA SNA /X<0 OR X=0?
04266 5765 JMP I ARTRAP /YES-TAKE ILLEGAL ARGUMENT TRAP
04267 7106 CLL RTL
04270 7450 SNA /NO-HORD=2000?
04271 1044 TAD EXP /YES=EXP=1?
04272 7041 CMA IAC
04273 7001 IAC
04274 7450 SNA
04275 1046 TAD LORD /YES=LORD=0?
04276 7640 SZA CLA
04277 5304 JMP POLYNL /NO-ARG IS LEGAL AND NOT 1
04300 3044 DCA EXP
04301 3046 DCA LORD
04302 3045 LTRPRT, DCA HORD
04303 5663 JMP I LOG /YES=LOG(1)=0
04304 1044 POLYNL, TAD EXP
04305 3200 DCA GTFLG /SAVE EXPONENT FOR LATER
04306 3044 DCA EXP /ISOLATE MANTISSA IN FAC
04307 4535 JMS I FPUTM /SAVE F
04310 1175 FPPTM1
04311 4757 JMS I FADDM /F+SQR(.5)
04312 4451 SGRP5
04313 4535 JMS I FPUTM /SAVE
04314 1172 FPPTM2
04315 4534 JMS I FGETM
04316 1175 FPPTM1
04317 4762 JMS I FSUBM /F-SQR(.5)
04320 4451 SGRP5
04321 4760 JMS I FDIVM /Z=F+SQR(.5)/F-SQR(.5)
04322 1172 FPPTM2
04323 4535 JMS I FPUTM
04324 1175 FPPTM1
04325 4764 JMS I FSQRM /Z**2
04326 4535 JMS I FPUTM
04327 1172 FPPTM2
04330 4756 JMS I FMPYM /C5(Z**2)
04331 4462 LOGC5
04332 4757 JMS I FADDM /C3+C5(Z**2)
04333 4457 LOGC3
04334 4756 JMS I FMPYM /C3(Z**2)+C5(Z**4)
04335 1172 FPPTM2
04336 4757 JMS I FADDM /C1+C3(Z**2)+C5(Z**4)
04337 4454 LOGC1
04340 4756 JMS I FMPYM /C1(Z)+C3(Z**3)+C5(Z**5)
04341 1175 FPPTM1
04342 4762 JMS I FSUBM /C1(Z)+C3(Z**3)+C5(Z**5)-1/2=LOG2(F)
04343 4465 ONEHAF
04344 4535 JMS I FPUTM /SAVE LOG2(F)

```

04345	1172	FPPTM2	
04346	1200	TAD GTFLG	/I
04347	3044	DCA EXP	/SET UP FLOAT
04350	4541	JMS I FLOATM	
04351	4757	JMS I FADDM	/I+LOG2(F)
04352	1172	FPPTM2	
04353	4756	JMS I FMPYM	/[I+LOG2(F)]*LOGE(2)=LOGE(X)
04354	4470	LN2	
04355	5663	JMP I LOG	/FAC=LN(X)
	4263	GT1FLG=LOG	
04356	5600	FMPYM, FFMPY	
04357	6000	FADDM, FFADD	
04360	5722	FDIVM, FFDIV	
04361	5412	FDIV1M, FFDIV1	
04362	6117	FSUBM, FFSUB	
04363	5400	FSUB1M, FFSUB1	
04364	6347	FSQRM, FFSQ	
04365	6360	ARTRAP, LM	
	0134	FGETM=FGETL	
	0141	FLOATM=FLOATL	
	0135	FPUTM=FPUTL	

/CONSTANTS USED BY VARIOUS FUNCTIONS

04366	0001	SINA1,	1	/1.5707949
04367	3110		3110	
04370	3747		3747	
04371	0000	SINA3,	0	/-.64592098
04372	5325		5325	
04373	1167		1167	
04374	7775	SINA5,	7775	/,07948766
04375	2426		2426	
04376	2466		2466	
04377	7771	SINA7,	7771	/-.004362476
04400	5610		5610	
04401	3164		3164	
04402	0001	PIOV2,	1	/1.5707963
04403	3110		3110	
04404	3756		3756	
04405	0001	LOG2E,	1	/1.442695
04406	2705		2705	
04407	2434		2434	
04410	7777	LN2OV2,	7777	/,34657359
04411	2613		2613	
04412	4415		4415	
04413	0006	EXPB1,	6	/60.090191
04414	3602		3602	
04415	7054		7054	
04416	0012	EXPA1,	12	/-601.80427
04417	5514		5514	
04420	3104		3104	
04421	0004	EXPA0,	4	/12.015017
04422	3001		3001	
04423	7301		7301	
04424	7776	ATANB0,	7776	/,17465544
04425	2626		2626	
04426	6157		6157	
04427	0002	ATANA1,	2	/3.7092563
04430	3553		3553	
04431	1071		1071	
04432	0003	ATANB1,	3	/6.762139
04433	3303		3303	
04434	0670		670	
04435	0003	ATANA2,	3	/-7.10676
04436	4344		4344	
04437	5267		5267	
04440	0002	ATANB2,	2	/3.3163354
04441	3241		3241	
04442	7554		7554	
04443	7777	ATANA3,	7777	/-,26476862
04444	5703		5703	
04445	4040		4040	
04446	0001	ATANB3,	1	/1.44863154
04447	2713		2713	
04450	3140		3140	
04451	0000	SQRP5,	0	/,7071068
04452	2650		2650	

04453	1170		1170	
04454	0002	LOGC1,	2	/2.8853913
04455	2705		2705	
04456	2440		2440	
04457	0000	LOGC3,	0	/ .9614706
04460	3661		3661	
04461	0566		566	
04462	0000	LOGC5,	0	/ .59897865
04463	2312		2312	
04464	5525		5525	
04465	0000	ONEHAF,	0	/ .5
04466	2000		2000	
04467	0000		0	
04470	0000	LN2,	0	/ .6931472
04471	2613		2613	
04472	4415		4415	

4000	FFSIN=SIN
4053	FFCOS=COS
4200	FFATN=ATAN
4263	FFLOG=LOG
4120	FFEXP=EXPON1

4500

*4500

/*****FIX*****/

/ROUTINE TO FIX ANY FLOATING NUMBER IN FAC BETWEEN -2047 AND +2047 TO
/A TWELVE BIT INTEGER AND LEAVE RESULT IN EXP (LOC 44)

```

04500 0000 FFIX, 0
04501 7200 CLA
04502 1044 TAD EXP /FETCH EXPONENT
04503 7540 SZA SMA /IS NUMBER <1?
04504 5307 JMP ,+3 /NO-CONTINUE ON
04505 7200 FTRPRT, CLA
04506 5326 JMP FIXDNE+1 /YES-FIX IT TO ZERO
04507 1330 TAD M13 /SET BINARY POINT AT 11
04510 7450 SNA /PLACES TO RIGHT OF CURRENT POINT?
04511 5325 JMP FIXDNE /NO-NUMBER IS ALREADY FIXED THEN.
04512 7500 SMA /YES-IS NUMBER TOO LARGE TO FIX?
04513 5732 JMP I OTRAPA /YES-TAKE OVERFLOW TRAP
04514 3044 DCA EXP /NO-SET SCALE COUNT
04515 7100 FIXLP, CLL /0 IN LINK
04516 1045 TAD HORD /GET HIGH MANTISSA
04517 7510 SPA /IS IT <0?
04520 7020 CML /YES-PUT A 1 IN LINK
04521 7010 RAR /SCALE RIGHT
04522 3045 DCA HORD /SAVE
04523 2044 ISZ EXP /DONE YET?
04524 5315 JMP FIXLP /NO
04525 1045 FIXDNE, TAD HORD /YES-ANSWER IN AC
04526 3044 DCA EXP /RETURN WITH ANSWER IN 44
04527 5700 JMP I FFIX /RETURN

04530 7765 M13, -13 /-11 DECIMAL
04531 0013 C13, 13 /11 DECIMAL
04532 1637 OTRAPA, FO /ADDRESS OF VECTOR FOR OVERFLOW TRAP

```

/*****FLOAT*****/

/ROUTINE TO FLOAT ANY INTEGER IN EXP (LOC 44) INTO FAC

```

04533 0000 FFLOAT, 0
04534 1044 TAD EXP
04535 3045 DCA HORD /PUT NUMBER IN HI MANTISSA
04536 3046 DCA LORD /CLEAR LOW MANTISSA
04537 1331 TAD C13 /11(10) INTO EXPONENT
04540 3044 DCA EXP
04541 4536 JMS I FNORL /NORMALIZE
04542 5733 JMP I FFLOAT /RETURN

```

/RANDOM NUMBER GENERATOR

```
04543 0000 RND, 0
04544 1762 TAD I RSEEDL /GET SEED
04545 3042 DCA TEMP3 /PUT IN MULTIPLY OPERAND
04546 1363 TAD K73
04547 4521 JMS I MPYLNK /MULTIPLY SEED BY 73 12 X 12 MPY
04550 3762 DCA I RSEEDL /USE LOW ORDER 12 BITS AS NEW SEED
04551 1762 TAD I RSEEDL /LOW ORDER OF PRODUCT ALSO SERVES
04552 7110 CLL RAR /AS RANDOM NUMBER
04553 3045 DCA HORD /SET SIGN TO 0 AND STORE AS HORD
04554 3044 DCA EXP
04555 7010 RAR
04556 3046 DCA LORD /USE 12 BITS AS MANTISSA
04557 3041 DCA AC1 /CLEAR FPP OVERFLOW
04560 4536 JMS I FNORL /AND NORMALIZE
04561 5513 JMP I ILOOPL /DONE

04562 2346 RSEEDL, RSEED
04563 0073 K73, 73
```

```

////////////////////
////////////////////
//////////////////// OVERLAY 2- STRING FUNCTIONS //////////////////
////////////////////
////////////////////

```

0001

2000

3400

FIELD 1

*2000

NOPUNCH

*OVERLAY

ENPUNCH

IFNZRO EAE <

NOPUNCH

>

/CHRS FUNCTION

/RETURNS 1 CHAR STRING FOR THE VALUE OF X

OR6 D 11
12000

13400	0000	CHR,	0	
13401	4514	JMS I INTL		/FIX X TO 12 BIT INTEGER
13402	4544	JMS I BSWL		/TREAT THE RIGHTMOST 6 BITS AS CHAR
13403	3621	DCA I SACL		/AND PUT INTO SAC
13404	7040	CMA		
13405	3032	DCA STRLEN		/SET SAC LENGTH TO 1
13406	5326	JMP RETMOD		/SET TO SMODE AND RETURN

/ASC FUNCTION

/RETURNS DECIMAL ASCII FOR 1 CHAR STRING IN FAC

13407	0000	ASC,	0	
13410	1621	TAD I SACL		/GET FIRST TWO CHARS OF STRING
13411	4544	JMS I BSWL		/WE WANT LEFT CHAR
13412	0075	AND K0077		/SO ISOLATE IT
13413	5620	JMP I FLOATB		/FLOAT RESULT INTO FAC AND RETURN

/LEN FUNCTION

/RETURNS LENGTH OF SAC IN FAC

13414	0000	LEN,	0	
13415	1032	TAD STRLEN		/LENGTH OF STRING IN SAC
13416	7041	CIA		/MAKE POSITIVE
13417	5620	JMP I FLOATB		/FLOAT RESULT AND RETURN

13420 4144 FLOATB, FLOATS

13421 0321 SACL, SAC

/STRS FUNCTION

/RETURNS ASCII STRING FOR NUMBER IN FAC

13422	0000	STR,	0	
13423	4657		JMS I ASCOLK	/GET ASCII FOR FAC INTO INTERMEDIATE BUFFER
13424	3032		DCA STRLEN	/ZERO FAC
13425	7101		CLL IAC	
13426	1111		TAD SACPTR	
13427	4526		JMS I STHINL	/INITIALIZE STH TO SAC
13430	4560		JMS I STHRST	/SET DF TO STH TO 0
13431	2061		ISZ TEMP10	/MOVE PAST LEADING SPACE
13432	1461		TAD I TEMP10	/GET SIGN OF NUMBER
13433	1261		TAD MINUSP	/IS IT "+"
13434	7640		SZA CLA	
13435	5240		JMP STSLP	/NO-IT IS "-" SO LEAVE IT ALONE
13436	1260		TAD CCC240	/YES-REPLACE IT WITH A SPACE
13437	3461		DCA I TEMP10	
13440	2061	STSLP,	ISZ TEMP10	/BUMP POINTER
13441	1461		TAD I TEMP10	/GET CHAR FROM INTERMEDIATE BUFFER
13442	1055		TAD M215	/IS IT CR?
13443	7450		SNA	
13444	5326		JMP RETMOD	/YES-RETURN IN SMODE
13445	1256		TAD MCRMAL	/IS IT ALTMODE?
13446	7650		SNA CLA	
13447	5240		JMP STSLP	/YES-IGNORE IT
13450	1461		TAD I TEMP10	/NO-GET CHAR AGAIN
13451	4530		JMS I STH	/PUT IN SAC
13452	7240		CLA CMA	
13453	1032		TAD STRLEN	/"BUMP" SAC COUNTER"
13454	3032		DCA STRLEN	
13455	5240		JMP STSLP	
13456	7616	MCRMAL,	7616	
13457	1260	ASCOLK,	ASCOUT	
13460	0240	CCC240,	240	
13461	7525	MINUSP,	-253	

/VAL FUNCTION

/RETURNS NUMBER IN FAC FOR STRING IN SAC

```

13462 0000 VAL, 0
13463 7101 CLL IAC
13464 1111 TAD SACPTR
13465 4527 JMS I LDHINL /INITIALIZE LDH TO SAC
13466 4557 JMS I LDHRST
13467 1032 TAD STRLEN
13470 3222 DCA VALCNT /COUNT OF CHARS TO INPUT
13471 1301 TAD STCGTJ /JMS TO VALGET
13472 3700 DCA I INPTCL /PUT IN INPUT ROUTINE IN PLACE OF KRB
13473 4677 JMS I FFINLK /CALL FPP INPUT ROUTINE
13474 1302 TAD GETCHG /JMS TO GETCH
13475 3700 DCA I INPTCL /RESTORE IN INPUT ROUTINE
13476 5513 JMP I ILOOPL /DONE

13477 5200 FFINLK, FFIN
13500 5350 INPTCL, INPUT+1
13501 4554 STCGTJ, JMS I VALLK
13502 4542 GETCHG, JMS I GETCHL

13503 0000 VALGET, 0
13504 1222 TAD VALCNT /GET # OF CHARS LEFT
13505 7650 SNA CLA /ANY MORE?
13506 5323 JMP ENVAL /NO-SEND A CR TO FPP INPUT ROUTINE
13507 4531 JMS I LDH /YES-HET CHAR
13510 3053 DCA CHAR /SAVE
13511 1053 TAD CHAR
13512 1106 TAD KM40 /SUBTRACT 40
13513 7710 SPA CLA /IS CHAR <40?
13514 1076 TAD K0100 /YES-IT IS IN 300 SERIES
13515 1077 TAD K200 /TURN ON PARITY BIT
13516 1053 TAD CHAR /BUILD 8 BIT CHAR
13517 3053 DCA CHAR
13520 2222 ISZ VALCNT /DECREASE COUNT
13521 7000 NOP
13522 5703 JMP I VALGET /RETURN WITH CHAR IN AC

13523 1054 ENVAL, TAD K215
13524 3053 DCA CHAR
13525 5703 JMP I VALGET

3422 VALCNT=STR

13526 7001 RETMOD, IAC
13527 3063 DCA MODESW /SET TO STRING MODE
13530 5513 JMP I ILOOPL /RETURN

```

```

2200      *2200
          NOPUNCH
3600      *OVERLAY+200
          ENPUNCH
          IFNZRO EAE <
          NOPUNCH
          >

```

/DATE FUNCTION

```

13600 0000 DATE, 0
13601 1020      TAD CDFIO
13602 3203      DCA ,+1      /CDF TO FIELD THAT 17600 SITS IN
13603 3603      .          /DF TO 17600 FIELD
13604 1031      TAD PSFLAG   /GET RESIDENT STATUS FLAG
13605 7104      CLL RAL      /TD8/E BIT TO LINK
13606 7620      SNL CLA      /IS PG 17600 AT N7400?
13607 5213      JMP N7666    /NO-GET DATE FROM N7666
13610 1667      TAD I L7466  /YES-GET DATE
13611 3040      DCA TEMP1    /SAVE
13612 5215      JMP DATCOM

13613 1670      N7666, TAD I L7666
13614 3040      DCA TEMP1    /SAVE
13615 1040      DATCOM, TAD TEMP1 /GET DATE AGAIN
13616 7640      SZA CLA      /IS IT EMPTY?
13617 1276      TAD KKM10    /NO-SET STRING COUNT TO 8
13620 3032      DCA STRLEN   /YES-RETURN NULL STRING
13621 6201      CDF
13622 1111      TAD SACPTR
13623 3015      DCA XR5      /POINTS TO SAC
13624 1040      TAD TEMP1
13625 7106      CLL RTL
13626 7006      RTL
13627 7004      RAL          /MONTH TO BITS 8-11
13630 0074      AND K0017    /ISOLATE
13631 4260      JMS ASCON    /CONVERT TO ASCII
13632 3415      DCA I XR5    /PUT IN SAC
13633 1040      TAD TEMP1    /DATE
13634 7112      CLL RTR
13635 7010      RAR          /DAY TO BITS7-11
13636 0271      AND K0037C   /ISOLATE
13637 4260      JMS ASCON    /CONVERT TO ASCII
13640 4544      JMS I BSWL   /SWAP DIGITS
13641 3007      DCA TEMP2
13642 1007      TAD TEMP2
13643 0075      AND K0077    /DAY DIGIT 1
13644 1274      TAD K5700    /"/N"
13645 3415      DCA I XR5    /PUT IN STRING
13646 1007      TAD TEMP2    /DAY DIGITS AGAIN
13647 0104      AND K7700    /DAY DIGIT 2
13650 1273      TAD K0057    /"/N/"
13651 3415      DCA I XR5    /ADD TO STRING
13652 1040      TAD TEMP1    /DATE
13653 0275      AND K0007C   /YEAR

```

13654	4260	JMS ASCON	
13655	1272	TAD K0700	/"7N"
13656	3415	DCA I XRS	/FINISH OFF STRING
13657	5665	JMP I RETMDL	/RETURN IN SMODE
13660	0000	ASCON, 0	
13661	1266	TAD DATABA	/ADDR OF DATE TABLE
13662	3042	DCA TEMP3	/POINTER TO RIGHT SET OF DIGITS
13663	1442	TAD I TEMP3	/GET TWO ASII DIGITS FROM TABLE
13664	5660	JMP I ASCON	
13665	3526	RETMDL, RETMOD	
13666	4525	DATABA, DATTAB-1	
13667	7466	L7466, 7466	
13670	7666	L7666, 7666	
13671	0037	K0037C, 37	
13672	0700	K0700, 700	
13673	0057	K0057, 57	
13674	5700	K5700, 5700	
13675	0007	K0007C, 7	
13676	7770	KKM10, -10	

/TRACE FUNCTION PRINTER, WHEN TRACE IS ENABLED, THIS ROUTINE
/PRINTS THE LINE # EACH TIME IT IS STORED

```

13677 0000 TPRINT, 0
13700 4741 JMS I LMAKEL /MAKE LINE # INTO FIVE DIGITS
13701 1342 TAD KEX
13702 4522 JMS I XPUT /PRINT "X"
13703 1347 TAD CC240
13704 4522 JMS I XPUT /PRINT A SPACE
13705 1345 TAD DIG1A /ADDR OF FIRST DIGIT-1
13706 3015 DCA XRS /IN XRS
13707 1415 IGS, TAD I XRS /GET DIGIT OF LINE NUMBER
13710 3277 DCA TPRINT /SAVE IT
13711 1346 TAD MM260
13712 1277 TAD TPRINT /COMPARE IT TO 0
13713 7650 SNA CLA /IS IT A 0?
13714 5307 JMP IGS /YES=IGNORE LEADING ZEROES
13715 1277 PREST, TAD TPRINT /NO-GET CHAR AGAIN
13716 1055 TAD M215
13717 7650 SNA CLA /IS IT A CR?
13720 5326 JMP TDONE /YES=LINE NUMBER IS PRINTED
13721 1277 TAD TPRINT /NO-GET CHAR A THIRD TIME
13722 4522 JMS I XPUT /TYPE IT
13723 1415 TAD I XRS /GET NEXT CHAR
13724 3277 DCA TPRINT
13725 5315 JMP PREST /AND LOOP
13726 1347 TDONE, TAD CC240
13727 4522 JMS I XPUT /FOLLOW LINE # WITH A SPACE
13730 1342 TAD KEX
13731 4522 JMS I XPUT /TYPE ANOTHER "X"
13732 1343 TAD CCR
13733 4522 JMS I XPUT /TYPE,CR,LF
13734 1344 TAD CLF
13735 4522 JMS I XPUT
13736 4510 JMS I PRINT /EMPTY RING BUFFER OF TRACE NUMBER
13737 5336 JMP ,=1
13740 5513 JMP I ILOOP /DONE

13741 4067 LMAKEL, LMAKE
13742 0245 KEX, 245
13743 0215 CCR, 215
13744 0212 CLF, 212
13745 4133 DIG1A, DIG1-1
13746 7520 MM260, -260
13747 0240 CC240, 240

```



```

2400      *2400
4000      NOPUNCH
          *OVERLAY+400
          ENPUNCH
          IFNZRO EAE <
          NOPUNCH
          >

```

/TRACE FUNCTION-ROUTINE TO TURN TRACE ON AND OFF

```

14000 0000 TRACE, 0
14001 1045      TAD HORD      /GET HI MANTISSA OF ARG
14002 7650      SNA CLA      /WHICH?
14003 5210      JMP TOFF     /FOR 0,TURN TRACE OFF
14004 1212      TAD KNOP     /TURN TRAC ON
14005 3607      DCA I HOOKL   /BY NOP ING INSTRUCTION AT TRHOOK
14006 5513 TRREST, JMP I ILOOPL

14007 1144 HOOKL, TRHOOK

14010 1206 TOFF, TAD TRREST   /TURN OFF TRACE
14011 5205      JMP TRREST-1  /BY RESTOREING JMP TO TRHOOK

14012 7000 KNOP, 7000

```

/ERROR ROUTINE

```

14013 0000 ERRORR, 0
14014 4510      JMS I PRINT   /PURGE TTY RING BUFFER
14015 5214      JMP , -1     /BEFORE PRINTING ERROR
14016 1354      TAD ETABA     /ADDR OF ERROR TABLE
14017 3014      DCA XR4       /POINTS INTO ERROR TABLE
14020 1414 FERRLP, TAD I XR4   /GET 2 CHAR ERROR CODE
14021 3040      DCA TEMP1     /SAVE
14022 1040      TAD TEMP1
14023 4544      JMS I BSWL    /FIRST CHAR TO RIGHT
14024 0075      AND K0077     /STRIP TO 6 BIT
14025 1266      TAD K0300     /MAKE 8 BIT (LETTERS ONLY ALLOWED)
14026 3321      DCA ESTRNG    /PUT IN MESSAGE
14027 1040      TAD TEMP1     /2 CHAR CODE AGAIN
14030 0075      AND K0077     /SECOND CHAR
14031 1266      TAD K0300     /MAKE LETTER
14032 3322      DCA ESTRNG+1   /PUT IN MESSAGE
14033 1414      TAD I XR4     /GET ERROR CODE +1
14034 1657      TAD I ERRET    /COMPARE AGAINST RETURN ADDRESS
14035 7640      SZA CLA       /MATCH?
14036 5220      JMP FERRLP    /NO-TRY NEXT ONE
14037 4267      JMS LMAKE     /MAKE THE LINE # INTO DECIMAL DIGITS
14040 1343      TAD ESTRA     /ADDR OF MESSAGE
14041 3015      DCA XR5
14042 1415 ETLOP, TAD I XR5    /GET MESSAGE CHAR
14043 7510      SPA          /DONE? (MESSAGE ENONS WITH - NUMBER
14044 5247      JMP FATCHK     /YES-DETERMINE ERROR TYPE

```

14045	4522	JMS I XPUT	/NO-PUT CHAR IN RING BUFFER
14046	5242	JMP ETLOP	
14047	7200	FATCHK, CLA	
14050	1657	TAD I ERRET	/GET RETURN ADDRESS
14051	3213	DCA ERRORR	/AND STORE IT
14052	1355	TAD MFATAL	/-ADDR OF FATAL ERRORS
14053	1014	TAD XR4	/ADDR OF THIS ERROR
14054	7700	SMA CLA	/FATAL ERROR?
14055	5613	JMP I ERRORR	/NO-NEXT INST
14056	5660	JMP I STOPI	/YES-TERMINATE RUN
14057	1460	ERRET, ERRDIS	
14060	0561	STOPI, FSTOPN	
14061	0000	MAKED, 0	
14062	0074	AND K0017	/ISOLATE BCD DIGIT
14063	1265	TAD K260	/MAKE ASCII DIGIT
14064	5661	JMP I MAKED	
14065	0260	K260, 260	
14066	0300	K0300, 300	

/SUBROUTINE LMAKE-MAKES THE CURRENT LINE NUMBER INTO FIVE DIGITS
/STARTING AT DIG1

14067	0000	LMAKE, 0	
14070	1065	TAD LINEHI	/YES:GET HI LINE #
14071	4261	JMS MAKED	/GET DIGIT 2
14072	3335	DCA DIG2	/PUT IN MESSAGE
14073	1065	TAD LINEHI	
14074	7112	CLL RTR	
14075	7012	RTR	
14076	4261	JMS MAKED	/GET DIGIT 1
14077	3334	DCA DIG1	/AND PUT IN MESSAGE
14100	1066	TAD LINELO	/DOGOTS 3,4, AND 5
14101	4261	JMS MAKED	/GET DIGIT 5
14102	3340	DCA DIG5	
14103	1066	TAD LINELO	
14104	7112	CLL RTR	
14105	7012	RTR	
14106	4261	JMS MAKED	/GET DIGIT 4
14107	3337	DCA DIG4	/AND PUT IN MESSAGE
14110	1066	TAD LINELO	
14111	7104	CLL RAL	
14112	7006	RTL	
14113	7006	RTL	
14114	4261	JMS MAKED	/GET DIGIT 3
14115	3336	DCA DIG3	/MESSAGE NOW COMPLETE
14116	5667	JMP I LMAKE	

/ERROR MESSAGE

14117	0215	EMESS,	215	
14120	0212		212	
14121	0000	ESTRNG,	0000	
14122	0000		0000	
14123	0240		240	
14124	0301		301	/A
14125	0324		324	/T
14126	0240		240	
14127	0314		314	/L
14130	0311		311	/I
14131	0316		316	/N
14132	0305		305	/E
14133	0240		240	
14134	0000	DIG1,	0	
14135	0000	DIG2,	0	
14136	0000	DIG3,	0	
14137	0000	DIG4,	0	
14140	0000	DIG5,	0	
14141	0215		215	
14142	0212		212	
14143	4116	ESTRA,	EMESS-1	/MINUS NUMBER TO END ABOVE MESSAGE

/ROUTINE TO FLOAT FAC AND RETURN

14144	3045	FLOATS,	DCA HORD	/NUMBER TO BE FLOATED IN HORD
14145	3046		DCA LORD	/CLEAR LORD
14146	3007		DCA TEMP2	/CLEAR FPP OVERFLOW
14147	1353		TAD CC13	/SET EXP TO 11
14150	3044		DCA EXP	
14151	4536		JMS I FNORL	/NORMALIZE
14152	5513		JMP I ILOOP	/RETURN
14153	0013	CC13,	13	

/ERROR TABLE
/ENTRY FORMAT-
/

2 CHAR 6-BIT ERROR CODE (LETTERS ONLY)
-(ADOR OF CALL)-1

14154	4155	ETABA,	ETAB-1	
14155	3524	MFATAL,	-EFATAL	
14156	0602	ETAB,	0602	/FB
14157	3765		-FB-1	/ATTEMPT TO OPEN AN ALREADY OPEN FILE
14160	0722		0722	/GR
14161	1240		-GR-1	/RETURN WITHOUT A GOSUB
14162	2622		2622	/VR
14163	4774		-VR-1	/ATTEMPT TO READ VARIABLE LENGTH FILE
14164	2325		2325	/SU
14165	7154		-SU-1	/SUBSCRIPT ERROR
14166	0405		0405	/DE
14167	7220		-DE-1	/DEVICE DRIVER ERROR
14170	0506		0506	/EF
14171	7217		-EF-1	/LOGICAL EOF
14172	1705		1705	/OE
14173	6264		-OE-1	/DRIVER ERROR WHILE OVERLAYING
14174	0615		0615	/FM
14175	6153		-FM-1	/ATTEMPT TO FIX MINUS NUMBER
14176	0617		0617	/FO
14177	6140		-FO-1	/ATTEMPT TO FIX NUMBER >4095
14200	0616		0616	/FN
14201	5772		-FN-1	/ILLEGAL FILE #
14202	2303		2303	/SC
14203	5545		-SC-1	/ATTEMPT TO OVERFLOW SAC ON CONCATENATE
14204	0611		0611	/FI
14205	5470		-FI-1	/ATTEMPT TO CLOSE OR USE UNOPENED FILE
14206	0401		0401	/DA
14207	5461		-DA-1	/ATTEMPT TO READ PAST END OF DATA LIST
14210	0723		0723	/GS
14211	5442		-GS-1	/TOO MANY NESTED GOSUBS
14212	2322		2322	/SR
14213	4647		-SR-1	/ATTEMPT TO READ STRING FROM NUMERIC FILE
14214	2327		2327	/SW
14215	4515		-SW-1	/ATTEMPT TO WRITE STRING INTO NUMERIC FILE
14216	2001		2001	/PA
14217	3362		-PA-1	/ILLEGAL ARG IN POS
14220	0603		0603	/FC
14221	4336		-FC-1	/OS/8 ERROR WHILE CLOSING TENTATIVE FILE
14222	0311		0311	/CI
14223	4155		-CI-1	/INQUIRE FAILURE IN CHAIN
14224	0314		0314	/CL
14225	4133		-CL-1	/LOOKUP FAILURE IN CHAIN
14226	1116		1116	/IN
14227	3745		-IN-1	/INQUIRE FAILURE IN OPEN
14230	0417		0417	/DO
14231	3722		-DO-1	/NO MORE ROOM FOR DRIVERS
14232	0605		0605	/FE
14233	3665		-FE-1	/FETCH ERROR IN OPEN
14234	0217		0217	/BO

624
12625

1720
2606

14235	3562	-BO=1	/NO MORE FILE BUFFERS AVAILABLE
14236	0516	0516	/EN
14237	3470	-EN=1	/ENTER ERROR IN OPEN
14240	1106	1106	/IF
14241	3353	-IF=1	/ILLEGAL DEV:FILENAME SPECIFICATION
14242	2314	2314	/SL
14243	7252	-8L=1	/STRING TOO LONG OR UNDEFINED
14244	1726	1726	/OV
14245	1423	-OO=1	/NUMERIC OR INPUT OVERFLOW
14246	1415	1415	/LM
14247	1417	-LM=1	/ATTEMPT TO TAKE LOG OF NEG # OR 0
14250	0515	0515	/EM
14251	4162	-EM=1	/ATTEMPT TO EXPONENTIATE A NEG NUMBER TO A REAL POWER
14252	1101	1101	/IA
14253	6312	-IA=1	/ILLEGAL ARGUMENT IN USER FUNCTION
/*****			
EFATAL,			
/*****			
/ERRORS BEFORE THIS LABEL ARE FATAL			
/*****			
14254	2205	2205	/RE
14255	4767	-RE=1	/ATTEMPT TO READ PAST EOF
14256	2705	2705	/WE
14257	4745	-WE=1	/ATTEMPT TO WRITE PAST EOF
14260	0426	0426	/DV
14261	1422	-DV=1	/ATTEMPT TO DIVIDE BY 0
14262	2324	2324	/ST
14263	5333	-ST=1	/STRING TRUNCATION ON INPUT
14264	1117	1117	/IO
14265	1103	-IO=1	/TTY INPUT BUFFER OVERFLOW

/SEGS FUNCTION

/RETURNS SEGMENT OF XS BETWEEN Y AND Z

/IF Y<=0, THEN Y TAKEN AS 1

/IF Y>LEN(XS), NULL STRING RETURNED

/IF Z<=0, NULL STRING RETURNED

/IF Z>LEN(XS), Z IS SET=LEN(XS)

/IF Z<Y, NULL STRING IS RETURNED

14266	0000	SEG,	0	
14267	7001		IAC	
14270	3063		DCA MODESW	/RETURN IN STRING MODE
14271	1045		TAD HORD	/IS Y>0?
14272	7740		SMA SZA CLA	
14273	5276		JMP ,+3	/YES
14274	4534		JMS I FGETL	/NO=SET Y TO 1
14275	4476		ONE1	
14276	4535		JMS I FPUTL	/SAVE Y
14277	1175		FPPTM1	
14300	4514		JMS I INTL	/FIX Y
14301	1032		TAD STRLEN	/COMPARE TO STRLEN
14302	7740		SMA SZA CLA	/Y>LEN(XS)?
14303	5365		JMP NULLST	/YES=RETURN THE NULL STRING
14304	3064		DCA INSAV	/FAKE POINTER TO SCALAR #0
14305	4707		JMS I ARGPLK	/GET ADDR OF Z
14306	4534		JMS I FGETL	/LOAD Z INTO FAC
14307	0307	ARGPLK,	ARGPRE	/LOC SKIPPED BY FPP SO WE PUT CONST HERE
14310	1045		TAD HORD	/HI MANTISSA OF Z
14311	7750		SPA SNA CLA	/IS Z<0?
14312	5365		JMP NULLST	/YES=RETURN THE NULL STRING
14313	4514		JMS I INTL	/NO=FIX Z
14314	1032		TAD STRLEN	/COMPARE TO STRING LENGTH
14315	7710		SPA CLA	
14316	5331		JMP ZMINY	/Z<=LEN(XS)
14317	3046		DCA LORD	/Z>LEN(XS) SO SET Z=LEN(XS)
14320	1330		TAD KK13	
14321	3044		DCA EXP	
14322	1032		TAD STRLEN	
14323	7041		CIA	/MAKE LENGTH POSITIVE
14324	3045		DCA HORD	
14325	4536		JMS I FNORL	/FLOAT LENGTH
14326	4707		JMS I ARGPLK	
14327	4535		JMS I FPUTL	/SAVE NEW Z
14330	0013	KK13,	13	
14331	6201	ZMINY,	CDF	
14332	4534		JMS I FGETL	/LOAD Y
14333	1175		FPPTM1	
14334	4707		JMS I ARGPLK	/GET ADDR OF Z
14335	4767		JMS I FISUBL	/Z=Y
14336	6201	CDF000,	CDF	
14337	1045		TAD HORD	/GET HI ORDER Z=Y
14340	7710		SPA CLA	/IS Y<Z?
14341	5365		JMP NULLST	/NO=RETURN NULL STRING
14342	4514		JMS I INTL	/FIX Z=Y
14343	7040		CMA	/ADD ONE AND NEGATE

14344	3071	DCA STRCNT	/STORE AS SEG LENGTH
14345	4534	JMS I PGETL	
14346	1175	FPPTM1	/RETRIEVE Y AGAIN
14347	4514	JMS I INTL	/FIX Y
14350	7110	CLL RAR	/DIVIDE BY TWO
14351	7430	SZL	
14352	7001	IAC	
14353	7020	CML	
14354	1111	TAD SACPTR	/USE Y/2 AS DISPLACEMENT FROM START OF SAC
14355	4527	JMS I LDHINL	/INITIALIZE LDH
14356	4557	JMS I LDHRST	
14357	1111	TAD SACPTR	
14360	7101	CLL IAC	
14361	4526	JMS I STHINL	/INITIALIZE STH TO SAC
14362	4560	JMS I STHRST	
14363	3032	DCA STRLEN	/ZERO SAC
14364	5770	JMP I SEGCML	/USE CODE IN CONCATENATE TO DO THE REST
14365	3032	NULLST, DCA STRLEN	/ZERO SAC
14366	5513	JMP I ILOOPL	/RETURN
14367	5400	FISUBL, FFSUB1	
14370	2222	SEGCML, SEGCOM	


```

3000      *3000
          NOPUNCH
4400      *OVERLAY+1000
          ENPUNCH
          IFNZRD EAE <
          NOPUNCH
          >

```

```

/POS FUNCTION
/RETURNS THE POSITION IN XS OF YS STARTING AFTER Z

```

```

14400 0000 POS, 0
14401 7100 CLL
14402 3064 DCA INSAV /FAKE AS STRING CALL TO STRING 0
14403 4704 JMS I STFINK /FIND YS
14404 3706 DCA I LDHCOL /GET YS CHARS FROM DF N
14405 1071 TAD STRCNT /# OF CHARS IN YS
14406 7650 SNA CLA /IS YS THE NULL STRING?
14407 5273 JMP ONERET /YES=RETURN 1 AS POSITION
14410 1032 TAD STRLEN /NO=# OF CHARS IN XS
14411 7650 SNA CLA /IS XS THE NULL STRING?
14412 5267 JMP ZRORET /YES=RETURN 0
14413 1045 TAD HORD /NO=GET HORD OF Z
14414 7710 SPA CLA /IS Z>=0?
14415 4516 PA, JMS I ERROR /NO=ILLEGAL ARGUMENT
14416 4514 JMS I INTL /FIX Z
14417 3301 DCA POSITN /USE IT AS POSITION TO START SEARCH
14420 1301 TAD POSITN
14421 1032 TAD STRLEN /COMPARE POSITION TO MAXIMUM LENGTH OF STRING
14422 7700 SNA CLA
14423 5215 JMP PA /Z IS PAST END OF STRING=ERROR
14424 1301 POSSET, TAD POSITN /SEARCH START POSITION IN XS
14425 7110 CLL RAR /DIVIDE BY 2
14426 7430 SZL
14427 7001 IAC
14430 7020 CML
14431 1111 TAD SACPTR /USE AS DISPLACEMENT OFF START OF SAC
14432 3302 DCA LDHPR /POINTS TO NEXT CHAR FROM XS
14433 7620 SNL CLA /IF LINK=0,GET RIGHT HALF
14434 7040 CMA /ELSE GET LEFT HALF
14435 3303 DCA LDHPSW
14436 1072 TAD STRPTR
14437 7101 CLL IAC /BUMP PAST CHAR COUNT
14440 4527 JMS I LDHINL /INITIALIZE LDH TO YS
14441 1032 TAD STRLEN /# OF CHARS IN XS
14442 3043 DCA TEMP4 /COUNTER
14443 1071 TAD STRCNT /# OF CHARS IN YS
14444 3042 DCA TEMP3 /COUNTER
14445 4307 SRCLP, JMS XDGET /GET CHAR FROM XS
14446 4531 JMS I LDH /GET CHAR FROM YS
14447 6201 CDF
14450 7041 CIA /NEGATE CHAR FROM YS
14451 1040 TAD TEMP1 /COMPARE WITH CHAR FROM XS
14452 7650 SNA CLA /DO THEY MATCH?

```

```

14453 5262      JMP SCONTU      /YES=CONTINUE MATCH TO NEXT CHAR IN XS AND YS
14454 2301      ISZ POSITN      /BUMP POSITION TO BE CHECKED
14455 1301      TAD POSITN      /GET POSITION NOW CHECKING
14456 1032      TAD STRLEN      /COMPARE AGAINST LENGTH OF STRING
14457 7740      SMA SZA CLA      /ANY MORE TO COME?
14460 5267      JMP ZRORRET     /NO-SEARCH FAILS
14461 5224      JMP POSSET      /YES=START COMPARING NEXT POSITION

14462 2042      SCONTU, ISZ TEMP3 /MORE CHARS IN YS?
14463 7410      SKP              /YES
14464 5271      JMP RETPOS      /NO-MATCH SUCCEEDS=RETURN POSITN
14465 2043      ISZ TEMP4      /MORE IN XS?
14466 5245      JMP SRCLP      /YES=CONTINUE MATCH
14467 4537      ZRORRET, JMS I FCLR /NO-SEARCH FAILS=RETURN 0
14470 5513      JMP I ILOOPL

14471 1301      RETPOS, TAD POSITN /GET POSITION OF MATCH
14472 5705      JMP I FLOABL     /FLOAT RESULT AND RETURN

14473 4534      ONERET, JMS I FGETL /1 INTO FAC
14474 4476      ONE1
14475 5513      JMP I ILOOPL

14476 0001      ONE1, 1
14477 2000      2000
14500 0000      0
14501 0000      POSITN, 0
14502 0000      LDHPR, 0
14503 0000      LDHPSW, 0
14504 1666      STFINK, STFIND
14505 4144      FLOABL, FLOATS
14506 2647      LDHCDL, LDHDF

/ROUTINE TO GET SUCCESSIVE HALFWORDS FROM XS

14507 0000      XDGET, 0
14510 1303      TAD LDHPSW      /HALFWORD SWITCH
14511 7650      SNA CLA        /LEFT OR RIGHT?
14512 5323      JMP XDWRITE     /RIGHT
14513 1702      TAD I LDHPR     /LEFT-GET CHARS
14514 4544      JMS I BSWL      /SWAP BYTES
14515 0075      XLCOM, AND K0077 /ISOLATE CHAR
14516 3040      DCA TEMP1      /SAVE
14517 1303      TAD LDHPSW      /HALFWORD SWITCH
14520 7040      CMA            /FLIP IT
14521 3303      DCA LDHPSW
14522 5707      JMP I XDGET     /RETURN

14523 1702      XDWRITE, TAD I LDHPR /GET 2 CHARS
14524 2302      ISZ LDHPR      /BUMP POINTER TO NEXT WORD
14525 5315      JMP XLCOM

```

/DATE TABLE-USED TO CONVERT BINARY NUMBERS<31 INTO ASCII CHARACTERS

14526	6061	DATTAB, 6061	/01
14527	6062	6062	/02
14530	6063	6063	/03
14531	6064	6064	/04
14532	6065	6065	/05
14533	6066	6066	/06
14534	6067	6067	/07
14535	6070	6070	/08
14536	6071	6071	/09
14537	6160	6160	/10
14540	6161	6161	/11
14541	6162	6162	/12
14542	6163	6163	/13
14543	6164	6164	/14
14544	6165	6165	/15
14545	6166	6166	/16
14546	6167	6167	/17
14547	6170	6170	/18
14550	6171	6171	/19
14551	6260	6260	/20
14552	6261	6261	/21
14553	6262	6262	/22
14554	6263	6263	/23
14555	6264	6264	/24
14556	6265	6265	/25
14557	6266	6266	/26
14560	6267	6267	/27
14561	6270	6270	/28
14562	6271	6271	/29
14563	6360	6360	/30
14564	6361	6361	/31

```

////////////////////
////////////////////
//////////////////// OVERLAY 3-FILE MANIPULATING
//////////////////// FUNCTIONS
////////////////////
////////////////////

```

3400

*3400

/FILE CLOSING ROUTINE

```

13400/ 3400 ANDPTR, ANDLST
13401 7776 ANDLST, 7776 /MASKS FOR CLEARING BUFFER AND HANDLER STATUS BITS B7+11
13402 7775 7775 B7+10
13403 7773 7773 B7+9
13404 7767 VSA 7767 B7+8 (Also Dummy for Sys Handler 7607)

13405 14625 CLOSE, TAD ENTND 6 /GET FILE #
13406 7650 SNA CLA /IS IT TTY?
13407 5513 5570 JMP I ILOOPI /YES-DON'T DO ANYTHING JMP E 170
13408 4523 4567 JMS I FIDLE=2304 /SEE IF FILE OPEN (Error if not open)
13409 4550 4540 JMS I FTYPL /IS FILE NUMERIC?
13410 5217 5220 JMP NOCZ /YES-DON'T OUTPUT ^Z
13411 4731 4777 JMS I FTYPSE /NO-IS FILE VARIABLE LENGTH?
13412 5217 5220 JMP NOCZ /NO-DON'T OUTPUT ^Z
13413 1330 1376 TAD K232 /YES
13414 4512 4550 JMS I PUTCHL /WRITE A ^Z IN FILE JMS I 150
13415 4732 NOCZ, JMS I WRBLKK /WRITE LAST BLOCK IF IT HAS CHANGED
13416 4556 JMS I PISWAP /RESTORE 17600
13417 4731 JMS I FTYPSE=2307 /IS FILE FIXED LENGTH?
13418 5242 JMP CLOSED /YES-NO NEED TO CLOSE THE FILE
13419 1571 TAD I WORD6 /NO-GET FILE LENGTH
13420 3240 DCA LENG /PUT IN CLOSE CALL
13421 1174 TAD WORD11
13422 3237 DCA FNAP /POINTER TO FILE NAME
13423 1563 TAD I WORD0
13424 7106 CLL RTL
13425 7006 RTL
13426 7004 RAL /GET DEVICE NUMBER INTO BITS 8-11
13427 0074 AND K0017 /ISOLATE IT
13428 6212 CIF 10
13429 4504 JMS I K7700 /CALL USR
13430 0004 4 /CLOSE
13431 3437 FNAP, . /POINTER TO FILE NAME
13432 3440 LENG, .
13433 4516 FC, JMS I ERROR /FILE CLOSING ERROR
13434 1564 CLOSED, TAD I WORD1 /GET BUFFER ADDRESS
13435 7106 CLL RTL
13436 7006 RTL /BUFFER NUMBER INTO AC
13437 7004 RAL /BITS 10,11
13438 0333 AND K0003 ✓ /STRIP
13439 1200 TAD ANDPTR /USE AS INDEX INTO MASKS
13440 3040 DCA TEMP1
13441 1036 TAD BMAP /BUFFER STATUS MAP

```

See next page for BMAP definition

/OS/8 BASIC RUNTIME SYSTEM

PAL8-V7 10/24/72 PAGE 79-1

13452 0440
13453 3036

AND I TEMP1
DCA BMAP

/CLEAR THE BIT FOR THIS BUFFER



/STARING ADDRESSES OF
/FILE TABLE ENTRIES

```

141153 PATCH, ISZ TEMP30)
JMP L (CHECKL
TAD M4444)
TAD J 34440
SEA CLA
JMP L PATCHL

```

13537 6746
13540 6763FILE3+V
FILE4+4

13541 3542 AN2PTR, ANDLS2

MM4,

13542 7774 ANDLS2, 7774

13543 7701 7701

Bits 10, 11

Bits 9, 10

~~13544 0001~~~~Root,~~

13544 1563 INIT, TAD I WORD0 / Header Word

13545 3350

PLA TMWDI

13546 1563

TAD I WORD0

13547 5755

JMP 00

13550 0 TMWDI, 0

3600

*3600

/CHAIN FUNCTION

/SETS UP COMMAND DECODER AREA, THEN CHAINS TO BCOMP.SV

13600	4510	CHAIN,	JMS I PRINT	/EMPTY TTY RING BUFFER
13601	5200		JMP ,=-1	
13602	4556		JMS I P1SWAP	/RESTORE PG 17600
13603	4617		JMS I DNA1	/RESTORE SYS RESIDENT
13604	4620		JMS I DNA2	/GET FILE NAME IN NAME AREA FROM CURRENT FILE
13605	6212		CIF 10	
13606	4504		JMS I K7700	/CALL USR
13607	0010		10	/LOCK IN CORE
13610	1572		TAD I WORD7	
13611	3217		DCA DNA1	/FIRST TWO CHARS OF DEV NAME
13612	1573		TAD I WORD10	/LAST TWO CHARS
13613	3220		DCA DNA2	
13614	6212		CIF 10	
13615	4477		JMS I USR	
13616	0012		12	/INQUIRE
13617	4334	DNA1,	PSWAP2	/DEVICE NAME
13620	4416	DNA2,	NAMEG	
13621	0000	CDIN,	0	
13622	4516	CI,	JMS I ERROR	/ERROR
13623	1221		TAD CDIN	/GET ENTRY POINT OF DRIVER FOR CAHIN FILE
13624	7640		SZA CLA	/IS IT IN CORE?
13625	5234		JMP DISIN	/YES-NO NEED TO FETCH IT
13626	1220		TAD DNA2	/NO-DEVICE # INTO AC
13627	6212		CIF 10	
13630	4477		JMS I USR	
13631	0001		1	/FETCH HANDLER
13632	7001		7001	/INTO PAGE 7000
13633	5222		JMP CI	/MAKE IT LOOK LIKE INQUIRE ERROR
13634	1174	DISIN,	TAD WORD11	
13635	3242		DCA STB	/POINTER TO FILE NAME
13636	1220		TAD DNA2	/GET DEVICE #
13637	6212		CIF 10	
13640	4477		JMS I USR	
13641	0002		2	/LOOKUP
13642	0000	STB,	0	/POINTER TO FILE NAME
13643	0000	FLN,	0	
13644	4516	CL,	JMS I ERROR	
13645	1242		TAD STB	/GET STARTING BLOCK
13646	6211		CDF 10	
13647	3665		DCA I L7620	/STARTING BLOCK IN CD AREA
13650	1243		TAD FLN	/FILE LENGTH
13651	7106		CLL RTL	
13652	7006		RTL	
13653	0266		AND K7760	/PUT IN BITS 0-7
13654	1220		TAD DNA2	/COMBINE WITH DEVICE #
13655	3676		DCA I CBLK	/PUT IN CD AREA
13656	1076		TAD K0100	/SET R SWITCH
13657	3664		DCA I L7644	
13660	1670		TAD I L7605K	/STARTING BLOCK OF COMPILER
13661	3276		DCA CBLK	/INTO COMPILER READ CODE


```

13662 5663      JMP I.+1
13663 4143      CSMOVE          /MOVE THE COMPILER READ TO FIELD 1 AND EXECUTE IT
13664 7644      L7644, 7644
13665 7620      L7620, 7620
13666 7760      K7760, 7760
13667 7621      L7621, 7621
13670 7605      L7605K, 7605
                /CODE TO READ IN COMPILER AND START IT
                /THIS CODE GETS MOVED TO FIELD 1 AND EXECUTED FROM
                /LOC 2001-2013 IN FIELD 1

```

```

13671 6211      CREAD, CDF 10
13672 6202      CIF 0
13673 4613      4613          /"JMS I L7607K"
13674 3700      3700          /31 PAGES
13675 0000      0            /0-7577
13676 7617      CBLK, 7617   /STARTING BLOCK OF COMPILER
13677 7402      HLT          /SYSTEM ERROR, AND SINCE WE ARE PROBABLY CLOBBERED, WE CAN'T REPORT IT
13700 6202      CIF 0
13701 5612      5612          /"JMP I .+1"-START THE COMPILER
13702 7001      7001          /STARTING ADDR OF COMPILER
13703 7607      K7607K, 7607 /LESS THAN THE DESIRED VALUE

```

/ROUTINE FOR INTERPRETER EXIT

```

13713 13704 6031 FSTOP, KSF /IS THE KEYBOARD FLAG SET?
4 13705 5325 33 JMP NOCTC /NO-THERE IS NO CHANGE "C" SENT US HERE
5 13706 1077 TAD K200 /YES-FORCE PARITY BIT
6 13707 6036 KRB /GET CHARACTER
7 13710 1350 61 TAD MCC /COMPARE AGAINST "C"
20 13711 7640 SZA CLA /WAS IT "C"?
21 13712 5325 33 JMP NOCTC /NO-THIS IS A NORMAL EXIT
3 13713 6041 TSF
23 13714 5313 22 JMP .-1
24 13715 1351 62 TAD KUPARO /YES -ECHO -
25 13716 6046 TLS
26 13717 7200 CLA
27 13720 6041 TSF
30 13721 5320 7 JMP .-1
31 13722 1352 63 TAD KC /ECHO "C"
32 13723 6046 TLS
33 13724 7200 CLA
37 13725 4556 NOCTC, JMS I PISWAP 1230 /RESTORE PG 17600
7 13726 4744 55 JMS I P2SWAL /RESTORE PG 2700
40 13727 6211 CDF 10
11 13730 1747 60 TAD I EDBLK /GET BLOCK # FOR EDITOR
2 13731 6201 CDF
3 13732 7450 SNA /SHALL WE CALL THE EDITOR?
4 13733 5745 56 JMP I KL7600 /NO JUST CALL OS/8
5 13734 3340 51 DCA EBLK /YES-PUT THE BLOCK # IN DRIVER CALL
6 13735 4746 57 JMS I LK7607 /CALL SYS DRIVER
7 13736 1600 1700 1600 /READ 7 BLOCKS
50 13737 0000 0 /INTO 0-3377
1 13740 3740 EBLK, . /BLOCK # OF EDITOR

```

PATCH 5335

PATCH 5335

JMP I 3760

USA

13732

13754

33 6041 TSF

JMS I P2SWAL

/RESTORE PG 2700

JMS I LK7607

/CALL SYS DRIVER

/READ 7 BLOCKS

/INTO 0-3377

/BLOCK # OF EDITOR

/INTO 0-3377

/BLOCK # OF EDITOR

17604

52 13741 7402 HLT
53 13742 5743 54 JMP I .+1
54 13743 3012 3012

/SYS ERROR, AND SINCE WE ARE PROBABLY CLOBBERED, WE CAN'T REPORT IT
/START THE EDITOR

55 13744 43344/P2SWAL, PSWAP2
6 13745 7600 KL7600, 7600
7 13746 7607 LK7607, 7607
60 13747 7604 EDBLK, 7604
1 13750 7575 MCC, -203
2 13751 0336 KUPARO, 336
3 13752 0303 KC, 303

4000

*4000

/FILE OPENING ROUTINE

/SITS IN THIS OVERLAY BECUASE THERE IS ROOM HERE, AND THE USR IS

/GOING TO SPIN SYS ANYWAY

14000 1340¹⁷¹ OPENAV, TAD C4 /ALPHANUMERIC,VARIABLE LENGTH
 14001 7001 OPENAF, IAC /ALPHANUMERIC,FIXED LENGTH
 14002 5204 JMP OPENNF
 14003 1340¹⁷¹ OPENNV, TAD C4 /NUMERIC,VARIABLE LENGTH
 14004 3563¹⁷¹ OPENNE, DCA I WORD0 /SET UP HEADER WORD
 14005 1162¹⁰⁶⁵ TAD ENTNO /IS FILE TTY?
 14006 7650¹⁰⁶⁵ SNA CLA
 14007 5513¹⁰⁶⁵ JMP I ILOOPL /YES-DON'T DO ANYTHING *jump 2 170*
 14010 1567¹⁰⁷² TAD I WORD4 /GET HANDLER ENTRY
 14011 7640¹⁰⁷² SZA CLA /IS FILE IDLE?
 14012 4516 FB, JMS I ERROR /ATTEMPT TO OPEN FILE ALREADY OPEN
 14013 4556 ~~JMS I PISWAP~~ /RESTORE 17600
 14014 4741 JMS I NAMEGL /GET FILE DEVICE NAME AND FILE NAME INTO WORDS 7-14 FROM SAC
 14015 6212 CIF 10
 14016 4504 JMS I K7700 /CALL TO USR
 14017 0010 10 /LOCK USR IN CORE
 14020 1572 TAD I WORD7
 14021 3227 DCA DEVNA1 /DEVICE NAME INTO INQUIRE CALL
 14022 1573 TAD I WORD10
 14023 3230 DCA DEVNA2
 14024 6212 CIF 10
 14025 4477 JMS I USR /CALL TO USR
 14026 0012 12 /INQUIRE
 14027 4027 DEVNA1, . /DEVICE NAME
 14030 4030 DEVNA2, .
 14031 0000 ENTRYN, 0 /ENTRY POINT
 14032 4516 IN, JMS I ERROR
 14033 1230 TAD DEVNA2 /GET DEVICE #
 14034 7110 CLL RAR
 14035 7012 RTR /PUT INTO BITS 0-3
 14036 7012 RTR
 14037 1563 TAD I WORD0
 14040 3563 DCA I WORD0 /STORE IN HEADER WORD
 14041 1231 TAD ENTRYN /GET DRIVER ADDRESS
 14042 7440 SZA /IS IT IN CORE?
 14043 5734 JMP I DRIVRL^(#) /YES-NO NEED TO FETCH IT
 14044 1035 TAD DMAP /NO-GET MAP OF DRIVER PAGES
 14045 7110 CLL RAR /PAGE 7000 BIT IN LINK *11 bit*
 14046 7420 SNL /IS PAGE 7000 FREE?
 14047 5262 JMP FREE70 /YES
 14050 7110 CLL RAR /NO-7200 BIT TO LINK *10 Bit*
 14051 7420 SNL /IS PAGE 7200 FREE?
 14052 5273 JMP FREE72 /YES

14053 7110 CLL RAR /NO=7400 BIT TO LINK 9 Bit
 14054 7630 SZL CLA /IS PAGE 7400 FREE?
 14055 4516 DO, JMS I ERROR /NO=NO MORE ROOM FOR DRIVERS
 14056 1103 TAD K7400 /YES=LOAD HANDLER INTO 7400
 14057 3311 DCA FETPAG /SET UP IN FETCH CALL
 14060 1340 TAD C4 /SET BIT 9 TO SHOW PAGE 7400 OCCUPIED
 14061 5303 JMP DFETCH /FETCH DRIVER

 14062 7110 FREE70, CLL RAR /PAGE 7200 BIT TO LINK
 14063 7620 SNL CLA /IS 7200 FREE?
 14064 7001 IAC /YES=THESE IS ROOM FOR A TWO PAGE HANDLER
 14065 1335 TAD K7000
 14066 3311 DCA FETPAG /SET UP FETCH TO USE PAGE 7000
 14067 7326 CLL CLA CML RTL /TURN ON BIT 10
 14070 3336 DCA TPH /SAVE IN TWO PAGE SET WORD
 14071 7001 IAC /SET BIT 11 TO SHOW PAGE 7000 OCCUPIED
 14072 5303 JMP DFETCH /FETCH HANDLER

 14073 7110 FREE72, CLL RAR /7400 BIT TO LINK
 14074 7620 SNL CLA /IS 7400 PAGE FREE?
 14075 7001 IAC /YES=THEN THERE IS ROOM FOR A 2 PAGE HANDLER
 14076 1342 TAD K7200
 14077 3311 DCA FETPAG /SET ADDRESS IN FETCH CALL
 14100 1340 TAD C4
 14101 3336 DCA TPH /IF TWO PAGE LOADED, SET BIT 9 ALSO
 14102 7326 CLL CLA CML RTL /TURN ON BIT 10 TO SHOW PAGE 7200 OCCUPIED
 14103 1035 DFETCH, TAD DMAP /TURN ON PAGE BIT FOR WHERE DRIVER WILL BE LOADED
 14104 3035 DCA DMAP
 14105 1230 TAD DEVNA2 /DEVICE # IN AC
 14106 6212 CIF 10
 14107 4477 JMS I USR /CALL TO USR
 14110 0001 I /FETCH
 14111 4111 FETPAG, /DRIVER ADDRESS
 14112 4516 FE, JMS I ERROR
 14113 6211 CDF 10
 14114 7240 CLA CMA
 14115 1737 TAD I L0037 /GET ADDR OF HANDLER INFO TABLE
 14116 1230 TAD DEVNA2 /USE THE DEVICE # AS AN INDEX INTO THAT TABLE
 14117 3040 DCA TEMP1 /SAVE POINTER
 14120 1440 TAD I TEMP1 /GET THE INFO WORD FOR THE HANDLER JUST FETCHED
 14121 6201 CDF
 14122 7700 SMA CLA /IS HANDLER 2 PAGES LONG?
 14123 5332 JMP DRAP /NO MAP IS COMPLETE
 14124 1336 TAD TPH /YES=UPDATE DRIVER MAP TO INCLUDE
 14125 1035 TAD DMAP /SECOND PAGE OF TWO PAGE HANDLERS
 14126 3035 DCA DMAP
 14127 1073 TAD K0010
 14130 1563 TAD I WORD0 /SET 2 PAGE BIT IN HEADER WORD
 14131 3563 DCA I WORD0
 14132 1311 DRAP, TAD FETPAG /HANDLER ENTRY ADDRESS
 14133 5734 JMP I DRIVRL /PAGE ESCAPE

 14134 4200 DRIVRL, DRIVRN
 14135 7000 K7000, 7000

14136 0000 TPH, 0
 14137 0037 L0037, 37
 14140 0004 C4, 4
 14141 4416 NAMEGL, NAMEG
 14142 7200 K7200, 7200

/ROUTINE TO MOVE THE COMPILER READER INTO FIELD 1 AND START IT

14135 1362 *1362* MOVE, TAD CSTA
 14144 3011 *3011* DCA XR1 /POINTES TO COMPILER STARTING CODE
 14145 1363 *1370* TAD CSTAC
 14146 3040 *3040* DCA TEMP1 /COUNTER
 14147 1361 *1367* TAD KK2000
 14150 3012 *3012* DCA XR2 /MOVE TO LOC 2001 IN FIELD 1
 14151 6201 *6201* CDF
 14152 1411 *1411* TAD I XR1 /GET WORD OF CODE
 14153 6211 *6211* CDF 10
 14154 3412 *3412* DCA I XR2 /MOVE IT
 14155 2040 *2040* ISZ TEMP1 /DONE?
 14156 5351 *5343* JMP .-5 /NO
 14157 6212 *6212* CIF 10 /YES-START IT
 14158 4761 *4767* JMS I .+1
 14160 2000 KK2000, 2000
 14161 3070 CSTA, CREAD-1 *3677*
 14163 7765 CSTAC, -13

Free in 5A
 14153 - 14166

VISA
Change
PK
1415

0
2001
3070
3040
1362
1376
7765
2000
3070
7765

	4200		*4200	
14200	3567	DRIVRN,	DCA I WORD4	/DRIVER ENTRY INTO I/O TABLE
14201	1036		TAD BMAP	/GET BUFFER MAP
14202	7110		CLL RAR	/BUFF1 BIT TO LINK
14203	7420		SNL	/IS IT FREE?
14204	5234		JMP B1	/YES=ASSIGN BUFF1
14205	7010		RAR	/BUFF2 BIT TO LINK
14206	7420		SNL	/IS IT FREE?
14207	5227		JMP B2	/YES=ASSIGN BUFF2
14210	7010		RAR	/BUFF3 BIT TO LINK
14211	7420		SNL	/IS IT FREE
14212	5222		JMP B3	/YES=ASSIGN BUFF3
14213	7010		RAR	/NO-BUFF4 BIT TO LINK
14214	7630		SZL CLA	/IS IT FREE?
14215	4516	B0,	JMS I ERROR	/NO-NO MORE BUFFERS AVAILABLE
14216	1326		TAD K1400	
14217	3564		DCA I WORD1	/SET BUFFER ADDRESS TO 1400
14220	1073		TAD K0010	/SET BUFF4 BIR IN MAP
14221	5237		JMP BUFASS	
14222	7200	B3,	CLA	
14223	1327		TAD K1000	
14224	3564		DCA I WORD1	/SET BUFFER ADDRESS TO 1000
14225	1330		TAD CC4	
14226	5237		JMP BUFASS	/SET BUFF3 BIT IN MAP
14227	7200	B2,	CLA	
14230	1102		TAD K0400	
14231	3564		DCA I WORD1	/SET BUFF ADDRESS TO 400
14232	7326		CLL CML CLA RTL	/SET BUFF2 BIT IN MAP
14233	5237		JMP BUFASS	
14234	7200	B1,	CLA	
14235	3564		DCA I WORD1	/SET BUFF ADDRESS TO 0000
14236	7201		CLA IAC	/TURN ON BUFF1 BIT IN MAP

14237	1036	BUFASS,	TAD BMAP	
14240	3036		DCA BMAP	/UPDATE BUFFER ASSIGNMENT MAP
14241	1563		TAD I WORD0	/GET HEADER WORD
14242	7112		CLL RTR	
14243	7010		RAR	/FIXED,VARIABLE BIT TO LINK
14244	7620		SNL CLA	/IS IT FIXED?
14245	5253		JMP FLOOK	/YES-DO A LOOKUP
14246	1331		TAD CC3	/NO-DO AN ENTER
14247	4275		JMS ENTLOK	/ENTER
14250	3572		DCA I WORD7	/MAXIMUM LENGTH IN WORD 7
14251	3571		DCA I WORD6	/ZERO ACTUAL LENGTH
14252	5260		JMP CLEANP	/FINALIZE I/O TABLE ENTRY
14253	7326	FLOOK,	CLL CML CLA RTL	/2
14254	4275		JMS ENTLOK	/LOOKUP
14255	3571		DCA I WORD6	/ACTUAL LENGTH
14256	1571		TAD I WORD6	
14257	3572		DCA I WORD7	/ALSO EQUALS MAXIMUM LENGTH
14260	3573	CLEANP,	DCA I WORD10	/ZERO COLUMN POINTER
14261	7040		CMA	/-1
14262	1570		TAD I WORD5	/STARTING BLOCK-1
14263	3565		DCA I WORD2	/CURRENT BLOCK #=STARTING BLOCK-1
14264	1564		TAD I WORD1	
14265	3566		DCA I WORD3	
14266	6212		CIF 10	/READ/WRITE POINTER AT BEGINNING OF BUFFER
14267	4477		JMS I USR	/CALL TO USR
14270	0011		11	/USROUT
14271	4556		JMS I PISWAP	/GET RID OF 17600
14272	4674		JMS I NEXRCK	/DO A NEXREC TO READ IN FIRST FILE BLOCK
14273	5513		JMP I ILOOPL	/DONE
14274	3302	NEXRCK,	NEXREC	

14275 0000 ENTLOK, 0
 14276 3304 ⁰² DCA FNOM /FUNCTION NUMBER IN PLACE
 14277 1174 TAD WORD11 /POINTER TO FILE NAME
 14300 3305 ⁰⁶ DCA STARTB /INTO CALL
 14301 1732 ⁰⁷ TAD I DEVNAL /DEVICE NUMBER
 14302 6212 CIF 10
 14303 4477 JMS I USR /CALL TO USR
 14304 4304 ⁰⁶ FNOM, . /ENTER OR LOOKUP
 14305 4305 ⁰⁶ STARTB, .
 14306 4306 ⁰⁷ FLEN, .
 14307 4516 EN, JMS I ERROR
 14310 1305 TAD STARTB /FILE STARTING BLOCK #
 14311 7640 SZA CLA /IS IT NON-ZERO?
 14312 5321 JMP FILSTU /YES-DEVICE IS FILE STRUCTURED
 14313 1306 TAD FLEN /NO-GET FILE LENGTH
 14314 7640 SZA CLA /IS IT EMPTY?
 14315 5321 JMP FILSTU /NO-DEVICE IS FILE STRUCTURED
 14316 1333 TAD C20 /NO-FILE IS READ/WRITE ONLY
 14317 1563 TAD I WORD0
 14320 3563 DCA I WORD0 /SET READ/WRITE ONLY BIT
 14321 1305 FILSTU, TAD STARTB /GET STARTING BLOCK # OF FILE
 14322 3570 DCA I WORD5 /PUT IN I/O TABLE
 14323 1306 TAD FLEN /FILE LENGTH
 14324 7041 CIA /MAKE FILE LENGTH POSITIVE
 14325 5675 JMP I ENTLOK /RETURN

 14326 1400 K1400, 1400
 14327 1000 K1000, 1000
 14330 0004 CC4, 4
 14331 0003 CC3, 3
 14332 4030 DEVNAL, DEVNA2
 14333 0020 C20, 20

V3
 V5A

/SUBROUTINE P2SWAP=RESTORE OS/8 RESIDENT MONITOR PRIOR TO EXIT FROM INTERPRETER
 /THIS IS DESTRUCTIVE CODE, AND ONCE THIS ROUTINE HAS BEEN EXECUTED
 /THERE IS NO PLACE TO GO BUT OUT,
 /HAS 3 FUNCTIONS:
 / 1) REMOVES CTRL/C HOOKS FROM SYS DRIVER
 / 2) RESTORES BATCH CONTROL WORDS TO 27774-27777
 / 3) IF SYS=TD/8E, RESTORES PAGE 27600 AND RETURNS COFS TO PAGE 07600

USA
 14330 3 14341 actual
 14334 0000 PSWAP2, 0
 14335 1363 TAD K4207K
 14336 3770 DCA I L7600K /REMOVE CTRL/C HOOKS
 14337 1364 TAD K6213K
 14340 3771 DCA I L7605P
 14341 1031 TAD PSFLAG /GET RESIDENT STATUS FLAG
 14342 7710 SPA CLA /IS THIS TD8/E SYS?
 14343 4762 JMS I TDFIXL /YES=RESTORE PAGE 27600 AND PAGE 07600
 14351 TAD 6211 14344 1365 TAD K7577K
 2 DCA 14345 3011 DCA XR1 /POINTER TO 17600
 3 CDF 14346 1366 TAD K7773K
 4 TAD 14347 3012 DCA XR2 /POINTER TO 27774
 5 CDF 14351 3040 TAD MIN4
 6 DCA 14352 6211 DCA TEMP1 /SET COUNTER TO -4
 7 ISC 14353 1411 CDF 10
 80 ISC 14354 6221 TAD I XR1 /FETCH BATCH WORD FROM SAVE AREA
 701 JMP 14355 3412 CDF 20
 302 CDF 14356 2040 DCA I XR2 /RESTORE IN FIELD 2
 13 JMP 14357 5352 ISZ TEMP1 /DONE?
 14361 6201 JMP , -5 /NO
 14362 9734 CDF
 14363 4544 JMP I PSWAP2 /YES=WE ARE FINISHED, SO RETURN
 14364 4207 TDFIXL, PSWAP2
 14365 6213 K4207K, 4207
 14366 7577 K6213K, 6213
 14367 7773 K7577K, 7577
 14370 7600 K7773K, 7773
 14371 7605 MIN4, -4
 L7600K, 7600
 L7605P, 7605

For 8K, Patch

5362

14351 1020 TAD 20 = 6221

2 3355 DCA , +3

Loop 3 6211 CDF 10

4 1767 TAD I (7577)

5 6221 CDF 20

6 3770 DCA I, (7773)

7 2367 ISZ 4367

60 2370 ISZ 4370

1 5353 JMP Loop

2 6201 CDF 0

3 5741 JMP I 4341

4 4544

5

4400

*4400

/NAMEG-ROUTINE TO TRANSLATE SAC INTO A 6 WORD FILE NAME BLOCK, THEN
/PUT THAT NAME BLOCK INTO THE BLOCK SPECIFIED BY THE AC ON ENTRY

14400	7706	MCOLON,	-72	
14401	0014	MCSPE,	14	
14402	4405	N3A,	N3	
14403	0000	N1,	0	/SCRATCH NAME BLOCK
14404	0000	N2,	0	/DEVICE NAME
14405	0000	N3,	0	
14406	0000	N4,	0	/FILE NAME
14407	0000	N5,	0	
14410	0000	N6,	0	/.EXT
14411	0423	DS,	0423	
14412	1300	K0,	1300	
14413	7772	M6,	-6	
14414	0016	CC16,	16	
14415	7774	MMM4,	-4	
<i>14400</i>	<i>0000</i>	NAMEG,	0	
14417	1172	TAD WORD7		/PUT THE NAME IN FILENAME AREA
14420	3042	DCA TEMP3		/SAVE DESTINATION BLOCK ADDRESS
01	14421	1032	TAD STRLEN	
	14422	1214	TAD CC16	/COMPARE STRING LENGTH TO 16
	14423	7710	SPA CLA	
	14424	4516	IF, JMS I ERROR	/TOO MANY CHARS IN DEV:FILENAME
	14425	1032	TAD STRLEN	
	14426	3007	DCA TEMP2	/STRING LENGTH COUNTER
	14427	1111	TAD SACPTR	
	14430	7101	CLL IAC	
	14431	4527	JMS I LDHINL	/INIT LDH TO PULL CHARS FROM SAC
	14432	4557	JMS I LDHRST	
	14433	1202	TAD N3A	
	14434	7100	CLL	
	14435	4526	JMS I STHINL	/INIT STH TO PUT CHARS IN SCRATCH BLOCK
	14436	4560	JMS I STHRST	
	14437	1211	TAD DS	
	14440	3203	DCA N1	
	14441	1212	TAD K0	
	14442	3204	DCA N2	/INITIALIZE DEV TO DSK:
	14443	3205	DCA N3	
	14444	3206	DCA N4	
	14445	3207	DCA N5	
	14446	3210	DCA N6	/ZERO FILE NAME
	14447	3043	DCA TEMP4	/ZERO INTERMEDIATE COUNTER
	14450	4531	JMS I LDH	/GET CHAR FROM SAC
	14451	3040	DCA TEMP1	/SAVE
	14452	1040	TAD TEMP1	
	14453	1200	TAD MCOLON	/IS IT A COLON?
	14454	7450	SNA	
	14455	5306	JMP CAD	/YES-CHARS SO FAR=DEVICE NAME
	14456	1201	TAD MCSPE	/NO-IS IT A PERIOD?
	14457	7650	SNA CLA	
	14460	5325	JMP SSAD	/YES-NEXT TWO CHARS=EXTENSION

14461	1040		TAD TEMP1	/NO-GET CHAR AGAIN
14462	4530		JMS I STH	/STORE IN NAME BLOCK
14463	2043		ISZ TEMP4	/BUMP COUNT FOR CURRENT SECTION
14464	2007	NCGS,	ISZ TEMP2	/END OF STRING YET?
14465	5250		JMP NCG	/NO-NEXT CHAR

VSA

/OS/8 BASIC RUNTIME SYSTEM

PAL8-V7 10/24/72 PAGE 89

14431

14466 1043 TAD TEMP4 /YES-GET CHAR COUNT FOR THIS SECTION (NAME)
 14467 1213 TAD M6
 14470 7740 SMA SZA CLA /IS IT >6?
 14471 5224 JMP IF /YES-TOO MANY CHARACTERS IN FILE NAME
 14472 1342 TAD N1A /NO-ADDRESS OF SCRATCH NAME BLOCK
 14473 3011 DCA XR1
 14474 7040 CMA /-1
 14475 1042 TAD TEMP3 /ADDRESS OF FINAL NAME BLOCK-1
 14476 3012 DCA XR2
 14477 1213 TAD M6 /MOVE 6 WORDS
 14500 3007 DCA TEMP2
 14501 1411 MML, TAD I XR1
 14502 3412 DCA I XR2 /MOVE NAME WORD FROM SCRATCH AREA TO FINAL DEST
 14503 2007 ISZ TEMP2 /DONE?
 14504 5301 JMP MML /NO
 14505 5616 JMP I NAMEG /YES-RETURN

14506 1043 CAD, TAD TEMP4 /GET CHAR COUNT FOR THIS SECTION
 14507 1215 TAD MMM4 /COMPARE AGAINST 4
 14510 7740 SMA SZA CLA /TOO MANY CHARS?
 14511 5224 JMP IF /YES-DEVICE NAME TOO LONG
 14512 1205 TAD N3
 14513 3203 DCA N1
 14514 1206 TAD N4
 14515 3204 DCA N2 /NO-MOVE NEW DEVICE NAME FROM FILE NAME WORDS TO PROPER PLACE
 14516 3205 DCA N3
 14517 3206 DCA N4 /CLEAR FILE NAME
 14520 1202 TAD N3A
 14521 7100 CLL
 14522 4526 JMS I STHINL /AND RE-INIT STH FOR NAME AREA
 14523 3043 DCA TEMP4 /ZERO COUNT
 14524 5264 JMP NCGS

14525 1043 SSAD, TAD TEMP4 /COUNT FOR THIS SECTION (FILE NAME)
 14526 1213 TAD M6
 14527 7740 SMA SZA CLA /TOO MANY?
 14530 5224 JMP IF /YES-FILE NAME TOO LONG
 14531 3043 DCA TEMP4 /NO-CLEAR COUNT
 14532 7332 CLA CLL CML RTR /2 IN AC
 14533 1007 TAD TEMP2 /COMPARE AGAINST # OF CHARS LEFT
 14534 7710 SPA CLA
 14535 5224 JMP IF /TOO MANY CHARS IN EXTENSION
 14536 1343 TAD N6A
 14537 7100 CLL
 14540 4526 JMS I STHINL /INIT STH TO PUT INTO EXTENSION
 14541 5264 JMP NCGS

14542 4402 N1A, N1-1
 14543 4410 N6A, N6

@ 14500 } 0
 14521 }

idove char
 Routine

lot of
 nodes
 through

14460

14464

14477

/SUBROUTINE TO RESTORE PAGE 27600 OF TDB/E DRIVER
/AND READJUST THE CDFS IN FIELD 0

14544
14545
14546
14547
14550
14551
14552
14553
14554
14555
14556
14557
14560
14561
14562
14563
14564
14565
14566
14567
14570
14571

0000 PSWP2P, 0
3031 DCA PSFLAG /CLEAR RESIDENT STATUS FLAG
1362 TAD CDF20
3763 DCA I P2CDFL /PUT CDF 20 IN SWAP ROUTINE
1362 TAD CDF20
3764 DCA I P2CDL1
4556 JMS I P1SWAP /MOVE DOWN PAGE 27600
1365 TAD K6223
3766 DCA I L7642
1367 TAD K6222
3770 DCA I L7721
1367 TAD K6222 /RESTORE CDFS IN PAGE 07600
3771 DCA I L7727
5744 JMP I PSWP2P /RETURN
6221 CDF20, CDF 20
1243 P2CDFL, P2CDF
1247 P2CDL1, P2CDF1
6223 K6223, 6223
7642 L7642, 7642
6222 K6222, 6222
7721 L7721, 7721
7727 L7727, 7727

14544
14545
14546
14547
14550
14551
14552
14553
14554
14555
14556
14557
14560
14561
14562
14563
14564
14565
14566
14567
14570
14571

0000 PSWP2P, 0
3031 DCA PSFLAG /CLEAR RESIDENT STATUS FLAG
1362 TAD CDF20
3763 DCA I P2CDFL /PUT CDF 20 IN SWAP ROUTINE
1362 TAD CDF20
3764 DCA I P2CDL1
4556 JMS I P1SWAP /MOVE DOWN PAGE 27600
1365 TAD K6223
3766 DCA I L7642
1367 TAD K6222
3770 DCA I L7721
1367 TAD K6222 /RESTORE CDFS IN PAGE 07600
3771 DCA I L7727
5744 JMP I PSWP2P /RETURN
6221 CDF20, CDF 20
1243 P2CDFL, P2CDF
1247 P2CDL1, P2CDF1
6223 K6223, 6223
7642 L7642, 7642
6222 K6222, 6222
7721 L7721, 7721
7727 L7727, 7727

IFNZRO EAE <
 NOPUNCH
 >
 IFZERO EAE <

4600

*4600

/FLOATING OUTPUT ROUTINE

```

04600 0000 FFOUT, 0
04601 7344 CLA CLL CMA RAL /MAKE A MINUS TWO
04602 3736 DCA I FFNGP /AND STORE IN SIGN WORD
04603 3337 DCA KNT /CLEAR COUNT WORD
04604 1056 TAD EPLG /IS THIS E FORMAT?
04605 7640 SZA CLA
04606 5213 JMP FFMT /NO=F FORMAT
04607 1340 TAD K6 /YES=GET A 6
04610 3060 DCA DADP /STORE AS # OF DIGITS AFT DEC PT
04611 1321 TAD K16 /SET FIELD WIDTH TO 14 ( DECIMAL)
04612 3057 DCA FLOW
04613 6201 FFMT, CDF /DF TO PACKAGE FIELD
04614 1332 TAD KM7 /SET # OF SIGNF. DIGITS
04615 3726 DCA I DCNTP /TO 6 (DON'T PRINT 7TH)
04616 1045 TAD ACH /DETERMINE IF #=0
04617 7450 SNA
04620 5274 JMP FOUT3 /YES=SKIP DOWN
04621 7700 SMA CLA /NO=IS IT NEGATIVE?
04622 5225 JMP .+3 /POSITIVE
04623 4736 JMS I FFNGP /NEGATE #
04624 3736 DCA I FFNGP /NEGATIVE=SET FLAG
04625 1044 FOUT1, TAD ACX /GET # INTO RANGE .1<=N<1
04626 7740 SMA SZA CLA /IS EXP. NEG.?
04627 5234 JMP FOUT2 /NO=GO ON
04630 4735 JMS I FFMPP /YES=MAKE # GREATER THAN 1
04631 5317 TEN /BY MULTIPLYING BY TEN (DEC.)
04632 2337 ISZ KNT /COUNT THE MULTIPLIES
04633 5225 JMP FOUT1 /SEE IF >1 YET
04634 4306 FOUT2, JMS SE /# IS >1-MAKE IT LESS THAN 1
04635 4535 JMS I FFPUP /STORE IN A TEMPORARY
04636 5155 TM3
04637 3044 DCA ACX /SET FAC TO .5
04640 7132 CLL CML RTR
04641 3045 DCA ACH
04642 3046 DCA ACLO
04643 1056 TAD EPLG /IS THIS E FORMAT?
04644 7640 SZA CLA
04645 1337 TAD KNT /NO=GET COUNT OF MULTIPLIES
04646 7041 CMA IAC /NEGATE IT
04647 1060 TAD DADP /AND ADD # OF DIGITS AFT. DC. PT.
04650 7500 SMA /MUST BE NEGATIVE
04651 7040 CMA
04652 1330 TAD KK7 /LIMIT # OF DIVS TO 7
04653 7510 SPA
04654 7200 CLA

```

04655	1332	TAD	KM7	/RESTORE
04656	3306	DCA	SE	/STORE AS COUNTER
04657	5262	JMP	.+3	


```

04660 4734      JMS I   FFDVP  /DIVIDE .5 BY TEN THAT # OF TIMES
04661 5317      TEN
04662 2306      ISZ     SE      /DONE?
04663 5260      JMP     .-3     /NO-GO ON
04664 4733      JMS I   FFA DP  /YES-ADD IN ORIG.#-THIS IS ROUNDING
04665 5155      TM3
04666 4306      JMS     SE      /INSURE THAT IT IS IN RANGE
04667 1044      TAD     ACX     /SHIFT MANTISSA ACCORDING TO EXP
                                /0=1 LEFT; 1=NO SHIFT;2=1 RIGHT,...
04670 7041      CMA     IAC
04671 4704      JMS I   ACSRPT  /SHIFT RIGHT (ACX+1) PLACES
04672 4705      JMS I   AL1PT  /SHIFT LEFT 2 TO CORRECT
04673 4705      JMS I   AL1PT  /WE ARE LOSING BITS!!
04674 1337      TAD     KNT     /DONE-GET COUNT OF MULS.
                                /PRESERVE IT
04675 3047      DCA     OPX
04676 1056      TAD     EFLG    /IS THIS E FORMAT OUT?
04677 7640      SZA     CLA
04700 5343      JMP     NOTE    /NO
04701 3337      DCA     KNT     /YES-ZERO COUNT
04702 1332      TAD     KM7     /GET MINUS 7-FOR 2 SIGNS,PT,+EXP
04703 5347      JMP     ADFW    /GO ADD FIELD WIDTH
04704 6072      ACSRPT, ACSR
04705 6057      AL1PT, AL1
/
/ROUTINE TO GET FAC<1
/
04706 0000      SE,      0
04707 1044      SE1,     TAD     ACX
04710 7750      SPA SNA CLA    /#>1?
04711 5706      JMP I   SE      /NO-RETN.
04712 4734      JMS I   FFDVP  /YES-DIV. BY TEN
04713 5317      TEN
04714 7040      CMA
04715 1337      TAD     KNT     /REDUCE KNT BY 1
04716 3337      DCA     KNT
04717 5307      JMP     SE1

```

/CONSTANTS AND POINTERS

```

04720 5150 OUTDGP, OUTDG
04721 0016 K16, 16
04722 5600 FLINK, JMP I FFOUT
04723 5160 PRNTP, PRNTX
04724 5172 PRZROP, PRZRO
04725 5076 DGTYP, DGTYP
04726 5137 DCNTP, DCNT
04727 7777 M1, 7777
04730 0007 KK7, 7
04731 7760 KM20, -20
04732 7771 KM7, -7
04733 6000 FFADP, FFADD
04734 5722 FFDVP, FFDIV
04735 0135 FFPUTP=FFPUTL
04735 5600 FFMPP, FFMPPY
04736 6135 FFNGP, FFNEG
04737 0000 KNT, 0
04740 0006 K6, 6

```

/CONTINUATION OF OUTPUT MAINLINE

```

4743 *4743
04743 1337 NOTE, TAD KNT /GET COUNT OF MULTIPLIES
04744 7500 SMA /IF NOT NEG-MAKE = -2
04745 7240 CLA CMA
04746 1327 TAD M1 /MINUS 1 FOR DEC.PT
04747 1057 ADFW, TAD FLDW /GET THE FIELD WIDTH
04750 7041 CMA IAC /NEGATE IT
04751 3734 DCA I FFDVP /STORE WHILE WE CHECK DADP
04752 1060 TAD DADP /GET DIGITS AFTER DEC. PT
04753 7450 SNA /DID HE SAY NO DEC. PLACES?
04754 7040 CMA /YES-TAKE AWAY 1 SINCE NO DEC. PT.
04755 1734 TAD I FFDVP /ADD IN REST
04756 7500 SMA /NEG?
04757 5723 JMP I PRNTP /NO-PRINT XS-NOT ENUFF ROOM
04760 3306 DCA SE /STORE AS CNT OF SPACES
04761 5364 JMP ,+3
04762 1331 TAD KM20
04763 4720 JMS I OUTDGP /PRINT A SPACE
04764 2306 ISZ SE /DONE?
04765 5362 JMP ,+3 /NO-GO ON
04766 7346 CLA CLL CMA RTL /MAKE A MINUS 3
04767 1736 TAD I FFNGP /YES-GET SIGN(=-2 OR 0)
04770 4720 JMS I OUTDGP /FOR PLUS OR MINUS-PRINT SIGN
04771 1337 TAD KNT /GET MUL COUNT
04772 7500 SMA
04773 5724 JMP I PRZROP /PRINT LEADING ZERO
04774 7041 CMA IAC
04775 4725 JMS I DGTYP /OUTPUT 'KNT' DIGITS
04776 1060 PROCP, TAD DADP /CHECK DADP FOR 0
04777 7650 SNA CLA /DON'T PRINT '.' IF DADP=0

```

```

/*****

```

```

/FALL THROUGH PAGE BOUNDARY!!!

```

```

/'SNA CLA' MUST BE LAST LOC. ON PAGE!!!

```

```

/((CURSE YOU B.C.))

```

```

/*****

```

```

      5000      PAGE
/*****FALL THROUGH PAGE BOUNDARY TO HERE*****/
05000 5203      JMP      GKNT      /MUST BE FIRST LOC. OF PAGE!!*****/
05001 7344      PDP,      CLA CLL CMA RAL
05002 4350      JMS      OUTDG      /PRINT DEC. PT.
05003 1754      GKNT,     TAD I      KNTP      /GET COUNT AGAIN
05004 7750      SPA SNA CLA
05005 5225      JMP      GD
05006 1754      TAD I      KNTP      /GET COUNT
05007 7040      CMA              /NEGATE
05010 3276      DCA      DGTYP      /STORE AS COUNTER
05011 1060      TAD      DADP
05012 7040      CMA              /SAME FOR DADP
05013 3357      DCA      SEP
05014 5216      JMP      PR      /GO ON
05015 4350      PZR,      JMS      OUTDG      /PRINT A ZERO
05016 2276      PR,      ISZ      DGTYP
05017 7410      SKP
05020 5223      JMP      PS
05021 2357      ISZ      SEP
05022 5215      JMP      PZR
05023 1754      PS,      TAD I      KNTP
05024 7041      CMA      IAC
05025 1060      GD,      TAD      DADP
05026 7540      SMA      SZA
05027 4276      JMS      DGTYP
05030 7200      CLA
05031 1056      TAD      EFLG
05032 7640      SZA      CLA
05033 5266      JMP      DONEF      /DONE
05034 4344      JMS      OUT
05035 0305      305          /PRINT 'E'
05036 1047      TAD      OPX      /GET PRESERVED COUNT OF MULS
05037 7740      SMA SZA CLA      /DETERMINE SIGN
05040 7326      CLA CLL CML RTL /MAKE A 2
05041 4344      JMS      OUT
05042 0253      253          /PRINT MINUS OR PLUS SIGN
05043 1334      TAD      KM144      /SET TO DIV BY 100
05044 3050      DCA      OPH
05045 7344      CLA CLL CMA RAL /SET LOOP COUNTER
05046 3276      DCA      DGTYP
05047 1047      TAD      OPX      /GET THE COUNT
05050 7510      SPA
05051 7041      CMA      IAC      /NEGATE IF NEGATIVE
05052 3046      LOOP,     DCA      ACLO      /STORE FOR DIV. ROUTINE
05053 3045      DCA      ACH      /HI ORD. MUST BE ZERO
05054 7100      CLL              /PREVENT DIVIDE OVERFLOW!!
05055 4736      JMS I      DV24PT      /DIVIDE BY 100
05056 1046      TAD      ACLO      /GET THE QUOTIENT
05057 4350      JMS      OUTDG      /OUTPUT HUNDREDS PLACE
05060 1335      TAD      KM12      /NOW DIV. BY 10
05061 3050      DCA      OPH
05062 1045      TAD      ACH      /DIV. REM. BY 10
05063 2276      ISZ      DGTYP      /DONE?
05064 5252      JMP      LOOP      /NO-GO DO CALCULATE , PRINT TENS PLACE

```

/08/8 BASIC RUNTIME SYSTEM

PAL8-V7 10/24/72 PAGE 95-1

05069 4350

JMS

OUTDG /YES-REM(ONES PLACE)IS IN AC-PRINTIT

```

05066 1055 DONEF, TAD SWIT2 /SHOULD WE PRINT CR/LF?
05067 7650 SNA CLA
05070 5742 JMP I FLING /NO
05071 4344 JMS OUT
05072 0215 215
05073 4344 JMS OUT
05074 0212 212
05075 5742 JMP I FLING

```

```

/
/OUTPUT DIGITS OF FAC BY MULTIPLYING BY TEN
/THE HIGH ORDER OVERFLOW IS THE DIGIT

```

```

05076 0000 DGTYP, 0
05077 7041 CMA IAC
05100 3357 DCA SEP /STORE COUNT PASSED
05101 1045 DT1, TAD ACH /GET FAC AND STORE FOR LATER
05102 3050 DCA OPH
05103 1046 TAD ACLO
05104 3051 DCA OPL
05105 4740 JMS I AL1PP /SHIFT FAC LEFT 1 = FAC*2
05106 7004 RAL /OVERFLOW TO TM3
05107 3355 DCA TM3
05110 4740 JMS I AL1PP /SHIFT LEFT AGAIN = FAC*4
05111 1355 TAD TM3 /SHIFT OUT OVERFLOW
05112 7004 RAL
05113 3355 DCA TM3
05114 3042 DCA AC2 /MUST BE 0 FOR OADD
05115 4741 JMS I OADDP /ADD ORIG FAC = FAC*5
05116 7004 RAL /ADD OVERFLOW TO TM3
05117 1355 TAD TM3
05120 3355 DCA TM3
05121 4740 JMS I AL1PP /SHIFT FAC 1 LEFT = FAC*10!!
05122 1355 TAD TM3 /OVERFLOW IN TM3 IS FIRST DIGIT
05123 7004 RAL
05124 2337 ISZ DCNT /DONE ALL SIGNIF. DIGS.?
05125 5330 JMP ,+3 /NO-GO ON
05126 7240 CLA CMA /YES=PRINT ZEROS
05127 3337 DCA DCNT /FROM NOW ON
05130 4350 JMS OUTDG /PRINT DIGIT (HI ORD. OVRFLOW)
05131 2357 ISZ SEP /DONE REQUIRED?
05132 5301 JMP DT1 /NOPE
05133 5676 JMP I DGTYP /YUP
05134 7634 KM144, -144
05135 7766 KM12, -12
05136 5745 DV24PT, DV24
05137 0000 DCNT, 0 /COUNT OF SIGNF. DIGITS
05140 6057 AL1PP, AL1
05141 6157 OADDP, OADD
05142 4722 FLING, FLINK

```

```
05143 4776 PROCPP, PROCP
/
/OUTPUT ROUTINE
/
05144 0000 OUT, 0
05145 1744 TAD I OUT /GET THE CHAR
05146 3413 DCA I XR3 /STORE CHAR IN INTERMEDIATE BUFFER
05147 5744 JMP I OUT

/
/OUTPUT DIGIT
/
05150 0000 OUTDG, 0
05151 4344 JMS OUT
05152 0260 260
05153 5750 JMP I OUTDG /RETN

05154 4737 KNTP, KNT
05155 0000 TM3, 0
05156 0000 0
05157 0000 SEP, 0
05160 7200 PRNTX, CLA
05161 1057 TAD FLDW /GET FIELD WIDTH
05162 7040 CMA /MUST BE NEGATIVE
05163 3357 DCA SEP /USE AS COUNTER
05164 2357 PRNTX1, ISZ SEP /DONE ALL?
05165 7410 SKP /NO-GO ON
05166 5266 JMP DONEF /YES-RETN.
05167 4344 JMS OUT /PRINT ASTERISK
05170 0252 252 /ASTERISK
05171 5364 JMP PRNTX1

/
/PRINT A LEADING ZERO
/
05172 7200 PRZRU, CLA
05173 4350 JMS OUTDG
05174 5743 JMP I PROCPP
```

/FLOATING POINT INPUT ROUTINE

```

5200      PAGE
05200 0000 FFIN, 0
05201 7240 CLA CMA
05202 3711 DCA I FDVPT /INITIALIZE PERIOD SWITCH TO -1
05203 7040 CMA /SET SIGN SWITCH TO -1
05204 3300 DCA SIGNF
05205 6201 COP /OF TO PACKAGE FIELD
05206 3052 DCA DSWIT /ZERO CONVERSION SWITCH
05207 3044 DECONV, DCA ACX /ZERO OUT THE FACI
05210 3046 DCA ACLO
05211 0200 P200, 200
05212 3045 DCA ACH
05213 3305 DECNV, DCA DNUMBR /ZERO # OF DIGITS SINCE DEC. PT.
05214 4322 DECON, JMS GCHR /GET A CHAR. FROM TTY.
05215 5232 JMP FFIN1 /TERMINATOR-
05216 2092 ISZ DSWIT /DIGIT-BUMP CONVERSION SWITCH
05217 2305 ISZ DNUMBR /BUMP # OF DIGITS-# IS STORED IN
05220 4710 JMS I FMPYLL /"FMPY TEN"
05221 5317 TEN
05222 4535 JMS I FPUTL /"FPUT I TM3PT"
05223 1175 FPPYM1
05224 4534 JMS I FGETL /"FGET TP"
05225 5314 TP
05226 4536 JMS I FNORL /"FNOR"
05227 4712 JMS I FADOLL /"FADD I TM3PT"
05230 1175 FPPYM1
05231 5214 JMP DECON /GO ON
05232 2711 FFIN1, ISZ I FDVPT /HAVE WE HAD A PERIOD YET?
05233 5241 JMP FIGO2 /YES-GO ON
05234 2315 ISZ TP1 /NO-IS THIS A PERIOD?
05235 2315 ISZ TP1
05236 7610 SKP CLA
05237 5213 JMP DECNV /YES-ZERO DIG. COUNT AFTER DEC. PT.
/AND GO CONVERT REST
05240 3305 DCA DNUMBR /NO-TERMINATOR-ZERO COUNT OF
/DIGITS AFTER DECIMAL POINT.
05241 2300 FIGO2, ISZ SIGNF /IS # NEGATIVE?(DID WE GET - SIGN?)
05242 4703 JMS I FFNEGP /YES-NEGATE IT
05243 7240 CLA CMA /RESET SIGN SWITCH FOR EXP.
05244 3300 DCA SIGNF
05245 1053 TAD CHAR /NO-WAS THE TERMINATOR AN 'E'?
05246 1306 TAD KME
05247 7650 SNA CLA
05250 4322 GETE, JMS GCHR /YES-GET A CHAR. OF EXPONENT
05251 5260 JMP EDON /END OF EXPONENT
05252 1043 TAD TM /GOT DIG. OF EXP-STORED IN TP1
05253 7106 CLL RTL /MULT. ACCUMULATED EXP BY 10
05254 1043 TAD TM
05255 7104 CLL RAL
05256 1315 TAD TP1 /ADD DIGIT
05257 5250 JMP GETE /CONTINUE

```

```

05260 1043 EDON, TAD TM /GET EXPONENT
05261 2300 ISZ SIGNF /WAS EXPONENT NEGATIVE?
05262 7041 CMA IAC /YES-NEGATE IT
05263 7041 CMA IAC /AND CALC. DNUMBR - EXPON.
05264 1305 TAD DNUMBR /GET # TIMES TO DIV MANTISSA BY TEN
05265 7141 CLL CMA IAC
05266 7510 SPA /RESULT POSITIVE?
05267 7161 CLL CMA CML IAC /NO-MAKE POS. AND SET LINK
05270 7040 CMA /NEGATE FOR COUNTER
05271 3305 DCA DNUMBR /AND STORE
05272 7004 RAL /LINK=1-DIV;=0-MUL. # BY TEN
05273 1307 TAD MDV /FORM CORRECT INSTRUCTION
05274 3300 DCA SIGNF /AND STORE FOR EXECUTION
05275 2305 FCNT, ISZ DNUMBR /DONE ALL OPERATIONS?
05276 5300 JMP SIGNF /NO
05277 5600 JMP I FFIN /YES-RETURN
05300 0000 SIGNF, 0 /NO- MUL OR DIV. MANTISSA
05301 5317 TEN /BY TEN
05302 5275 JMP FCNT /GO ON
05303 6135 FFNEGP, FFNEG
05304 5155 TM3PT, TM3
05305 0000 DNUMBR, 0
05306 7473 KME, -305
05307 4710 MDV, JMS I .+1 /THESE 3 WDS. MUST BE IN THIS ORDER
05310 5600 FMPYLL, FFMPY
05311 5722 FDVPT, FFDIV
05312 6000 FADDLL, FFAOD /!!!!!!!!!!!!!!!!!!!!

05313 0012 KK12, 12
05314 0013 TP, 13
05315 0000 TP1, 0
05316 0000 0
05317 0004 TEN, 4
05320 2400 2400
05321 0000 0

```


/ROUTINE TO GET A CHAR FROM THE TTY AND SEE IF IT IS DIGIT
 /OR A TERMINATOR.
 /RETURN TO CALL + 1 IF TERMINATOR, TO CALL + 2 IF DIGIT
 /THIS ROUTINE MUST NOT MODIFY THE MQ!!

```

05322 0000 GCHR, 0
05323 3043 DCA TM /STORE ACCUMULATED EXPONENT (MAYBE)
05324 4347 JMS INPUT /GET A CHAR FROM TTY.
05325 1053 TAD CHAR /PICK IT UP
05326 1362 TAD PLUS /WAS IT PLUS SIGN?
05327 7450 SNA
05330 5335 JMP DECON1 /YES-GET ANOTHER CHAR.
05331 1363 TAD MINUS /NO WAS IT MINUS SIGN?
05332 7640 SZA CLA
05333 5336 JMP .+3
05334 3300 DCA SIGNF /YES-FLIP SWITCH
05335 4347 DECON1, JMS INPUT /GET A CHAR.
05336 1053 TAD CHAR
05337 1346 TAD K7506 /SEE IF ITS A DIGIT
05340 7100 CLL
05341 1313 TAD KK12
05342 3315 DCA TP1 /STORE FOR LATER
05343 7430 SZL /DIGIT?
05344 2322 ISZ GCHR /YES-RETN. TO CALL+2
05345 5722 JMP I GCHR /NO-RETN. TO CALL+1
05346 7506 K7506, 7506

```

/INPUT ROUTINE-IGNORES LEADING SPACES

```

05347 0000 INPUT, 0
05350 4542 JMS I GETCHL /USE OUR ROUTINE TO GET CHAR
05351 1052 TAD DSWIT /GET TERMINATOR
05352 7640 SZA CLA /VALID INPUT YET?
05353 5360 JMP IOUT /YES-CONTINUE
05354 1053 TAD CHAR /NO-GET CHAR
05355 1361 TAD M240 /COMPARE AGAINST SPACE
05356 7650 SNA CLA /IS IT A SPACE?
05357 5350 JMP INPUT+1 /YES-IGNORE IT
05360 5747 IOUT, JMP I INPUT /RETURN
05361 7540 M240, -240
05362 7525 PLUS, -253
05363 7776 MINUS, 253-255

```

/ROUTINE TO DECIDE CALLING MODE IN LIEU OF "SPECIAL MODE" PROBLEMS

```

05364 0000 PATCHF, 0
05365 7440 SZA /IS AC EMPTY
05366 5371 JMP RTN2 /NO-THIS IS ALWAYS SI MODE WITH ADDR IN AC
05367 1037 TAD FF /YES-GET SPECIAL MODE FLIP-FLOP
05370 7640 SZA CLA /IF ON, THE ZERO AC MEANS ADDRESS OF 0
05371 2364 RTN2, ISZ PATCHF /USE AC AS ADDRESS OF OPERAND
05372 5764 JMP I PATCHF /RETURN

```

5400

PAGE

/INVERSE FLOATING SUBTRACT-USES FLOATING ADD
/IIFSW111-THIS IS OP-FAC

```

05400 0000 FFSUB1, 0
05401 4555 JMS I PATCHP /WHICH MODE?
05402 1600 TAD I FFSUB1 /CALLED BY USER-GET ADDR. OF OP.
05403 4644 JMS I ARGETL /GO PICK UP OPERAND
05404 6201 CDF
05405 4610 JMS I FPNEGA /NEGATE FAC
05406 1200 TAD FFSUB1 /AND GO ADD
05407 5611 JMP I SUBOP
05410 6135 FPNEGA, FPNEG
05411 6125 SUBOP, SUBO

```

/INVERSE FLOATING DIVIDE
/FSWITCH=1
/THIS IS OP/FAC

```

05412 0000 FFDIV1, 0
05413 4555 JMS I PATCHP /WHICH MODE OF CALL?
05414 1612 TAD I FFDIV1 /CALLED BY USER-GET ADDR.
05415 4644 JMS I ARGETL /PICK UP OPERAND
05416 1046 TAD ACLO /SWAP THE FAC AND OPERAND
05417 3051 DCA OPL /THERE IS A POINTER TO OPL
05420 1442 TAD I AC2 /IN AC2 LEFT FROM ARGET SUBR.
05421 3046 DCA ACLO
05422 1044 TAD ACX /MIGHT AS WELL SUBTRACT THE
05423 7141 CLL CMA IAC /EXPONENTS HERE (SAVES A WORD)
05424 1047 TAD OPX /THEN ZERO OPX SO WILL NOT
05425 3044 DCA ACX /MESS UP WHEN ITS DONE AGAIN
05426 3047 DCA OPX /LATER (SEE DIV. ROUTINE)
05427 1045 TAD ACH
05430 3042 DCA AC2 /NOW SWAP HIGH ORDER MANTISSAS
05431 1050 TAD OPH
05432 3045 DCA ACH
05433 1042 TAD AC2
05434 3050 DCA OPH
05435 6201 CDF /DF TO PACKAGE FIELD
05436 1212 TAD FFDIV1 /NOW KLUDGE UP SUBROUTINE LINKAGE
05437 3646 DCA I FFDP
05440 1247 TAD KFD1
05441 3645 DCA I MDSETP
05442 5643 JMP I MD1P /GO SET UP AND DIVIDE

```

```

05443 5452 MD1P, MD1
05444 6200 ARGETL, ARGET
05445 5450 MDSETP, MDSET
05446 5722 FFDP, FFDIV
05447 5726 KFD1, FFD1

```

/MDSET-SETS UP SIGNS FOR MULTIPLY AND DIVIDE
 /ALSO SHIFTS OPERAND ONE BIT TO THE LEFT.
 /EXIT WITH EXPONENT OF OPERAND IN AC FOR EXPONENT
 /CALCULATION-CALLED WITH ADDRESS OF OPERAND IN AC AND
 /DATA FIELD SET PROPERLY FOR OPERAND.

```

05450 0000 MDSET, 0
05451 4703 JMS I ARGETK /GET ARGUMENT
05452 6201 MD1, CDF /OF TO PACKAGE FIELD
05453 7344 CLA CLL CMA RAL /SET SIGN CHECK TO -2
05454 3043 DCA TM
05455 1050 TAD OPH /IS OPERAND NEGATIVE?
05456 7700 SMA CLA
05457 5262 JMP .+3 /NO
05460 4702 JMS I OPNEGP /YES-NEGATE IT
05461 2043 ISZ TM /BUMP SIGN CHECK
05462 1051 TAD OPL /AND SHIFT OPERAND LEFT ONE BIT
05463 7104 CLL RAL
05464 3051 DCA OPL
05465 1050 TAD OPH
05466 7004 RAL
05467 3050 DCA OPH
05470 3041 DCA AC1 /CLR. OVERFLOW WORK OF FAC
05471 1045 TAD ACH /IS FAC NEGATIVE
05472 7700 SMA CLA
05473 5277 JMP LEV /NO-GO ON
05474 4701 JMS I FFNEGK /YES-NEGATE IT
05475 2043 ISZ TM /BUMP SIGN CHECK
05476 7000 NOP /MAY SKIP
05477 1047 LEV, TAD OPX /EXIT WITH OPERAND EXPONENT IN AC
05500 5650 JMP I MDSET

05501 6135 FFNEGK, FFNEG
05502 6146 OPNEGP, OPNEG
05503 6200 ARGETK, ARGET

```

/CONTINUATION OF FLOATING DIVIDE ROUTINE

```

05504 1042 FD1, TAD AC2 /NEGATE HI ORDER PRODUCT
05505 7141 CLL CMA IAC
05506 1045 TAD ACH /COMPARE WITH REMAINDER OF FIRST DIV.
05507 7420 SNL /WELL?
05510 5733 JMP I DVOPSP /GREATER THAN REM.-ADJUST QUOT OF 1ST DIV.
05511 7100 CLL /OK-DO (REM-(Q*OPL))/OPH
05512 3045 DCA ACH /FIRST STORE ADJUSTED PRODUCT
05513 4732 JMS I DV24P /DIVIDE BY OPH (HI ORDER OPERAND)
05514 1041 DVL1, TAD AC1 /GET QUOT. OF FIRST DIV.
05515 7500 SMA /IF HI ORDER BIT SET-MUST SHIFT 1 RIGHT
05516 5327 JMP FD /NO-ITS NORMALIZED-DONE
05517 7110 CLL RAR /MUST SHIFT RIGHT 1
05520 3045 DCA ACH /STORE IN FAC

```

```
05521 1046      TAD      ACLO      /SHIFT LOW ORDER RIGHT
05522 7010      RAR
05523 3046      DCA      ACLO      /STORE BACK
05524 2044      ISZ      ACX      /BUMP EXPONENT
05525 7000      NOP
05526 1045      TAD      ACH
05527 3045      FD,     DCA      ACH      /STORE HIGH ORDER RESULT
05530 5731      JMP I    FDDONP     /GO LEAVE DIVIDE

05531 5742      FDDONP, FDDON      /END OF FLTG, DIV. ROUTINE
05532 5745      DV24P,  DV24      /ROUTINE TO DO A 24X12BIT DIVIDE
05533 6315      DVOPSP, DVOP8     /ROUTINE TO ADJUST QUOT OF FIRST DIV.
/
/CONTINUATION OF ROUTINE TO ADJUST QUOT. OF FIRST DIV.
/DBAD1 IS ONLY EXECUTED ON DIVIDE OVERFLOW-OTHERWISE THE
/ROUTINE STARTS AT DVOP2
/
05534 3044      DBAD1,  DCA      ACX      /DIVIDE OVERFLO-ZERO ALL
05535 7450      DVOP2,  SNA      /IS IT ZERO?
05536 3046      DCA      ACLO      /YES-MAKE WHOLE THING ZERO
05537 3045      DCA      ACH
05540 4732      JMS I    DV24P     /DIVIDE EXTENDED REM. BY HI DIVISOR
05541 1046      TAD      ACLO      /NEGATE THE RESULT
05542 7141      CLL CMA IAC
05543 3046      DCA      ACLO
05544 7420      SNL      /IF QUOT. IS NON-ZERO, SUBTRACT
05545 7040      CMA      /ONE FROM HIGH ORDER QUOT.
05546 5314      JMP      DVL1     /GO TO IT
```

/ROUTINE TO BUMP CHARACTER NUMBER
/USED BY CHAR PACKING ROUTINES,BUT ITS HERE AS A PATCH

```
5573      *5573
05573 0000      CNOBML, 0
05574 1563      TAD I WORD0      /HEADER WORD
05575 1076      TAD K0100      /ADD 1 TO THE COUNT BITS
05576 3563      DCA I WORD0
05577 5773      JMP I CNOBML      /DONE
```

5600 PAGE
/FLOATING MULTIPLY-DOES 2 24X12 BIT MULTIPLIES

05600	0000	FFMPY, 0		
05601	4555	23 JMS I	PATCHP	/WHICH MODE OF CALL?
05602	1600	TAD I	FFMPY	/CALLED BY USER-GET OPERAND ADDR.
05603	4774	JMS I	MDSETK	/SET UP FOR MPY-OPX IN AC ON RETN.
05604	1044	TAD	ACX	/DO EXPONENT ADDITION
05605	3044	DCA	ACX	/STORE FINAL EXPONENT
05606	3345	DCA	DV24	/ZERO TEM STORAGE FOR MPY ROUTINE
05607	3042	DCA	AC2	
05610	1045	TAD	ACH	/IS FAC=0?
05611	7650	SNA	CLA	
05612	3044	DCA	ACX	/YES-ZERO EXPONENT
05613	4243	JMS	MP24	/NO-MULTIPLY FAC BY LOW ORDER OPR.
05614	1050	TAD	OPH	/NOW MULTIPLY FAC BY HI ORDER MULTIPLIER
05615	3051	DCA	OPL	
05616	4243	JMS	MP24	
05617	1042	TAD	AC2	/STORE RESULT BACK IN FAC
05620	3046	RTZRO, DCA	ACLO	/LOW ORDER
05621	1345	TAD	DV24	/HIGH ORDER
05622	3045	DCA	ACH	
05623	1045	TAD	ACH	/DO WE NEED TO NORMALIZE?
05624	7004	RAL		
05625	7700	SMA	CLA	
05626	5235	JMP	SHLFT	/YES-DO IT FAST
05627	3041	MOONE, DCA	AC1	/NO-ZERO OVERFLOW WD(DO I NEED THIS???)
05630	2200	ISZ	FFMPY	/BUMP RETURN POINTER
05631	2043	ISZ	TM	/SHOULD RESULT BE NEGATIVE?
05632	5600	JMP I	FFMPY	/NOPE-RETN.
05633	4773	JMS I	FFNEGR	/YES-NEGATE IT
05634	5600	JMP I	FFMPY	/RETURN
05635	7040	SHLFT, CMA		/SUBTRACT 1 FROM EXP.
05636	1044	TAD	ACX	
05637	3044	DCA	ACX	
05640	4642	JMS I	AL1PTR	/SHIFT FAC LEFT 1 BIT
05641	5230	JMP	MOONE+1	/DONE.
05642	6057	AL1PTR, AL1		

/24 BIT BY 12 BIT MULTIPLY. MULTIPLIER IS IN OPL
/MULTIPLICAND IS IN ACH AND ACLO
/RESULT LEFT IN DV24, AC2, AND AC1

05643	0000	MP24, 0		
05644	1375	TAD	KKM12	/SET UP 12 BIT COUNTER
05645	3047	DCA	OPX	
05646	1051	TAD	OPL	/IS MULTIPLIER=0?
05647	7440	SZA		
05650	5254	JMP	MPLP1	/NO-GO ON
05651	3041	DCA	AC1	/YES-INSURE RESULT=0
05652	5643	JMP I	MP24	/RETURN
05653	1051	MPLP, TAD	OPL	/SHIFT A BIT OUT OF LOW ORDER
05654	7010	MPLP1, RAR		/OF MULTIPLIER AND INTO LINK
05655	3051	DCA	OPL	
05656	7420	SNL		/WAS IT A 1?
05657	5266	JMP	MPLP2	/NO-0-JUST SHIFT PARTIAL PRODUCT

EAE

need by Divide Rtn

overlays

by
EAE
Routine

V3
V4
V5A

05660	7100	CLL		/YES-ADD MULTIPLICAND TO PARTIAL PRODUCT
05661	1042	TAD	AC2	
05662	1046	TAD	ACLO	/LOW ORDER
05663	3042	DCA	AC2	
05664	7004	RAL		/PROPAGATE CARRY
05665	1045	TAD	ACH	/HI ORDER
05666	1345	MPLP2, TAD	DV24	
05667	7010	RAR		/NOW SHIFT PARTIAL PROD. RIGHT 1 BIT
05670	3345	DCA	DV24	
05671	1042	TAD	AC2	
05672	7010	RAR		
05673	3042	DCA	AC2	
05674	7010	RAR		/1 BIT OF OVERFLOW TO AC1
05675	3041	DCA	AC1	
05676	2047	ISZ	OPX	/DONE ALL 12 MULTIPLIER BITS?
05677	5253	JMP	MPLP	/NO-GO ON
05700	5643	JMP I	MP24	/YES-RETURN

V3
V4
V5A
FREE

05701	3051	/PART OF DIVIDE ROUTINE-FFDIV MUST BE AT LOC. 5722		
05702	1042	MP12L, DCA	OPL	/STORE BACK MULTIPLIER
05703	7420	TAD	AC2	/GET PRODUCT SO FAR
05704	5307	SNL		/WAS MULTIPLIER BIT A 1?
05705	7100	JMP	.+3	/NO-JUST SHIFT THE PARTIAL PRODUCT
05706	1046	CLL		/YES-CLEAR LINK AND ADD MULTIPLICAND
05707	7010	TAD	ACLO	/TO PARTIAL PRODUCT
05710	3042	RAR		/SHIFT PARTIAL PRODUCT-THIS IS HI ORDER
05711	1051	DCA	AC2	/RESULT-STORE BACK
05712	7010	MPLP1, TAD	OPL	/SHIFT A BIT OUT OF MULTIPLIER
05713	2200	RAR		/AND A BIT OR RESULT INTO IT (LO ORDER PROD.)
05714	5301	ISZ	FFMPY	/DONE ALL BITS?
05715	7141	JMP	MP12L	/NO-LOOP BACK
05716	3046	CLL CMA	IAC	/YES-LOW ORDER PROD. OF QUOT. X OPL IN AC
05717	7024	DCA	ACLO	/NEGATE AND STORE
05720	5721	CML	RAL	/PROPAGATE CARRY
05721	5504	JMP I	FD1P	/GO ON
		FD1P, FD1		/POINTER TO REST OF DIVIDE ROUTINE

AC2 contains H!
order Product

EAE

V3
V4
V5A
FREE

/FLOATING DIVIDE ROUTINE				
/USES THE METHOD OF TRIAL DIVISION BY HI ORDER				
05722	0000	FFDIV, 0		(USED AS A TEM. BY I/O ROUTINES)
05723	4500	JMS I	PATCHP	/WHICH MODE OF CALL?
05724	1722	TAD I	FFDIV	/CALLED BY USER-GET ARG. ADDR.
05725	4774	JMS I	MOSETK	/GO SET UP FOR DIVIDE-OPX IN AC ON RETN.
05726	7041	FFD1, CMA	IAC	/NEGATE EXP. OF OPERAND
05727	1044	TAD	ACX	/ADD EXP OF FAC
05730	3044	DCA	ACX	/STORE AS FINAL EXPONENT
05731	1050	TAD	OPH	/NEGATE HI ORDER OP. FOR USE
05732	7141	CLL CMA	IAC	/AS DIVISOR
05733	3050	DCA	OPH	
05734	4345	JMS	DV24	/CALL DIV.--(ACH+ACLO)/OPH
05735	1046	TAD	ACLO	/SAVE QUOT. FOR LATER
05736	3041	DCA	AC1	
05737	1376	TAD	KM13	/SET COUNTER FOR 12 BIT MULTIPLY
05740	3200	DCA	FFMPY	/TO MULTIPLY QUOT. OF DIV. BY
05741	5311	JMP	DVLP1	/LOW ORDER OF OPERAND (OPL)

→ MP12L

NO

```

/END OF FLOATING DIVIDE-FUDGE SOME
/STUFF THEN JUMP INTO MULTIPLY
/
05742 1322 FDDON, TAD FFDIV /STORE RETN. ADDR. IN MULT ROUTINE
05743 3200 DCA FFMPY
05744 5227 JMP MDONE /GO CLEAN UP
/
/DIVIDE ROUTINE--24 BITS IN ACH, ACLO ARE DIVIDED BY 12 BITS
/IN OPH. OPH IS ASSUMED NEGATIVE AND .GT. ACH IN ABSOLUTE VALUE
/ELSE-DIVIDE OVERFLOW--WE RETURN NORMALLY WITH QUOTIENT
/IN ACLO AND REM. IN ACH, (AC2=0 ON RETN.)
/
05745 0000 DV24, 0
05746 1045 TAD ACH /CHECK THAT DIVISOR IS .GT. DIVIDEND
05747 1050 TAD OPH /DIVISOR IN OPH (NEGATIVE)
05750 7630 SZL CLA /IS IT?
05751 5777 JMP I DVOVR /NO-DIVIDE OVERFLOW
05752 1376 TAD KM13 /YES-SET UP 12 BIT LOOP
05753 3042 DCA AC2
05754 5365 JMP DV1 /GO BEGIN DIVIDE
05755 1045 DV2, TAD ACH /CONTINUE SHIFT OF FAC LEFT
05756 7004 RAL
05757 3045 DCA ACH /RESTORE HI ORDER
05760 1045 TAD ACH /NOW SUBTRACT DIVISOR FROM HI ORDER
05761 1050 TAD OPH /DIVIDEND
05762 7430 SZL /GOOD SUBTRACT?
05763 3045 DCA ACH /YES-RESTORE HI DIVIDEND
05764 7200 CLA /NO-DON'T RESTORE--OPH.GT.ACH
05765 1046 DV1, TAD ACLO /SHIFT FAC LEFT 1 BIT-ALSO SHIFT
05766 7004 RAL /1 BIT OF QUOT. INTO LOW ORD OF ACLO
05767 3046 DCA ACLO
05770 2042 ISZ AC2 /DONE 12 BITS OF QUOT?
05771 5355 JMP DV2 /NO-GO ON
05772 5745 JMP I DV24 /YES-RETN W/AC2=0
05773 6135 FFNEGR, FFNEG
05774 5450 MDSETK, MDSET
05775 7764 KKM12, -14
05776 7763 KKM13, -15
05777 6355 DVOVR, DV

```

*Replace
by EAE*

*V3
V4
V5A
FREE*

6000

PAGE

/FLOATING ADD

```

06000 0000 FFADD, 0
06001 4555-23 JMS I PATCHP /WHICH MODE FO CALL?
06002 1600 TAD I FFADD /CALLED BY USER-GET ADDR. OF OPR.
06003 4727 JMS I ARGETP /PICK UP OPERAND
06004 6201 FAD1, CDF /OF TO PACKAGE FIELD
06005 1050 TAD OPH /IS OPERAND = 0
06006 7650 SNA CLA
06007 5227 JMP DONA /YES-DONE
06010 1045 TAD ACH /NO-IS FAC=0?
06011 7650 SNA CLA
06012 5223 JMP DOADD /YES-DO ADD
06013 1044 TAD ACX /NO-DO EXPONENT CALCULATION
06014 7141 CLL CMA IAC
06015 1047 TAD OPX
06016 7540 SMA SZA /WHICH EXP. GREATER?
06017 5231 JMP FACR /OPERANDS-SHIFT FAC
06020 7041 CMA IAC /FAC'S-SHIFT OPERAND=DIFFRNC+1
06021 4234 JMS OPSR
06022 4272 JMS ACSR /SHIFT FAC ONE PLACE RIGHT
06023 1047 DOADD, TAD OPX /SET EXPONENT OF RESULT
06024 3044 DCA ACX
06025 4357 JMS OADD /DO THE ADDITION
06026 4776 JMS I FNORP /NORMALIZE RESULT
06027 2200 DONA, ISZ FFADD /BUMP RETURN
06030 5600 JMP I FFADD /RETURN
06031 4272 FACR, JMS ACSR /SHIFT FAC = DIFF.+1
06032 4234 JMS OPSR /SHIFT OPR. 1 PLACE
06033 5223 JMP DOADD /DO ADDITION

```

/OPERAND SHIFT RIGHT-ENTER WITH POSITIVE COUNT-1

/IN AC

```

06034 0000 OPSR, 0-
06035 7040 CMA
06036 3040 DCA AC0 /- (COUNT+1) TO SHIFT COUNTER
06037 1050 LOP2, TAD OPH /GET SIGN BIT
06040 7004 RAL /TO LINK
06041 7200 CLA
06042 1050 TAD OPH /GET HI MANTISSA
06043 7010 RAR /SHIFT IT RIGHT, PROPAGATING SIGN
06044 3050 DCA OPH /STORE BACK
06045 1051 TAD OPL
06046 7010 RAR
06047 3051 DCA OPL /STORE LO ORDER BACK
06050 7010 RAR /SAVE 1 BIT OF OVERFLOW
06051 3042 DCA AC2 /IN AC2
06052 2047 ISZ OPX /INCREMENT EXPONENT
06053 7000 NOP2, NOP
06054 2040 ISZ AC0 /DONE ALL SHIFTS?
06055 5237 JMP LOP2 /NO-LOOP
06056 5634 JMP I OPSR /YES-RETN.

```

-1

EAE

/SHIFT FAC LEFT 1 BIT

06057 0000 AL1, 0
 06060 1041 TAD AC1 /GET OVERFLOW BIT
 06061 7104 CLL RAL /SHIFT LEFT
 06062 3041 DCA AC1 /STORE BACK
 06063 1046 TAD ACLO /GET LOW ORDER MANTISSA
 06064 7004 RAL /SHIFT LEFT
 06065 3046 DCA ACLO /STORE BACK
 06066 1045 TAD ACH /GET HI ORDER
 06067 7004 RAL
 06070 3045 DCA ACH /STORE BACK
 06071 5657 JMP I AL1 /RETN.

/SHIFT FAC RIGHT-ENTER WITH COUNT=1 IN AC (POSITIVE)

06072 0000 ACSR, 0
 06073 7040 CMA /AC CONTAINS COUNT=1
 06074 3040 DCA AC0 /STORE COUNT
 06075 1045 LOP1, TAD ACH /GET SIGN BIT OF MANTISSA
 06076 7004 RAL /SET UP SIGN PROPAGATION
 06077 7200 CLA
 06100 1045 TAD ACH /GET HIGH ORDER MANTISSA
 06101 7010 RAR /SHIFT RIGHT 1, PROPAGATING SIGN
 06102 3045 DCA ACH /STORE BACK
 06103 1046 TAD ACLO /GET LOW ORDER
 06104 7010 RAR /SHIFT IT
 06105 3046 DCA ACLO /STORE BACK
 06106 7010 RAR
 06107 3041 DCA AC1 /SAVE 1 BIT OF OVERFLOW
 06110 2044 ISZ ACX /INCREMENT EXPONENT
 06111 7000 NOP1, NOP
 06112 2040 ISZ AC0 /DONE?
 06113 5275 JMP LOP1 /NO-LOOP
 06114 5672 JMP I ACSR /YES-RETN-AC=L=0

/DIVIDE OVERFLOW-ZERO ACX,ACH,ACLO

06115 7300 DBAD, CLA CLL /NECESSARY SO WE DON'T GET OVRFLO AGAIN
 06116 5775 JMP I DBAD1P /GO ZERO ALL

/FLOATING SUBTRACT

06117 0000 FFSUB, 0
 06120 4555 JMS I PATCHP /WHICH MODE OF CALL?
 06121 1717 TAD I FFSUB /CALLED BY USER-GET ADDR. OF OP
 06122 4727 JMS I ARGETP /PICK UP THE OP.
 06123 4346 JMS OPNEG /NEGATE OPERAND
 06124 1317 TAD FFSUB /JMP INTO FLTG. ADD
 06125 3200 SUB0, DCA FFAOD /AFTER SETTING UP RETURN
 06126 5204 JMP FAD1
 06127 6200 ARGETP, ARGET

6130-6134 0000

6135

*6135

/FLOATING NEGATE

06135	0000	FFNEG, 0		/(USED AS A TEM. BY OUTPUT ROUTINE)
06136	1046	TAD	ACLO	/GET LOW ORDER FAC
06137	7141	CLL CMA	IAC	/NEGATE IT
06140	3046	DCA	ACLO	/STORE BACK
06141	7024	CML	RAL	/ADJUST OVERFLOW BIT AND
06142	1045	TAD	ACH	/PROPAGATE CARRY-GET HI ORD
06143	7141	CLL CMA	IAC	/NEGATE IT
06144	3045	DCA	ACH	/STORE BACK
06145	5735	JMP I	FFNEG	

/NEGATE OPERAND

06146	0000	OPNEG, 0		
06147	1051	TAD	OPL	/GET LOW ORDER
06150	7141	CLL CMA	IAC	/NEGATE AND STORE BACK
06151	3051	DCA	OPL	
06152	7024	CML	RAL	/PROPAGATE CARRY
06153	1050	TAD	OPH	/GET HI ORDER
06154	7141	CLL CMA	IAC	/NEGATE AND STORE BACK
06155	3050	DCA	OPH	
06156	5746	JMP I	OPNEG	

/ADD OPERAND TO FAC

06157	0000	OADD, 0		
06160	7100	CLL		
06161	1042	TAD	AC2	/ADD OVERFLOW WORDS
06162	1041	TAD	AC1	
06163	3041	DCA	AC1	
06164	7004	RAL		/ROTATE CARRY
06165	1051	TAD	OPL	/ADD LOW ORDER MANTISSAS
06166	1046	TAD	ACLO	
06167	3046	DCA	ACLO	
06170	7004	RAL		
06171	1050	TAD	OPH	/ADD HI ORDER MANTISSAS
06172	1045	TAD	ACH	
06173	3045	DCA	ACH	
06174	5757	JMP I	OADD	/RETN.
06175	5524	DBAD1P, DBAD1		
06176	6215	FNORP, FFNOR		

IFNZRO EAE <

/EAE FLOATING POINT PACKAGE
/FOR PDP8/E WITH KE8-E EAE/
/W.J. CLOHER

/DEFINITIONS OF EAE INSTRUCTIONS

SWP=7521

CAM=7621

MQA=7501

MQL=7421

SGT=6006

SWAB=7431

SWBA=7447

SCA=7441

MUY=7405

DVI=7407

NMI=7411

SHL=7413

ASR=7415

LSR=7417

ACS=7403

SAM=7457

DAD=7443

OLD=7663

DST=7445

DPIC=7573

DCM=7575

OPSZ=7451

/

ACLO=LORD

TM=TEMP4

ENPUNCH

*4600

/
/FLOATING OUTPUT ROUTINE
/

```

FFOUT, 0
      SWAB          /ALSO DOES MQL TO CLR, AC
      DCA          SIGN /CLEAR SIGN AND COUNT WORDS
      DCA          KNT
      TAD          EFLG /IS THIS E FORMAT?
      SZA          CLA
      JMP          FFMT /NO-F FORMAT
      CLL CML IAC RTL /YES-MAKE A 6
      DCA          DADP /STORE AS # OF DIGITS AFT DEC PT
      TAD          K16 /SET FIELD WIDTH TO 14 ( DECIMAL)
      DCA          FLDW
FFMT,  CDF          /CHANGE TO FIELD OF PACKAGE
      TAD          KM7 /SET # OF SIGNF, DIGITS
      DCA I        DCNTP /TO 6 (DON'T PRINT 7TH)
      TAD          ACH /DETERMINE IF #=0
      SNA
      JMP          FOUT3 /YES-SKIP DOWN
      SMA          CLA /NO-IS IT NEGATIVE?
      JMP          .+3 /POSITIVE
      ISZ          SIGN /NEGATIVE-SET FLAG
      JMS I        FFNGP /AND NEGATE #
FOUT1, TAD          ACX /GET # INTO RANGE .1<=N<1
      SMA SZA CLA /IS EXP. NEG.?
      JMP          FOUT2 /NO-GO ON
      JMS I        FFMPP /YES-MAKE # GREATER THAN 1
      TEN          /BY MULTIPLYING BY TEN (DEC.)
      ISZ          KNT /COUNT THE MULTIPLIES
      JMP          FOUT1 /SEE IF >1 YET
FOUT2, JMS I        SEP /* IS >1-MAKE IT LESS THAN 1
      JMS I        FFPUTP /STORE IN A TEMPORARY
      TM3
      DCA          ACX /SET FAC TO .5
      CLL CML RTR
      DCA          ACH
      DCA          ACLO
      TAD          EFLG /IS THIS E FORMAT?
      SZA          CLA
      TAD          KNT /NO-GET COUNT OF MULTIPLIES
      CMA          IAC /NEGATE IT
      TAD          DADP /AND ADD # OF DIGITS AFT. DC. PT.
      SMA          /MUST BE NEGATIVE
      CMA
      TAD          KK7 /LIMIT # OF DIVS TO 7
      SPA
      CLA
      TAD          KM7 /RESTORE
      DCA I        SEP /STORE AS COUNTER
      JMP          .+3

```

```

JMS I  FFDVP  /DIVIDE .5 BY TEN THAT # OF TIMES
TEN
ISZ I  SEP    /DONE?
JMP    .-3    /NO-GO ON
JMS I  FFAOP  /YES-ADD IN ORIG,#-THIS IS ROUNDING
TM3
JMS I  SEP    /INSURE THAT IT IS IN RANGE
FOUT4, TAD    ACX  /GET EXPONENT
CMA    IAC    /USE AS COUNT FOR SHIFTING MANT.
DCA    FOUT5
DLD
ACH
SWP    SHL    /PUT IN CORRECT ORDER
1      /SHIFT LEFT 1(FOR 0 EXP.)
LSR    /NOW SHIFT RIGHT ACCORD TO EXP.
FOUT5, 0
DCA    ACH    /STORE BACK
SWP
DCA    ACLO
FOUT3, TAD    KNT  /DONE-GET COUNT OF MULS.
DCA    OPX    /PRESERVE IT
TAD    EFLG   /IS THIS E FORMAT OUT?
SZA    CLA
JMP    NOTE   /NO
DCA    KNT    /YES-ZERO COUNT
TAD    KM7    /GET MINUS 7-FOR 2 SIGNS,PT,+EXP
JMP    ADFW   /GO ADD FIELD WIDTH
NOTE,  TAD    KNT  /GET COUNT OF MULTIPLIES
SMA    /IF NOT NEG-MAKE = -2
CLA    CMA
TAD    M1
ADFW,  TAD    FLDW  /GET THE FIELD WIDTH
CMA    IAC    /NEGATE IT
TAD    DADP   /ADD DIGITS AFTER DEC. PT
SMA    /NEG?
JMP I  PRNTXP /NO-PRINT XS-NOT ENUFF ROOM
DCA I  SEP    /STORE AS CNT OF SPACES
JMP    .+3
TAD    KK240
JMS I  OUTP   /PRINT A SPACE
ISZ I  SEP    /DONE?
JMP    .-3    /NO-GO ON
TAD    SIGN   /YES-GET SIGN
CLL    RAL    /MAKE A ZERO OR 2
TAD    K253   /FOR PLUS OR MINUS
JMS I  OUTP   /PRINT SIGN
TAD    KNT    /GET MUL COUNT
SMA
JMP I  PRZROP /PRINT LEADING ZERO
CMA    IAC
JMS I  DGIYPP /OUTPUT 'KNT' DIGITS
PRDCP, TAD    DADP  /DON'T PRINT DEC. PT
SNA    CLA    /IF DADP IS 0
JMP I  GKNTP
JMP I  POPP

```

PRZROP,	PRZRO	
PDP,	PDP	
K16,	16	
GKNT,	GKNT	
FLINK,	JMP I	FFOUT
PRNTXP,	PRNTX	
K253,	253	
PRP,	PR	
DCNT,	DCNT	
M1,	7777	
KK7,	7	
DGTYP,	DGTYP	
OUTP,	OUT	
KK240,	240	
KM7,	-7	
FFADP,	FFADD	
FFOVP,	FFDIV	
FFPUT,	FFPUT	
SEP,	SE	
FFMPP,	FFMPY	
FFNGP,	FFNEG	
KNT,	0	
SIGN,	0	

	PAGE		
PDP,	CLA CLL	CMA RAL	
	JMS	OUTDG	/PRINT DEC. PT.
GKNT,	TAD I	KNTF	/GET COUNT AGAIN
	SPA SNA	CLA	
	JMP	GD	
	TAD I	KNTF	/GET COUNT
	CMA		/NEGATE
	DCA	DGTYP	/STORE AS COUNTER
	TAD	DADP	
	CMA		/SAME FOR DADP
	DCA	SE	
	JMP	PR	/GO ON
PZR,	JMS	OUTDG	/PRINT A ZERO
PR,	ISZ	DGTYP	
	SKP		
	JMP	PS	
	ISZ	SE	
	JMP	PZR	
PS,	TAD I	KNTF	
	CMA	IAC	
GD,	TAD	DADP	
	SMA	SZA	
	JMS	DGTYP	
	TAD	EFLG	
	SZA	CLA	
	JMP	DONEF	/DONE
	TAD	K305	/PRINT 'E'
	JMS	OUT	
	TAD	OPX	/GET PRESERVED COUNT OF MULS
	SMA SZA	CLA	/DETERMINE SIGN
	CLA IAC	RAL	/MAKE A 2
	TAD	P253	/PRINT MINUS OR PLUS SIGN
	JMS	OUT	
	TAD	OPX	/GET THE COUNT
	SPA		
	CMA	IAC	/NEGATE IF NEGATIVE
	MQL	DVI	/DIVIDE BY ONE HUNDRED
	K144		
	SWP		/QUOT TO AC, REM TO MQ
	JMS	OUTDG	/THIS IS FIRST DIG-PRINT IT
	DVI		/DIVIDE REM BY TEN
	K12		
	SWP		/GET SECOND DIGIT
	JMS	OUTDG	/PRINT IT
	SWP		
	JMS	OUTDG	/PRINT LAST
DONEF,	TAD	SWIT2	/SHOULD WE PRINT CR/LF?
	SNA	CLA	
	JMP I	FLING	/NO
	TAD	KK215	
	JMS	OUT	
	TAD	K212	
	JMS	OUT	
	JMP I	FLING	

/

/ROUTINE TO GET FAC<1

/

SE, 0

SE1, TAD ACX

SPA SNA CLA /#>1?

JMP I SE /NO-RETN,

JMS I FFDV /YES-DIV. BY TEN

TEN

CMA

TAD I KNTD /REDUCE KNT BY 1

DCA I KNTD

JMP SE1

/

/OUTPUT DIGITS OF FAC BY MULTIPLYING BY TEN

/THE HIGH ORDER OVERFLOW IS THE DIGIT

DGTYP, 0

CMA IAC

DCA SE /STORE COUNT PASSED

SWAB /MODE 8 OF EAE

DT1, TAD ACLO /GET LOW ORDER FAC

SQL MUY /MUL BY TEN

K12

SWP /NEW ACLO TO AC

DCA ACLO /STORE IT BACK

TAD ACH /GET ACH-SEND TO MQ, AND

SWP MUY /HI ORD. OVERFLO OF MUY TO AC

K12 /MULT BY TEN, OVRFLO IS ADDED

ISZ DCNT /DONE ALL SIGNIF. DIGS.?

JMP .+3 /NO-GO ON

CLA CMA /YES-PRINT ZEROS

DCA DCNT /FROM NOW ON

JMS OUTDG /PRINT DIGIT (HI ORD. OVRFLOW)

SWP /NEW ACH IS IN MQ

DCA ACH /STORE IT

ISZ SE /DONE REQUIRED?

JMP DT1 /NOPE

JMP I DGTYP /YUP

PRNTX, CLA

TAD FLOW /GET FIELD WIDTH

CMA /MUST BE NEGATIVE

DCA SE /USE AS COUNTER

PRNTX1, ISZ SE /DONE ALL?

SKP /NO-GO ON

JMP DDEF /YES-RETN,

TAD K252

JMS OUT /PRINT ASTERISK

JMP PRNTX1

K252, 252 /ASTERISK


```
PRZRO, CLA      /CLR. GARBAGE
      JMS      OUTDG /PRINT ZERO
      JMP I    PROCPP /PRINT DEC. PT. (MAYBE)
PROCPP, PROCP
/
/OUTPUT ROUTINE
/
OUT, 0
      DCA I XR3      /STORE IN INTERMEDIATE BUFFER
      JMP I  OUT

/
/OUTPUT DIGIT
/
OUTDG, 0
      TAD      P260
      JMS      OUT
      JMP I    OUTDG /RETN

KNTP, KNT
KK215, 215
K212, 212
TM3, 0
      0
      0
DCNT, 0      /COUNT OF SIGNF. DIGITS
K305, 305
P260, 260
FFDV, FFDIV
P253, 253
FLING, FLINK
K144, 144
```

/

/FLOATING POINT INPUT ROUTINE

/

FFIN,	PAGE		
	0		
	CLA	CMA	
	DCA	PRSW	/INITIALIZE PERIOD SWITCH TO -1
	CMA		/SET SIGN SWITCH TO -1
	DCA	SIGNF	
	COF		/CHANGE TO DF OF PACKAGE
	DCA	DSWIT	/ZERO CONVERSION SWITCH
DECONV,	DCA	ACX	/ZERO OUT THE FACI
	DCA	ACLO	
	DCA	ACH	
DECNV,	DCA	DNUMBR	/ZERO # OF DIGITS SINCE DEC. PT.
DECON,	JMS	GCHR	/GET A CHAR. FROM TTY.
	JMP	FFIN1	/TERMINATOR-
	ISZ	DSWIT	/DIGIT-BUMP CONVERSION SWITCH
	ISZ	DNUMBR	/BUMP # OF DIGITS
	DCA	TP1	/STORE IT IN FORM EASILY FLOATIBLE
	JMS I	FMPYLL	/MULTIPLY # BY 10
	TEN		
	JMS I	FPUTL	/STORE IT AWAY
	FPPTM1		
	JMS I	FGETL	/GET NEW DIGIT
	TP		
	JMS I	FNORL	/FLOAT IT
	JMS I	FADDLL	/ADD IT TO THE ACCUMULATED #
	FPPTM1		
	JMP	DECON	/GO ON
FFIN1,	ISZ	PRSW	/HAVE WE HAD A PERIOD YET?
	JMP	FIGO2	/YES-GO ON
	TAD	K2	/NO-IS THIS A PERIOD?
	SNA	CLA	
	JMP	DECNV	/YES-ZERO DIG. COUNT AFTER DEC. PT.
			/AND GO CONVERT REST
	DCA	DNUMBR	/NO-TERMINATOR-ZERO COUNT OF
			/DIGITS AFTER DECIMAL POINT.
FIGO2,	CLA	MQL	/0 TO MQ FOR LATER MULTIPLY
	ISZ	SIGNF	/IS # NEGATIVE?(DID WE GET - SIGN?)
	JMS I	FFNEGP	/YES-NEGATE IT
	SWAB		
	CMA		/RESET SIGN SWITCH FOR EXP.
	DCA	SIGNF	
	TAD	CHAR	/NO-WAS THE TERMINATOR AN 'E'?
	TAD	KME	
	SNA	CLA	
GETE,	JMS	GCHR	/YES-GET A CHAR. OF EXPONENT
	JMP	EDON	/END OF EXPONENT
	MUY		/GOT DIGIT OF EXP-MULT ACCUMULATED
	K12		/EXPONENT BY TEN AND ADD DIGIT
	JMP	GETE	/CONTINUE

```

EDON,  ISZ  SIGNF  /WAS EXPONENT NEGATIVE?
      DCM           /YES=NEGATE IT
      CLA  CLL     /CLEAR AC AND LINK
      TAD  DNUMBR  /GET # TIMES TO DIV MANTISSA BY TEN
      SAM           /SUBTRACT FROM EXPONENT
      CLL
      SPA           /RESULT POSITIVE?
      CLL CMA CML IAC /NO=MAKE POS. AND SET LINK
      CMA           /NEGATE FOR COUNTER
      DCA  DNUMBR  /AND STORE
      RAL           /LINK=1-DIV;=0=MUL. # BY TEN
      TAD  MDV     /FORM CORRECT INSTRUCTION
      DCA  FINST   /AND STORE FOR EXECUTION
FCNT,  ISZ  DNUMBR  /DONE ALL OPERATIONS?
      JMP  FINST   /NO
      JMP I  FFIN   /YES=RETURN
FINST,  0          /NO= MUL OR DIV. MANTISSA
      TEN         /BY TEN
      JMP  FCNT    /GO ON
FFNEGP, FFNEG
PRSW,  0
DNUMBR, 0
SIGNF, 0
K2,    2
KME,   -305
MDV,   JMS I      ,+1  /THESE 3 WDS, MUST BE IN THIS ORDER
FFPYLL, FFMPY
      FFDIV       /!!!!!!!!!!!!!!!!!!!!
FADDLL, FFADD

K12,   12
TP,    13
TP1,   0
      0
TEN,   4
      2400
      0

```

/ROUTINE TO GET A CHAR FROM THE TTY AND SEE IF IT IS DIGIT
 /OR A TERMINATOR.
 /RETURN TO CALL + 1 IF TERMINATOR, TO CALL + 2 IF DIGIT
 /THIS ROUTINE MUST NOT MODIFY THE MQ!!

GCHR, 0
 JMS INPUT /GET A CHAR FROM TTY.
 TAD CHAR /PICK IT UP
 TAD PLUS /WAS IT PLUS SIGN?
 SNA
 JMP DECON1 /YES-GET ANOTHER CHAR.
 TAD MINUS /NO WAS IT MINUS SIGN?
 SZA CLA
 JMP .+3
 DCA SIGNF /YES-FLIP SWITCH
 DECON1, JMS INPUT /GET A CHAR.
 TAD CHAR
 TAD K7506 /SEE IF ITS A DIGIT
 CLL
 TAD K12
 SZL /DIGIT?
 ISZ GCHR /YES-RETN. TO CALL+2
 JMP I GCHR /NO-RETN. TO CALL+1
 K7506, 7506
 PLUS, -253
 MINUS, 253-255

/

/INPUT ROUTINE-IGNORES LEADING SPACES

INPUT, 0
 JMS I GETCHL /USE OUR ROUTINE TO GET CHAR
 TAD DSWIT /GET TERMINATOR
 SZA CLA /VALID INPUT YET?
 JMP IOUT /YES-CONTINUE
 TAD CHAR /NO-GET CHAR
 TAD M240 /COMPARE AGAINST SPACE
 SNA CLA /IS IT A SPACE?
 JMP INPUT+1 /YES-IGNORE IT
 IOUT, JMP I INPUT /RETURN
 M240, -240

/

/ROUTINE TO DECIDE CALLING MODE IN LIEU OF "SPECIAL MODE" PROBLEMS

/

 *5364
 PATCHF, 0
 SZA /IS AC EMPTY
 JMP RTN2 /NO-THIS IS ALWAYS SI MODE WITH ADUR IN AC
 TAD FF /YES-GET SPECIAL MODE FLIP-FLOP
 SZA CLA /IF ON, THE ZERO AC MEANS ADDRESS OF 0
 RTN2, ISZ PATCHF /USE AC AS ADDRESS OF OPERAND
 JMP I PATCHF /RETURN

PAGE

/FLOATING SUBTRACT-USES FLOATING ADD

/FSW111

FFSUB1, 0

JMS I PATCHP /WHICH MODE?
 TAD I FFSUB1 /CALLED BY USER-GET ADDR. OF OP
 JMS I ARGETL /PICK UP ARGUMENT
 CDF
 JMS I FFNEGA /NEGATE FAC!
 TAD FFSUB1
 JMP I SUBOP

FFNEGA, FFNEG
 SUBOP, SUB0

/FLOATING DIVIDE

/FSWITCH=1

/THIS IS OP/FAC

/

FFDIV1, 0

JMS I PATCHP /WHICH MODE OF CALL?
 TAD I FFDIV1 /CALLED BY USER-GET ADDR.
 JMS I ARGETL /((INTERP.)-GET OPRND.-ADDR. IN AC
 CDF /CDF TO FIELD OF PACKAGE
 TAD ACH /SWAP FAC AND OPRND-OPH IN MQ!
 DCA OPH /STORE ACH IN OPH
 TAD ACX /GET EXP OF FAC
 SWP /OPH TO AC, ACX TO MQ
 DCA ACH /STORE OPH IN ACH
 TAD OPX /STORE OPX IN ACX
 DCA ACX
 TAD OPL /OPL TO MQ, ACX TO AC
 SWP
 DCA OPX /STORE ACX IN OPX
 TAD ACLO
 DCA OPL /STORE ACLO IN OPL
 TAD OPH /OPH TO MQ FOR LATER
 SWP
 DCA ACLO /STORE OPL IN ACLO
 TAD FFDIV1 /SET UP SO WE RETN TO
 DCA I FFDP /NORMAL DIVIDE ROUTINE
 TAD FD1
 DCA I MDSETP
 JMP I MD1P /GO ARRANGE OPERANDS

MD1P, MD1
 ARGETL, ARGET
 MDSETP, MDSET
 FFDP, FFDIV
 FD1, FFD1

/PATCH TO EAE ADD ROUTINE

```
ADDPCH, 0
      TAD AC1
      TAD RB4000
      DPSZ
      JMP ADDP1
      CLL CML RTR
      ISZ ACX
      NOP
ADDP1,  TAD RB4000
      JMP I ADDPCH
RB4000, 4000
```

/ROUTINE TO BUMP CHARACTER NUMBER
/USED BY CHAR PACKING ROUTINES, BUT ITS HERE AS A PATCH

```
      +5573
CNOBML, 0
      TAD I WORD0      /HEADER WORD
      TAD K0100        /ADD 1 TO THE COUNT BITS
      DCA I WORD0
      JMP I CNOBML     /DONE
```

/FLOATING MULTIPLY--DOES 4 SINGLE MULTIPLIES WITH EAE
 /THIS USES THE FACT THAT IF AC IS NON-ZERO WHEN YOU DO
 /A MUY INSTR, THE AC IS ADDED TO RESULT OF THE MULTIPLY.
 /((IN THE LOW ORDER, NATCHERLY))

PAGE

FFMPY, 0

JMS I	PATCHP	/WHICH MODE?
TAD I	FFMPY	/CALLED BY USER-GET ADDRESS
JMS	MDSET	/SET UP FOR MULT
CLA	MUY	/MULTIPLY-LOW ORDER FAC STILL IN MQ
OPH		/THIS IS PRODUCT OF LOW ORDERS
MOI		/ZAP LOW ORDER RESULT-INSIGNIFICANT
TAD	ACH	/GET LOW ORDER(!) OF FAC
SWP	MUY	/TO MQ-HIGH ORD, RESLT OF LAST MPY
OPL		/TO AC-WILL BE ADDED TO RESLT-THIS
DST		/IS PRODUCT-LOW ORD FAC, HI ORD OP
ACB		/STORE RESULT
OLD		/HIGH ORDER FAC TO MQ, OPX TO AC
ACLO		
TAD	ACX	/ADD FAC EXPONENT-GET SUM OF EXPS.
DCA	ACX	/STORE RESULT
MUY		/MUL. HIGH ORDER FAC BY LOW ORD OP.
OPH		/HIGH ORDER FAC WAS IN MQ
DAD		/ADD IN RESULT OF SECOND MULTIPLY
ACB		
DCA	ACH	/STORE HIGH ORDER RESULT
TAD	ACLO	/GET HIGH ORDER FAC
SWP		/SEND IT TO MQ AND LOW ORD. RESULT
DCA	ACB	/OF ADD TO AC-STORE IT
RAL		/ROTATE CARRY TO AC
DCA	ACLO	/STORE AWAY
MUY		/NOW DO PRODUCT OF HIGH ORDERS
OPL		/FAC HIGH IN MQ, OP HIGH IN OPL
DAD		/ADD IN THE ACCUMULATED #
ACH		
SNA		/ZERO?
JMP	RTZRO,	/YES-GO ZERO EXPONENT
NMI		/NO-NORMALIZE (1 SHIFT AT MOST!)
DCA	ACH	/STORE HIGH ORDER RESULT
CLA	SCA	/GET STEP CNTR-DID WE NEED A SHIFT?
SNA	CLA	
JMP	SNCK	/NO-JUST CHECK SIGN
CLA	CMA	/YES-MUST DECREASE EXP. BY 1
TAD	ACX	
DCA	ACX	/STORE BACK

RTZRO,

```

      TAD      AC0
      SPA      CLA      /IS HIGH ORDER OF OVERFLO WD. 1?
      DPIC      /YES-ADD 1 TO LOW ORDER-STILL IN MQ
SNCK,  ISZ      MSIGN    /RESULT NEGATIVE?
      JMP      MPOS      /NO-GO ON
      TAD      ACH      /YES-GET HIGH ORDER BACK
      DCM      /LOW ORDER STILL IN MQ-NEGATE
      DCA      ACH      /STORE HIGH ORDER BACK
MPOS,  SWP      /LOW ORDER TO AC
      DCA      ACLO     /STORE AWAY
      ISZ      FFMPY    /BUMP RETURN
      JMP I     FFMPY    /RETURN
MSIGN, 0
ARGETK, ARGET
DVOFL,  DV

```

```

/
/ROUTINE TO SET UP FOR MULTIPLY AND DIVIDE
/

```

```

MDSET, 0
      JMS I     ARGETK    /GET OPERAND (ADDR. IN AC)
      CDF      /CHANGE TO DATA FIELD OF PACKAGE
MD1,   CLA CLL CMA RAL   /MAKE A MINUS TWO
      DCA      MSIGN     /AND STORE IN MSIGN.
      TAD      OPL       /GET LOW ORDER MANTISSA OF OP.
      SWP      /GET INTO RIGHT ORDER ( OPH IN MQ)
      SMA      /NEGATIVE?
      JMP      .+3       /NO
      DCM      /YES-NEGATE IT
      ISZ      MSIGN     /BUMP SIGN COUNTER
      SHL      /SHIFT OPRND LEFT 1 TO AVOID OVRFLO
      1
      DST      /STORE BACK-OPH CONTAINS LOW ORDER
      OPH      /          OPL CONTAINS HIGH ORDER
      DLD      /GET THE MANTISSA OF THE FAC
      ACH
      SWP      /MAKE IT CORRECT ORDER
      SMA      /NEGATIVE?
      JMP      FPOS      /NO
      DCM      /YES-NEGATE IT
      ISZ      MSIGN     /BUMP SIGN COUNTER (MAY SKIP)
      NOP
FPOS,  DST      /STORE BACK-ACH CONTAINS LOW ORDER
      ACH      /          ACLO CONTAINS HIGH ORDER
      JMP I     MDSET    /RETURN

```

OPL-51
 1/2

/FLOATING DIVIDE

*5722
 PFDIV, 0
 JMS I PATCHP /WHICH MODE?
 TAD I FFDIV /CALLED BY USER-GET ARG. ADDRESS
 JMS MDSET /GET ARG. AND SET UP SIGNS
 PFD1, DVI /DIVIDE-ACH AND ACLO IN AC, MQ
~~OPL~~ /THIS IS HI (1) ORDER DIVISOR
 DST /QUOT TO AC0, REM TO AC1
 AC0
 SZL CLA /DIVIDE ERROR?
 JMP I DVOFL /YES-HANDLE IT
 TAD OPX /DO EXPONENT CALCULATION
 CMA IAC /EXP. OF FAC = EXP. OF OP
 TAD ACX
 DCA ACX
 DPSZ /IS QUOT = 0?
 SKP /NO-GO ON
 DCA ACX /YES-ZERO EXPONENT
 DVLP, MUY /NO-THIS IS Q*OPL*2**-12
 OPH
 DCM /NEGATE IT
 TAD AC1 /SEE IF GREATER THAN REMAINDER
 SNL
 JMP I DVOPSP /YES-ADJUST FIRST DIVIDE
 DVI /NO-DO Q*OPL*2**-12/OPH
 OPL
 SZL CLA /DIV ERROR?
 JMP I DVOFL /YES
 DVLP1, TAD AC0 /NO-GET QUOT OF FIRST DIV.
 SMA /NEGATIVE?
 JMP .+5 /NO-REMEMBER-QUOT OF 2ND DIV. IN MQ
 LSR /YES-MUST SHIFT IT RIGHT 1
 1
 ISZ ACX /ADJUST EXPONENT
 NOP
 ISZ MSIGN /SHOULD SIGN BE MINUS?
 SKP /NO
 DCM /YES-DO IT
 DBAD1, DCA ACH /STORE IT BACK
 SWP
 DCA ACLO
 ISZ FFDIV
 JMP I FFDIV /BUMP RETN. AND RETN.
 DVOPSP, DVOPS
 DBAD, CAM
 DCA ACX /ZERO EXPONENT
 JMP DBAD1 /GO ZERO MANTISSA

AC0 = d0
 1 = d1

/FLOATING ADDITION-IN ORDER NOT TO LOSE BITS, WE DO NOT
 /SHIFT BOTH NUMBERS RIGHT 1 BIT BEFORE ADD-ONLY SHIFTS DONE
 /ARE TO ALIGN EXPONENTS.

```

/
PAGE
FFADD, 0
JMS I PATCHP /WHICH MODE OF CALLING
TAD I FFAOD /CALLED DIRECTLY BY USER
JMS I ARGETP /PICK UP ARGUMENTS
CDF /CHANGE TO CURRENT DATA FIELD
FAD1, TAD OPX /PICK UP EXPONENT OF OPERAND
MQL /SEND IT TO MQ FOR SUBTRACT
TAD ACX /GET EXPONENT OF FAC
SAM /SUBTRACT-RESULT IN AC
SPA /NEGATIVE RESULT?
CMA IAC /YES-MAKE IT POSITIVE
DCA CNT /STORE IT AS A SHIFT COUNT
TAD CNT /COUNT TOO BIG?(CAN'T BE ALIGNED)
TAD M27
SPA SNA CLA
CMA /NO-OK
DCA AC0 /YES-MAKE IT A LOAD OF LARGEST #
DLD /GET ADDRESSES TO SEE WHO'S SHIFTED
ADDRS
SGT /WHICH EXP GREATER(GT FLG SET
/ BY SUBTR. OF EXPS.)
SWP /OPERAND'S-SHIFT THE FAC
DCA SHFBG /STORE ADDRESS OF WHO GETS SHIFTED
SWP /GET ADDRESS OF OTHER (0 TO MQ)
DCA DADR /THIS ONE JUST GETS ADDED
SGT /WHICH EXPONENT WAS GREATER?
JMP .+3 /FAC'S - DO NOTHING
TAD OPX /OPERAND'S-PUT FINAL EXP. IN ACX
DCA ACX
DLD /GET THE LARGER # TO AC,MQ
DADR, 0
SWP /PUT IN THE RIGHT ORDER
ISZ AC0 /COULD EXPONENTS BE ALIGNED?
JMP LOD /NO-JUST LEAVE LARGER IN AC,MQ
DST /YES-STORE THIS TEMPORARILY
AC0 /((IF ONLY FAC STORAGE WAS REVERSED)
DLD /GET THE SMALLER #
SHFBG, 0
SWP /PUT IT IN RIGHT ORDER
ASR /DO THE ALIGNMENT SHIFT
CNT, 0

```

```

DAD          /ADD THE LARGER #
AC0
DST          /STORE RESULT
AC0
SZL          /OVERFLOW?(L NOT = SIGN BIT)
CMA          /NOTE-WE DIDN'T SHIFT BOTH RIGHT 1
SMA          CLA
JMP          NOOV /NOPE
CLA CLL CML RAR /MAYBE-SEE IF 2 #S HAD SAME SIGN
AND          ACH
TAD          OPH
SMA          CLA /SIGNS ALIKE?
JMP          OVRFLO /YES=OVERFLOW
NOOV, JMS I ADDPCL /JUMP TO PATCH FOR THIS ROUTINE
LOD, NMI       /NORMALIZE (LOW ORDER STILL IN MQ)
DCA          ACH /STORE FINAL RESULT
SWP          /GET AND STORE LOW ORDER
DCA          ACLO
SCA          /GET SHIFT COUNTER(# OF NMI SHIFTS)
CMA          IAC /NEGATE IT
TAD          ACX /AND ADJUST FINAL EXPONENT
DCA          ACX
ADON, ISZ     FFADD /BUMP RETURN PAST ADDRESS
JMP I        FFADD /RETURN
OVRFLO, TAD   AC1 /OVERFLOW-GET HIGH ORDER RESLT BACK
ASR          /SHIFT IT RIGHT 1
1
TAD          KK4000 /REVERSE SIGN BIT
DCA          ACH /AND STORE
SWP
DCA          ACLO /STORE LOW ORDER
ISZ          ACX /BUMP EXPONENT
NOP
JMP          ADON /DONE
KK4000, 4000
M27, -27
ADDRS, OPH
ACH
ARGETP, ARGET
/FLOATING SUBTRACT-USES FLOATING ADD
/FSW0!!
FFSUB, 0
JMS I PATCHP /WHICH MODE?
TAD I FFSUB /CALLED BY USER-GET ADDRESS OF OP.
JMS I ARGETP
CDF
TAD          OPL /OPH IS IN MQ!
SWP          /PUT IT IN RIGHT ORDER
DCM          /NEGATE IT
DCA          OPH /STORE BACK
MGA
DCA          OPL
TAD          FFSUB /GO TO ADD
SUB0, DCA     FFADD
JMP        FAD1

```

```

/
/FLOATING NEGATE--NEGATE FLOATING AC
/
FFNEG, 0
      SWAB          /MUST BE MODE 8
      DLD           /GET MANTISSA
      ACH
      SWP           /CORRECT ORDER PLEASE!
      DCM           /NEGATE IT
      DCA          ACH /RESTORE
      SWP           /SEND 0 TO MQ
      DCA          ACLO
      JMP I        FFNEG

```

```

/
/CONTINUATION OF DIVIDE ROUTINE
/WE ARE ADJUSTING THE RESULT OF THE
/FIRST DIVIDE.
/
DVOPS, CMA      IAC
      DCA      AC1      /ADJUST REMAINDER
      TAD      OPL      /WATCH FOR OVERFLOW
      CLL CMA IAC
      TAD      AC1
      SNL
      JMP      DVOP1    /DON'T ADJUST QUOT.
      DCA      AC1
      CMA
      TAD      AC0
      DCA      AC0      /REDUCE QUOT BY 1
DVOP1, CLA      CLL
      TAD      AC1      /GET REMAINDER
      SNA      /ZERO?
      CAM      /YES-ZERO EVERYTHING
      DVI      /NO
      OPL
      SZL      CLA      /DIV. OVERFLOW?
      JMP I    DVOVR    /YES
      DCM      /NO-ADJUST HI QUOT (MAYBE)
      JMP I    DVLPI1   /GO BACK
DVLPI1, DVLPI
DVOVR,  DV
ADDPCL, ADPPCH
NOPUNCH
>

```

6200

PAGE

/

/ARGUMENT PICK UP ROUTINE-ENTER WITH DATA FIELD SET TO EITHER
/FLTG. DATA FIELD OR FLTG. INSTR. FIELD.
/ADDRESS OF OPERAND IS IN THE AC ON ENTRY.
/ON RETURN, THE AC IS CLEAR

/

06200	0000	ARGET, 0	
06201	3042	DCA AC2	/STORE ADDRESS OF OPERAND
06202	1442	TAD I AC2	/PICK UP EXPONENT
06203	3047	DCA OPX	
06204	4310	JMS ISZAC2	/MOVE POINTER TO WORD, WATCH FOR FIELD OVERLAP
06205	1442	TAD I AC2	/PICK IT UP
		IFZERO EAE <	
06206	7000	NOP	
06207	7000	NOP	

>

IFNZRO EAE <
ENPUNCH

*,

SWAB /OPH INTO MQ BECAUSE EAE ROUTINES
MQA /EXPECT TO FIND IT THERE
NOPUNCH

>

06210	3050	DCA OPH	/STORE
06211	4310	JMS ISZAC2	/MOVE POINTER TO WORD, WATCHING FOR OVERLAP
06212	1442	TAD I AC2	/PICK IT UP
06213	3051	DCA OPL	/STORE IT
06214	5600	JMP I ARGET	/RETURN

IFZERO EAE <

/ROUTINE TO NORMALIZE THE FAC

06215 0000 FFNOR, 0
 06216 1045 TAD ACH /GET THE HI ORDER MANTISSA
 06217 7450 SNA /ZERO?
 06220 1046 TAD ACLO /YES-HOW ABOUT LOW?
 06221 7450 SNA
 06222 1041 TAD AC1 /LOW=0, IS OVRFLO BIT ON?
 06223 7650 SNA CLA
 06224 5341 JMP ZEXP /#=0=ZERO EXPONENT
 06225 7332 NORMLP, CLA CLL CML RTR /NOT 0=MAKE A 2000 IN AC
 06226 1045 TAD ACH /ADD HI ORDER MANTISSA
 06227 7440 SZA /HI ORDER = 6000
 06230 5233 JMP .+3 /NO-CHECK LEFT MOST DIGIT
 06231 1046 TAD ACLO /YES=6000 OK IF LOW=0
 06232 7640 SZA CLA
 06233 7710 SPA CLA /2,3,4,5, ARE LEGAL LEFT MOST DIGS.
 06234 5236 JMP FFNORR /FOR NORMALIZED *-(+2000=4,5,6,7)
 06235 5334 JMP FNLP /JUMP SO FFGET AND PUT ARE ORGED RIGHT

06236 3041 FFNORR, DCA AC1 /DONE W/NORMALIZE-CLEAR AC1
 06237 5615 JMP I FFNOR /RETURN
~~06240 6057 ALIP, AL1~~

>
 IFNZRO EAE <
 ENPUNCH

/ROUTINE TO NORMALIZE THE FAC

*6215
 FFNOR, 0
 CDF /CHANGE D.F. TO FIELD OF PACKAGE
 SWAB /FORCE MODE B
 OLD /PICK UP MANTISSA
 ACH
 SWP /PUT IT IN CORRECT ORDER
 NMI /NORMALIZE IT
 SNA /IS THE # ZERO?
 DCA ACX /YES=INSURE ZERO EXPONENT
 DCA ACH /STORE HIGH ORDER BACK
 SWP /STORE LOW ORDER BACK
 DCA ACLO
 CLA SCA /STEP COUNTER TO AC
 CMA IAC /NEGATE IT
 TAD ACX /AND ADJUST EXPONENT
 DCA ACX
 JMP I FFNOR /RETURN
 NOPUNCH
 >

- FNLP can
 return to NORMLP
 or FFNORR

V3
 V4
 V5A

FALL

/FLOATING GET

```

06241 0000 *6241
06241 0000 FFGET, 0
06242 *555 4523 JMS I PATCHP /WHICH MODE OF CALL
06243 1641 TAD I FFGET /CALLED BY USER-GET ADDR. OF OP
06244 4200 JMS ARGET /PICK UP OPERAND
06245 1047 TAD OPX
06246 3044 DCA ACX /LOAD THE OPERAND INTO FAC
06247 1051 TAD OPL
06250 3046 DCA ACLO
06251 1050 TAD OPH
06252 3045 DCA ACH
06253 2241 ISZ FFGET
06254 6201 CDF
06255 5641 JMP I FFGET /RETN. TO CALL +2

```

/FLOATING PUT

```

06256 0000 FFPUT, 0
06257 4555 4523 JMS I PATCHP /WHICH MODE OF CALL?
06260 1656 TAD I FFPUT /CALLED BY USER-GET OPR. ADDR
06261 3241 DCA FFGET /STORE IN A TEMP
06262 1044 TAD ACX /GET FAC AND STORE IT
06263 3641 DCA I FFGET /AT SPECIFIED ADDRESS
06264 4275 JMS ISZFGT /BUMP POINTER, WATCHING FOR FIELD OVERLAP
06265 1045 TAD ACH
06266 3641 DCA I FFGET
06267 4275 JMS ISZFGT
06270 1046 TAD ACLO
06271 3641 DCA I FFGET
06272 2256 ISZ FFPUT /BUMP RETN.
06273 6201 CDF
06274 5656 JMP I FFPUT /RETN. TO CALL+2

```

/ROUTINES TO BUMP ARGET AND FPUT POINTERS AND INCREMENT THE
/DATA FIELD IF THE POINTER CROSSES A FIELD BOUNDARY

```

06275 0000 ISZFGT, 0
06276 2241 ISZ FFGET /BUMP POINTER
06277 5675 JMP I ISZFGT /NO SKIP MEANS JUST RETURN
06300 7410 SKP /SKIP MEANS WE HAVE TO INCREMENT DATA FIELD
06301 3275 NEWCDF, DCA ISZFGT /THIS INST EXECUTED ONLY BY ISZAC2
06302 6214 RDP /GET THE DATA FIELD
06303 1307 TAD CDF10 /BUMP BY 1 AND MAKE A CDF
06304 3305 DCA ,+1 /PUT IN LINE
06305 6305
06306 5675 JMP I ISZFGT /RETURN

```

```

06307 6211 CDF10, CDF 10

```

```

06310 0000 ISZAC2, 0
06311 2042 ISZ AC2 /BUMP POINTER
06312 5710 JMP I ISZAC2 /NOTHING HAPPENED

```

/OS/8 BASIC RUNTIME SYSTEM

PAL8-V7 10/24/72 PAGE 129-1

06313 1310
06314 5301

TAD ISZAC2
JMP NEWCOF

/NEED NEW DF. GET RETURN ADDR
/AND BUMP DF

IFZERO EAE <

/ROUTINE TO ADJUST QUOTINET OF FIRST DIVIDE (MAYBE) WHEN THE
 /REMAINDER OF THE FIRST DIVIDE IS LESS THAN QUOT*OPL
 /USED BY FLTG. DIVIDE ROUTINE

06315	7041	DVOPS,	CMA	IAC	/NEGATE AND STORE REVISED REMAINDER
06316	3045		DCA	ACH	
06317	7100		CLL		
06320	1050		TAD	OPH	
06321	1045		TAD	ACH	/WATCH FOR OVERFLOW
06322	7420		SNL		
06323	5330		JMP	DVOP1	/OVERFLOW-DON'T ADJUST QUOT. OF 1ST DIV.
06324	3045		DCA	ACH	/NO OVERFLOW-STORE NEW REM.
06325	7040		CMA		/SUBTRACT 1 FROM QUOT OF
06326	1041		TAD	AC1	/FIRST DIVIDE
06327	3041		DCA	AC1	
06330	7300	DVOP1,	CLA	CLL	
06331	1045		TAD	ACH	/GET HI ORD OF REMAINDER
06332	5733		JMP I	DVOP2P	/GO ON
06333	5535	DVOP2P,	DVOP2		
06334	7160	FNLP,	CLL CML	CMA	/-1
06335	1044		TAD	ACX	/SUBTR. 1 FROM EXPONENT
06336	3044		DCA	ACX	
06337	4640		JMS I	ALIP	/SHIFT FAC LEFT 1
06340	5225		JMP	NORMLP	/GO BACK AND SEE IF NORMALIZED
06341	3044	ZEXP,	DCA	ACX	
06342	5236		JMP	FFNORR	

/
/FSQUARE-SQUARE FAC-CALLS MULTIPLY TO MUL, FAC BY ITSELF
/

```

6347      *6347
A,
06347 0000 FFSQ, 0
06350 4753      JMS I   TMPY   /CALL MULTIPLY TO MULTIPLY
06351 0044      ACX          /FAC BY ITSELF
06352 5747      JMP I   FFSQ   /DONE
06353 5600      TMPY, FFMPY
/
/
06354 4516 00,   JMS I ERROR /OVERFLOW
06355 4516 DV,   JMS I ERROR /DIVISION ERROR
06356 4537      JMS I FCLR   /RETURN 0 IN FAC
06357 5513      JMP I ILOOPL
06360 4516 LM,   JMS I ERROR /ILLEGAL ARGUMENT

```

/ROUTINE TO COMPARE 2 ABSOLUTE 12 BIT INTEGERS TO DETERMINE WHICH IS
/THE LARGER. WHEN CALLED, NUMBER "A" IS IN THE AC, NUMBER "B" IS IN
/CORE LOCATION BABS ON PAGE 0. RETURNS TO CALL+1 IF B>=A, CALL+2 IF
/A>B

```

06361 0000 NUMCOL, 0
06362 3347 DCA A          /SAVE A
06363 1015 TAD BABS       /GET B
06364 7710 SPA CLA       /B>0?
06365 5374 JMP BNEG      /NO-SPECIAL CASE CHECK
06366 1347 MOFTEN, TAD A  /YES-GET A AGAIN
06367 7141 CLL CIA       /NEGATE
06370 1015 TAD BABS       /IF B>=A, THERE WILL BE A CARRY
06371 7620 SNL CLA
06372 2361 ISZ NUMCOL     /B<A
06373 5761 JMP I NUMCOL
/
06374 1347 BNEG, TAD A    /GET A AGAIN
06375 7710 SPA CLA       /IA A NEG ALSO?
06376 5366 JMP MOFTEN    /YES-USE THE CODE ABOVE TO COMPARE
06377 5761 JMP I NUMCOL  /NO-THEN B> A

```

6361

Ø

2

1563

TAD I INWORD Ø

3

7112

CLL RTR

4

7620

SNL CLA

5

4550

JMS I FTYPE

6

5513

JMP I ILOOP

6400

+OVERLAY+3000

/TELETYPE "DRIVER"-WHEN CALLED,GRABS CHARACTERS FROM THE
 /TELETYPE UNTIL A CR IS SENT OR THE BUFFER IS FULL. ASSUMES TTY ENTRY
 /IS IN I/O WORK AREA.

06400	0000	TTYDRI, 0	
06401	7410	SKP	/CRLF ONLY NECESSARY ON FLUSH
06402	4553	LFLUSH, JMS I CRLF	/PRINT A CR,LF
06403	1253	TAD K277	/PRINT A ? SIGNIFYING WAIT FOR INPUT
06404	4522	JMS I XPUT	
06405	1564	TAD I WORD1	/BUFFER ADDRESS
06406	3566	DCA I WORD3	/INITIALIZE POINTER TO START OF BUFFER
06407	4546	JMS I CNOCLL	/INITIALIZE CHAR # TO 1
06410	4510	TTYIN, JMS I PRINT	/EMPTY TTY BUFFER BEFORE AWAITING INPUT
06411	5210	JMP ,=1	
06412	1252	TAD K5252	/DESIGN INTO AC
06413	6031	KSPA, KSF	/CHAR READY?
06414	5321	JMP SPIN	/NO-DIDDLE WHILE WE WAIT
06415	7300	CLA CLL	/FLUSH SPINNER OUT OF AC
06416	1077	TAD K0200	/FORCE PARITY BIT
06417	6034	KRS	/GET CHAR
06420	3053	DCA CHAR	/SAVE
06421	1053	TAD CHAR	
06422	4522	JMS I XPUT	/ECHO IT
06423	6032	KCC	/CLEAR KEYBOARD FLAG AND SET READER RUN
06424	1053	TAD CHAR	
06425	1246	TAD MCTRLU	/IS IT CTRL/U?
06426	7650	SNA CLA	
06427	5202	JMP LFLUSH	/YES-START AGAIN
06430	1053	TAD CHAR	/NO
06431	1250	TAD CRUBOT	/IS IT RUBOUT?
06432	7450	SNA	
06433	5254	JMP BACKUP	/YES-BACK UP BUFFER POINTER
06434	1247	TAD MCR	/NO-IS IT CR?
06435	7650	SNA CLA	
06436	5306	JMP CR	/YES-DONE
06437	1053	TAD CHAR	
06440	4545	JMS I PACKL	/PACK CHAR IN BUFFER
06441	4547	JMS I BUFCHL	/BUFFER FULL?
06442	5651	JMP I IOLK	/YES-ERROR
06443	7000	NOP	/NO-CHAR 3 LEFT
06444	7000	NOP	/NO-2 AND 3 LEFT
06445	5210	JMP TTYIN	/NO-NEXT CHAR
06446	7553	MCTRLU, -225	
06447	0162	MCR, 377-215	
06450	7401	CRUBOT, -377	
06451	6674	IOLK, IO	
06452	5252	K5252, 5252	
06453	0277	K277, 277	

06454 1566 BACKUP, TAD I WORD3 /BUFFER POINTER

06455	7041		CIA	/NEGATE
06456	1564		TAD I WORD1	/COMPARE AGAINST START OF BUFFER
06457	7650		SNA CLA	/BUFFER EMPTY?
06460	5210		JMP TTYIN	/YES-THERE IS NOTHING TO RUBOUT
06461	1273		TAD K334	
06462	4522		JMS I XPUT	/ECHO "\"
06463	4551		JMS I CHRNL	/GET CHAR # OF NEXT CHAR (LAST #+1)
06464	5274		JMP C1B	/1
06465	5301		JMP C3B	/3
06466	4546		JMS I CNOCLL	/IT WAS 2-MAKE IT 1
06467	7240	PBACK,	CLA CMA	/-1
06470	1566		TAD I WORD3	/BACK UP BUFFER POINTER
06471	3566		DCA I WORD3	
06472	5210		JMP TTYIN	/NEXT CHAR
06473	0334	K334,	334	
06474	1563	C1B,	TAD I WORD0	
06475	0105		AND K7477	
06476	1077		TAD K0200	/IT WAS 1-MAKE IT 3
06477	3563		DCA I WORD0	
06500	5210		JMP TTYIN	/NO NEED TO BACK UP POINTER
06501	1563	C3B,	TAD I WORD0	
06502	0105		AND K7477	
06503	1076		TAD K0100	/IT WAS 3,MAKE IT 2
06504	3563		DCA I WORD0	
06505	5267		JMP PBACK	/BACK UP POINTER
06506	4553	CR,	JMS I CRLF	/ECHO A CR,LF
06507	1320		TAD K4	
06510	1200		TAD TTYDRI	/BUMP DRIVE RETURN TO NORMAL
06511	3200		DCA TTYDRI	
06512	1053		TAD CHAR	
06513	4545		JMS I PACKL	/PACK CHAR IN BUFFER
06514	1564		TAD I WORD1	
06515	3566		DCA I WORD3	/INITAILZE BUFFER POINTERS
06516	4546		JMS I CNOCLL	
06517	5600		JMP I TTYDRI	/RETURN
06520	0004	K4,	4	
06521	2017	SPIN,	ISZ SPINNR	/SPIN RANDOM # SEED
06522	7410		SKP	
06523	7064		CMA CML RAL	/MARCH TO THE LEFT
06524	5213		JMP KSFA	/CHECK FOR CHAR YET

/SUBROUTINE FBITGT-ROUTINE TO PUT FUNCTION BITS FROM INSTRUCTION INTO AC

```
06525 0000 FBITGT, 0
06526 1064      TAD INSAV
06527 7112      CLL RTR
06530 7012      RTR          /PUT FUNCTION BITS IN BITS 8-11
06531 0074      AND K0017    /MASK THEM OFF
06532 5725      JMP I FBITGT  /RETURN
```

/GOSUB POP ROUTINE-ROUTINE TO POP ELEMENT OFF GOSUB STACK

```
06533 0000 POPG, 0
06534 1067      TAD GSP      /GET GOSUB STACK POINTER
06535 1347      TAD MSTTOP    /COMPARE AGAINST TOP OF STACK
06536 7710      SPA CLA      /ATTEMPT TO POP OF EMPTY STACK?
06537 4516      GR, JMS I ERROR /YES-RETURN WITHOUT A GOSUB
06540 1467      TAD I GSP     /GET TOP STACK ELEMENT
06541 3040      DCA TEMP1     /SAVE
06542 7240      CLA CMA      /=1 IN AC
06543 1067      TAD GSP      /BACK UP GOSUB STACK POINTER
06544 3067      DCA GSP
06545 1040      TAD TEMP1     /GET POPPED ELEMENT IN AC
06546 5733      JMP I POPG    /RETURN
06547 7325      MSTTOP, -GSTCK
```

/GOSUB RETURN

```
06550 4333 RETRNI, JMS POPG    /POP PC OFF GOSUB STACK
06551 7001      IAC          /BUMP OVER SECOND WORD OF GOSUB INST
06552 3774      DCA I INTPLK  /USE AS NEW PSEUDO-PC
06553 4333      JMS POPG     /POP CDF OFF STACK
06554 3515      DCA I CDFPSL  /PUT IN LINE IN PWFECH
06555 5513      JMP I ILOOPL  /RETURN TO ILOOP
```

/DATA LIST READ (NUMERIC)

```
06556 4765 RDLIST, JMS I DLRELK /FETCH WORD FROM LIST
06557 3044      DCA EXP      /STORE AS EXPONENT
06560 4765      JMS I DLRELK
06561 3045      DCA HORD      /HIGH MANTISSA
06562 4765      JMS I DLRELK
06563 3046      DCA LORD      /LOW MANTISSA
06564 5513      JMP I ILOOPL
06565 2311      DLRELK, DLREAD
```

/SUBROUTINE FTYPE-RETURNS TO CALL+1 IF FILE NUMERIC,CALL+2 IF ASCII

```
06566 0000 FTYPE, 0
06567 1563      TAD I WORD0   /GET HEADER
06570 7110      CLL RAR      /TYPE TO LINK
06571 7630      SZL CLA      /IS IT NUMERIC?
06572 2366      ISZ FTYPE     /NO-BUMP RETURN
06573 5766      JMP I FTYPE   /RETURN
```

/05/8 BASIC RUNTIME SYSTEM

PAL8-V7 10/24/72 PAGE 133-1

06574 0302 INTPLK, INTPC

6600

PAGE

/LAST PAGE OF BRTS-CONTAINS SAC,I/O TABLE, AND SOME MISCELLANEOUS CODE
 /*****
 /TELETYPE INPUT BUFFER (74 CHARACTERS LONG)
 /THIS BUFFER CONTAINS ONCE ONLY START CODE WHEN LOADED

```

TTYBUF,
06600 1025 START4, TAD CDFPS      /DF FOR BOTTOM OF PSEUDO-CODE
06601 1242      TAD MCDP1      /COMPARE TO A CDF 10
06602 7640      SZA CLA        /DO THEY MATCH?
06603 5513      JMP I ILOOPL    /NO-ALL BUFFERS ARE FREE-START INTERPRETER
06604 1026      TAD PSSTRY
06605 3015      DCA BABS
06606 1102      TAD K0400
06607 4405      JMS I NUMCOM    /IS START OF PSEUDO-CODE BELOW 400
06610 5213      JMP CHKB2      /NO-CHECK FOR 1000
06611 1074      TAD K0017      /YES-SET ALL BUFFERS BUSY
06612 5237      JMP BAS
06613 1026 CHKB2, TAD PSSTRY
06614 3015      DCA BABS
06615 1246      TAD C1000
06616 4405      JMS I NUMCOM    /IS START OF PSEUDO-CODE BELOW 1000
06617 5222      JMP CHKB3      /NO-CHECK 1400
06620 1245      TAD C16        /YES-ONLY BUFFER 1 IS AVAILABLE
06621 5237      JMP BAS
06622 1026 CHKB3, TAD PSSTRY
06623 3015      DCA BABS
06624 1247      TAD C1400
06625 4405      JMS I NUMCOM    /IS START OF CODE BELOW 1400?
06626 5231      JMP CHKB4      /YES-CHECK 2000
06627 1244      TAD C14        /YES-ONLY BUFFER 1 AND 2 AVAILABLE
06630 5237      JMP BAS
06631 1026 CHKB4, TAD PSSTRY
06632 3015      DCA BABS
06633 1243      TAD K2000
06634 4405      JMS I NUMCOM    /IS CODE START BELOW 2000?
06635 5513      JMP I ILOOPL    /NO-START INTERPRETER-ALL BUFFER FREE
06636 1073      TAD K0010      /YES-BUFFERS 1,2, AND 3 AVAILABLE
06637 3036 BAS,  DCA BMAP
06640 5513      JMP I ILOOPL    /START INTERPRETER
06641 0000      0
06642 1567 MCDP1, -6211
06643 2000 K2000, 2000
06644 0014 C14, 14
06645 0016 C16, 16
06646 1000 C1000, 1000
06647 1400 C1400, 1400
06650 0000      0
06651 0000      0
06652 0000      0
06653 0000      0
06654 0000      0
06655 0000      0

```

```
06656 0000      0
06657 0000      0
06660 0000 TTYEND, 0
0103      KM400=K7400
```

/*****

```
/SUBROUTINE CHARNO=RETURNS TO CALL+1 IF CHAR #=1,CALL+2 IF 3,CALL+3
/IF 2
```

```
06661 0000 CHARNO, 0
06662 1563 TAD I WORD0 /HEADER
06663 0273 AND K300 /ISOLATE CHAR #
06664 7106 CLL RTL
06665 7006 RTL /CHAR # TO BITS 0,1
06666 7540 SMA SZA /IS IT 2?
06667 2261 ISZ CHARNO /YES=BUMP RETURN
06670 7640 SZA CLA /IS IT 2 OR 3?
06671 2261 ISZ CHARNO /YES=BUMP RETURN
06672 5661 JMP I CHARNO /RETURN
06673 0300 K300, 300
```

/ERROR MESSAGE FOR TTY INPUT OVERFLOW

```
06674 4516 ID, JMS I ERROR /LINE FULL
06675 5676 JMP I .+1 /FLUSH BUFFER AND TRY AGAIN
06676 6402 LFLUSH
```


6677

*OVERLAY+3277

////////////////////////////////////
//////// I/O TABLE 5 13-WORD ENTRIES //////////
////////////////////////////////////

06677	0001	TTYF,	1	/TELETYPE ENTRY-FILE IS ASCII
06700	6600	TTYBUF		/BUFFER ADDRESS
06701	0000	0		/CURRENT BLOCK IN BUFFER
06702	6600	TTYBUF		/READ WRITE POINTER
06703	6400	TTYDRI		/HANDLER ENTRY
06704	0000	0		
06705	0000	0		
06706	0000	0		
06707	0000	0		
06710	0000	0		
06711	0000	0		
06712	0000	0		
06713	0000	0		
06714	0000	FILE1,	0	/FILE #1
06715	0000	0		
06716	0000	0		
06717	0000	0		
06720	0000	0		
06721	0000	0		
06722	0000	0		
06723	0000	0		
06724	0000	0		
06725	0000	0		
06726	0000	0		
06727	0000	0		
06730	0000	0		
06731	0000	FILE2,	0	/FILE #2
06732	0000	0		
06733	0000	0		
06734	0000	0		
06735	0000	0		
06736	0000	0		
06737	0000	0		
06740	0000	0		
06741	0000	0		
06742	0000	0		
06743	0000	0		
06744	0000	0		
06745	0000	0		
06746	0000	FILE3,	0	/FILE #3
06747	0000	0		
06750	0000	0		
06751	0000	0		
06752	0000	0		
06753	0000	0		
06754	0000	0		
06755	0000	0		

06756	0000	0
06757	0000	0
06760	0000	0
06761	0000	0
06762	0000	0
06763	0000	0
06764	0000	0
06765	0000	0
06766	0000	0
06767	0000	0
06770	0000	0
06771	0000	0
06772	0000	0
06773	0000	0
06774	0000	0
06775	0000	0
06776	0000	0
06777	0000	0

FILE4,

/FILE #4

S

A	6347	ATANA1	4427	CHAIN	3600	DA	2316
ABSV	3575	ATANA2	4435	CHAR	0053	DADP	0060
ABSVL	2366	ATANA3	4443	CHARNO	6661	DATABA	3666
ABSVLL	1376	ATANB0	4424	CHAR3P	2746	DATCOM	3615
ACH	0045	ATANB1	4432	CHAR3U	3064	DATE	3600
ACL	0046	ATANB2	4440	CHECKL	3461	DATTAB	4526
ACLO	0046	ATANB3	4446	CHKB2	6613	DBAD	6115
ACSR	6072	A999	2514	CHKB3	6622	DBAD1	5534
ACSRPT	4704	BABS	0015	CHKB4	6631	DBAD1P	6175
ACX	0044	BACKUP	6454	CHR	3400	DCNT	5137
AC0	0040	BAS	6637	CHRNOL	0151	DCNTP	4726
AC1	0041	BCGET	3035	CHRSV	0040	DE	0557
AC2	0042	BCGETL	2776	CI	3622	DECNV	5213
ADCALC	0640	BCNT	1030	CL	3644	DECON	5214
ADFC	0707	BCPUT	0747	CLEANP	4260	DECONV	5207
ADFW	4747	BCPUTL	3034	CLENG	3440	DECON1	5335
AJT	0714	BEND	1137	CLF	3744	DEVCAL	0124
AL1	6057	BLINIT	3361	CLOSE	3405	DEVNAL	4332
AL1K	3774	BLREAD	3344	CLOSED	3442	DEVNA1	4027
AL1P	6240	BMAP	0036	CNOBMK	2765	DEVNA2	4030
AL1PP	5140	BNEG	6374	CNOBML	5573	DFETCH	4103
AL1PT	4705	BO	4215	CNOBMP	3104	DGTYP	5076
AL1PTR	5642	BOUTRS	1064	CNOCLL	0146	DGTYP	4725
AMODE	2324	BSTRT	1070	CNOCLR	3016	DIG1	4134
ANDLST	3401	BSTRTA	1027	COMBNE	2766	DIG1A	3745
ANDLS2	3542	BWL	0144	COMLOP	2535	DIG2	4135
ANDPTR	3400	BWP	2240	COMMA	2517	DIG3	4136
AN1	3775	BUFASS	4237	COMMAP	3250	DIG4	4137
AN2	3776	BUFCHK	2706	COMMAS	2557	DIG5	4140
AN2PTR	3541	BUFCHL	0147	COS	4053	DISIN	3634
AP0001	2175	BUFIN	1025	COUNCK	1363	DLCDF	2317
ARGET	6200	BUFOUT	1026	CPLOOP	3234	DLCDFL	1176
ARGETK	5503	B1	4234	CR	6506	DLPTR	0016
ARGETL	5444	B2	4227	CREAD	3671	DLREAD	2311
ARGETP	6127	B3	4222	CRETN	3516	DLREAL	2416
ARGPLK	4307	CAD	4506	CRFUNC	0373	DLRELK	6565
ARGPLL	3511	CBLK	3676	CRLF	0153	DLSTOP	0027
ARGPOL	4220	CCC240	3460	CRLFR	2347	DLSTRT	0030
ARGPRE	0307	CCR	3743	CRREP	1360	OMAP	0035
ARGPRL	1420	CC13	4153	CRUBOT	6450	DNA1	3617
ARITHA	1531	CC16	4414	CSFN	2001	DNA2	3620
ARJMP	0711	CC240	3747	CSMOVE	4143	DNUMBR	5305
ARRAYI	0600	CC3	4331	CSTA	4162	DU	4055
ARSTRT	0022	CC4	4330	CSTAC	4163	DOADD	6023
ARTRAP	4365	COFINL	0541	C18	6474	DONA	6027
ASC	3407	COFIO	0020	C1000	6646	DONE	3765
ASCHR	3123	COFPS	0025	C13	4531	DONEF	5066
ASCNDE	3244	COFPSL	0115	C14	6644	DRAP	4132
ASCOLK	3457	COFPSU	0206	C1400	6647	DRARG1	0551
ASCON	3660	COF0	0761	C16	6645	DRARG2	0552
ASCOUL	3216	COF000	4336	C20	4333	DRARG3	0553
ASCOUT	1260	COF10	6307	C240	2565	DRCALL	0537
ATABDF	0610	COF20	4562	C3	4070	DRERR	0556
ATABDL	1173	COIN	3621	C3B	6501	DRIVRL	4134
ATAN	4200	CFETCH	1322	C4	4140	DRIVRN	4200

DS	4411	FACCLR	0365	FFNEG	6135	FNEGL	0140
DSWIT	0052	FACR	6031	FFNEGA	5410	FNLP	6334
DT1	5101	FACREL	0133	FFNEGK	5501	FNOM	4304
DV	6355	FACRES	2373	FFNEGP	5303	FNORL	0136
DVLP1	5711	FACRSAL	0132	FFNEGR	5773	FNORP	6176
DVL1	5514	FACSAV	3371	FFNGP	4736	FO	1637
DVOPS	6315	FADDL	4060	FFNOR	6215	FOTYPE	2357
DVOPSP	5533	FAUDLK	3473	FFNORR	6236	FOTYPL	3343
DVOP1	6330	FADDLL	5312	FFORMT	1302	FOUT1	4625
DVOP2	5535	FADDM	4357	FFOUT	4600	FOUT2	4634
DVOP2P	6333	FAD1	6004	FFOUTL	1373	FOUT3	4674
DVOVR	5717	FATCHK	4047	FFPUT	6256	FOUT4	4667
DVTRAP	3574	FB	4012	FFPUTP	0135	FPPTM1	1175
DV1	5765	FBITGT	6525	FFSIN	4000	FPPTM2	1172
DV2	5755	FBITS	0117	FFSQ	6347	FPPTM3	1167
DV24	5745	FC	3441	FFSUB	6117	FPPTM4	1164
DV24P	5532	FCLR	0137	FFSUB1	5400	FPPTM5	1161
DV24PT	5136	FCNT	5275	FGETL	0134	FPUTL	0135
EAE	0000	FD	5527	FGETM	0134	FPUTLL	0722
EBC	2730	FDDON	5742	FI	2307	FPUTM	0135
EBLK	3740	FDDONP	5531	FIDLE	0123	FRACT	4071
EDBLK	3747	FDIVL	4062	FIDVP	3631	FRANDM	2342
EDON	5260	FDIVM	4360	FIGO2	5241	FREE70	4062
EF	0560	FDIV1L	4067	FILEFA	1533	FREE72	4073
EFATAL	4254	FDIV1M	4361	FILE1	6714	FR00T	3646
EFLG	0056	FDVPT	5311	FILE2	6731	FSQRL	4065
EM	3615	FD1	5504	FILE3	6746	FSQRM	4364
EMDONE	3600	FD1P	5721	FILE4	6763	FSTOP	3704
EMDONL	3570	FE	4112	FILSTR	3316	FSTOPI	0563
EMESS	4117	FERRLP	4020	FILSTU	4321	FSTOPN	0561
EMLOOP	3545	FETPAG	4111	FISUBL	4367	FSTOP1	0161
EN	4307	FF	0037	FIXDNE	4525	FSUBL	4064
ENTLOK	4275	FFADD	6000	FIXL	4066	FSUBLK	1371
ENTNO	0162	FFADP	4733	FIXLP	4515	FSUBLL	3573
ENTRYN	4031	FFATN	4200	FJOCI	0400	FSUBM	4362
ENVAL	3523	FFCOS	4053	FLOW	0057	FSUB1L	4063
EOFSEL	0143	FFDIV	5722	FLEN	4306	FSUB1M	4363
EOFSET	2252	FFDIV1	5412	FLING	5142	FSWITC	2060
ERRDIS	1460	FFDP	5446	FLINK	4722	FTCOM	2424
ERRET	4057	FFOVP	4734	FLN	3643	FTRPRT	4505
ERROR	0116	FFD1	5726	FLOABL	4505	FTYL	3015
ERRORR	4013	FFEXP	4120	FLOATB	3420	FTYPE	6566
ESTRA	4143	FFEXPL	3625	FLOATL	0141	FTYPL	0150
ESTRNG	4121	FFGET	6241	FLOATM	0141	FTYPSE	3531
ETAB	4156	FFIN	5200	FLOATS	4144	FUJUMP	1523
ETABA	4154	FFINL	3125	FLOOK	4253	FUNC1I	1473
ETLOP	4042	FFINLK	3477	FM	1624	FUNC2I	1472
EXP	0044	FFIN1	5232	FMPYL	3572	FUNC3I	1542
EXPA0	4421	FFIX	4500	FMPYLK	4061	FUNC4I	1466
EXPA1	4416	FFLOAT	4533	FMPYLL	5310	FUNC5I	1471
EXPB1	4413	FFLOG	4263	FMPYLV	3627	GCHR	5322
EXPON	3477	FFLOGL	3626	FMPYM	4356	GO	5025
EXPONK	3630	FFMPP	4735	FN	2005	GETCH	3126
EXPON1	4120	FFMPY	5600	FNAP	3437	GETCHG	3502
E20P10	1307	FFMT	4613	FNEGI	1226	GETCHL	0142

GETE	5250	JMSI4	1421	K0377	0101	LDHPR	4502
GKNT	5003	JMSI5	1434	K0400	0102	LDMPST	2143
GON	3761	JMSI7	1545	K0700	3672	LDHPSW	4503
GOSUB	0415	JMSSI	0306	K1000	4327	LDHR	2705
GR	6537	JMSTAD	1535	K110	2236	LDHRST	0157
GS	2335	JMSUSR	1557	K1400	4326	LDHSWT	2704
GSP	0067	JNEG	3466	K16	4721	LEN	3414
GSSTOP	0476	JSL	2627	K20	3272	LEV	5477
GSTCK	0453	JUSNEG	3464	K200	0077	LFLUSH	6402
GSTCKT	2341	KC	3752	K2000	6643	LINEHI	0065
GTFLG	4200	KC240	1151	K212L	2356	LINEI	1140
GT1FLG	4263	KEX	3742	K215	0054	LINELO	0066
HOOKL	4007	KFD1	5447	K232	3530	LK7607	3746
HORD	0045	KKM10	3676	K240	3204	LM	6360
IA	1465	KKM12	5775	K260	4065	LMAKE	4067
IDLE	2304	KK12	5313	K277	6453	LMAKEL	3741
IF	4424	KK13	4330	K300	6673	LN2	4470
IGS	3707	KK2000	4161	K334	6473	LN20V2	4410
ILOOP	0212	KK40	3271	K4	6520	LOADDF	3165
ILOOPF	0240	KK7	4730	K40	0767	LOG	4263
ILOOPPL	0113	KK7600	1253	K4207K	4363	LOGC1	4454
IN	4032	KL7600	3745	K4210	3360	LOGC3	4457
INPTCL	3500	KME	5306	K5252	6452	LOGC5	4462
INPUT	5347	KM10	0250	K5700	3674	LOG2E	4405
INSAV	0064	KM12	5135	K6	4740	LOOP	5052
INSC	3413	KM13	5776	K6000	3772	LOP01	3722
INT	3400	KM144	5134	K6213K	4364	LOP02	3746
INTERB	1152	KM20	4731	K6222	4567	LOP1	6075
INTL	0114	KM22	3777	K6223	4565	LOP2	6037
INTPC	0302	KM4	2031	K7000	4135	LORD	0046
INTPCK	1175	KM40	0106	K7200	4142	LRESET	0567
INTPCL	0444	KM400	0103	K73	4563	LRSCOM	2655
INTPLK	6574	KM7	4732	K7400	0103	LSUB1I	1403
INTPOS	3420	KNOP	4012	K7435	1665	LSUB2I	1413
INTRB	1372	KNT	4737	K7437	3370	LS1I	1400
IO	6674	KNTP	5154	K7477	0105	LS2I	1410
IOLK	6451	KSFA	6413	K7506	5346	LTRPRT	4302
IOTAB	2035	KUPARO	3751	K7554	0431	L0037	4137
IOTPTR	2034	K0	4412	K7577K	4365	L7466	3667
IOUT	5360	K0003	3533	K7600	0432	L7600	0366
ISZAC2	6310	K0004	2365	K7607K	3703	L7600K	4370
ISZFGT	6275	K0007C	3675	K7700	0104	L7605K	3670
JEOFI	0445	K0010	0073	K7737	0770	L7605P	4371
JFAIL	0413	K0017	0074	K7760	3666	L7607	1527
JFAILL	2053	K0037	0742	K7773K	4366	L7620	3665
JFOR	2042	K0037C	3671	LASTB	3332	L7621	3667
JFORL	0443	K0037L	1755	LDH	0131	L7642	4566
JMPFIL	1450	K0057	3673	LDHC	2105	L7644	3664
JMPI	0247	K0077	0075	LDHCDF	2237	L7666	3670
JMPISA	0736	K0100	0076	LDHCDL	4506	L7721	4570
JMPI5N	4012	K0177	1776	LDHDC	0573	L7727	4571
JMPI2	0746	K0200	0077	LDHDF	2647	L7746	0357
JMPI3	1205	K0210	3205	LDHINI	2676	MAKED	4061
JMPI6	1604	K0300	4066	LDHINL	0127	MASKL	3427
JMSI	0246	K0340	0100	LDHL	2646	MBEND	1031

MCC	3750	NAMEG	4416	OPSR	6034	PUTCH	3253
MCDF1	6642	NAMEGL	4141	OPX	0047	PUTCHL	0112
MCOLON	4400	NCG	4450	OTRPA	4532	PUTCHR	1010
MCR	6447	NCGS	4464	OUT	5144	PWFEC	0200
MCRMAL	3456	NCHK	4113	OUTDG	5150	PWFEC	0120
MCSPE	4401	NCHKL	4262	OUTDGP	4720	PZR	5015
MCTRLC	1032	NEWCDF	6301	OVADD	1512	P1CDF	1240
MCTRLU	6446	NEXRCK	4274	OVDNE	1516	P1CDF1	1245
MCTRLZ	3103	NEXREC	3302	OVERLA	3400	P1SWAP	0156
MDECPT	1375	NEXREL	0152	OVML	0346	P2CDF	1243
MDNE	3227	NFLAG	4120	OVLAY	1530	P2CDFL	4563
MDONE	5627	NFLGST	4111	OD	6354	P2CDF1	1247
MDSET	5450	NGT	4257	PA	4415	P2CDL1	4564
MDSETK	5774	NHNDLE	4103	PACKCH	2735	P2SWAL	3744
MDSETP	5445	NHNDLL	4261	PACKL	0145	P200	5211
MDV	5307	NOCC	1044	PATCHF	5364	QUAD2	4017
MD1	5452	NOCTC	3725	PATCHP	0155	QUAD3	4022
MD1P	5443	NOCZ	3417	PBACK	6467	QUAD4	4024
MFATAL	4155	NOP1	6111	PDNE	3217	RDLIST	6556
MINUS	5363	NOP2	6053	PDP	5001	RE	3010
MINUSP	3461	NORMLP	6225	PINFO	0361	READFL	3000
MINUS4	0360	NOTE	4743	PIOV2	4402	READFW	3147
MIN4	4367	NTTY	3135	PLUS	5362	READI	3105
MK61	2727	NULLST	4365	PNT	1770	RECP2	1066
MLF	2456	NUM	4113	POLYNL	4304	RELCOM	3512
MML	4501	NUMCOL	6361	POLYSN	4026	RESOLS	2567
MMM4	4415	NUMCOM	0005	POPG	6533	RESTI	1650
MM110	2513	N1	4403	POS	4400	RESTIL	2577
MM260	3746	N1A	4542	POSITN	4501	RESTOR	2572
MM4	3542	N2	4404	POSPTA	2560	RETMDL	3665
MODESW	0063	N3	4405	POSSET	4424	RETMOD	3526
MOFTEN	6366	N3A	4402	POSTP	2561	RETPOS	4471
MPLP	5653	N4	4406	POVTAB	0362	RETRNI	6550
MPLP1	5654	N5	4407	PR	5016	RETRNO	3571
MPLP2	5666	N6	4410	PROCP	4776	RETRN1	3610
MPLUS	3251	N6A	4543	PROCPP	5143	RET0	3565
MPY	2260	N7666	3613	PREST	3715	RIGHTL	2665
MPYLNK	0121	OADD	6157	PRINT	0110	RIGHTS	2633
MP12L	5701	OADDP	5141	PRNTX	5160	RND	4543
MP12LP	2265	OATADI	1526	PRNTXP	4723	RONLY	3314
MP24	5643	OCDF	2146	PRNTX1	5164	RSEED	2346
MRUBOT	3301	OE	1513	PRZRO	5172	RSEEDL	4562
MSPACE	1216	ONE	3474	PRZROP	4724	RTN2	5371
MSTTOP	6547	ONEHAF	4465	PS	5023	RTZRO	5620
M1	4727	ONERET	4473	PSFLAG	0031	RWONC	3312
M1R	3470	ONE1	4476	PSSTRT	0026	SAC	0321
M110	3252	OPENAF	4001	PSTCHK	3476	SACCHK	2132
M13	4530	OPENAV	4000	PSWAP	1230	SACEM	2216
M14	0107	OPENNF	4004	PSWAP2	4334	SACL	3421
M215	0055	OPENNV	4003	PSWP2P	4544	SACPTR	0111
M240	5361	OPERI	1200	PS1L	0363	SAC40C	2116
M260	1374	OPH	0050	PS2L	0364	SAD	0735
M4	2566	OPL	0051	PTRBMP	2166	SAFIND	1747
M50	1033	OPNEG	6146	PUSHG	2326	SARRAY	0724
M6	4413	OPNEGP	5502	PUSHGL	0442	SASTRT	0024

SC	2232	STARTB	4305	TEMP12	2311	WE	3032
SCALDF	0317	START1	0321	TEMP13	2260	WORD0	0163
SCALDL	1177	START3	1152	TEMP14	0003	WORD0A	2032
SCASE	0243	START4	6600	TEMP15	0004	WORD1	0164
SCDF	1737	STB	3642	TEMP17	3023	WORD1A	2033
SCOMP	2061	STCGTJ	3501	TEMP18	3016	WORD10	0173
SCONTU	4462	STCOM	1700	TEMP19	3371	WORD11	0174
SCON1	2200	STC40C	2134	TEMP2	0007	WORD12	0175
SCSTRT	0021	STDF	1701	TEMP21	2636	WORD13	0176
SDIS	0276	STDFL	1174	TEMP24	0531	WORD14	0177
SE	4706	STFILK	0743	TEMP3	0042	WORD2	0165
SEG	4266	STFIND	1666	TEMP30	0771	WORD3	0166
SEGCML	4370	STFINK	4504	TEMP4	0043	WORD4	0167
SEGCOM	2222	STFINL	0305	TEMP5	0047	WORD5	0170
SEP	5157	STM	0130	TEMP6	0050	WORD6	0171
SEP1	0261	STHCOF	0536	TEMP7	0051	WORD7	0172
SETF	2054	STHCDL	2457	TEN	5317	WRBLK	3350
SE1	4707	STHDF	2603	TM	0043	WRBLKK	3532
SPN	2000	STHOKK	0775	TMPI	6353	WRBLKL	1664
SPNLP	2023	STHINI	2636	TM3	5155	WRITEI	3200
SGN	3632	STHINL	0126	TM3PT	5304	WRITFL	3023
SHLFT	5635	STHL	2600	TOFF	4010	WRITFW	0125
SIGNF	5300	STHR	2645	TOUT	3276	WOPTR	3535
SIN	4000	STHRST	0160	TOVPI	4160	WOPTRA	3534
SINA1	4366	STHSWT	2644	TP	5314	XOGET	4507
SINA3	4371	STOPI	4060	TPH	4136	XDRITE	4523
SINA5	4374	STR	3422	TPREL	3521	XLCOM	4515
SINA7	4377	STRCNT	0071	TPRINT	3677	XPRINT	1034
SL	0525	STRLEN	0032	TP1	5315	XPUT	0122
SLOAD	3150	STRMAX	0070	TRACE	4000	XPUTCH	1000
SLOOP	3713	STRNGA	1532	TRHOOK	1144	XPUT1	1002
SLOVEL	1767	STRPTR	0072	TRREST	4006	XR0	0010
SLOVER	2547	STSLP	3440	TTEST	3143	XR1	0011
SLRCOM	2614	STSTRT	0023	TTYBUF	6600	XR2	0012
SMODE	0271	SU	0623	TTYDRI	6400	XR3	0013
SNEQ	2127	SUBERL	1756	TTYEND	6660	XR4	0014
SNEQ1	2126	SUB0	6125	TTYF	6677	XR5	0015
SPFUNC	1600	SUBOP	5411	TTYIN	6410	ZCNT	3773
SPIN	6521	SUCJML	2052	UNPACK	3047	ZEXP	6341
SPINNR	0017	SUCJMP	0421	UNSFIX	1615	ZMINY	4331
SQRPS	4451	SW	3262	UNSLP	1642	ZR	1342
SR	3130	SWCLP	2477	UNSOUT	1646	ZRCONT	1352
SRCLP	4445	SWITCC	0042	USE	0574	ZROFF	1334
SREAD	2417	SWIT1	0054	USECON	0006	ZRORET	4467
SRESET	0771	SWIT2	0055	USELOG	3613		
SRFIN	2447	SWRITE	2460	USELOL	3567		
SRLIST	2400	S1	0033	USR	0077		
SRLOOP	2411	S2	0034	USRA	1534		
SSAD	4525	TAB	1757	U123C	3056		
SSLOOP	0514	TADTAB	1536	VAL	3462		
SSLP	3160	TDFIXL	4362	VALCNT	3422		
SSMODE	2323	TDONE	3726	VALGET	3503		
SSTEX	0527	TEMP1	0040	VALLK	0154		
SSTORE	0477	TEMP10	0061	VR	3003		
ST	2444	TEMP11	0062	WDONE	3246		



