

000000 .REPT 0

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DFKAC-A-D
PRODUCT NAME: 11/34 EIS INSTRUCTION TESTS
DATE CREATED: DECEMBER 22, 1975
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: GLENN JOHNSON

COPYRIGHT (C) 1975
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102

CONTENTS

1.	ABSTRACT
2.	REQUIREMENTS
2.1	EQUIPMENT
2.2	STORAGE
2.3	PRELIMINARY PROGRAMS
3.	LOADING PROCEDURE
4.	STARTING PROCEDURE
4.1	CONTROL SWITCH SETTINGS
4.2	STARTING ADDRESS
4.3	PROGRAM AND/OR OPERATOR ACTION
5.	OPERATING PROCEDURE
5.1	SWITCH SETTINGS
5.2	SUBROUTINE ABSTRACTS
6.	ERRORS
6.1	ERROR PRINTOUT
6.2	ERROR RECOVERY
7.	RESTRICTIONS
8.	MISCELLANEOUS
8.1	EXECUTION TIME
8.2	STACK POINTER
8.3	PASS COUNTER
8.4	TEST NUMBER
8.5	POWER FAIL
9.	PROGRAM DESCRIPTION

1. ABSTRACT

THIS PROGRAM TESTS THE 11/34 EXTENDED INSTRUCTION SET
<ASH, ASHC, MUL, AND DIV> USING REGISTERS 0-5 AT
LEAST ONCE WITH EACH INSTRUCTION.
THE PROGRAM SHOULD BE RUN FOR
AT LEAST 2 PASSES WITH ALL SWITCHES LOW. THE PROGRAM IS
DESIGNED TO RUN UNDER APT, AND ACT, SYSTEMS.
THIS PROGRAM IS A MODIFICATION OF THE LSI-11 EIS TEST.
IT HAS BEEN MODIFIED TO ACCOUNT FOR ANY LSI-11 - 11/34 DIFFERENCES.

2. REQUIREMENTS

2.1 EQUIPMENT

11/34 STANDARD COMPUTER
AND 4K OF MEMORY

2.2 STORAGE

PROGRAM STORAGE - THE ROUTINES USE MEMORY 0 - 17500

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1 (ALL LOW FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158

159 AFTER LOADING THE PROGRAM IT SHOULD ALWAYS BE STARTED AT 200.
160 IF IT IS DESIRED TO SAVE THE PASS COUNTER THEN THE PROGRAM
161 SHOULD BE RESTARTED AT LOCATION RESTRT [I.E. 222] OTHERWISE THE
162 PROGRAM CAN BE RESTARTED AT 200
163
164
165
166
167

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 STAND ALONE

- 1) LOAD PROGRAM INTO MEMORY USING ABS LOADER.
- 2) SET SWITCHES (SEE SEC 5.1) ALL LOW FOR WORST CASE.
- 3) START AT 200.
- 4) THE PROGRAM WILL LOOP AND "END PASS" WILL BE TYPED AFTER COMPLETION OF FIRST PASS AND EVERY 4TH PASS. HOWEVER TYPE OUT WILL BE SUPPRESSED IF BIT 5 OF LOCATION \$ENVM IS HIGH
- 5) A MINIMUM OF TWO PASSES SHOULD ALWAYS BE RUN.

4.3.2 UNDER APT

LOAD THE PROGRAM AND START AFTER SETTING THE DESIRED SWITCHES (SEE SEC. 5.1).

5. OPERATING PROCEDURE

5.1 SWITCH SETTINGS

IF NO HARDWARE SWITCH REGISTER IS AVAILABLE, THE PROGRAM AUTOMATICALLY USES THE CONTENTS OF LOC. 176 AS THE SOFTWARE SWITCH REGISTER. THE USER SHOULD SET THIS LOCATION BEFORE STARTING THE PROGRAM.

BIT #	OCTAL VALUE	FUNCTION
15	10000	HALT ON ERROR
13	02000	INHIBIT PRINTOUT

AN 8 BIT BYTE \$ENVM [I.E. LOCATION 421] HAS BEEN USED TO DEFINE THE OPERATING MODE. ALL TYPEOUTS CAN BE SUPPRESSED BY MAKING BIT 5 OF BYTE \$ENVM HIGH, IN OTHER WORDS BY PLICING A 20000 IN LOCATION 420

168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205

5.2 SUBROUTINE ABSTRACTS

5.2.1 HALT ROUTINE

THIS ROUTINE CALLED VIA JSR INSTRUCTION IS USED EACH TIME AN ERROR IS SEEN AND AN ERROR MESSAGE OF THE FORMAT GIVEN IN SEC. 6.1 IS TYPED OUT UNLESS SUPPRESSED BY THE SWITCHES DEFINED IN SEC. 5.1

5.2.2 TRAP CATCHER

A " +2" = "HALT" SEQUENCE IS REPEATED FROM 0=776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR +2.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

ADR ERRNM

WHERE:

ADR = ADDRESS OF ERROR
ERRNM = ERROR NUMBER

IN MOST CASES THE COMMENT BESIDE THE CALL FOR HALT SUBROUTINE TELLS WHAT WAS BEING CHECKED AND WHAT WAS EXPECTED. ALL PRINTOUTS WILL BE SUPPRESSED WHEN BIT 5 OF LOCATION \$ENVM IS HIGH. WHILE RUNNING UNDER APT THE DIAGNOSTIC WILL NOT SUPPORT SPOOLING OF CONSOLE OUTPUTS.

6.2 ERROR RECOVERY

RESTART AT 200 OR 222 (SEE SEC 4.2)

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261

262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308

8.1 EXECUTION TIME

NORMALLY FIRST "END PASS" WILL BE TYPED WITHIN 1 SECOND AND WITHIN 10 SECONDS FOR EVERY CONSECUTIVE 400 PASSES

8.2 STACK POINTER

STACK IS INITIALLY SET TO 600

8.3 PASS COUNT

A 16 BIT LOCATION "SPASS" (I.E. LOCATION 406) IS USED TO KEEP PASS COUNT. IT CAN BE CLEARED BY RESTARTING THE PROGRAM AT 200

8.4 TEST NUMBER

A 16 BIT LOCATION "STESTN" (I.E. LOCATION 404) IS USED TO KEEP TRACK OF THE TEST NUMBER, UPPER BYTE OF THIS LOCATION GIVES THE ITERATION NUMBER AND THE LOWER BYTE THE TEST THAT WAS BEING EXECUTED

8.5 POWER FAIL

THE DIAGNOSTIC CAN BE POWER FAILED WITH NO ERRORS. TO USE, START THE TEST AS USUAL AND POWER DOWN THEN UP AT ANY TIME, THE PROGRAM SHOULD RESTART FROM TEST 0 AFTER TYPING "POWER" WITH NO ERRORS. HOWEVER IF THE PROGRAM IS STORED IN A MOS MEMORY THAT CAN NOT HOLD DATA WITH POWER DOWN THEN THE PROGRAM WILL NOT RECOVER FROM A POWER FAIL

9. PROGRAM DESCRIPTION

THIS PROGRAM TESTS ALL THE EIS INSTRUCTIONS OF THE 11/34 FOR ASH AND ASHC INSTRUCTIONS EVERY EVEN PASS IS EXECUTED WITH DESTINATION MODE 0 FOR ALL REGISTERS AND EVERY ODD PASS WITH DESTINATION MODE OF 67. THE DIAGNOSTIC DOES NOT MAKE A PASS WITH T BIT SET.

.ENDR

DFKACA MACY11 3W(1046) 04-AUG-77 14:09 PAGE 7
DFKACA.SRC 18-NOV-75 00:00

309

```

310
311
312          ,ABS
313          ,NLIST MD,MC,CND
314          ,LIST ME
315          ,TITLE DFKACA
316          ,*COPYRIGHT (C) DECEMBER 1975
317          ,*DIGITAL EQUIPMENT CORP.
318          ,*MAYNARD, MASS, 01754
319          ,*
320          ,*PROGRAM BY GLENN JOHNSON
321          ,*
322          ,*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
323          ,*PACKAGE (MAINDEC-11-DZQAC-B1),AUG 29,1975.
324          ,*
325          $TN=1
326          $SHR=160000      ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT
327

```

```

328
329
330          ;*****
331          000000          ,=0          ;TRAP CATCHER 0 - 776
332
333          ;*****
334
335          ;*****
336
337          ,SBTTL ACT11 HOOKS
338          ;HOOKS REQUIRED BY ACT11
339          001000          $SVPC=,          ;SAVE PC
340          000046          ,=46
341          000046 016356          $ENDAD          ;;1)SET LOC,46 TO ADDRESS OF $ENDAD IN ,SEOP
342          000052          ,=52
343          000052 000000          ,WORD 0          ;;2)SET LOC,52 TO ZERO
344          001000          ,=$SVPC          ;; RESTORE PC
345
346          DUMMY= 0
347          ERRNM= 1
348          F= 51
349          N= 176
350          PC= %7
351          SP= %6
352          SCOPE= 10701
353          SCOPE1= 10701
354          SCOPE3= 10703
355          SW09= 1000
356          SW10= 2000
357          SW11= 4000
358          SW12= 10000
359          TYPE= IOT
360
361
362          ,=20
363          000020 016624          $TYPE

```

```
364          000400          .S400
365          ;*****
366
367          .SBTTL  APT MAILBOX=ETABLE
368
369
370
371          .EVEN
372          $MAIL:          ;;APT MAILBOX
373          $MSGTY: .WORD  AMSGTY ;;MESSAGE TYPE CODE
374          $FATAL: .WORD  AFATAL ;;FATAL ERROR NUMBER
375          $TESTN: .WORD  ATESTN ;;TEST NUMBER
376          $PASS: .WORD  APASS  ;;PASS COUNT
377          $DEVCT: .WORD  ADEVCT ;;DEVICE COUNT
378          $UNIT: .WORD  AUNIT  ;;I/O UNIT NUMBER
379          $MSGAD: .WORD  AMSGAD ;;MESSAGE ADDRESS
380          $MSGLG: .WORD  AMSGLG ;;MESSAGE LENGTH
381          $ETABLE:          ;;APT ENVIRONMENT TABLE
382          $ENV: .BYTE  AENV   ;;ENVIRONMENT BYTE
383          $ENVM: .BYTE  AENVM  ;;ENVIRONMENT MODE BITS
384          $SWREG: .WORD  ASWREG ;;APT SWITCH REGISTER
385          $USWR: .WORD  AUSWR  ;;USER SWITCHES
386          $CPUOP: .WORD  ACPUOP ;;CPU TYPE,OPTIONS
387          ;*          BITS 15-11=CPU TYPE
388          ;*          11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
389          ;*          11/70=06,PDG=07,Q=10
390          ;*          BIT 10=REAL TIME CLOCK
391          ;*          BIT 9=FLOATING POINT PROCESSOR
392          ;*          BIT 8=MEMORY MANAGEMENT
393          $ETEND:          ;
394          .MEXIT
395
396          ;*****
397
398          .SBTTL  APT PARAMETER BLOCK
399          ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
400          ;*****
401          .S4=          ;;SAVE CURRENT LOCATION
402          .S24=        ;;SET POWER FAIL TO POINT TO START OF PROGRAM
403          200          ;;FOR APT START UP
404          .S44=        ;;POINT TO APT INDIRECT ADDRESS PNTR.
405          $APTHDR      ;;POINT TO APT HEADER BLOCK
406          .S.SX        ;;RESET LOCATION COUNTER
407          ;*****
408          ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
409          ;INTERFACE SPEC.
410
411          $APTHD:
412          $HIBTS: .WORD  0      ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
413          $MBADR: .WORD  $MAIL  ;;ADDRESS OF APT MAILBOX (BITS 0-15)
414          $STMT: .WORD  3      ;;RUN TIM OF LONGEST TEST
415          $PASTM: .WORD  5      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
416          $UNITM: .WORD  3      ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
417          .WORD  $ETEND-$MAIL/2 ;;LENGTH MAILBOX=ETABLE(WORDS)
418
419          .=$APTHD
```

```
420          $COUNT:          .=$COUNT+2
421
422          $PSWORD:          .=$PSWORD+2
423
424          $TEMP1:          .=$TEMP1+2
425
426          $TEMP2:          .=$TEMP2+2
427
428          $TEMP3:          .=$TEMP3+2
429
430          $TEMP4:          .=$TEMP4+2
431
432          $TEMP5: .WORD
433          $TEMP6: .WORD
434          $TYPCNT: .BYTE  000
435          $STPCNT: .BYTE
436          $S0: 7
437          $S1: -7
438          $S2: S1
439          $S3: -6
440          $S4: -1
441          $S5: 40000
442          $S6: S5
443          $S7: 40000
444          $S8: -2
445          $S9: 2
446          $S10: S9
447          $S11: 2
448          $SWR: 177570
449          $DISPLAY: 177570
450          $TTYOUT: 64
451          $TPB: 177566
452          $TPS: 177564
453          $CRLF: .ASCIZ <15><12>/ /
454          $POWER: .ASCIZ <12><15>/POWER/
455
456
457
458
459
460
461
462
463
464
465
```

```

466      000200      012737      016410      000024      =200      MOV      #SPWRDN,#24      ;PREPARE TO SERVICE POWER DOWN ROUTINE
467      000200      012737      000410      28:      MOV      #DEVCT,R0      ;PREPARE TO INITIALIZE THE STACK
468      000212      015040      28:      CLR      -(R0)
469      000214      022700      000400      CMP      #SMAIL,R0
470      000220      001374      BNE      28
471      030222      000167      000352      RESTRT:  JMP      BEGIN
472
473
474      000600      =600
475
476      030600      012705      000404      BEGIN:  MOV      #STESTN,R5      ;MAKE R5 POINT TO THE LOCATION STESTN
477      000604      005037      000430      CLR      #COUNT      ;CLEAR THE COUNTER
478      000610      012715      000001      MOV      #1,(R5)      ;INITIALIZE TEST NUMBER
479      000614      012706      000600      MOV      #BEGIN,SP      ;** STACK AT BEGIN **
480      000620      013746      000004      MOV      #4,-(SP)      ;SAVE ERROR VECTOR
481      000624      013746      000006      MOV      #6,-(SP)
482      000630      012767      000644      177146      MOV      #16,4      ;SET UP TIME OUT VECTOR
483      000636      005777      177640      TST      #SWR      ;TRY TO REFERENCE HARDWARE SWR
484      000642      000407      BR      36      ;BRANCH IF NO TIMEOUT TRAP OCCURS
485      000644      012767      000176      177630      18:      MOV      #SWREG,SWR      ;POINT TO SOFTWARE SWR
486      000652      012767      000174      177624      MOV      #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY REG
487      000660      022626      CMP      (SP)+,(SP)+      ;RESTORE STACK
488      000662      012637      000006      36:      MOV      (SP)+,#6      ;RESTORE ERROR VECTOR
489      000666      012637      000004      MOV      (SP)+,#4
490      000672      106427      000000      MTPS     #0      ;PLACE #0 IN PSW
491      000676      132737      000001      000420      BITB     #1,#ENV      ;ARE WE UNDER APT ?
492      000704      001403      BFC     28      ;IF NOT THEN GO TO 28
493      000706      012767      000422      177566      MOV      #SSWREG,SWR      ;USE APT SWITCH REGISTER
494      000714      012737      000001      000434      28:      MOV      #1,#TEMP1      ;TEMP1=1
495      000722      005037      000436      CLR      #TEMP2      ;TEMP2=0
496      000726      012737      000001      000440      MOV      #1,#TEMP3      ;TEMP3=1
497      000734      005037      000442      CLR      #TEMP4      ;TEMP4=0
498
499
    
```

```

500
501
502
503
504
505
506
507
508
509
510
511
512
513      000740      010701      000434      START:  SCOPE1
514      000742      013700      000434      MOV      #TEMP1,%0      ;LOAD R0 WITH THE CONTENTS OF TEMP1
515      000746      032737      000001      000406      BIT      #1,#PASS      ;IS IT AN EVEN PASS ?
516      000754      001004      BNE      28      ;IF NOT THEN GO TO 28
517      000756      013701      000436      MOV      #TEMP2,R1      ;OTHERWISE EXECUTE THE INSTRUCTION
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
    
```

556	001130	072167	177302	281	ASH	TEMP2,%1	;SHIFT R1 BY THE NUMBER SPECIFIED BY TEMP2
557	001134	106737	000432	481	MFPS	0#PSWORD	;SAVE PS
558	001140	123737	000442	000432	CMPB	0#TEMP4,0#PSWORD	;IS THE PS = TEMP4 ?
559	001146	001403			BEQ	+10	
560	001150	004767	015264		JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
561							;THE PS IS NOT EQUAL TO 0
562	001154	000003			3		
563	001156	005237	000430		INC	0#COUNT	;INCREMENT THE COUNTER
564	001162	023701	000440		CMP	0#TEMP3,%1	;IS THE RESULT IN R1 EQUAL TO TEMP3?
565	001166	001403			BEQ	+10	
566	001170			681			
567	001170	004767	015244		JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
568							;EITHER INCORRECT R1 OR INCORRECT SEQUENCE
569	001174	000004			4		
570	001176	021537	000430		CMP	(R5),0#COUNT	;IS THE TEST NUMBER EQUAL TO THE COUNTER?
571	001202	001372			BNE	68	;IF NOT GO TO THE HLT ABOVE
572	001204	005215			INC	(R5)	
573	001206	010703			SCOPE3		
574	001210	021527	000037		CMP	(R5),#37	;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT
575							;BY 14, AND RIGHT BY 14,?
576	001214	002011			BGE	08	
577	001216	005237	000436		INC	0#TEMP2	
578	001222	006367	177212		ASL	TEMP3	;SHIFT TEMP3 LEFT
579	001226	021527	000020		CMP	(R5),#20	;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14,?
580	001232	001004			BNE	REG2	
581	001234	000167	000622		JMP	NEGAT	;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
582	001240	004767	000644		JSR	PC,TST37	;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
583	001244	010701		881	REG2:	SCOPE1	
584	001246	013702	000434		MOV	0#TEMP1,%2	;LOAD R2 WITH THE CONTENTS OF TEMP1
585	001252	032737	000001	000406	BIT	#1,0#SPASS	;IS IT AN EVEN PASS ?
586	001260	001004			BNE	28	;IF NOT THEN GO TO 28
587	001262	013703	000436		MOV	0#TEMP2,R3	;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
588	001266	072703			ASH	R3,R2	;USING R2
589	001270	000402			BR	48	
590	001272	072267	177140	281	ASH	TEMP2,%2	;SHIFT R2 BY THE NUMBER SPECIFIED BY TEMP2
591	001276	106737	000432	481	MFPS	0#PSWORD	;SAVE PS
592	001302	123737	000442	000432	CMPB	0#TEMP4,0#PSWORD	;IS THE PS = TEMP4 ?
593	001310	001403			BEQ	+10	
594	001312	004767	015122		JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
595							;THE PS IS NOT EQUAL TO 0
596	001316	000005			5		
597	001320	005237	000430		INC	0#COUNT	
598	001324	023702	000440		CMP	0#TEMP3,%2	;IS THE RESULT IN R2 EQUAL TO TEMP3?
599	001330	001403			BEQ	+10	
600	001332			681			
601	001332	004767	015102		JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
602							;EITHER INCORRECT R2 OR INCORRECT SEQUENCE
603	001336	000006			6		
604	001340	021537	000430		CMP	(R5),0#COUNT	;IS THE TEST NUMBER EQUAL TO THE COUNTER?
605	001344	001372			BNE	68	;IF NOT GO TO THE HLT ABOVE
606	001346	005215			INC	(R5)	
607	001350	010701			SCOPE1		
608	001352	021527	000037		CMP	(R5),#37	;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
609							;LEFT BY 14, AND RIGHT BY 14,?
610	001356	002011			BGE	08	
611	001360	005237	000436		INC	0#TEMP2	

612	001364	006367	177050		ASL	TEMP3	;SHIFTED TEMP3 LEFT
613	001370	021527	000020		CMP	(R5),#20	;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14,?
614	001374	001004			BNE	REG3	
615	001376	000167	000460		JMP	NEGAT	;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
616	001402	004767	000502		JSR	PC,TST37	;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
617	001406	010701		881	REG3:	SCOPE1	
618	001410	013703	000434		MOV	0#TEMP1,%3	;LOAD R3 WITH THE CONTENTS OF TEMP1
619	001414	032737	000001	000406	BIT	#1,0#SPASS	;IS IT AN EVEN PASS ?
620	001422	001004			BNE	28	;IF NOT THEN GO TO 28
621	001424	013704	000436		MOV	0#TEMP2,R4	;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
622	001430	072702			ASH	R4,R3	;USING R3
623	001432	000402			BR	48	
624	001434	072567	176776	281	ASH	TEMP2,%3	;SHIFT R3 BY THE NUMBER SPECIFIED BY TEMP2
625	001440	106737	000432	481	MFPS	0#PSWORD	;SAVE PS
626	001444	123737	000442	000432	CMPB	0#TEMP4,0#PSWORD	;IS THE PS = TEMP4 ?
627	001452	001403			BEQ	+10	
628	001454	004767	014760		JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
629							;THE PS IS NOT EQUAL TO 0.
630	001460	000007			7		
631	001462	005237	000430		INC	0#COUNT	
632	001466	023703	000440		CMP	0#TEMP3,%3	;IS THE RESULT IN R3 EQUAL TO TEMP3?
633	001472	001403			BEQ	+10	
634	001474			681			
635	001474	004767	014740		JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
636							;EITHER INCORRECT R3 OR INCORRECT SEQUENCE
637	001500	000010			10		
638	001502	021537	000430		CMP	(R5),0#COUNT	;IS THE TEST NUMBER EQUAL TO THE COUNTER?
639	001506	001372			BNE	68	;IF NOT GO TO THE HLT ABOVE
640	001510	005215			INC	(R5)	
641	001512	010701			SCOPE1		
642	001514	021527	000037		CMP	(R5),#37	;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
643							;LEFT BY 14, AND RIGHT BY 14,?
644	001520	002010			BGE	08	
645	001522	005237	000436		INC	0#TEMP2	
646	001526	006367	176706		ASL	TEMP3	;SHIFT TEMP3 LEFT?
647	001532	021527	000020		CMP	(R5),#20	;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14,?
648	001536	001003			BNE	REG4	
649	001540	000550			BR	NEGAT	;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
650	001542	004767	000342		JSR	PC,TST37	;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
651	001546	010703		881	REG4:	SCOPE3	
652	001550	013704	000434		MOV	0#TEMP1,%4	;LOAD R4 WITH THE CONTENTS OF TEMP1
653	001554	010501			MOV	R5,R1	;SAVE R5
654	001556	032737	000001	000406	BIT	#1,0#SPASS	;IS IT AN EVEN PASS ?
655	001564	001004			BNE	28	;IF NOT THEN GO TO 28
656	001566	013705	000436		MOV	0#TEMP2,R5	;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
657	001572	072405			ASH	R5,R4	;USING R4
658	001574	000402			BR	48	
659	001576	072467	176634	281	ASH	TEMP2,%4	;SHIFT R4 BY THE NUMBER SPECIFIED BY TEMP2
660	001602	106737	000432	481	MFPS	0#PSWORD	;SAVE PS
661	001606	123737	000442	000432	CMPB	0#TEMP4,0#PSWORD	;IS PS = TEMP4 ?
662	001614	001403			BEQ	+10	
663	001616	004767	014616		JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE
664							;THE PS IS NOT EQUAL TO 0
665	001622	000011			11		
666	001624	005237	000430		INC	0#COUNT	
667	001630	023704	000440		CMP	0#TEMP3,%4	;IS THE RESULT IN R4 EQUAL TO TEMP3?

668	001634	001403		BEQ	,+10	
669	001636		68:	JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE ;EITHER INCORRECT R4 OR INCORRECT SEQUENCE
670	001636	004767	014576			
671				12		
672	001642	000012		MOV	R1,R5	;RESTORE R5
673	001644	010105		CMP	(R5),#COUNT	;IS THE TEST NUMBER EQUAL TO THE COUNTER?
674	001646	021537	000430	BNE	68	;IF NOT GO TO THE HLT ABOVE
675	001652	001371		INC	(R5)	
676	001654	005215		SCOPE1		
677	001656	010701		CMP	(R5),#37	;HAS THE CONTENTS OF REGISTERS BEEN ;SHIFTED LEFT BY 14, AND RIGHT BY 14,?
678	001660	021527	000037			
679				BGE	00	
680	001664	002010		INC	0#TEMP2	
681	001666	005237	000436	ASL	TEMP3	;SHIFT TEMP3 LEFT
682	001672	006367	176542	CMP	(R5),#20	;HAS THE CONTENTS OF REGISTER BEEN SHIFTED BY 14,?
683	001676	021527	000020	BNE	REGS	
684	001702	001003		BR	NEGAT	;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
685	001704	009466		JSR	PC,TST37	;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
686	001706	004767	000176	88:		
687	001712	010701		SCOPE1		
688	001714	010501		MOV	R5,R1	;SAVE R5
689	001716	013705	000434	MOV	0#TEMP1,#5	;LOAD R5 WITH THE CONTENTS OF TEMP1
690	001722	032737	000001	BIT	#1,#PSW	;IS IT AN EVEN PASS ?
691	001730	001004	000406	BNE	20	;IF NOT THEN GO TO 20
692	001732	013700	000436	MOV	0#TEMP2,R0	;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
693	001736	072500		ASH	R0,R5	;USING R5
694	001740	000402		BR	40	
695	001742	072567	176470	ASH	TEMP2,#5	;SHIFT R5 BY THE NUMBER SPECIFIED BY TEMP2
696	001746	106737	000432	MFFPS	0#PSWORD	;SAVE PS
697	001752	123737	000442	CMPB	0#TEMP4,0#PSWORD	;IS PS = TEMP4 ?
698	001760	001403		BEQ	,+10	
699	001762	004767	014452	JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE ;THE PS IS NOT EQUAL TO 0.
700				13		
701	001766	000013		INC	0#COUNT	
702	001770	005237	000430	CMP	0#TEMP3,#5	;IS THE RESULT IN R5 EQUAL TO TEMP3?
703	001774	023705	000440	BEQ	,+10	
704	002000	001403		68:		
705	002002			JSR	PC,\$HLT	;SEEN AN ERROR, GO TO TH HALT ROUTINE ;EITHER INCORRECT R5 OR INCORRECT SEQUENCE
706	002002	004767	014432			
707				14		
708	002006	000014		CMP	(R1),#COUNT	;IS THE TEST NUMBER EQUAL TO THE COUNTER?
709	002010	021137	000430	BNE	68	;IF NOT GO TO THE HLT ABOVE
710	002014	001372		MOV	R1,R5	;RESTORE R5
711	002016	010105		INC	(R5)	
712	002020	005215		SCOPE1		
713	002022	010701		CMP	(R5),#37	;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED ;LEFT BY 14, AND RIGHT BY 14,?
714	002024	021527	000037			;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
715				BGE	00	
716	002030	002010		INC	0#TEMP2	
717	002032	005237	000436	ASL	TEMP3	;SHIFT TEMP3 LEFT
718	002036	006367	176376	CMP	(R5),#20	;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14,?
719	002042	021527	000020	BEQ	NEGAT	;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
720	002046	001405		BR	100	
721	002050	000402		JSR	PC,TST37	
722	002052	004767	000032	88:		
723	002056	000167	176656	100:	JMP	START

724	002062	012737	040000	000434	NEGAT:	MOV	#40000,0#TEMP1	;TEMP1=40000
725	002070	012737	177762	000436	MOV	#177762,0#TEMP2	;TEMP2=177762	
726	002076	012737	000001	000440	MOV	#1,0#TEMP3	;TEMP3=1	
727	002104	000167	176630		JMP	START		
728	002110	021527	000037		TST37:	CMP	(R5),#37	;IS IT TEST 37?
729	002114	001013			BNE	TST40	;IF NOT THEN TRY TEST 40	
730	002116	005037	000434		CLR	0#TEMP1	;0	
731	002122	012737	000020	000436	MOV	#16,0#TEMP2	;SHIFTED BY 16	
732	002130	005037	000440		CLR	0#TEMP3	;IS=0	
733	002134	012737	000004	000442	MOV	#4,0#TEMP4	;AND PS=4	
734	002142	000207			RTS	PC		
735	002144	021527	000040		TST40:	CMP	(R5),#40	;IS IT TEST 40?
736	002150	001003			BNE	TST41	;IF NOT THEN TRY TEST 41	
737	002152	005037	000436		CLR	0#TEMP2	;0 SHIFTED BY 0=0 AND PS=4	
738	002156	000207			RTS	PC		
739	002160	021527	000041		TST41:	CMP	(R5),#41	;IS IT TEST 41?
740	002164	001004			BNE	TST42	;IF NOT THEN TRY TEST 42	
741	002166	012737	177760	000436	MOV	#-16,0#TEMP2	;0 SHIFTED BY -16,=0 AND PS=4	
742	002174	000207			RTS	PC		
743	002176	021527	000042		TST42:	CMP	(R5),#42	;IS IT TEST 42?
744	002202	001013			BNE	TST43	;IF NOT THEN TRY TEST 43	
745	002204	012737	100000	000434	MOV	#100000,0#TEMP1	;100000	
746	002212	005237	000436		INC	0#TEMP2	;SHIFTED BY -15	
747	002216	005337	000440		DEC	0#TEMP3	;IS=-1	
748	002222	012737	000010	000442	MOV	#10,0#TEMP4	;AND PS=10	
749	002230	000207			RTS	PC		
750	002232	021527	000043		TST43:	CMP	(R5),#43	;IS IT TEST 43?
751	002236	001012			BNE	TST44	;IF NOT THEN TRY TEST 44	
752	002240	012737	125252	000434	MOV	#125252,0#TEMP1	;125252	
753	002246	012737	177777	000436	MOV	#-1,0#TEMP2	;SHIFTED BY -1	
754	002254	012737	152525	000440	MOV	#152525,0#TEMP3	;IS=152525 AND PS=10	
755	002262	000207			RTS	PC		
756	002264	021527	000044		TST44:	CMP	(R5),#44	;IS IT TEST 44?
757	002270	001012			BNE	TST45	;IF NOT THEN TRY TEST 45	
758	002272	012737	000001	000436	MOV	#1,0#TEMP2	;125252 SHIFTED BY 1	
759	002300	012737	052524	000440	MOV	#52524,0#TEMP3	;IS=52524	
760	002306	012737	000003	000442	MOV	#3,0#TEMP4	;AND PS=3	
761	002314	000207			RTS	PC		
762	002316	021527	000045		TST45:	CMP	(R5),#45	;IS IT TEST 45?
763	002322	001012			BNE	TST46	;IF NOT THEN TRY TEST 46	
764	002324	012737	177776	000436	MOV	#-2,0#TEMP2	;125252 SHIFTED BY -2	
765	002332	012737	165252	000440	MOV	#165252,0#TEMP3	;IS=165252	
766	002340	012737	000011	000442	MOV	#11,0#TEMP4	;AND PS=11	
767	002346	000207			RTS	PC		
768	002350	021527	000046		TST46:	CMP	(R5),#46	;IS IT TEST 46?
769	002354	001014			BNE	TST47	;IF NOT THEN TRY TEST 47	
770	002356	012737	177777	000434	MOV	#-1,0#TEMP1	;1	
771	002364	012737	000020	000436	MOV	#16,0#TEMP2	;SHIFTED BY 15.	
772	002372	005037	000440		CLR	0#TEMP3	;IS=0	
773	002376	012737	000007	000442	MOV	#7,0#TEMP4	;AND PS=7	
774	002404	000207			RTS	PC		
775	002406	021527	000047		TST47:	CMP	(R5),#47	;IS IT TEST 47?
776	002412	001011			BNE	TST50	;IF NOT THEN TRY TEST 50	
777	002414	005337	000436		DEC	0#TEMP2	;1 SHIFTED BY 15	
778	002420	012737	100000	000440	MOV	#100000,0#TEMP3	;IS=100000	
779	002426	012737	000011	000442	MOV	#11,0#TEMP4	;AND PS=11	


```

780 002434 000207          RTS      PC
781 002436 021527 000050    TST50:  CMP      (R5),#50      ;IS IT TEST 50
782 002442 001007          BNE      ENT51             ;IF NOT THEN TRY TEST 51
783 002444 012737 137777 000434  MOV      #137777,0*TEMP1  ;137777 SHIFTED BY 15, IS=100000
784 002452 012737 000013 000442  MOV      #13,0*TEMP4     ;AND PS=13
785 002460 000207          RTS      PC
786 002462 021527 000051    ENT51:  CMP      (R5),#51      ;IS IT ENTERING TEST 51?
787 002466 001403          BEQ      ,+10
788 002470 004767 013744    JSR      PC,$HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
789                                ;TEST NUMBER GOOFED
790 002474 000015          15
791
792 002476 005726          TST      (SP)+           ;RESTORE STACK POINTER
793 002500 012704 177771    MOV      #-7,%4
794 002504 012702 000454    MOV      #S1,%2
795 002510 012703 000456    MOV      #S2,%3
    
```

```

796                                ;*****
797                                ;TEST:51 11/34 ASH 125252 SHIFTED BY #5 = 52500 PS = 3
798                                ;*****
799
800 002514 010701          TST51:  SCOPE1
801 002516 012701 125252    MOV      #125252,%1      ;LOAD R1 WITH 125252
802 002522 072127 000005    ASH      #5,%1          ;SHIFT R1 BY #5
803 002526 106737 000432    MFPS     0*PSWORD       ;SAVE PS
804 002532 122737 000003 000432  CMPB     #3,0*PSWORD     ;IS THE PS 3?
805 002540 001403          BEQ      ,+10
806 002542 004767 013672    JSR      PC,$HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
807                                ;THE PS IS NOT EQUAL TO 3
808 002546 000016          16
809 002550 022701 052500    CMP      #52500,%1      ;IS THE RESULT 52500?
810 002554 001403          BEQ      ,+10
811 002556          15:
812 002556 004767 013656    JSR      PC,$HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
813                                ;R1 IS NOT EQUAL TO 52500 OR INCORRECT SEQUENCE
814 002562 000017          17
815 002564 021527 000051    CMP      (R5),#51      ;IS $TESTN = #51
816 002570 001372          BNE      15             ;IF NOT THEN GO TO HLT ABOVE
817 002572 005215          INC      (R5)
818
819
820
821                                ;*****
822                                ;TEST:52 11/34 ASH 125252 SHIFTED BY #S2 = 177525 PS = 10
823                                ;*****
824
825 002574 010701          TST52:  SCOPE1
826 002576 012700 125252    MOV      #125252,%0     ;LOAD R0 WITH 125252
827 002602 072077 175650    ASH      #S2,%0         ;SHIFT R0 BY #S2
828 002606 106737 000432    MFPS     0*PSWORD       ;SAVE PS
829 002612 122737 000010 000432  CMPB     #10,0*PSWORD    ;IS THE PS 10?
830 002620 001403          BEQ      ,+10
831 002622 004767 013612    JSR      PC,$HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
832                                ;THE PS IS NOT EQUAL TO 10
833 002626 000020          20
834 002630 022700 177525    CMP      #177525,%0     ;IS THE RESULT 177525?
835 002634 001403          BEQ      ,+10
836 002636          15:
837 002636 004767 013576    JSR      PC,$HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
838                                ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
839 002642 000021          21
840 002644 021527 000052    CMP      (R5),#52      ;IS $TESTN = #52
841 002650 001372          BNE      15             ;IF NOT THEN GO TO HLT ABOVE
842 002652 005215          INC      (R5)
843
844
845
    
```

```
*****
;TEST:53 11/34 ASH 125252 SHIFTED BY 0#S1 = 177525 PS = 10
*****
846 ;
847 ;
848 ;
849 ;
850 002654 010701 TST53: SCOPE1
851 002656 012700 MOV #125252,%0 ;LOAD R0 WITH 125252
852 002662 072037 125252 ASH 0#S1,%0 ;SHIFT R0 BY 0#S1
853 002666 106737 000432 MFPS 0#PSWORD ;SAVE PS
854 002672 122737 000010 000432 CMPB #10,0#PSWORD ;IS THE PS 10?
855 002700 001403 BEQ .+10
856 002702 004767 013532 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
857 ; ;THE PS IS NOT EQUAL TO 10
858 002706 000022 22
859 002710 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
860 002714 001403 BEQ .+10
861 002716 18: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
862 002716 004767 013516 ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
863 ;
864 002722 000023 23
865 002724 021527 000053 CMP (R5),#53 ;IS $TESTN = #53
866 002730 001372 BNE 18 ;IF NOT THEN GO TO HLT ABOVE
867 002732 005215 INC (R5)
868 ;
869 ;
870 ;
871 ;
872 ;
873 ;
874 ;
875 002734 010701 TST54: SCOPE1
876 002736 012700 MOV #125252,%0 ;LOAD R0 WITH 125252
877 002742 072012 125252 ASH (2),%0 ;SHIFT R0 BY (2)
878 002744 106737 000432 MFPS 0#PSWORD ;SAVE PS
879 002750 122737 000010 000432 CMPB #10,0#PSWORD ;IS THE PS 10?
880 002756 001403 BEQ .+10
881 002760 004767 013454 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
882 ; ;THE PS IS NOT EQUAL TO 10
883 002764 000024 24
884 002766 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
885 002772 001403 BEQ .+10
886 002774 18: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
887 002774 004767 013440 ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
888 ;
889 003000 000025 25
890 003002 021527 000054 CMP (R5),#54 ;IS $TESTN = #54
891 003006 001372 BNE 18 ;IF NOT THEN GO TO HLT ABOVE
892 003010 005215 INC (R5)
893 ;
894 ;
895 ;
```

```
*****
;TEST:55 11/34 ASH 125252 SHIFTED BY (2)+ = 177525 PS = 10
*****
896 ;
897 ;
898 ;
899 ;
900 003012 010701 TST55: SCOPE1
901 003014 012700 MOV #125252,%0 ;LOAD R0 WITH 125252
902 003020 072022 125252 ASH (2)+,%0 ;SHIFT R0 BY (2)+
903 003022 106737 000432 MFPS 0#PSWORD ;SAVE PS
904 003026 122737 000010 000432 CMPB #10,0#PSWORD ;IS THE PS 10?
905 003034 001403 BEQ .+10
906 003036 004767 013376 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
907 ; ;THE PS IS NOT EQUAL TO 10
908 003042 000026 26
909 003044 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
910 003050 001403 BEQ .+10
911 003052 18: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
912 003052 004767 013362 ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
913 ;
914 003056 000027 27
915 003060 021527 000055 CMP (R5),#55 ;IS $TESTN = #55
916 003064 001372 BNE 18 ;IF NOT THEN GO TO HLT ABOVE
917 003066 005215 INC (R5)
918 ;
919 ;
920 ;
921 ;
922 ;
923 ;
924 ;
925 003070 010701 TST56: SCOPE1
926 003072 012700 MOV #125252,%0 ;LOAD R0 WITH 125252
927 003076 072042 125252 ASH -(2),%0 ;SHIFT R0 BY -(2)
928 003100 106737 000432 MFPS 0#PSWORD ;SAVE PS
929 003104 122737 000010 000432 CMPB #10,0#PSWORD ;IS THE PS 10?
930 003112 001403 BEQ .+10
931 003114 004767 013320 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
932 ; ;THE PS IS NOT EQUAL TO 10
933 003120 000030 30
934 003122 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
935 003126 001403 BEQ .+10
936 003130 18: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
937 003130 004767 013304 ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
938 ;
939 003134 000031 31
940 003136 021527 000056 CMP (R5),#56 ;IS $TESTN = #56
941 003142 001372 BNE 18 ;IF NOT THEN GO TO HLT ABOVE
942 003144 005215 INC (R5)
943 ;
944 ;
945 ;
```

```
*****  
;TEST:57 11/34 ASH 125252 SHIFTED BY 2(3) = 177252 PS = 11  
*****  
946  
947  
948  
949  
950 003146 010701 TST57: SCOPE1  
951 003150 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252  
952 003154 072063 000002 ASH 2(3),%0 ;SHIFT R0 BY 2(3)  
953 003160 106737 000432 MFPS 0#PSWORD ;SAVE PS  
954 003164 127737 000011 000432 CMPB #11,0#PSWORD ;IS THE PS 11?  
955 003172 001403 BEQ ,+10  
956 003174 004767 013240 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
957 ;THE PS IS NOT EQUAL TO 11  
958 003200 000032 32 ;  
959 003202 022700 177252 CMP #177252,%0 ;IS THE RESULT 177252?  
960 003206 001403 BEQ ,+10  
961 003210 18: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
962 003214 004767 013224 ;R0 IS NOT EQUAL TO 177252 OR INCORRECT SEQUENCE  
963  
964 003214 000033 33 ;  
965 003216 021527 000057 CMP (R5),#57 ;IS $TESTN = #57  
966 003222 001372 BNE 18 ;IF NOT THEN GO TO HLT ABOVE  
967 003224 005215 INC (R5)  
968  
969  
970  
971  
972  
973  
974  
975 003226 010701 TST60: SCOPE1  
976 003230 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252  
977 003234 072073 000000 ASH 0(3),%0 ;SHIFT R0 BY 0(3)  
978 003240 106737 000432 MFPS 0#PSWORD ;SAVE PS  
979 003244 127737 000010 000432 CMPB #10,0#PSWORD ;IS THE PS 10?  
980 003252 001403 BEQ ,+10  
981 003254 004767 013160 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
982 ;THE PS IS NOT EQUAL TO 10  
983 003260 000034 34 ;  
984 003262 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?  
985 003266 001403 BEQ ,+10  
986 003270 18: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
987 003274 004767 013144 ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE  
988  
989 003274 000035 35 ;  
990 003276 021527 000060 CMP (R5),#60 ;IS $TESTN = #60  
991 003302 001372 BNE 18 ;IF NOT THEN GO TO HLT ABOVE  
992 003304 005215 INC (R5)  
993  
994  
995
```

```
*****  
;TEST:61 11/34 ASH 125252 SHIFTED BY 0(3)+ = 177525 PS = 10  
*****  
996  
997  
998  
999  
1000 003306 010701 TST61: SCOPE1  
1001 003310 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252  
1002 003314 072033 000000 ASH 0(3)+,%0 ;SHIFT R0 BY 0(3)+  
1003 003316 106737 000432 MFPS 0#PSWORD ;SAVE PS  
1004 003322 127737 000010 000432 CMPB #10,0#PSWORD ;IS THE PS 10?  
1005 003330 001403 BEQ ,+10  
1006 003332 004767 013102 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
1007 ;THE PS IS NOT EQUAL TO 10  
1008 003336 000036 36 ;  
1009 003340 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?  
1010 003344 001403 BEQ ,+10  
1011 003346 18: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
1012 003346 004767 013066 ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE  
1013  
1014 003352 000037 37 ;  
1015 003354 021527 000061 CMP (R5),#61 ;IS $TESTN = #61  
1016 003360 001372 BNE 18 ;IF NOT THEN GO TO HLT ABOVE  
1017 003362 005215 INC (R5)  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025 003364 010701 TST62: SCOPE1  
1026 003366 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252  
1027 003372 072053 000000 ASH 0-(3),%0 ;SHIFT R0 BY 0-(3)  
1028 003374 106737 000432 MFPS 0#PSWORD ;SAVE PS  
1029 003400 127737 000010 000432 CMPB #10,0#PSWORD ;IS THE PS 10?  
1030 003406 001403 BEQ ,+10  
1031 003410 004767 013024 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
1032 ;THE PS IS NOT EQUAL TO 10  
1033 003414 000040 40 ;  
1034 003416 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?  
1035 003422 001403 BEQ ,+10  
1036 003424 18: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
1037 003424 004767 013010 ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE  
1038  
1039 003430 000041 41 ;  
1040 003432 021527 000062 CMP (R5),#62 ;IS $TESTN = #62  
1041 003436 001372 BNE 18 ;IF NOT THEN GO TO HLT ABOVE  
1042 003440 005215 INC (R5)  
1043  
1044  
1045
```

```

1046 ;*****
1047 ; ASHC INSTRUCTION TESTS
1048 ;*****
1049
1050
1051
1052 ;*****
1053 ;TESTS 63-157
1054 ;*****
1055
1056
1057
1058 003442 012737 000062 000430 MOV #62,#COUNT
1059 003450 005037 000434 CLR #TEMP1 ;TEMP1=0
1060 003454 012737 000001 000436 MOV #1,#TEMP2 ;TEMP2=1
1061 003467 005037 000440 CLR #TEMP3 ;TEMP3=0
1062 003466 005037 000442 CLR #TEMP4 ;TEMP4=0
1063 003472 012737 000001 000444 MOV #1,#TEMP5 ;TEMP5=1
1064 003500 005037 000446 CLR #TEMP6 ;0 1 SHIFTED BY 0=0 1, PS=0
1065
1066 003504 010703 REG01: SCOPE3
1067 003506 010522 MOV R5,R2 ;SAVE R5
1068 003510 013700 MOV #TEMP1,%0 ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 0
1069 003514 013701 000436 MOV #TEMP2,%0,1 ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 1
1070 003520 000241 CLC
1071 003522 032737 000001 000406 BIT #1,#SPASS ;IS IT AN EVEN PASS ?
1072 003530 001004 BNE 20 ;IF NOT THEN GO TO 20
1073 003532 013705 000440 MOV #TEMP3,R5 ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
1074 003536 073005 ASHC R5,R0 ;USING R0
1075 003540 000402 BR 40
1076 003542 073067 174672 20: ASHC TEMP3,%0 ;ASHC REGISTER 0 BY THE CONTENTS OF TEMP3
1077 003546 006737 000432 40: MFPS #PSWORD ;SAVE PS
1078 003552 123737 000446 000432 CMPB #TEMP6,#PSWORD ;COMPARE PS WITH THE CONTENTS OF TEMP6
1079 003560 001403 BEQ .+10
1080 003562 004767 012652 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1081 ;WRONG PS
1082 003566 000042 42
1083 003570 005237 000430 INC #COUNT
1084 003574 023700 000442 CMP #TEMP4,%0 ;IS THE RESULT IN R0 SAME AS TEMP4?
1085 003600 001403 BEQ .+10
1086 003602 004767 012632 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1087 ;WRONG RESULT IN R0
1088 003606 000043 43
1089 003610 023701 000444 CMP #TEMP5,%1 ;IS THE RESULT IN R1 SAME AS TEMP5?
1090 003614 001403 BEQ .+10 ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMPS
1091 ;AND PS=TEMP6
1092 003616 004767 012616 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1093 ;WRONG RESULT IN R1
1094 003622 000044 44
1095 003624 012005 MOV R2,R5 ;RESTORE R5
1096 003626 021537 000430 CMP (R5),#COUNT ;IS TEST NUMBER=COUNTER?
1097 003632 001403 BEQ .+10
1098 003634 004767 012600 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1099 ;NO
1100 003640 000045 45
1101 003642 005215 INC (R5)
    
```

```

1102 003644 021527 000160 CMP (R5),#160 ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
1103 003650 002014 BGE 60 ;YES
1104 003652 005237 000440 INC #TEMP3
1105 003656 000241 CLC
1106 003660 006137 000444 ROL #TEMP5 ;ROTATE TEMPS LEFT BY 1 PLACE
1107 003664 006137 000442 ROL #TEMP4 ;INTRODUCE CARRY FROM TEMP4 IN TEMPS
1108 003670 021527 000121 CMP (R5),#121 ;IS IT TEST 121?
1109 003674 001004 BNE REG23
1110 003676 004467 000410 JSR R4,R1TSH ;IF SO THEN GO AND INITIATE RIGHT SHIFT
1111 003702 004767 000440 60: JSR #7,TST160
1112 003706 010701 60: REG23: SCOPE1
1113 003710 013702 000434 MOV #TEMP1,%2 ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 2
1114 003714 013703 000436 MOV #TEMP2,%2,1 ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 3
1115 003720 000241 CLC
1116 003722 032737 000001 000406 BIT #1,#SPASS ;IS IT AN EVEN PASS ?
1117 003730 001004 BNE 20 ;IF NOT THEN GO TO 20
1118 003732 013704 000440 MOV #TEMP3,R4 ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
1119 003736 073204 ASHC R4,R2 ;USING R2
1120 003740 000402 BR 46
1121 003742 073267 174472 20: ASHC TEMP3,%2 ;ASHC REGISTER 2 BY THE CONTENTS OF TEMP3
1122 003746 006737 000432 40: MFPS #PSWORD ;SAVE PS
1123 003752 123737 000446 000432 CMPB #TEMP6,#PSWORD ;COMPARE PS WITH THE CONTENTS OF TEMP6
1124 003760 001403 BEQ .+10
1125 003762 004767 012452 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1126 ;WRONG PS
1127 003766 000046 46
1128 003770 005237 000430 INC #COUNT
1129 003774 023702 000442 CMP #TEMP4,%2 ;IS THE RESULT IN R2 SAME AS TEMP4?
1130 004000 001403 BEQ .+10
1131 004002 004767 012432 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1132 ;WRONG RESULT IN R2
1133 004006 000047 47
1134 004010 023703 000444 CMP #TEMP5,%3 ;IS THE RESULT IN R3 SAME AS TEMP5?
1135 004014 001403 BEQ .+10 ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMPS
1136 ;AND PS=TEMP6
1137 004016 004767 012416 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1138 ;WRONG RESULT IN R1
1139 004022 000050 50
1140 004024 021537 000430 CMP (R5),#COUNT ;IS TEST NUMBER=COUNTER?
1141 004030 001403 BEQ .+10
1142 004032 004767 012402 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1143 ;NO
1144 004036 000051 51
1145 004040 005215 INC (R5)
1146 004042 021527 000160 CMP (R5),#160 ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
1147 004046 002014 BGE 60 ;YES
1148 004050 005237 000440 INC #TEMP3
1149 004054 000241 CLC
1150 004056 006137 000444 ROL #TEMP5 ;ROTATE TEMPS LEFT BY 1 PLACE
1151 004062 006137 000442 ROL #TEMP4 ;INTRODUCE CARRY FROM TEMPS IN TEMP4
1152 004066 021527 000121 CMP (R5),#121 ;IS IT TEST 121?
1153 004072 001004 BNE REG45
1154 004074 004467 000212 JSR R4,R1TSH ;IF SO THEN GO AND INITIATE RIGHT SHIFT
1155 004100 004767 000242 60: JSR #7,TST160
1156 004104 010701 60: REG45: SCOPE1
1157 004106 010501 MOV R5,R1 ;SAVE R5
    
```

ASHC INSTRUCTION TESTS

```

1158 004110 013704 000434      MOV    @TEMP1,%4      ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 4
1159 004114 013705 000436      MOV    @TEMP2,%4,11  ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 5
1160 004120 000241                CLC
1161 004122 032737 000001 000406    BIT    #1,@#PSPASS    ;IS IT AN EVEN PASS ?
1162 004130 001004                BNE    26              ;IF NOT THEN GO TO 26
1163 004132 013700 000440      MOV    @TEMP3,R0     ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
1164 004136 073400                R0,R4                ;USING R4
1165 004140 000402                BR     46
1166 004142 073467 174272        28: ASHC  TEMP3,%4      ;ASHC REGISTER 4 BY THE CONTENTS OF TEMP3
1167 004146 106737 000432        48: MFPS  @#PSWORD     ;SAVE PS
1168 004152 123737 000446 000432    CMPB  @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
1169 004160 001403                BEQ    ,+10
1170 004162 004767 012252        JSR    PC,$HLT       ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1171                                ;WRONG PS
1172 004166 000052                52
1173 004170 005237 000430        INC    @#COUNT
1174 004174 023704 000442        CMP    @#TEMP4,%4    ;IS THE RESULT IN R4 SAME AS TEMP4?
1175 004200 001403                BEQ    ,+10
1176 004202 004767 012232        JSR    PC,$HLT       ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1177                                ;WRONG RESULT IN R4
1178 004206 000053                53
1179 004210 023705 000444        CMP    @#TEMP5,%5    ;IS THE RESULT IN R5 SAME AS TEMP5?
1180 004214 001403                BEQ    ,+10          ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5
1181                                ;AND PS=TEMP6
1182 004216 004767 012216        JSR    PC,$HLT       ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1183                                ;WRONG RESULT IN R5
1184 004222 000054                54
1185 004224 021137 000430        CMP    (R1),@#COUNT ;IS TEST NUMBER=COUNTER?
1186 004230 001403                BEQ    ,+10
1187 004232 004767 012202        JSR    PC,$HLT       ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1188                                ;NO
1189 004236 000055                55
1190 004240 010105 000440        MOV    P1,R5         ;RESTORE R5
1191 004242 005215                INC    (R5)
1192 004244 021527 000160        CMP    (R5),#160    ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
1193 004250 002011                BGE    68             ;YES
1194 004252 005237 000440        INC    @#TEMP3

```

ASHC INSTRUCTION TESTS

```

1195 004256 000241                CLC
1196 004260 006137 000444        ROL    @#TEMP5        ;ROTATE TEMPS LEFT BY 1 PLACE
1197 004264 006137 000442        ROL    @#TEMP4        ;INTRODUCE CARRY FROM TEMPS IN TEMP4
1198 004270 021527 000121        CMP    (R5),#121     ;IS IT TEST 121?
1199 004274 001004                BNE    86
1200 004276 004467 000010        JSR    R4,RITSH      ;IF SO THEN GO AND INITIATE RIGHT SHIFT
1201 004302 004767 000040        68: JSR    $7,TST160
1202 004306 000167 177172        86: JMP    REG01
1203 004312 022424                RITSH: CMP    (R4)+,(R4)+  ;MAKE R4 POINT TO THE NEXT REG TAG
1204 004314 012737 040000 000434    MOV    #40000,@#TEMP1 ;TEMP1=4000
1205 004322 005037 000436        CLR    @#TEMP2        ;TEMP2=0
1206 004326 012737 177742 000440    MOV    #-30,@#TEMP3   ;TEMP3=-30
1207 004334 005037 000442        CLR    @#TEMP4        ;TEMP4=0
1208 004340 005237 000444        INC    @#TEMP5        ;TEMP5=1
1209 004344 000204                RTS    R4
1210 004346 021527 000160        TST160: CMP    (R5),#160 ;IS IT TEST 160
1211 004352 001010                BNE    TST161        ;IF NOT THEN TRY TEST 161
1212 004354 005037 000434        CLK    @#TEMP1        ;0 0 SHIFTED BY 0
1213 004360 005037 000442        CLR    @#TEMP4        ;IS EQUAL TO 0 0
1214 004364 012737 000004 000446    MOV    #4,@#TEMP6     ;AND PS=4
1215 004372 000207                RTS    %7
1216 004374 021527 000161        TST161: CMP    (R5),#161 ;IS IT TEST 161
1217 004400 001004                BNE    TST162
1218 004402 012737 177746 000440    MOV    #-32,@#TEMP3   ;0 0 SHIFTED BY -32=0 0, PS=4
1219 004410 000207                RTS    %7
1220 004412 021527 000162        TST162: CMP    (R5),#162 ;IS IT TEST 162
1221 004416 001004                BNE    TST163        ;IF NOT THEN TRY TEST 163
1222 004420 012737 000032 000440    MOV    #32,@#TEMP3   ;0 0 SHIFTED BY 32=0 0, PS=4
1223 004426 000207                RTS    %7
1224 004430 021527 000163        TST163: CMP    (R5),#163 ;IS IT TEST 163?
1225 004434 001016                BNE    TST164        ;IF NOT THEN TRY TEST 164
1226 004436 012737 052525 000434    MOV    #52525,@#TEMP1 ;52525 0
1227 004444 012737 177760 000440    MOV    #-16,@#TEMP3  ;SHIFTED BY -16.
1228 004452 005037 000442        CLR    @#TEMP4
1229 004456 012737 052525 000444    MOV    #52525,@#TEMP5 ;IS EQUAL TO 0 52525
1230 004464 005037 000446        CLR    @#TEMP6        ;AND PS = 0
1231 004470 000207                RTS    %7
1232 004472 021527 000164        TST164: CMP    (R5),#164 ;IS IT TEST 164?
1233 004476 001014                BNE    TST165        ;IF NOT THEN TRY TEST 165
1234 004500 012737 125252 000434    MOV    #125252,@#TEMP1 ;125252 0 SHIFTED BY -16.
1235 004506 005337 000442        DEC    @#TEMP4
1236 004512 012737 125252 000444    MOV    #125252,@#TEMP5 ;IS EQUAL TO -1 125252
1237 004520 012737 000010 000446    MOV    #10,@#TEMP6   ;AND PS=10
1238 004526 000207                RTS    %7
1239 004530 021527 000165        TST165: CMP    (R5),#165 ;IS IT TEST 165?
1240 004534 001007                BNE    TST166        ;IF NOT THEN TRY TEST 166
1241 004536 012737 177777 000434    MOV    #-1,@#TEMP1   ;-1 0 SHIFTED BY -16
1242 004544 012737 177777 000444    MOV    #-1,@#TEMP5   ;IS EQUAL TO -1 -1, AND PS=10
1243 004552 000207                RTS    %7
1244 004554 021527 000166        TST166: CMP    (R5),#166 ;IS IT TEST 166?
1245 004560 001011                BNE    TST167        ;IF NOT THEN TRY TEST 167
1246 004562 012737 100000 000434    MOV    #100000,@#TEMP1 ;100000 0
1247 004570 012737 177740 000440    MOV    #-32,@#TEMP3  ;SHIFTED BY -32 IS EQUAL TO -1 -1
1248 004576 005237 000446        INC    @#TEMP6        ;AND PS=11
1249 004602 000207                RTS    %7
1250 004604 021527 000167        TST167: CMP    (R5),#167 ;IS IT TEST 167?

```

```
1251 004610 001014 BNE TST170 ;IF NOT THEN TRY TEST 170
1252 004612 005037 CLR #TEMP1
1253 004616 005337 DEC #TEMP2 ;0 -1
1254 004622 012737 MOV #16,,#TEMP3 ;SHIFTED BY 16,
1255 004630 005037 CLK #TEMP5 ;IS EQUAL TO -1 0
1256 004634 005237 INC #TEMP6 ;AND PS=12
1257 004640 000207 RTS #7
1258 004642 021527 TST170: CMP (R5),#170 ;IS IT TEST 170?
1259 004646 001007 BNE TST171 ;IF NOT THEN TRY TEST 171
1260 004650 012737 MOV #125252,#TEMP2 ;0 125252 SHIFTED BY 16
1261 004656 012737 MOV #125252,#TEMP4 ;IS EQUAL TO 125252 0, AND PS=12
1262 004664 000207 RTS #7
1263 004666 021527 TST171: CMP (R5),#171 ;IS IT TEST 171?
1264 004672 001010 BNE TST172 ;IF NOT THEN TRY TEST 172
1265 004674 005337 DEC #TEMP3 ;0 125252 SHIFTED BY 15
1266 004700 012737 MOV #52525,#TEMP4 ;IS EQUAL TO 52525 0
1267 004706 005037 CLR #TEMP6 ;AND PS=0
1268 004712 000207 RTS #7
1269 004714 021527 TST172: CMP (R5),#172 ;IS IT TEST 172?
1270 004720 001006 BNE TST173 ;IF NOT THEN TRY TEST 173
1271 004722 012737 MOV #52525,#TEMP2 ;0 52525
1272 004730 005237 INC #TEMP3 ;SHIFTED BY 16, IS EQUAL TO 52525 0, AND PS=0
1273 004734 000207 RTS #7
1274 004736 021527 TST173: CMP (R5),#173 ;IS IT TEST 173?
1275 004742 001014 BNE TST174 ;IF NOT THEN TRY TEST 174
1276 004744 012737 MOV #-1,#TEMP2 ;0 -1
1277 004752 005337 DEC #TEMP3 ;SHIFTED BY 15,
1278 004756 012737 MOV #77777,#TEMP4
1279 004764 012737 MOV #100000,#TEMP5 ;IS EQUAL TO 77777 100000, AND PS=0
1280 004772 000207 RTS #7
1281 004774 021527 TST174: CMP (R5),#174 ;IS IT TEST 174?
1282 005000 001013 BNE TST175 ;IF NOT THEN TRY TEST 175
1283 005002 012737 MOV #100000,#TEMP1
1284 005010 005337 DEC #TEMP2 ;100000 -2 SHIFTED BY 15,
1285 005014 005037 CLR #TEMP5 ;IS EQUAL TO 77777 0
1286 005020 012737 MOV #2,#TEMP6 ;AND PS=2
1287 005026 000207 RTS #7
1288 005030 021527 TST175: CMP (R5),#175 ;IS IT TEST 175?
1289 005034 001015 BNE ENT176 ;IF NOT THEN TRY TEST 176
1290 005036 012737 MOV #-1,#TEMP1
1291 005044 005037 CLR #TEMP2 ;-1 0
1292 005050 005237 INC #TEMP3 ;SHIFTED BY 16,
1293 005054 005037 CLR #TEMP4 ;IS EQUAL TO 0 0
1294 005060 012737 MOV #7,#TEMP6 ;AND PS=7
1295 005066 000207 RTS #7
1296 005070 021527 ENT176: CMP (R5),#176 ;IS THE PROGRAM ENTERING TEST 176?
1297 005074 001403 BEQ #+10
1298 005076 004767 JSK PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1299 ;TEST NUMBER GOOFED
1300 005102 000056 56
1301
1302 005104 005726 TST (SP)+ ;RESTORE STACK POINTER
1303
```

```
1304 ;*****
1305 ;TEST:176 1 SHIFTED BY 0, = 400 PS = 0
1306 ;*****
1307
1308 005106 010701 TST176: SCOPE1
1309 005110 012701 MOV #DUMMY,#1 ;LOAD R1 WITH DUMMY
1310 005114 012701 MOV #1,#11 ;LOAD R11 WITH 1
1311 005120 000241 CLC
1312 005122 073127 ASHC #8,,#1 ;SHIFT R1,R11 BY 8,
1313 005126 106737 MPPS #PSWORD ;SAVE PS
1314 005132 122737 CMPB #0,#PSWORD ;IS THE PS 0?
1315 005140 001403 BEQ #+10
1316 005142 004767 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1317 ;THE PS IS NOT EQUAL TO 0
1318 57
1319 005150 022701 CMP #400,#1 ;IS THE RESULT 400?
1320 005154 001403 BEQ #+10
1321 005156 004767 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1322 ;R1 IS NOT EQUAL TO 400
1323 60
1324 005162 000060 60
1325 005164 021527 CMP (R5),#176 ;IS $TESTN = #176?
1326 005170 001403 BEQ #+10 ;IF NOT THEN GO TO HLT
1327 005172 004767 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1328 ;TEST IS IN WRONG SEQUENCE
1329 61
1330 INC (R5)
1331
1332 ;*****
1333 ;TEST:177 -1 SHIFTED BY 15, = 100000 PS = 11
1334 ;*****
1335
1336 005202 010701 TST177: SCOPE1
1337 005204 012703 MOV #DUMMY,#3 ;LOAD R3 WITH DUMMY
1338 005210 012703 MOV #-1,#311 ;LOAD R311 WITH -1
1339 005214 000241 CLC
1340 005216 073327 ASHC #15,,#3 ;SHIFT R3,R311 BY 15,
1341 005222 106737 MPPS #PSWORD ;SAVE PS
1342 005226 122737 CMPB #11,#PSWORD ;IS THE PS 11?
1343 005234 001403 BEQ #+10
1344 005236 004767 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1345 ;THE PS IS NOT EQUAL TO 11
1346 62
1347 005242 000062 62
1348 005244 022703 CMP #100000,#3 ;IS THE RESULT 100000?
1349 005250 001403 BEQ #+10
1350 005252 004767 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1351 ;R3 IS NOT EQUAL TO 100000
1352 63
1353 005256 021527 CMP (R5),#177 ;IS $TESTN = #177?
1354 005264 001403 BEQ #+10 ;IF NOT THEN GO TO HLT
1355 005266 004767 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1356 ;TEST IS IN WRONG SEQUENCE
1357 64
1358 INC (R5)
1359
```

```

1360 ;*****
1361 ;TEST:200 52525 SHIFTED BY 0 = 52525 PS = 0
1362 ;*****
1363
1364 005276 010701 TST200: SCOPE1
1365 005300 010501 MOV R5,R1 ;SAVE R5
1366 005302 012705 000000 MOV #DUMMY,%5 ;LOAD R5 WITH DUMMY
1367 005306 012705 052525 MOV #52525,%511 ;LOAD R511 WITH 52525
1368 005312 000241 CLC
1369 005314 073527 000000 ASHC #0,%5 ;SHIFT R5,R511 BY 0
1370 005320 106737 000432 MFPS #PSWORD ;SAVE PS
1371 005324 122737 000000 000432 CMPB #0,#PSWORD ;IS THE PS 0?
1372 005332 001403 BEQ ,+10
1373 005334 004767 011100 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1374 ;THE PS IS NOT EQUAL TO 0
1375 005340 000065 65
1376 005342 022705 052525 CMP #52525,%5 ;IS THE RESULT 52525?
1377 005346 001403 BEQ ,+10
1378 005350 004767 011064 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1379 ;R5 IS NOT EQUAL TO 52525
1380 005354 000066 66
1381 005356 010105 MOV R1,R5 ;RESTORE R5
1382 005360 021527 000200 CMP (R5),#200 ;IS $TESTN = #200?
1383 005364 001403 BEQ ,+10 ;IF NOT THEN GO TO HLT
1384 005366 004767 011046 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1385 ;TEST IS IN WRONG SEQUENCE
1386 005372 000067 67
1387 005374 005215 INC (R5)
1388
1389
1390 ;*****
1391 ;TEST:201 20010 SHIFTED BY -13, = 101 PS = 0
1392 ;*****
1393
1394 005376 010701 TST201: SCOPE1
1395 005400 012701 000000 MOV #DUMMY,%1 ;LOAD R1 WITH DUMMY
1396 005404 012701 020010 MOV #20010,%111 ;LOAD R111 WITH 20010
1397 005410 000241 CLC
1398 005412 073127 177763 ASHC #-13,%1 ;SHIFT R1,R111 BY -13.
1399 005416 136737 000432 MFPS #PSWORD ;SAVE PS
1400 005422 122737 000000 000432 CMPB #0,#PSWORD ;IS THE PS 0?
1401 005430 001403 BEQ ,+10
1402 005432 004767 011002 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1403 ;THE PS IS NOT EQUAL TO 0
1404 005436 000070 70
1405 005440 022701 000101 CMP #101,%1 ;IS THE RESULT 101?
1406 005444 001403 BEQ ,+10
1407 005446 004767 010766 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1408 ;R1 IS NOT EQUAL TO 101
1409 005452 000071 71
1410 005454 021527 000201 CMP (R5),#201 ;IS $TESTN = #201?
1411 005460 001403 BEQ ,+10 ;IF NOT THEN GO TO HLT
1412 005462 004767 010752 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1413 ;TEST IS IN WRONG SEQUENCE
1414 005466 000072 72
1415 005470 005215 INC (R5)

```

1416
 1417

```

1418 ;*****
1419 ;TEST:202 -1 SHIFTED BY 16, = 0 PS = 11
1420 ;*****
1421
1422 TST202: SCOPE1
1423 MOV #DUMMY,%3 ;LOAD R3 WITH DUMMY
1424 MOV #-1,%311 ;LOAD R311 WITH -1
1425 CLC
1426 ASHC #16,%3 ;SHIFT R3,R311 BY 16.
1427 MFPS #PSWORD ;SAVE PS
1428 CMPB #1,%PSWORD ;IS THE PS 11?
1429 BEQ ,+10
1430 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1431 ;THE PS IS NOT EQUAL TO 11
1432
1433 73
1434 CMP #0,%3 ;IS THE RESULT 0?
1435 BEQ ,+10
1436 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1437 ;R3 IS NOT EQUAL TO 0
1438
1439 74
1440 CMP (R5),#202 ;IS $TESTN = #202?
1441 BEQ ,+10 ;IF NOT THEN GO TO HLT
1442 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1443 ;TEST IS IN WRONG SEQUENCE
1444
1445 75
1446 INC (R5)
1447
1448 ;*****
1449 ;TEST:203 1 SHIFTED BY -1 = 100000 PS = 1
1450 ;*****
1451 TST203: SCOPE1
1452 MOV R5,R1 ;SAVE R5
1453 MOV #DUMMY,%5 ;LOAD R5 WITH DUMMY
1454 MOV #1,%511 ;LOAD R511 WITH 1
1455 CLC
1456 ASHC #-1,%5 ;SHIFT R5,R511 BY -1
1457 MFPS #PSWORD ;SAVE PS
1458 CMPB #1,%PSWORD ;IS THE PS 1?
1459 BEQ ,+10
1460 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1461 ;THE PS IS NOT EQUAL TO 1
1462
1463 76
1464 CMP #100000,%5 ;IS THE RESULT 100000?
1465 BEQ ,+10
1466 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1467 ;R5 IS NOT EQUAL TO 100000
1468
1469 77
1470 MOV R1,R5 ;RESTORE R5
1471 CMP (R5),#203 ;IS $TESTN = #203?
1472 BEQ ,+10 ;IF NOT THEN GO TO HLT
1473 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1474 ;TEST IS IN WRONG SEQUENCE
1475
1476 100
1477 INC (R5)
  
```

1474
 1475


```

1476 ;*****
1477 ;TEST:204 125252 SHIFTED BY -16, = 125252 PS = 11
1478 ;*****
1479
1480 TST204: SCOPE1
1481 005666 010701 MOV #DUMMY,%1 ;LOAD R1 WITH DUMMY
1482 005670 012701 000000 MOV #125252,%111 ;LOAD R111 WITH 125252
1483 005674 012701 125252 CLC
1484 005700 000241 ASHC #-16,%1 ;SHIFT R1,R111 BY -16,
1485 005702 073127 177760 MFPS #0#PASSWORD ;SAVE PS
1486 005706 106737 000432 000011 000432 CMPB #11,0#PASSWORD ;IS THE PS 11?
1487 005712 122737 000011 BEQ ,+10
1488 005720 001403 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1489 005722 004767 010512 ;THE PS IS NOT EQUAL TO 11
1490
1491 005726 000101 101
1492 005730 022701 125252 CMP #125252,%1 ;IS THE RESULT 125252?
1493 005734 001403 BEQ ,+10
1494 005736 004767 010476 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1495 ;R1 IS NOT EQUAL TO 125252
1496
1497 005742 000102 102
1498 005744 021527 000204 CMP (R5),#204 ;IS $TESTN = #204?
1499 005750 001403 BEQ ,+10 ;IF NOT THEN GO TO HLT
1500 005752 004767 010462 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1501 ;TEST IS IN WRONG SEQUENCE
1502
1503 INC (R5)
1504
1505 ;*****
1506 ;TEST:205 125252 125252 SHIFTED BY 21, = 52500 000000 PS = 3
1507 ;*****
1508 TST205: SCOPE1
1509 005762 010701 MOV #125252,%2 ;LOAD R2 WITH 125252
1510 005764 012702 125252 MOV #125252,%211 ;LOAD R211 WITH 125252
1511 005770 000241 CLC
1512 005776 073227 000025 ASHC #21,%2 ;SHIFT R2,R211 BY 21,
1513 006002 106737 000432 MFPS #0#PASSWORD ;SAVE PS
1514 006006 122737 000003 000432 CMPB #3,0#PASSWORD ;IS THE PS 3?
1515 006014 001403 BEQ ,+10
1516 006016 004767 010416 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1517 ;THE PS IS NOT EQUAL TO 3
1518
1519 006022 000104 104
1520 006024 022702 052500 CMP #52500,%2 ;IS THE RESULT 52500?
1521 006030 001403 BEQ ,+10
1522 006032 004767 010402 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1523 ;R2 IS NOT EQUAL TO 52500
1524
1525 006036 000105 105
1526 006040 022703 000000 CMP #000000,%211 ;IS THE RESULT 000000?
1527 006044 001403 BEQ ,+10
1528 006046 004767 010366 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1529 ;R2:1 IS NOT EQUAL TO 000000
1530
1531 006052 000106 106
1532 006054 021527 000205 CMP (R5),#205 ;IS $TESTN = #205?
1533 006060 001403 BEQ ,+10 ;IF NOT THEN GO TO HLT
1534 006062 004767 010352 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE

```

```

1532 ;TEST IS IN WRONG SEQUENCE
1533 006066 000107 107
1534 006070 005215 INC (R5)
1535
1536
1537
1538 006072 012702 177771 MOV #-7,%2
1539 006076 012703 000454 MOV #51,%3
1540 006102 012704 000456 MOV #52,%4
1541

```

```

1542 ;*****
1543 ;TEST:206 125252 125252 SHIFTED BY S1 = 177525 52525 PS = 10
1544 ;*****
1545
1546 #06106 010701 TST06: SCOPE1
1547 006110 012700 MOV #125252,%0 ;LOAD R0 WITH 125252
1548 006114 012701 MOV #125252,%011 ;LOAD R011 WITH 125252
1549 006120 000241 CLC
1550 006122 073067 ASHC S1,%0 ;SHIFT R0,R011 BY S1
1551 006126 106737 MFPB #0,0 ;SAVE PS
1552 006132 122737 CMFB #10,%0 ;IS THE PS 10?
1553 006140 001403 BEQ .+10
1554 006142 004767 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1555 ;THE PS IS NOT EQUAL TO 10
1556 #06146 000110 110
1557 006150 022700 CMP #177525,%0 ;IS THE RESULT 177525?
1558 006154 001403 BEQ .+10
1559 006156 004767 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1560 ;R0 IS NOT EQUAL TO 177525
1561 006162 000111 111
1562 006164 022701 CMP #52525,%011 ;IS THE RESULT 52525?
1563 006170 001403 BEQ .+10
1564 006172 18: JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1565 006172 004767 ;R011 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
1566
1567 006176 000112 112
1568 006200 021527 CMP (R5),#206 ;IS THE $TESTN = #206?
1569 006204 001372 BNE 18 ;IF NOT THEN GO TO HLT ABOVE
1570 006206 005215 INC (R5)
1571
1572
1573 ;*****
1574 ;TEST:207 125252 125252 SHIFTED BY 0S2 = 177525 52525 PS = 10
1575 ;*****
1576
1577 006210 010701 TST07: SCOPE1
1578 006212 012700 MOV #125252,%0 ;LOAD R0 WITH 125252
1579 006216 012701 MOV #125252,%011 ;LOAD R011 WITH 125252
1580 006222 000241 CLC
1581 006224 073077 ASHC 0S2,%0 ;SHIFT R0,R011 BY 0S2
1582 006230 106737 MFPB #0,0 ;SAVE PS
1583 006234 122737 CMFB #10,%0 ;IS THE PS 10?
1584 006242 001403 BEQ .+10
1585 006244 004767 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1586 ;THE PS IS NOT EQUAL TO 10
1587 006250 000113 113
1588 006252 022700 CMP #177525,%0 ;IS THE RESULT 177525?
1589 006256 001403 BEQ .+10
1590 006260 004767 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1591 ;R0 IS NOT EQUAL TO 177525
1592 006264 000114 114
1593 006266 022701 CMP #52525,%011 ;IS THE RESULT 52525?
1594 006272 001403 BEQ .+10
1595 006274 18: JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1596 006274 004767 ;R011 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
1597

```

```

1598 006300 000115 115
1599 006302 021527 CMP (R5),#207 ;IS THE $TESTN = #207?
1600 006306 001372 BNE 18 ;IF NOT THEN GO TO HLT ABOVE
1601 006310 005215 INC (R5)
1602
1603

```

```
1604 ;*****  
1605 ;TEST:210 125252 125252 SHIFTED BY 0#S1 = 177525 52525 PS = 10  
1606 ;*****  
1607  
1608 006312 010701 TST210: SCOPE1  
1609 006314 012700 MOV #125252,%0 ;LOAD R0 WITH 125252  
1610 006320 012701 MOV #125252,%011 ;LOAD R011 WITH 125252  
1611 006324 000241 CLC  
1612 006326 073037 000454 ASHC 0#S1,%0 ;SHIFT R0,R011 BY 0#S1  
1613 006332 106737 000432 MFPS 0#PSWORD ;SAVE PS  
1614 006336 122737 000010 000432 CMPB #10,0#PSWORD ;IS THE PS 10?  
1615 006344 001403 BEQ ,+10  
1616 006346 004767 010066 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
1617 ;THE PS IS NOT EQUAL TO 10  
1618 006352 000116 116  
1619 006354 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?  
1620 006360 001403 BEQ ,+10  
1621 006362 004767 010052 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
1622 ;R0 IS NOT EQUAL TO 177525  
1623 006366 000117 117  
1624 006370 022701 052525 CMP #52525,%011 ;IS THE RESULT 52525?  
1625 006374 001403 BEQ ,+10  
1626 006376 162  
1627 006376 004767 010036 161: JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
1628 ;R011 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE  
1629 006402 000120 120  
1630 006404 021527 000210 CMP (R5),#210 ;IS THE $TESTN = #210?  
1631 006410 001372 BNE 16 ;IF NOT THEN GO TO HLT ABOVE  
1632 006412 005215 INC (R5)  
1633  
1634  
1635 ;*****  
1636 ;TEST:211 125252 125252 SHIFTED BY (3) = 177525 52525 PS = 10  
1637 ;*****  
1638  
1639 006414 010701 TST211: SCOPE1  
1640 006416 012700 MOV #125252,%0 ;LOAD R0 WITH 125252  
1641 006422 012701 MOV #125252,%011 ;LOAD R011 WITH 125252  
1642 006426 000241 CLC  
1643 006430 073013 ASHC (3),%0 ;SHIFT R0,R011 BY (3)  
1644 006432 106737 000432 MFPS 0#PSWORD ;SAVE PS  
1645 006436 122737 000010 000432 CMPB #10,0#PSWORD ;IS THE PS 10?  
1646 006444 001403 BEQ ,+10  
1647 006446 004767 007766 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
1648 ;THE PS IS NOT EQUAL TO 10  
1649 006452 000121 121  
1650 006454 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?  
1651 006460 001403 BEQ ,+10  
1652 006462 004767 007752 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
1653 ;R0 IS NOT EQUAL TO 177525  
1654 006466 000122 122  
1655 006470 022701 052525 CMP #52525,%011 ;IS THE RESULT 52525?  
1656 006474 001403 BEQ ,+10  
1657 006476 161: JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE  
1658 006476 004767 007736 ;R011 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE  
1659
```

```
1660 006502 000123 123  
1661 006504 021527 000211 CMP (R5),#211 ;IS THE $TESTN = 211?  
1662 006510 001372 BNE 16 ;IF NOT THEN GO TO HLT ABOVE  
1663 006512 005215 INC (R5)  
1664  
1665
```

```

1666 ;*****
1667 ;TEST:212 125252 125252 SHIFTED BY (3)+ = 177525 52525 PS = 10
1668 ;*****
1669
1670 006514 010701 TST212: SCOPE1
1671 006516 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252
1672 006522 012701 125252 MOV #125252,%011 ;LOAD R011 WITH 125252
1673 006526 000241 CLC
1674 006530 073023 ASHC (3)+,%0 ;SHIFT R0,R011 BY (3)+
1675 006532 106737 000432 MFPS #PSWORD ;SAVE PS
1676 006536 122737 000010 000432 CMPB #10,#PSWORD ;IS THE PS 10?
1677 006544 001403 BEQ ,+10
1678 006546 004767 007666 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1679 ;THE PS IS NOT EQUAL TO 10
1680 006552 000124 124
1681 006554 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
1682 006560 001403 BEQ ,+10
1683 006562 004767 007652 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1684 ;R0 IS NOT EQUAL TO 177525
1685 006566 000125 125
1686 006570 022701 052525 CMP #52525,%011 ;IS THE RESULT 52525?
1687 006574 001403 BEQ ,+10
1688 006576 101
1689 006576 004767 007636 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1690 ;R011 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
1691 006602 000126 126
1692 006604 021527 000212 CMP (R5),#212 ;IS THE $TESTN = #212?
1693 006610 001372 BNE 10 ;IF NOT THEN GO TO HLT ABOVE
1694 006612 005215 INC (R5)
1695
1696
1697 ;*****
1698 ;TEST:213 125252 125252 SHIFTED BY -(3) = 177525 52525 PS = 10
1699 ;*****
1700
1701 006614 010701 IST213: SCOPE1
1702 006616 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252
1703 006622 012701 125252 MOV #125252,%011 ;LOAD R011 WITH 125252
1704 006626 000241 CLC
1705 006630 073043 ASHC -(3),%0 ;SHIFT R0,R011 BY -(3)
1706 006632 106737 000432 MFPS #PSWORD ;SAVE PS
1707 006636 122737 000010 000432 CMPB #10,#PSWORD ;IS THE PS 10?
1708 006644 001403 BEQ ,+10
1709 006646 004767 007566 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1710 ;THE PS IS NOT EQUAL TO 10
1711 006652 000127 127
1712 006654 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
1713 006660 001403 BEQ ,+10
1714 006662 004767 007552 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1715 ;R0 IS NOT EQUAL TO 177525
1716 006666 000130 130
1717 006670 022701 052525 CMP #52525,%011 ;IS THE RESULT 52525?
1718 006674 001403 BEQ ,+10
1719 006676 101
1720 006676 004767 007536 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1721 ;R011 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE

```

```

1722 006702 000131 131
1723 006704 021527 000213 CMP (R5),#213 ;IS THE $TESTN = #213?
1724 006710 001372 BNE 10 ;IF NOT THEN GO TO HLT ABOVE
1725 006712 005215 INC (R5)
1726
1727

```

```

1728 ;*****
1729 ;TEST:214 125252 125252 SHIFTED BY 2(4) = 177252 125252 PS = 11
1730 ;*****
1731
1732 006714 010701 TST214: SCOPE1
1733 006716 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252
1734 006722 012701 125252 MOV #125252,%011 ;LOAD R011 WITH 125252
1735 006726 000241 CLC
1736 006730 073064 000000 ASHC 2(4),%0 ;SHIFT R0,R011 BY 2(4)
1737 006734 106737 000432 MFPS 0*PSWORD ;SAVE PS
1738 006740 122737 000011 000432 CMPB #11,0*PSWORD ;IS THE PS 11?
1739 006746 001403 BEQ ,+10
1740 006750 004767 007464 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1741 ;THE PS IS NOT EQUAL TO 11
1742 006754 000132 132
1743 006756 022700 177252 CMP #177252,%0 ;IS THE RESULT 177252?
1744 006762 001403 BEQ ,+10
1745 006764 004767 007450 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1746 ;R0 IS NOT EQUAL TO 177252
1747 006770 000133 133
1748 006772 022701 125252 CMP #125252,%011 ;IS THE RESULT 125252?
1749 006776 001403 BEQ ,+10
1750 007000 18:
1751 007000 004767 007434 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1752 ;R011 IS NOT EQUAL TO 125252 OR INCORRECT SEQUENCE
1753 007004 000134 134
1754 007006 021527 000214 CMP (R5),#214 ;IS THE $TESTN = #214?
1755 007012 001372 BNE 18 ;IF NOT THEN GO TO HLT ABOVE
1756 007014 005215 INC (R5)
1757
1758 ;*****
1759 ;TEST:215 125252 125252 SHIFTED BY 0(4) = 177525 52525 PS = 10
1760 ;*****
1761
1762 TST215: SCOPE1
1763 007016 010701 125252 MOV #125252,%0 ;LOAD R0 WITH 125252
1764 007020 012700 125252 MOV #125252,%011 ;LOAD R011 WITH 125252
1765 007024 000241 CLC
1766 007030 000241 ASHC 0(4),%0 ;SHIFT R0,R011 BY 0(4)
1767 007032 073074 000000 MFPS 0*PSWORD ;SAVE PS
1768 007036 106737 000432 000432 CMPB #10,0*PSWORD ;IS THE PS 10?
1769 007042 122737 000010 000432 BEQ ,+10
1770 007050 001403 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1771 007052 004767 007362 ;THE PS IS NOT EQUAL TO 10
1772 007056 000135 135
1773 007060 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
1774 007064 001403 BEQ ,+10
1775 007066 004767 007346 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1776 ;R0 IS NOT EQUAL TO 177525
1777 007072 000136 136
1778 007074 022701 052525 CMP #52525,%011 ;IS THE RESULT 52525?
1779 007100 001403 BEQ ,+10
1780 007102 004767 007332 18: JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1781 ;R011 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
1782
1783

```

```

1784 007106 000137 137
1785 007110 021527 000215 CMP (R5),#215 ;IS THE $TESTN = #215?
1786 007114 001372 BNE 18 ;IF NOT THEN GO TO HLT ABOVE
1787 007116 005215 INC (R5)
1788
1789

```

```

1790 ;*****
1791 ;TEST:216 125252 125252 SHIFTED BY 0(4)+ = 177525 52525 PS = 10
1792 ;*****
1793
1794 007120 010701 TST216: SCOPE1
1795 007122 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252
1796 007126 012701 125252 MOV #125252,%011 ;LOAD R011 WITH 125252
1797 007132 000241 CLC
1798 007134 073034 ASHC 0(4)+,%0 ;SHIFT R0,R011 BY 0(4)+
1799 007136 106737 000432 MFPS 0#PSWORD ;SAVE PS
1800 007142 122737 000010 000432 CMPB #10,0#PSWORD ;IS THE PS 10?
1801 007150 001403 BEQ ,+10
1802 007152 004767 007262 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1803 ;THE PS IS NOT EQUAL TO 10
1804 007156 000140 140
1805 007160 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
1806 007164 001403 BEQ ,+10
1807 007166 004767 007246 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1808 ;R0 IS NOT EQUAL TO 177525
1809 007172 000141 141
1810 007174 022701 052525 CMP #52525,%011 ;IS THE RESULT 52525?
1811 007200 001403 BEQ ,+10
1812 007202 18: JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1813 007202 004767 007232 ;R011 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
1814
1815 007206 000142 142
1816 007210 021527 000216 CMP (R5),#216 ;IS THE $TESTN = #216?
1817 007214 001372 BNE 18 ;IF NOT THEN GO TO HLT ABOVE
1818 007216 005215 INC (R5)
1819
1820 ;*****
1821 ;TEST:217 125252 125252 SHIFTED BY 0(4) = 177525 52525 PS = 10
1822 ;*****
1823
1824
1825 007220 010701 TST217: SCOPE1
1826 007222 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252
1827 007226 012701 125252 MOV #125252,%011 ;LOAD R011 WITH 125252
1828 007232 000241 CLC
1829 007234 073054 ASHC 0-(4),%0 ;SHIFT R0,R011 BY 0-(4)
1830 007236 106737 000432 MFPS 0#PSWORD ;SAVE PS
1831 007242 122737 000010 000432 CMPB #10,0#PSWORD ;IS THE PS 10?
1832 007250 001403 BEQ ,+10
1833 007252 004767 007162 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1834 ;THE PS IS NOT EQUAL TO 10
1835 007256 000143 143
1836 007260 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
1837 007264 001403 BEQ ,+10
1838 007266 004767 007146 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1839 ;R0 IS NOT EQUAL TO 177525
1840 007272 000144 144
1841 007274 022701 052525 CMP #52525,%011 ;IS THE RESULT 52525?
1842 007300 001403 BEQ ,+10
1843 007302 18: JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1844 007302 004767 007132 ;R011 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
1845

```

```

1846 007306 000145 145
1847 007310 021527 000217 CMP (R5),#217 ;IS THE $TESTN = #217?
1848 007314 001372 BNE 18 ;IF NOT THEN GO TO HLT ABOVE
1849 007316 005215 INC (R5)
1850
1851
1852
1853
1854
1855
1856
1857
1858

```

```

1859 ;*****
1860 ; MUL INSTRUCTION TESTS
1861 ;*****
1862
1863
1864
1865
1866 ;*****
1867 ;TEST:220 MUL 1 * #0 = 0 0 PS = 4
1868 ;*****
1869
1870 007320 010701 TST220: SCOPE
1871 007322 012700 MOV #1,%0 ;LOAD MULTIPLICAND WITH 1
1872 007326 070027 000001 MUL #0,%0 ;MULTIPLY 1 * #0
1873 007332 106737 000432 MFPS #PSWORD ;SAVE PS
1874 007336 122737 000004 000432 CMPB #4,#PSWORD ;IS PS = 4
1875 007344 001403 BEQ ,+10
1876 007346 004767 007066 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1877 ;PS IS WRONG
1878 007352 000146 146
1879 007354 022700 000000 CMP #0,%0 ;IS HIGH ORDER = 0
1880 007360 001403 BEQ ,+10
1881 007362 004767 007052 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1882 ;HIGH ORDER IS WRONG
1883 007366 000147 147
1884 007370 022701 000000 CMP #0,%011 ;IS LOW ORDER = 0
1885 007374 001403 BEQ ,+10
1886 007376 004767 007036 18: JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1887 007376 004767 007036 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
1888
1889 007402 000150 150
1890 007404 021527 000220 CMP (R5),#220
1891 007410 001372 BNE 18 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
1892 007412 005215 INC (R5)
1893
1894

```

```

1895 ;*****
1896 ;TEST:221 MUL -1 * #1 = -1 -1 PS = 10
1897 ;*****
1898
1899 007414 010701 TST221: SCOPE
1900 007416 012700 177777 MOV #-1,%0 ;LOAD MULTIPLICAND WITH -1
1901 007422 070027 000001 MUL #1,%0 ;MULTIPLY -1 * #1
1902 007426 106737 000432 MFPS #PSWORD ;SAVE PS
1903 007432 122737 000010 000432 CMPB #10,#PSWORD ;IS PS = 10
1904 007440 001403 BEQ ,+10
1905 007442 004767 006772 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1906 ;PS IS WRONG
1907 007446 000151 151
1908 007450 022700 177777 CMP #-1,%0 ;IS HIGH ORDER = -1
1909 007454 001403 BEQ ,+10
1910 007456 004767 006756 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1911 ;HIGH ORDER IS WRONG
1912 007462 000152 152
1913 007464 022701 177777 CMP #-1,%011 ;IS LOW ORDER = -1
1914 007470 001403 BEQ ,+10
1915 007472 004767 006742 18: JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1916 007472 004767 006742 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
1917
1918 007476 000153 153
1919 007500 021527 000221 CMP (R5),#221
1920 007504 001372 BNE 18 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
1921 007506 005215 INC (R5)
1922
1923

```

```

1924 ;*****
1925 ;TEST:222 MUL 2 * #2 = 0 4 PS = 0
1926 ;*****
1927
1928 007510 010701 TST222: SCOPE
1929 007512 012702 MOV #2,#2 ;LOAD MULTIPLICAND WITH 2
1930 007516 070227 MUL #2,#2 ;MULTIPLY 2 * #2
1931 007522 106737 MFPS #PSWORD ;SAVE PS
1932 007526 122737 CMPB #0,#PSWORD ;IS PS = 0
1933 007534 001403 BEQ ,+10
1934 007536 004767 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1935 ;PS IS WRONG
1936 007542 000154 154
1937 007544 022702 CMP #0,#2 ;IS HIGH ORDER = 0
1938 007550 001403 BEQ ,+10
1939 007552 004767 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1940 ;HIGH ORDER IS WRONG
1941 007556 000155 155
1942 007560 022703 CMP #4,#211 ;IS LOW ORDER = 4
1943 007564 001403 BEQ ,+10
1944 007566 004767 18: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1945 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
1946
1947 007572 000156 156
1948 007574 021527 CMP (R5),#222
1949 007600 001372 BNE 18 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
1950 007602 005215 INC (R5)
1951
1952
    
```

```

1953 ;*****
1954 ;TEST:223 MUL 1000 * #200 = 1 0 PS = 1
1955 ;*****
1956
1957 007604 010701 TST223: SCOPE
1958 007606 010501 MOV R5,R1 ;SAVE R5
1959 007610 012704 MOV #1000,#4 ;LOAD MULTIPLICAND WITH 1000
1960 007614 070427 MUL #200,#4 ;MULTIPLY 1000 * #200
1961 007620 106737 MFPS #PSWORD ;SAVE PS
1962 007624 122737 CMPB #1,#PSWORD ;IS PS = 1
1963 007632 001403 BEQ ,+10
1964 007634 004767 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1965 ;PS IS WRONG
1966 007640 000157 157
1967 007642 022704 CMP #1,#4 ;IS HIGH ORDER = 1
1968 007646 001403 BEQ ,+10
1969 007650 004767 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1970 ;HIGH ORDER IS WRONG
1971 007654 000160 160
1972 007656 022705 CMP #0,#411 ;IS LOW ORDER = 0
1973 007662 001403 BEQ ,+10
1974 007664 004767 18: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1975 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
1976
1977 007670 000161 161
1978 007672 021127 CMP (R1),#223 ;CHECK THE TEST NUMBER
1979 007676 001372 BNE 18 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
1980 007700 010105 MOV R1,R5 ;RESTORE R5
1981 007702 005215 INC (R5)
1982
1983
    
```



```

1984 ;*****
1985 ;TEST:224 MUL 2 * #7777 = 0 17776 PS = 1
1986 ;*****
1987
1988 007704 010701 TST224: SCOPE
1989 007706 012700 MOV #2,%0 ;LOAD MULTIPLICAND WITH 2
1990 007712 070027 MUL #7777,%0 ;MULTIPLY 2 * #7777
1991 007716 106737 MFPS @#PSWORD ;SAVE PS
1992 007722 122737 CMFB #1,@#PSWORD ;IS PS = 1
1993 007730 001403 BEQ ,+10
1994 007732 004767 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
1995 ;PS IS WRONG
1996 007736 000162 162
1997 007740 022700 CMP #0,%0 ;IS HIGH ORDER = 0
1998 007744 001403 BEQ ,+10
1999 007746 004767 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2000 ;HIGH ORDER IS WRONG
2001 007752 000163 163
2002 007754 022701 CMP #17776,%01 ;IS LOW ORDER = 17776
2003 007760 001403 BEQ ,+10
2004 007762 161
2005 007762 004767 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2006 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2007 007766 000164 164
2008 007770 021527 CMP (R5),#224
2009 007774 001372 BNE 16 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2010 007776 005215 INC (R5)
2011
2012
  
```

```

2013 ;*****
2014 ;TEST:225 MUL 777 * #10 = 0 7770 PS = 0
2015 ;*****
2016
2017 010000 010701 TST225: SCOPE
2018 010002 012702 MOV #777,%2 ;LOAD MULTIPLICAND WITH 777
2019 010006 070227 MUL #10,%2 ;MULTIPLY 777 * #10
2020 010012 106737 MFPS @#PSWORD ;SAVE PS
2021 010016 122737 CMFB #0,@#PSWORD ;IS PS = 0
2022 010024 001403 BEQ ,+10
2023 010026 004767 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2024 ;PS IS WRONG
2025 010032 000165 165
2026 010034 022702 CMP #0,%2 ;IS HIGH ORDER = 0
2027 010040 001403 BEQ ,+10
2028 010042 004767 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2029 ;HIGH ORDER IS WRONG
2030 010046 000166 166
2031 010050 022703 CMP #7770,%211 ;IS LOW ORDER = 7770
2032 010054 001403 BEQ ,+10
2033 010056 161
2034 010056 004767 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2035 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2036 010062 000167 167
2037 010064 021527 CMP (R5),#225
2038 010070 001372 BNE 16 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2039 010072 005215 INC (R5)
2040
2041
  
```

```

2042 ;*****
2043 ;TEST:226 MUL 7777 * 77777 = 37777 1 PS = 1
2044 ;*****
2045
2046 010074 010701 TST226: SCOPE
2047 010076 010501 MOV R5,R1 ;SAVE R5
2048 010100 012704 077777 MOV #77777,%4 ;LOAD MULTIPLICAND WITH 77777
2049 010104 070427 077777 MUL #77777,%4 ;MULTIPLY 77777 * 77777
2050 010110 106737 000432 MFPS 0#PSWORD ;SAVE PS
2051 010114 122737 000001 000432 CMPB #1,0#PSWORD ;IS PS = 1
2052 010122 001403 BEQ .+10
2053 010124 004767 006310 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2054 ;PS IS WRONG
2055 010130 000170 170
2056 010132 022704 037777 CMP #37777,%4 ;IS HIGH ORDER = 37777
2057 010136 001403 BEQ .+10
2058 010140 004767 006274 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2059 ;HIGH ORDER IS WRONG
2060 010144 000171 171
2061 010146 022705 000001 CMP #1,%4+11 ;IS LOW ORDER = 1
2062 010152 001403 BEQ .+10
2063 010154 16:
2064 010154 004767 006260 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2065 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2066 010160 000172 172
2067 010162 021127 000226 CMP (R1),#226 ;CHECK THE TEST NUMBER
2068 010166 001372 BNE 16 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2069 010170 010105 MOV R1,R5 ;RESTORE R5
2070 010172 005215 INC (R5)
2071
2072
    
```

```

2073 ;*****
2074 ;TEST:227 MUL -1 * 77777 = -1 100001 PS = 10
2075 ;*****
2076
2077 010174 010701 TST227: SCOPE
2078 010176 012702 177777 MOV #-1,%2 ;LOAD MULTIPLICAND WITH -1
2079 010202 070227 077777 MUL #77777,%2 ;MULTIPLY -1 * 77777
2080 010206 106737 000432 MFPS 0#PSWORD ;SAVE PS
2081 010212 122737 000010 000432 CMPB #10,0#PSWORD ;IS PS = 10
2082 010220 001403 BEQ .+10
2083 010222 004767 006212 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2084 ;PS IS WRONG
2085 010226 000173 173
2086 010230 022702 177777 CMP #-1,%2 ;IS HIGH ORDER = -1
2087 010234 001403 BEQ .+10
2088 010236 004767 006176 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2089 ;HIGH ORDER IS WRONG
2090 010242 000174 174
2091 010244 022703 100001 CMP #100001,%2+11 ;IS LOW ORDER = 100001
2092 010250 001403 BEQ .+10
2093 010252 16:
2094 010252 004767 006162 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2095 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2096 010256 000175 175
2097 010260 021527 000227 CMP (R5),#227 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2098 010264 001372 BNE 16
2099 010266 005215 INC (R5)
2100
2101
    
```

```

2102 ;*****
2103 ;TEST:230 MUL -2 * #77777 = -1 2 PS = 11
2104 ;*****
2105
2106 #10270 010701 TST230: SCOPE
2107 #10272 012700 MOV #-2,%0 ;LOAD MULTIPLICAND WITH -2
2108 #10276 070027 177777 MUL #77777,%0 ;MULTIPLY -2 * #77777
2109 #10302 106737 000432 MFPS #PSWORD ;SAVE PS
2110 #10306 122737 000011 000432 CMPB #11,%PSWORD ;IS PS = 11
2111 #10314 001403 BEQ ,+10
2112 #10316 004767 006116 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2113 ;PS IS WRONG
2114 #10322 000176 176
2115 #10324 022700 177777 CMP #-1,%0 ;IS HIGH ORDER = -1
2116 #10330 001403 BEQ ,+10
2117 #10332 004767 006102 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2118 ;HIGH ORDER IS WRONG
2119 #10336 000177 177
2120 #10340 022701 000002 CMP #2,%011 ;IS LOW ORDER = 2
2121 #10344 001403 BEQ ,+10
2122 #10346 004767 006066 18: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2123 #10352 000200 200 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2124 #10354 021527 000230 CMP (R5),#230
2125 #10360 001372 BNE 18 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2126 #10362 005215 INC (R5)
2127
2128
2129
2130
    
```

```

2131 ;*****
2132 ;TEST:231 MUL 125252 * #2 = -1 52524 PS = 11
2133 ;*****
2134
2135 #10364 010701 TST231: SCOPE
2136 #10366 012702 125252 MOV #125252,%2 ;LOAD MULTIPLICAND WITH 125252
2137 #10372 070227 000002 MUL #2,%2 ;MULTIPLY 125252 * #2
2138 #10376 106737 000432 MFPS #PSWORD ;SAVE PS
2139 #10402 122737 000011 000432 CMPB #11,%PSWORD ;IS PS = 11
2140 #10410 001403 BEQ ,+10
2141 #10412 004767 006022 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2142 ;PS IS WRONG
2143 #10416 000201 201
2144 #10420 022702 177777 CMP #-1,%2 ;IS HIGH ORDER = -1
2145 #10424 001403 BEQ ,+10
2146 #10426 004767 006006 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2147 ;HIGH ORDER IS WRONG
2148 #10432 000202 202
2149 #10434 022703 052524 CMP #52524,%211 ;IS LOW ORDER = 52524
2150 #10440 001403 BEQ ,+10
2151 #10442 004767 005772 18: JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2152 #10446 000203 203 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2153 #10450 021527 000231 CMP (R5),#231
2154 #10454 001372 BNE 18 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2155 #10456 005215 INC (R5)
2156
2157
2158
2159
    
```

```

2160 ;*****
2161 ;TEST:232 MUL 125252 * #40000 = 165252 100000 PS = 11
2162 ;*****
2163
2164 #10460 #10701 TST232: SCOPE
2165 #10462 #10501 MOV R5,R1 ;SAVE R5
2166 #10464 #12704 125252 MOV #125252,%4 ;LOAD MULTIPLICAND WITH 125252
2167 #10470 #70427 040000 MUL #40000,%4 ;MULTIPLY 125252 * #40000
2168 #10474 106737 000432 MFPS #PSWORD ;SAVE PS
2169 #10500 122737 000011 000432 CMPB #1,%PSWORD ;IS PS = 11
2170 #10506 #01403 BEQ ,+10
2171 #10510 #04767 005724 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2172 ;PS IS WRONG
2173 #10514 #00204 204
2174 #10516 #22704 165252 CMP #165252,%4 ;IS HIGH ORDER = 165252
2175 #10522 #01403 BEQ ,+10
2176 #10524 #04767 005710 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2177 ;HIGH ORDER IS WRONG
2178 #10530 #00205 205
2179 #10532 #22705 100000 CMP #100000,%4,1 ;IS LOW ORDER = 100000
2180 #10536 #01403 BEQ ,+10
2181 #10540 10: JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2182 #10540 #04767 005674 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2183
2184 #10544 #00206 206
2185 #10546 #21127 000232 CMP (R1),#232 ;CHECK THE TEST NUMBER
2186 #10552 #01372 BNE 10 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2187 #10554 #10105 MOV R1,R5 ;RESTORE R5
2188 #10556 #05215 INC (R5)
2189
2190

```

```

2191 ;*****
2192 ;TEST:233 MUL 107070 * #107070 = 31222 26100 PS = 1
2193 ;*****
2194
2195 #10560 #10701 TST233: SCOPE
2196 #10562 #12700 107070 MOV #107070,%0 ;LOAD MULTIPLICAND WITH 107070
2197 #10566 #70027 107070 MUL #107070,%0 ;MULTIPLY 107070 * #107070
2198 #10572 106737 000432 MFPS #PSWORD ;SAVE PS
2199 #10576 122737 000001 000432 CMPB #1,%PSWORD ;IS PS = 1
2200 #10604 #01403 BEQ ,+10
2201 #10606 #04767 005626 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2202 ;PS IS WRONG
2203 #10612 #00207 207
2204 #10614 #22700 031222 CMP #31222,%0 ;IS HIGH ORDER = 31222
2205 #10620 #01403 BEQ ,+10
2206 #10622 #04767 005612 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2207 ;HIGH ORDER IS WRONG
2208 #10626 #00210 210
2209 #10630 #22701 026100 CMP #26100,%0,1 ;IS LOW ORDER = 26100
2210 #10634 #01403 BEQ ,+10
2211 #10636 10: JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2212 #10636 #04767 005576 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2213
2214 #10642 #00211 211
2215 #10644 #21527 000233 CMP (R5),#233 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2216 #10650 #01372 BNE 10
2217 #10652 #05215 INC (R5)
2218
2219

```

```

2220 ;*****
2221 ;TEST:234 MUL -1 * #1 = -1 -1 PS = 10
2222 ;*****
2223
2224 010654 010701 TST234: SCOPE
2225 010656 012701 177777 MOV #-1,%1 ;LOAD MULTIPLICAND WITH -1
2226 010662 070127 000001 MUL #1,%1 ;MULTIPLY -1 * #1
2227 010666 106737 000432 MFPS ##PSWORD ;SAVE PS
2228 010672 122737 000010 000432 CMPB #10,##PSWORD ;IS PS = 10
2229 010700 001403 BEQ ,+10
2230 010702 004767 005532 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2231 ;PS IS WRONG
2232 010706 000212 212
2233 010710 022701 177777 CMP #-1,%1 ;IS HIGH ORDER = -1
2234 010714 001403 BEQ ,+10
2235 010716 004767 005516 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2236 ;HIGH ORDER IS WRONG
2237 010722 000213 213
2238 010724 022701 177777 CMP #-1,%11 ;IS LOW ORDER = -1
2239 010730 001403 BEQ ,+10
2240 010732 18:
2241 010732 004767 005502 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2242 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2243 010736 000214 214
2244 010740 021527 000234 CMP (R5),#234
2245 010744 001372 BNE 18 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2246 010746 005215 INC (R5)
2247
2248

```

```

2249 ;*****
2250 ;TEST:235 MUL -1 * #0 = 0 0 PS = 4
2251 ;*****
2252
2253 010750 010701 TST235: SCOPE
2254 010752 012703 177777 MOV #-1,%3 ;LOAD MULTIPLICAND WITH -1
2255 010756 070327 000000 MUL #0,%3 ;MULTIPLY -1 * #0
2256 010762 106737 000432 MFPS ##PSWORD ;SAVE PS
2257 010766 122737 000004 000432 CMPB #4,##PSWORD ;IS PS = 4
2258 010774 001403 BEQ ,+10
2259 010776 004767 005436 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2260 ;PS IS WRONG
2261 011002 000215 215
2262 011004 022703 000000 CMP #0,%3 ;IS HIGH ORDER = 0
2263 011010 001403 BEQ ,+10
2264 011012 004767 005422 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2265 ;HIGH ORDER IS WRONG
2266 011016 000216 216
2267 011020 022703 000000 CMP #0,%311 ;IS LOW ORDER = 0
2268 011024 001403 BEQ ,+10
2269 18:
2270 011026 004767 005406 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2271 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2272 011032 000217 217
2273 011034 021527 000235 CMP (R5),#235
2274 011040 001372 BNE 18 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2275 011042 005215 INC (R5)
2276
2277

```

```

2278
2279 ;*****
2280 ;TEST:236 MUL 77777 * #100000 = 100000 100000 PS = 11
2281 ;*****
2282 011044 010701 TST236: SCOPE
2283 011046 010501 MOV R5,R1 ;SAVE R5
2284 011050 012705 077777 MOV #77777,%5 ;LOAD MULTIPLICAND WITH 77777
2285 011054 070527 100000 MUL #100000,%5 ;MULTIPLY 77777 * #100000
2286 011060 106737 000432 MFPS 0#PSWORD ;SAVE PS
2287 011064 172737 000011 000432 CMPB #11,0#PSWORD ;IS PS = 11
2288 011072 001403 BEQ ,+10
2289 011074 004767 005340 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2290 ;PS IS WRONG
2291 011100 000220 220
2292 011102 022705 100000 CMP #100000,%5 ;IS HIGH ORDER = 100000
2293 011106 001403 BEQ ,+10
2294 011110 004767 005324 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2295 ;HIGH ORDER IS WRONG
2296 011114 000221 221
2297 011116 022705 100000 CMP #100000,%511 ;IS LOW ORDER = 100000
2298 011122 001403 BEQ ,+10
2299 011124
2300 011124 004767 005310 1$: JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2301 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2302 011130 000222 222
2303 011132 021127 000236 CMP (R1),#236 ;CHECK THE TEST NUMBER
2304 011136 001372 BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2305 011140 010105 MOV R1,R5 ;RESTORE R5
2306 011142 005215 INC (R5)
2307
2308

```

```

2309 ;*****
2310 ;TEST:237 MUL -1 * #77777 = 100001 100001 PS = 10
2311 ;*****
2312
2313 011144 010701 TST237: SCOPE
2314 011146 012701 177777 MOV #-1,%1 ;LOAD MULTIPLICAND WITH -1
2315 011152 070127 077777 MUL #77777,%1 ;MULTIPLY -1 * #77777
2316 011156 106737 000432 MFPS 0#PSWORD ;SAVE PS
2317 011162 122737 000010 000432 CMPB #10,0#PSWORD ;IS PS = 10
2318 011170 001403 BEQ ,+10
2319 011172 004767 005242 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2320 ;PS IS WRONG
2321 011176 000223 223
2322 011200 022701 100001 CMP #100001,%1 ;IS HIGH ORDER = 100001
2323 011204 001403 BEQ ,+10
2324 011206 004767 005226 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2325 ;HIGH ORDER IS WRONG
2326 011212 000224 224
2327 011214 022701 100001 CMP #100001,%111 ;IS LOW ORDER = 100001
2328 011220 001403 BEQ ,+10
2329 011222
2330 011222 004767 005212 1$: JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2331 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2332 011226 000225 225
2333 011230 021527 000237 CMP (R5),#237
2334 011234 001372 BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2335 011236 005215 INC (R5)
2336
2337

```

```

2330 ;*****
2339 ;TEST:240 MUL 77777 * #77777 = 1 1 PS = 1
2340 ;*****
2341
2342 011240 010701 TST240: SCOPE
2343 011242 012703 077777 MOV #77777,%3 ;LOAD MULTIPLICAND WITH 77777
2344 011246 070327 077777 MUL #77777,%3 ;MULTIPLY 77777 * #77777
2345 011252 106737 000432 MFPS #PSWORD ;SAVE PS
2346 011256 122737 000001 000432 CMPB #1,#PSWORD ;IS PS = 1
2347 011264 001403 BEQ ,+10
2348 011266 004767 005146 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2349 ;PS IS WRONG
2350 011272 000226 226 CMP #1,%3 ;IS HIGH ORDER = 1
2351 011274 022703 000001 BEQ ,+10
2352 011300 001403 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2353 011302 004767 005132 ;HIGH ORDER IS WRONG
2354
2355 011306 000227 227 CMP #1,%311 ;IS LOW ORDER = 1
2356 011310 022703 000001 BEQ ,+10
2357 011314 001403 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2358 011316 004767 005116 10: ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2359 011316 004767 005116 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2360
2361 011322 000230 230 CMP (R5),#240
2362 011324 021527 000240 BNE 10
2363 011330 001372 INC (R5)
2364 011332 005215
2365
2366
    
```

```

2367 ;*****
2368 ;TEST:241 MUL 2 * #2 = 4 4 PS = 0
2369 ;*****
2370
2371 011334 010701 TST241: SCOPE
2372 011336 010501 MOV R5,R1 ;SAVE R5
2373 011340 012705 000002 MOV #2,%5 ;LOAD MULTIPLICAND WITH 2
2374 011344 070527 000002 MUL #2,%5 ;MULTIPLY 2 * #2
2375 011350 106737 000432 MFPS #PSWORD ;SAVE PS
2376 011354 122737 000000 000432 CMPB #0,#PSWORD ;IS PS = 0
2377 011362 001403 BEQ ,+10
2378 011364 004767 005050 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2379 ;PS IS WRONG
2380 011370 000231 231 CMP #4,%5 ;IS HIGH ORDER = 4
2381 011372 022705 000004 BEQ ,+10
2382 011376 001403 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2383 011400 004767 005034 ;HIGH ORDER IS WRONG
2384
2385 011404 000232 232 CMP #4,%511 ;IS LOW ORDER = 4
2386 011406 022705 000004 BEQ ,+10
2387 011412 001403 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2388 011414 004767 005020 10: ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2389 011414 004767 005020 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2390
2391 011420 000233 233 CMP (R1),#241 ;CHECK THE TEST NUMBER
2392 011422 021127 000241 BNE 10 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2393 011426 001372 MOV R1,R5 ;RESTORE R5
2394 011430 010105 INC (R5)
2395 011432 005215
2396
2397
    
```

```

2398 011434 012702 040000      MOV    #40000,%2
2399 011440 012703 000464      MOV    #85,%3
2400 011444 012704 000466      MOV    #86,%4
2401
2402
2403 ;*****
;TEST:242      MUL    125252 * 85 = 165252 100000      PS = 11
;*****
2404
2405
2406 011450 010701      TST242: SCOPE
2407 011452 012700 125252      MOV    #125252,%0      ;LOAD MULTIPLICAND WITH 125252
2408 011456 070067 167002      MUL    85,%0          ;MULTIPLY 125252 * 85
2409 011462 106737 000432      MFPS  0#PSWORD        ;SAVE PS
2410 011466 122737 000011 000432      CMPB  #11,0#PSWORD     ;IS PS = 11
2411 011474 001403      BEQ    ,+10
2412 011476 004767 004736      JSR    PC,#HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2413                          ;PS IS WRONG
2414 011502 000234      234
2415 011504 022700 165252      CMP    #165252,%0      ;IS HIGH ORDER = 165252
2416 011510 001403      BEQ    ,+10
2417 011512 004767 004722      JSR    PC,#HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2418                          ;HIGH ORDER IS WRONG
2419 011516 000235      235
2420 011520 022701 100000      CMP    #100000,%011    ;IS LOW ORDER = 100000
2421 011524 001403      BEQ    ,+10
2422 011526
2423 011526 004767 004706      18: JSR    PC,#HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2424                          ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2425 011532 000236      236
2426 011534 021527 000242      CMP    (R5),#242
2427 011540 001372      BNE   18
2428 011542 005215      INC   (R5)              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2429
2430

```

```

2431 ;*****
2432 ;TEST:243      MUL    125252 * 86 = 165252 100000      PS = 11
2433 ;*****
2434
2435 011544 010701      TST243: SCOPE
2436 011546 012700 125252      MOV    #125252,%0      ;LOAD MULTIPLICAND WITH 125252
2437 011552 070077 166710      MUL    86,%0          ;MULTIPLY 125252 * 86
2438 011556 106737 000432      MFPS  0#PSWORD        ;SAVE PS
2439 011562 122737 000011 000432      CMPB  #11,0#PSWORD     ;IS PS = 11
2440 011570 001403      BEQ    ,+10
2441 011572 004767 004642      JSR    PC,#HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2442                          ;PS IS WRONG
2443 011576 000237      237
2444 011600 022700 165252      CMP    #165252,%0      ;IS HIGH ORDER = 165252
2445 011604 001403      BEQ    ,+10
2446 011606 004767 004626      JSR    PC,#HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2447                          ;HIGH ORDER IS WRONG
2448 011612 000240      240
2449 011614 022701 100000      CMP    #100000,%011    ;IS LOW ORDER = 100000
2450 011620 001403      BEQ    ,+10
2451 011622
2452 011622 004767 004612      18: JSR    PC,#HLT         ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2453                          ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2454 011626 000241      241
2455 011630 021527 000243      CMP    (R5),#243
2456 011634 001372      BNE   18
2457 011636 005215      INC   (R5)              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2458
2459

```



```

2460 ;*****
2461 ;TEST:244 MUL 125252 * 0#S5 = 165252 100000 PS = 11
2462 ;*****
2463
2464 011640 010701 TST244: SCOPE
2465 011642 012700 125252 MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
2466 011646 070037 000464 MUL #0S5,%0 ;MULTIPLY 125252 * 0#S5
2467 011652 106737 000432 MFPS 0#PSWORD ;SAVE PS
2468 011656 122737 000011 000432 CMPB #11,0#PSWORD ;IS PS = 11
2469 011664 001403 BEQ ,+10
2470 011666 004767 004546 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2471 ;PS IS WRONG
2472 011672 000242 242
2473 011674 022700 165252 CMP #165252,%0 ;IS HIGH ORDER = 165252
2474 011700 001403 BEQ ,+10
2475 011702 004767 004532 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2476 ;HIGH ORDER IS WRONG
2477 011706 000243 243
2478 011710 022701 100000 CMP #100000,%011 ;IS LOW ORDER = 100000
2479 011714 001403 BEQ ,+10
2480 011716 18: JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2481 011716 004767 004516 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2482
2483 011722 000244 244
2484 011724 021527 000244 CMP (R5),#244
2485 011730 001372 BNE 18 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2486 011732 005215 INC (R5)
2487
2488

```

```

2489 ;*****
2490 ;TEST:245 MUL 125252 * %2 = 165252 100000 PS = 11
2491 ;*****
2492
2493 011734 010701 TST245: SCOPE
2494 011736 012700 125252 MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
2495 011742 070002 000002 MUL %2,%0 ;MULTIPLY 125252 * %2
2496 011744 106737 000432 MFPS 0#PSWORD ;SAVE PS
2497 011750 122737 000011 000432 CMPB #11,0#PSWORD ;IS PS = 11
2498 011756 001403 BEQ ,+10
2499 011760 004767 004454 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2500 ;PS IS WRONG
2501 011764 000245 245
2502 011766 022700 165252 CMP #165252,%0 ;IS HIGH ORDER = 165252
2503 011772 001403 BEQ ,+10
2504 011774 004767 004440 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2505 ;HIGH ORDER IS WRONG
2506 012000 000246 246
2507 012002 022701 100000 CMP #100000,%011 ;IS LOW ORDER = 100000
2508 012006 001403 BEQ ,+10
2509 012010 18: JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2510 012010 004767 004424 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2511
2512 012014 000247 247
2513 012016 021527 000245 CMP (R5),#245
2514 012022 001372 BNE 18 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2515 012024 005215 INC (R5)
2516
2517

```

```

2518 ;*****
2519 ;TEST:246 MUL 125252 * (3)+ = 165252 100000 PS = 11
2520 ;*****
2521
2522 012026 010701 TST246: SCOPE
2523 012030 012700 125252 MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
2524 012034 070023 MUL (3)+,%0 ;MULTIPLY 125252 * (3)+
2525 012036 106737 000432 MFPS 0#PSWORD ;SAVE PS
2526 012042 122737 000011 000432 CMPB #11,0#PSWORD ;IS PS = 11
2527 012050 001403 BEQ ,+10
2528 012052 004767 004362 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2529 ;PS IS WRONG
2530 012056 000250 250
2531 012060 022700 165252 CMP #165252,%0 ;IS HIGH ORDER = 165252
2532 012064 001403 BEQ ,+10
2533 012066 004767 004346 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2534 ;HIGH ORDER IS WRONG
2535 012072 000251 251
2536 012074 022701 100000 CMP #100000,%011 ;IS LOW ORDER = 100000
2537 012100 001403 BEQ ,+10
2538 012102 18:
2539 012102 004767 004332 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2540 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2541 012106 000252 252
2542 012110 021527 000246 CMP (R5),#246
2543 012114 001372 BNE 18 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2544 012116 005215 INC (R5)
2545
2546
    
```

```

2547 ;*****
2548 ;TEST:247 MUL 125252 * -(3) = 165252 100000 PS = 11
2549 ;*****
2550
2551 012120 010701 TST247: SCOPE
2552 012122 012700 125252 MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
2553 012126 070043 MUL -(3),%0 ;MULTIPLY 125252 * -(3)
2554 012130 106737 000432 MFPS 0#PSWORD ;SAVE PS
2555 012134 122737 000011 000432 CMPB #11,0#PSWORD ;IS PS = 11
2556 012142 001403 BEQ ,+10
2557 012144 004767 004270 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2558 ;PS IS WRONG
2559 012150 000253 253
2560 012152 022700 165252 CMP #165252,%0 ;IS HIGH ORDER = 165252
2561 012156 001403 BEQ ,+10
2562 012160 004767 004254 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2563 ;HIGH ORDER IS WRONG
2564 012164 000254 254
2565 012166 022701 100000 CMP #100000,%011 ;IS LOW ORDER = 100000
2566 012172 001403 BEQ ,+10
2567 012174 18:
2568 012174 004767 004240 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2569 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2570 012200 000255 255
2571 012202 021527 000247 CMP (R5),#247
2572 012206 001372 BNE 18 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2573 012210 005215 INC (R5)
2574
2575
    
```

```

2576 ;*****
2577 ;TEST:250 MUL 125252 * 2(4) = 165252 100000 PS = 11
2578 ;*****
2579
2580 012212 010701 TST250: SCOPE
2581 012214 012700 125252 MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
2582 012220 070064 000000 MUL 2(4),%0 ;MULTIPLY 125252 * 2(4)
2583 012224 106737 000432 MFPS 0#PSWORD ;SAVE PS
2584 012230 122737 000011 000432 CMPB #11,0#PSWORD ;IS PS = 11
2585 012236 001403 BEQ ,+10
2586 012240 004767 004174 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2587 ;PS IS WRONG
2588 012244 000256 256
2589 012246 022700 165252 CMP #165252,%0 ;IS HIGH ORDER = 165252
2590 012252 001403 BEQ ,+10
2591 012254 004767 004160 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2592 ;HIGH ORDER IS WRONG
2593 012260 000257 257
2594 012262 022701 100000 CMP #100000,%011 ;IS LOW ORDER = 100000
2595 012266 001403 BEQ ,+10
2596 012270 101
2597 012270 004767 004144 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2598 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2599 012274 000260 260
2600 012276 021527 000250 CMP (R5),#250
2601 012302 001372 BNE 10 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2602 012304 005215 INC (R5)
2603
2604
    
```

```

2605 ;*****
2606 ;TEST:251 MUL 125252 * 0(4) = 165252 100000 PS = 11
2607 ;*****
2608
2609 012306 010701 TST251: SCOPE
2610 012310 012700 125252 MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
2611 012314 070074 000000 MUL 0(4),%0 ;MULTIPLY 125252 * 0(4)
2612 012320 106737 000432 MFPS 0#PSWORD ;SAVE PS
2613 012324 122737 000011 000432 CMPB #11,0#PSWORD ;IS PS = 11
2614 012332 001403 BEQ ,+10
2615 012334 004767 004100 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2616 ;PS IS WRONG
2617 012340 000261 261
2618 012342 022700 165252 CMP #165252,%0 ;IS HIGH ORDER = 165252
2619 012346 001403 BEQ ,+10
2620 012350 004767 004064 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2621 ;HIGH ORDER IS WRONG
2622 012354 000262 262
2623 012356 022701 100000 CMP #100000,%011 ;IS LOW ORDER = 100000
2624 012362 001403 BEQ ,+10
2625 012364 101
2626 012364 004767 004050 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2627 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2628 012370 000263 263
2629 012372 021527 000251 CMP (R5),#251
2630 012376 001372 BNE 10 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2631 012400 005215 INC (R5)
2632
2633
    
```

```

2634 ;*****
2635 ;TEST:252 MUL 125252 * 0(4)+ = 165252 100000 PS = 11
2636 ;*****
2637
2638 012402 010701 TST252: SCOPE
2639 012404 012700 125252 MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
2640 012410 070034 MUL 0(4)+,%0 ;MULTIPLY 125252 * 0(4)+
2641 012412 106737 000432 MFPS 0#PSWORD ;SAVE PS
2642 012416 122737 000011 000432 CMPB #11,0#PSWORD ;IS PS = 11
2643 012424 001403 BEQ .+10
2644 012426 004767 004006 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2645 ;PS IS WRONG
2646 012432 000264 264 CMP #165252,%0 ;IS HIGH ORDER = 165252
2647 012434 022700 165252 BEQ .+10
2648 012440 001403 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2649 012442 004767 003772 ;HIGH ORDER IS WRONG
2650
2651 012446 000265 265 CMP #100000,%011 ;IS LOW ORDER = 100000
2652 012450 022701 100000 BEQ .+10
2653 012454 001403 18: JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2654 012456 004767 003756 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2655
2656 012462 000266 266 CMP (R5),#252
2657 012464 021527 000252 BNE 18 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2658 012470 001372 INC (R5)
2659 012472 005215
2660
2661
2662
  
```

```

2663 ;*****
2664 ;TEST:253 MUL 125252 * 0-(4) = 165252 100000 PS = 11
2665 ;*****
2666
2667 012474 010701 TST253: SCOPE
2668 012476 012700 125252 MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
2669 012502 070054 MUL 0-(4),%0 ;MULTIPLY 125252 * 0-(4)
2670 012504 106737 000432 MFPS 0#PSWORD ;SAVE PS
2671 012510 122737 000011 000432 CMPB #11,0#PSWORD ;IS PS = 11
2672 012516 001403 BEQ .+10
2673 012520 004767 003714 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2674 ;PS IS WRONG
2675 012524 000267 267 CMP #165252,%0 ;IS HIGH ORDER = 165252
2676 012526 022700 165252 BEQ .+10
2677 012532 001403 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2678 012534 004767 003700 ;HIGH ORDER IS WRONG
2679
2680 012540 000270 270 CMP #100000,%011 ;IS LOW ORDER = 100000
2681 012542 022701 100000 BEQ .+10
2682 012546 001403 18: JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2683 012550 004767 003664 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
2684 012552 004767 003664
2685
2686 012554 000271 271 CMP (R5),#253
2687 012556 021527 000253 BNE 18 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
2688 012562 001372 INC (R5)
2689 012564 005215
2690
2691
  
```

```

2692 ;*****
2693 ; DIV INSTRUCTION TESTS
2694 ;*****
2695
2696
2697
2698 ;*****
2699 ;TEST:254 DIV 0 4 / #2 = 2 REM = 0 PS = 0
2700 ;*****
2701
2702 012566 010701 TST254: SCOPE
2703 012570 012700 000000 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
2704 012574 012701 000004 MOV #4,%0+1 ;LOAD LOW ORDER WITH 4
2705 012600 071027 000002 DIV #2,%0 ;DIVIDE BY #2
2706 012604 106737 000432 MFPS 0#PSWORD ;SAVE PS
2707
2708 012610 122737 000000 000432 CMPB #0,0#PSWORD ;IS PS = 0
2709 012616 001403 BEQ .+10
2710 012620 004767 003614 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2711 ;PS IS WRONG
2712 012624 000272 272
2713
2714 012626 022700 000002 CMP #2,%0 ;IS QUOTIENT = 2
2715 012632 001403 BEQ .+10
2716 012634 004767 003600 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2717 ;QUOTIENT IS WRONG
2718 012640 000273 273
2719
2720 012642 022701 000000 CMP #0,%0+1 ;IS REMAINDER = 0
2721 012646 001403 BEQ .+10
2722 012650 004767 003564 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2723 ;WRONG REMAINDER
2724 012654 000274 274
2725 012656 021527 000254 CMP (R5),#254
2726 012662 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
2727 012664 004767 003550 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2728 ;TEST IS IN WRONG SEQUENCE
2729 012670 000275 275
2730 012672 005215 INC (R5)
2731
    
```

```

2732 ;*****
2733 ;TEST:255 DIV -1 -9, / #3 = -3 REM = 0 PS = 10
2734 ;*****
2735
2736 012674 010701 TST255: SCOPE
2737 012676 012702 177777 MOV #-1,%2 ;LOAD HIGH ORDER WITH -1
2738 012702 012703 177767 MOV #-9,%2+1 ;LOAD LOW ORDER WITH -9,
2739 012706 071227 000003 DIV #3,%2 ;DIVIDE BY #3
2740 012712 106737 000432 MFPS 0#PSWORD ;SAVE PS
2741
2742 012716 122737 000010 000432 CMPB #10,0#PSWORD ;IS PS = 10
2743 012724 001403 BEQ .+10
2744 012726 004767 003506 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2745 ;PS IS WRONG
2746 012732 000276 276
2747
2748 012734 022702 177775 CMP #-3,%2 ;IS QUOTIENT = -3
2749 012740 001403 BEQ .+10
2750 012742 004767 003472 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2751 ;QUOTIENT IS WRONG
2752 012746 000277 277
2753
2754 012750 022703 000000 CMP #0,%2+1 ;IS REMAINDER = 0
2755 012754 001403 BEQ .+10
2756 012756 004767 003456 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2757 ;WRONG REMAINDER
2758 012762 000300 300
2759 012764 021527 000255 CMP (R5),#255
2760 012770 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
2761 012772 004767 003442 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2762 ;TEST IS IN WRONG SEQUENCE
2763 012776 000301 301
2764 013000 005215 INC (R5)
2765
    
```

```

2766 ;*****
2767 ;TEST:256 DIV 0 9. / #2 = 4 REM = 1 PS = 0
2768 ;*****
2769
2770 #13002 010701 TST256: SCOPE
2771 013004 010501 MOV R5,R1 ;SAVE R5
2772 013006 012704 000000 MOV #0,%4 ;LOAD HIGH ORDER WITH 0
2773 013012 012705 000011 MOV #9,%4+1 ;LOAD LOW ORDER WITH 9.
2774 013016 071427 000002 DIV #2,%4 ;DIVIDE BY #2
2775 013022 106737 000432 MFPS 0#PSWORD ;SAVE PS
2776
2777 #13026 122737 000000 000432 CMPB #0,0#PSWORD ;IS PS = 0
2778 013034 001403 BEQ .+10
2779 013036 004767 003376 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2780 ;PS IS WRONG
2781 013042 000302 302
2782
2783 013044 022704 000004 CMP #4,%4 ;IS QUOTIENT = 4
2784 013050 001403 BEQ .+10
2785 013052 004767 003362 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2786 ;QUOTIENT IS WRONG
2787 013056 000303 303
2788
2789 013060 022705 000001 CMP #1,%4+1 ;IS REMAINDER = 1
2790 013064 001403 BEQ .+10
2791 013066 004767 003346 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2792 ;WRONG REMAINDER
2793 013072 000304 304
2794 013074 010105 MOV R1,R5 ;RESTORE R5
2795 013076 021527 000256 CMP (R5),#256
2796 013102 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
2797 013104 004767 003330 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2798 ;TEST IS IN WRONG SEQUENCE
2799 #13110 000305 305
2800 013112 005215 INC (R5)
2801

```

```

2802 ;*****
2803 ;TEST:257 DIV -1 -9. / #2 = -4 REM = -1 PS = 10
2804 ;*****
2805
2806 013114 010701 TST257: SCOPE
2807 013116 012700 177777 MOV #1,%0 ;LOAD HIGH ORDER WITH -1
2808 013122 012701 177767 MOV #-9,%0+1 ;LOAD LOW ORDER WITH -9.
2809 013126 071027 000002 DIV #2,%0 ;DIVIDE BY #2
2810 013132 106737 000432 MFPS 0#PSWORD ;SAVE PS
2811
2812 013136 122737 000010 000432 CMPB #10,0#PSWORD ;IS PS = 10
2813 013144 001403 BEQ .+10
2814 013146 004767 003266 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2815 ;PS IS WRONG
2816 #13152 000306 306
2817
2818 013154 022700 177774 CMP #-4,%0 ;IS QUOTIENT = -4
2819 013160 001403 BEQ .+10
2820 013162 004767 003252 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2821 ;QUOTIENT IS WRONG
2822 013166 000307 307
2823
2824 013170 022701 177777 CMP #-1,%0+1 ;IS REMAINDER = -1
2825 013174 001403 BEQ .+10
2826 013176 004767 003236 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2827 ;WRONG REMAINDER
2828 013202 000310 310
2829 013204 021527 000257 CMP (R5),#257
2830 013210 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
2831 013212 004767 003222 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2832 ;TEST IS IN WRONG SEQUENCE
2833 013216 000311 311
2834 013220 005215 INC (R5)
2835

```

```

2836 ;*****
2837 ;TEST:260 DIV 0 2 / *3 = 0 REM = 2 PS = 4
2838 ;*****
2839
2840 013222 010701 TST260: SCOPE
2841 013224 012702 MOV #0,%2 ;LOAD HIGH ORDER WITH 0
2842 013230 012703 MOV #2,%2+1 ;LOAD LOW ORDER WITH 2
2843 013234 071227 DIV #-3,%2 ;DIVIDE BY #-3
2844 013240 106737 MFPS 0#PSWORD ;SAVE PS
2845
2846 013244 122737 000004 000432 CMPB #4,0#PSWORD ;IS PS = 4
2847 013252 001403 BEQ ,+10
2848 013254 004767 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2849 ;PS IS WRONG
2850 013260 000312 312
2851
2852 013262 022702 000000 CMP #0,%2 ;IS QUOTIENT = 0
2853 013266 001403 BEQ ,+10
2854 013270 004767 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2855 ;QUOTIENT IS WRONG
2856 013274 000313 313
2857
2858 013276 022703 000002 CMP #2,%2+1 ;IS REMAINDER = 2
2859 013302 001403 BEQ ,+10
2860 013304 004767 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2861 ;WRONG REMAINDER
2862 013310 000314 314
2863 013312 021527 000260 CMP (R5),#260
2864 013316 001403 BEQ ,+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
2865 013320 004767 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2866 ;TEST IS IN WRONG SEQUENCE
2867 013324 000315 315
2868 013326 005215 INC (R5)
2869

```

```

2870 ;*****
2871 ;TEST:261 DIV -1 -2 / *3 = 0 REM = -2 PS = 4
2872 ;*****
2873
2874 013330 010701 TST261: SCOPE
2875 013332 010501 MOV R5,R1 ;SAVE R5
2876 013334 012704 177777 MOV #-1,%4 ;LOAD HIGH ORDER WITH -1
2877 013340 012705 177776 MOV #-2,%4+1 ;LOAD LOW ORDER WITH -2
2878 013344 071427 000003 DIV #3,%4 ;DIVIDE BY #3
2879 013350 106737 000432 MFPS 0#PSWORD ;SAVE PS
2880
2881 013354 122737 000004 000432 CMPB #4,0#PSWORD ;IS PS = 4
2882 013362 001403 BEQ ,+10
2883 013364 004767 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2884 ;PS IS WRONG
2885 013370 000316 316
2886
2887 013372 022704 000000 CMP #0,%4 ;IS QUOTIENT = 0
2888 013376 001403 BEQ ,+10
2889 013400 004767 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2890 ;QUOTIENT IS WRONG
2891 013404 000317 317
2892
2893 013406 022705 177776 CMP #-2,%4+1 ;IS REMAINDER = -2
2894 013412 001403 BEQ ,+10
2895 013414 004767 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2896 ;WRONG REMAINDER
2897 013420 000320 320
2898 013422 010105 MOV R1,R5 ;RESTORE R5
2899 013424 021527 000261 CMP (R5),#261
2900 013430 001403 BEQ ,+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
2901 013432 004767 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2902 ;TEST IS IN WRONG SEQUENCE
2903 013436 000321 321
2904 013440 005215 INC (R5)
2905

```

```

2906 ;*****
2907 ;TEST:262 DIV -1 -1 / #1 = -1 REM = 0 PS = 10
2908 ;*****
2909
2910 TST262: SCOPE
2911 013442 010701 MOV #1,%0 ;LOAD HIGH ORDER WITH -1
2912 013444 012700 177777 MOV #1,%0+1 ;LOAD LOW ORDER WITH -1
2913 013450 012701 177777 DIV #1,%0 ;DIVIDE BY #1
2914 013454 071027 000001 MFPS 0#PSWORD ;SAVE PS
2915
2916 013464 122737 000010 000432 CMPB #10,0#PSWORD ;IS PS = 10
2917 013472 001403 BEQ ,+10'
2918 013474 004767 002740 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2919 ;PS IS WRONG
2920 322
2921
2922 013500 000322 322
2923 013502 022700 177777 CMP #1,%0 ;IS QUOTIENT = -1
2924 013506 001403 BEQ ,+10
2925 013510 004767 002724 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2926 ;QUOTIENT IS WRONG
2927 323
2928 013514 000323
2929 013516 022701 000000 CMP #0,%0+1 ;IS REMAINDER = 0
2930 013522 001403 BEQ ,+10
2931 013524 004767 002710 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2932 ;WRONG REMAINDER
2933 324
2934 013530 000324 CMP (R5),#262
2935 013532 021527 000262 BEQ ,+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
2936 013536 001403 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2937 ;TEST IS IN WRONG SEQUENCE
2938 325
2939 013544 000325 INC (R5)
2940 013546 005215

```

```

2940 ;*****
2941 ;TEST:263 DIV 0 0 / #1 = 0 REM = 0 PS = 4
2942 ;*****
2943
2944 TST263: SCOPE
2945 013550 010701 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
2946 013552 012700 000000 MOV #0,%0+1 ;LOAD LOW ORDER WITH 0
2947 013556 012701 000000 DIV #1,%0 ;DIVIDE BY #1
2948 013562 071027 000001 MFPS 0#PSWORD ;SAVE PS
2949
2950 013572 122737 000004 000432 CMPB #4,0#PSWORD ;IS PS = 4
2951 013600 001403 BEQ ,+10'
2952 013602 004767 002632 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2953 ;PS IS WRONG
2954 326
2955
2956 013606 000326 326
2957 013610 022700 000000 CMP #0,%0 ;IS QUOTIENT = 0
2958 013614 001403 BEQ ,+10
2959 013616 004767 002616 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2960 ;QUOTIENT IS WRONG
2961 327
2962 013622 000327
2963 013624 022701 000000 CMP #0,%0+1 ;IS REMAINDER = 0
2964 013630 001403 BEQ ,+10
2965 013632 004767 002602 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2966 ;WRONG REMAINDER
2967 330
2968 013636 000330 CMP (R5),#263
2969 013640 021527 000263 BEQ ,+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
2970 013644 004767 002566 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2971 ;TEST IS IN WRONG SEQUENCE
2972 331
2973 013652 000331 INC (R5)
2974 013654 005215

```



```

2974 ;*****
2975 ;TEST:264 DIV -1 125252 / #2 = 152525 REM = 0 PS = 10
2976 ;*****
2977
2978 013656 010701 TST264: SCOPE
2979 013660 012702 177777 MOV #=1,%2 ;LOAD HIGH ORDER WITH -1
2980 013664 012703 125252 MOV #125252,%2+1 ;LOAD LOW ORDER WITH 125252
2981 013670 071227 000002 DIV #2,%2 ;DIVIDE BY #2
2982 013674 106737 000432 MFPS @#PSWORD ;SAVE PS
2983
2984 013700 122737 000010 000432 CMPB #10,@#PSWORD ;IS PS = 10
2985 013706 001403 BEQ .+10
2986 013710 004767 002524 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2987 ;PS IS WRONG
2988 013714 000332 332
2989
2990 013716 022702 152525 CMP #152525,%2 ;IS QUOTIENT = 152525
2991 013722 001403 BEQ .+10
2992 013724 004767 002510 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2993 ;QUOTIENT IS WRONG
2994 013730 000333 333
2995
2996 013732 022703 000000 CMP #0,%2+1 ;IS REMAINDER = 0
2997 013736 001403 BEQ .+10
2998 013740 004767 002474 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
2999 ;WRONG REMAINDER
3000 013744 000334 334
3001 013746 021527 000264 CMP (R5),#264
3002 013752 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3003 013754 004767 002460 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3004 ;TEST IS IN WRONG SEQUENCE
3005 013760 000335 335
3006 013762 005215 INC (R5)
3007

```

```

3008 ;*****
3009 ;TEST:265 DIV -1 -1 / #-1 = 1 REM = 0 PS = 0
3010 ;*****
3011
3012 013764 010701 TST265: SCOPE
3013 013766 010501 MOV R5,R1 ;SAVE R5
3014 013770 012704 177777 MOV #=-1,%4 ;LOAD HIGH ORDER WITH -1
3015 013774 012705 177777 MOV #=-1,%4+1 ;LOAD LOW ORDER WITH -1
3016 014000 071427 177777 DIV #=-1,%4 ;DIVIDE BY #-1
3017 014004 106737 000432 MFPS @#PSWORD ;SAVE PS
3018
3019 014010 122737 000000 000432 CMPB #0,@#PSWORD ;IS PS = 0
3020 014016 001403 BEQ .+10
3021 014020 004767 002414 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3022 ;PS IS WRONG
3023 014024 000336 336
3024
3025 014026 022704 000001 CMP #1,%4 ;IS QUOTIENT = 1
3026 014032 001403 BEQ .+10
3027 014034 004767 002400 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3028 ;QUOTIENT IS WRONG
3029 014040 000337 337
3030
3031 014042 022705 000000 CMP #0,%4+1 ;IS REMAINDER = 0
3032 014046 001403 BEQ .+10
3033 014050 004767 002364 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3034 ;WRONG REMAINDER
3035 014054 000340 340
3036 014056 010105 MOV R1,R5 ;RESTORE R5
3037 014060 021527 000265 CMP (R5),#265
3038 014064 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3039 014066 004767 002346 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3040 ;TEST IS IN WRONG SEQUENCE
3041 014072 000341 341
3042 014074 005215 INC (R5)
3043

```

```

3044 ;*****
3045 ;TEST:266 DIV 25253 1 / #125252 = 100000 REM = 1 PS = 10
3046 ;*****
3047
3048 TST266: SCOPE
3049 014076 010701 MOV #25253,%0 ;LOAD HIGH ORDER WITH 25253
3050 014100 012700 025253 MOV #1,%0+1 ;LOAD LOW ORDER WITH 1
3051 014104 012701 000001 DIV #125252,%0 ;DIVIDE BY #125252
3052 014114 106737 000432 MFPS #PSWORD ;SAVE PS
3053
3054 014120 122737 000010 000432 CMPB #0,#PSWORD ;IS PS = 10
3055 014126 001403 BEQ .+10
3056 014130 004767 002304 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3057 ;PS IS WRONG
3058 014134 000342 342
3059
3060 014136 022700 100000 CMP #100000,%0 ;IS QUOTIENT = 100000
3061 014142 001403 BEQ .+10
3062 014144 004767 002270 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3063 ;QUOTIENT IS WRONG
3064 014150 000343 343
3065
3066 014152 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
3067 014156 001403 BEQ .+10
3068 014160 004767 002254 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3069 ;WRONG REMAINDER
3070 014164 000344 344
3071 014166 021527 000266 CMP (R5),#266
3072 014172 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3073 014174 004767 002240 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3074 ;TEST IS IN WRONG SEQUENCE
3075 014200 000345 345
3076 014202 005215 INC (R5)
3077

```

```

3078 ;*****
3079 ;TEST:267 DIV 37777 77777 / #77777 = 77777 REM = 77776 PS = 0
3080 ;*****
3081
3082 TST267: SCOPE
3083 014204 010701 MOV #37777,%2 ;LOAD HIGH ORDER WITH 37777
3084 014212 012703 077777 MOV #77777,%2+1 ;LOAD LOW ORDER WITH 77777
3085 014216 071227 077777 DIV #77777,%2 ;DIVIDE BY #77777
3086 014222 106737 000432 MFPS #PSWORD ;SAVE PS
3087
3088 014226 122737 000000 000432 CMPB #0,#PSWORD ;IS PS = 0
3089 014234 001403 BEQ .+10
3090 014236 004767 002176 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3091 ;PS IS WRONG
3092 014242 000346 346
3093
3094 014244 022702 077777 CMP #77777,%2 ;IS QUOTIENT = 77777
3095 014250 001403 BEQ .+10
3096 014252 004767 002162 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3097 ;QUOTIENT IS WRONG
3098 014256 000347 347
3099
3100 014260 022703 077776 CMP #77776,%2+1 ;IS REMAINDER = 77776
3101 014264 001403 BEQ .+10
3102 014266 004767 002146 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3103 ;WRONG REMAINDER
3104 014272 000350 350
3105 014274 021527 000267 CMP (R5),#267
3106 014300 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3107 014302 004767 002132 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3108 ;TEST IS IN WRONG SEQUENCE
3109 014306 000351 351
3110 014310 005215 INC (R5)
3111

```

```

3112 ;*****
3113 ;TEST:270 DIV 0 100000 / #2 = 40000 REM = 0 PS = 0
3114 ;*****
3115
3116 TST270: SCOPE
3117 014312 010701 MOV R5,R1 ;SAVE R5
3118 014314 010501 MOV #0,#4 ;LOAD HIGH ORDER WITH 0
3119 014316 012704 000000 MOV #100000,#4+1 ;LOAD LOW ORDER WITH 100000
3120 014322 012705 100000 DIV #2,#4 ;DIVIDE BY #2
3121 014326 071427 000002 MFPS #PSWORD ;SAVE PS
3122
3123 014332 122737 000000 000432 CMPB #0,#PSWORD ;IS PS = 0
3124 014344 001403 ,+10 BEQ ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3125 014346 004767 002066 JSR PC,$HLT ;PS IS WRONG
3126
3127 014352 000352 352
3128
3129 014354 022704 040000 CMP #40000,#4 ;IS QUOTIENT = 40000
3130 014360 001403 ,+10 BEQ ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3131 014362 004767 002052 JSR PC,$HLT ;QUOTIENT IS WRONG
3132
3133 014366 000353 353
3134
3135 014370 022705 000000 CMP #0,#4+1 ;IS REMAINDER = 0
3136 014374 001403 ,+10 BEQ ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3137 014376 004767 002036 JSR PC,$HLT ;WRONG REMAINDER
3138
3139 014402 000354 354
3140 014404 010105 MOV R1,R5 ;RESTORE R5
3141 014406 021527 000270 CMP (R5),#270
3142 014412 001403 ,+10 BEQ ;IF IN WRONG SEQUENCE GO TO THE HLT
3143 014414 004767 002020 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3144 ;TEST IS IN WRONG SEQUENCE
3145 014420 000355 355
3146 014422 005215 INC (R5)
3147

```

```

3148 ;*****
3149 ;TEST:271 DIV 17777 77777 / #17776 = 40000 REM = 17777 PS = 0
3150 ;*****
3151
3152 TST271: SCOPE
3153 014424 010701 MOV #17777,#0 ;LOAD HIGH ORDER WITH 17777
3154 014426 012700 177777 MOV #77777,#0+1 ;LOAD LOW ORDER WITH 77777
3155 014432 012701 077777 DIV #17776,#0 ;DIVIDE BY #17776
3156 014436 071027 177776 MFPS #PSWORD ;SAVE PS
3157
3158 014442 106737 000432 CMPB #0,#PSWORD ;IS PS = 0
3159 014444 001403 ,+10 BEQ ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3160 014446 004767 001756 JSR PC,$HLT ;PS IS WRONG
3161
3162 014462 000356 356
3163
3164 014464 022700 040000 CMP #40000,#0 ;IS QUOTIENT = 40000
3165 014470 001403 ,+10 BEQ ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3166 014472 004767 001742 JSR PC,$HLT ;QUOTIENT IS WRONG
3167
3168 014476 000357 357
3169
3170 014500 022701 177777 CMP #17777,#0+1 ;IS REMAINDER = 17777
3171 014504 001403 ,+10 BEQ ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3172 014506 004767 001726 JSR PC,$HLT ;WRONG REMAINDER
3173
3174 014512 000360 360
3175 014514 021527 000271 CMP (R5),#271
3176 014520 001403 ,+10 BEQ ;IF IN WRONG SEQUENCE GO TO THE HLT
3177 014522 004767 001712 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3178 ;TEST IS IN WRONG SEQUENCE
3179 014526 000361 361
3180 014530 005215 INC (R5)
3181

```

```

3182 ;*****
3183 ;TEST:272 DIV 0 52525 / #52525 = 1 REM = 0 PS = 0
3184 ;*****
3185
3186 014532 010701 TST272: SCOPE
3187 014534 012702 000000 MOV #0,#2 ;LOAD HIGH ORDER WITH 0
3188 014540 012703 052525 MOV #52525,#2+1 ;LOAD LOW ORDER WITH 52525
3189 014544 071227 052525 DIV #52525,#2 ;DIVIDE BY #52525
3190 014550 106737 000432 MFPS 0#PSWORD ;SAVE PS
3191
3192 014554 122737 000000 000432 CMPB #0,#PSWORD ;IS PS = 0
3193 014562 001403 BEQ ,+10
3194 014564 004767 001650 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3195 ;PS IS WRONG
3196 014570 000362 362
3197
3198 014572 022702 000001 CMP #1,#2 ;IS QUOTIENT = 1
3199 014576 001403 BEQ ,+10
3200 014600 004767 001634 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3201 ;QUOTIENT IS WRONG
3202 014604 000363 363
3203
3204 014606 022703 000000 CMP #0,#2+1 ;IS REMAINDER = 0
3205 014612 001403 BEQ ,+10
3206 014614 004767 001620 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3207 ;WRONG REMAINDER
3208 014620 000364 364
3209 014622 021527 000272 CMP (R5),#272
3210 014626 001403 BEQ ,+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3211 014630 004767 001604 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3212 ;TEST IS IN WRONG SEQUENCE
3213 014634 000365 365
3214 014636 005215 INC (R5)
3215

```

```

3216 ;*****
3217 ;TEST:273 DIV 0 77777 / #0 = DUMMY REM = DUMMY PS = 3
3218 ;*****
3219
3220 014640 010701 TST273: SCOPE
3221 014642 010501 MOV R5,R1 ;SAVE R5
3222 014644 012704 000000 MOV #0,#4 ;LOAD HIGH ORDER WITH 0
3223 014650 012705 077777 MOV #77777,#4+1 ;LOAD LOW ORDER WITH 77777
3224 014654 071427 000000 DIV #0,#4 ;DIVIDE BY #0
3225 014660 106737 000432 MFPS 0#PSWORD ;SAVE PS
3226 014664 042737 000014 000432 BIC #14,0#PSWORD
3227
3228 014672 122737 000003 000432 CMPB #3,0#PSWORD ;IS PS = 3
3229 014700 001403 BEQ ,+10
3230 014702 004767 001532 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3231 ;PS IS WRONG
3232 014706 000366 366
3233
3234 014710 010105 MOV R1,R5 ;RESTORE R5
3235 014712 021527 000273 CMP (R5),#273
3236 014716 001403 BEQ ,+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3237 014720 004767 001514 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3238 ;TEST IS IN WRONG SEQUENCE
3239 014724 000367 367
3240 014726 005215 INC (R5)
3241

```

```

3242 ;*****
3243 ;TEST:274 DIV 7777 17777 / #2 = DUMMY REM = DUMMY PS = 2
3244 ;*****
3245
3246 014730 010701 TST274: SCOPE
3247 014732 012700 MOV #7777,%0 ;LOAD HIGH ORDER WITH 7777
3248 014736 012701 177777 MOV #17777,%0+1 ;LOAD LOW ORDER WITH 17777
3249 014742 071027 000002 DIV #2,%0 ;DIVIDE BY #2
3250 014746 106737 000432 MFPS 0#PSWORD ;SAVE PS
3251 014752 042737 000014 000432 BIC #14,0#PSWORD
3252
3253 014760 122737 000002 000432 CMPB #2,0#PSWORD ;IS PS = 2
3254 014766 001403 BEQ .+10
3255 014770 004767 001444 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3256 ;PS IS WRONG
3257 014774 000370 370
3258
3259 014776 021527 000274 CMP (R5),#274
3260 015002 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3261 015004 004767 001430 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3262 ;TEST IS IN WRONG SEQUENCE
3263 015010 000371 371
3264 015012 005215 INC (R5)
3265

```

```

3266 015014 012702 000002 MOV #2,%2
3267 015020 012703 000474 MOV #S9,%3
3268 015024 012704 000476 MOV #S10,%4
3269
3270 ;*****
3271 ;TEST:275 DIV 0 52525 / S9 = 25252 REM = 1 PS = 0
3272 ;*****
3273
3274 015030 010701 TST275: SCOPE
3275 015032 012700 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
3276 015036 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
3277 015042 071067 163426 DIV S9,%0 ;DIVIDE BY S9
3278 015046 106737 000432 MFPS 0#PSWORD ;SAVE PS
3279
3280 015052 122737 000000 000432 CMPB #0,0#PSWORD ;IS PS = 0
3281 015060 001403 BEQ .+10
3282 015062 004767 001352 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3283 ;PS IS WRONG
3284 015066 000372 372
3285
3286 015070 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
3287 015074 001403 BEQ .+10
3288 015076 004767 001336 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3289 ;QUOTIENT IS WRONG
3290 015102 000373 373
3291
3292 015104 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
3293 015110 001403 BEQ .+10
3294 015112 004767 001322 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3295 ;WRONG REMAINDER
3296 015116 000374 374
3297 015120 021527 000275 CMP (R5),#275
3298 015124 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3299 015126 004767 001306 JSR PC,#HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3300 ;TEST IS IN WRONG SEQUENCE
3301 015132 000375 375
3302 015134 005215 INC (R5)
3303

```

```

3304 ;*****
3305 ;TEST:276 DIV 0 52525 / 0S10 = 25252 REM = 1 PS = 0
3306 ;*****
3307
3308 #15136 010701 TST276: SCOPE
3309 015140 012700 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
3310 015144 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
3311 015150 071077 163322 DIV 0S10,%0 ;DIVIDE BY 0S10
3312 015154 106737 000432 MFPS 0#PSWORD ;SAVE PS
3313
3314 015160 122737 000000 000432 CMPB #0,0#PSWORD ;IS PS = 0
3315 015166 001403 BEQ .+10
3316 015170 004767 001244 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3317 ;PS IS WRONG
3318 015174 000376 376
3319
3320 015176 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
3321 015202 001403 BEQ .+10
3322 015204 004767 001230 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3323 ;QUOTIENT IS WRONG
3324 015210 000377 377
3325
3326 015212 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
3327 015216 001403 BEQ .+10
3328 015220 004767 001214 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3329 ;WRONG REMAINDER
3330 015224 000400 400
3331 015226 021527 000276 CMP (R5),#276
3332 015232 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3333 015234 004767 001200 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3334 ;TEST IS IN WRONG SEQUENCE
3335 015240 000401 401
3336 015242 005215 INC (R5)
3337

```

```

3338 ;*****
3339 ;TEST:277 DIV 0 52525 / 0#S9 = 25252 REM = 1 PS = 0
3340 ;*****
3341
3342 015244 010701 TST277: SCOPE
3343 015246 012700 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
3344 015252 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
3345 015256 071037 000474 DIV 0#S9,%0 ;DIVIDE BY 0#S9
3346 015262 106737 000432 MFPS 0#PSWORD ;SAVE PS
3347
3348 015266 122737 000000 000432 CMPB #0,0#PSWORD ;IS PS = 0
3349 015274 001403 BEQ .+10
3350 015276 004767 001136 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3351 ;PS IS WRONG
3352 015302 000402 402
3353
3354 015304 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
3355 015310 001403 BEQ .+10
3356 015312 004767 001122 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3357 ;QUOTIENT IS WRONG
3358 015316 000403 403
3359
3360 015320 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
3361 015324 001403 BEQ .+10
3362 015326 004767 001106 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3363 ;WRONG REMAINDER
3364 015332 000404 404
3365 015334 021527 000277 CMP (R5),#277
3366 015340 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3367 015342 004767 001072 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3368 ;TEST IS IN WRONG SEQUENCE
3369 015346 000405 405
3370 015350 005215 INC (R5)
3371

```

```

3372 ;*****
3373 ;TEST:300 DIV 0 52525 / %2 = 25252 REM = 1 PS = 0
3374 ;*****
3375
3376 #15352 010701 TST300: SCOPE
3377 015354 012700 000000 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
3378 015360 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
3379 015364 071002 DIV %2,%0 ;DIVIDE BY %2
3380 015366 106737 000432 MFPS #PPSWORD ;SAVE PS
3381
3382 015372 122737 000000 000432 CMPB #0,#PPSWORD ;IS PS = 0
3383 015400 001403 BEQ ,+10
3384 015402 004767 001032 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3385 ;PS IS WRONG
3386 015406 000406 406
3387
3388 015410 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
3389 015414 001403 BEQ ,+10
3390 015416 004767 001016 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3391 ;QUOTIENT IS WRONG
3392 015422 000407 407
3393
3394 015424 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
3395 015430 001403 BEQ ,+10
3396 015432 004767 001002 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3397 ;WRONG REMAINDER
3398 015436 000410 410
3399 015440 021527 000300 CMP (R5),#300
3400 015444 001403 BEQ ,+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3401 015446 004767 000766 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3402 ;TEST IS IN WRONG SEQUENCE
3403 015452 000411 411
3404 015454 005215 INC (R5)
3405

```

```

3406 ;*****
3407 ;TEST:301 DIV 0 52525 / (3)+ = 25252 REM = 1 PS = 0
3408 ;*****
3409
3410 015456 010701 TST301: SCOPE
3411 015460 012700 000000 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
3412 015464 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
3413 015470 071023 DIV (3)+,%0 ;DIVIDE BY (3)+
3414 015472 106737 000432 MFPS #PPSWORD ;SAVE PS
3415
3416 015476 122737 000000 000432 CMPB #0,#PPSWORD ;IS PS = 0
3417 015504 001403 BEQ ,+10
3418 015506 004767 000726 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3419 ;PS IS WRONG
3420 015512 000412 412
3421
3422 015514 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
3423 015520 001403 BEQ ,+10
3424 015522 004767 000712 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3425 ;QUOTIENT IS WRONG
3426 015526 000413 413
3427
3428 015530 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
3429 015534 001403 BEQ ,+10
3430 015536 004767 000676 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3431 ;WRONG REMAINDER
3432 015542 000414 414
3433 015544 021527 000301 CMP (R5),#301
3434 015550 001403 BEQ ,+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3435 015552 004767 000662 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3436 ;TEST IS IN WRONG SEQUENCE
3437 015556 000415 415
3438 015560 005215 INC (R5)
3439

```

DIV INSTRUCTION TESTS

```

3440 ;*****
3441 ;TEST:302 DIV 0 52525 / -(3) = 25252 REM = 1 PS = 0
3442 ;*****
3443
3444 TST302: SCOPE
3445 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
3446 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
3447 DIV -(3),%0 ;DIVIDE BY -(3)
3448 MFPS #PSWORD ;SAVE PS
3449
3450 CMPB #0,%PSWORD ;IS PS = 0
3451 BEQ ,+10
3452 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3453 ;PS IS WRONG
3454 416
3455
3456 CMP #25252,%0 ;IS QUOTIENT = 25252
3457 BEQ ,+10
3458 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3459 ;QUOTIENT IS WRONG
3460 417
3461
3462 CMP #1,%0+1 ;IS REMAINDER = 1
3463 BEQ ,+10
3464 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3465 ;WRONG REMAINDER
3466 420
3467 CMP (R5),#302
3468 BEQ ,+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3469 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3470 ;TEST IS IN WRONG SEQUENCE
3471 421
3472 INC (R5)
3473
  
```

DIV INSTRUCTION TESTS

```

3474 ;*****
3475 ;TEST:303 DIV 0 52525 / 2(4) = 25252 REM = 1 PS = 0
3476 ;*****
3477
3478 TST303: SCOPE
3479 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
3480 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
3481 DIV 2(4),%0 ;DIVIDE BY 2(4)
3482 MFPS #PSWORD ;SAVE PS
3483
3484 CMPB #0,%PSWORD ;IS PS = 0
3485 BEQ ,+10
3486 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3487 ;PS IS WRONG
3488 422
3489
3490 CMP #25252,%0 ;IS QUOTIENT = 25252
3491 BEQ ,+10
3492 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3493 ;QUOTIENT IS WRONG
3494 423
3495
3496 CMP #1,%0+1 ;IS REMAINDER = 1
3497 BEQ ,+10
3498 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3499 ;WRONG REMAINDER
3500 424
3501 CMP (R5),#303
3502 BEQ ,+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3503 JSR PC,%HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3504 ;TEST IS IN WRONG SEQUENCE
3505 425
3506 INC (R5)
3507
  
```



```

3508 ;*****
3509 ;TEST:304 DIV 0 52525 / 0(4) = 25252 REM = 1 PS = 0
3510 ;*****
3511
3512 015774 010701 TST304: SCOPE
3513 015776 012700 000000 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
3514 016002 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
3515 016006 071074 000000 DIV 0(4),%0 ;DIVIDE BY 0(4)
3516 016012 106737 000432 MFPS 0#PSWORD ;SAVE PS
3517
3518 016016 122737 000000 000432 CMPB #0,0#PSWORD ;IS PS = 0
3519 016024 001403 BEQ .+10
3520 016026 004767 000406 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3521 ;PS IS WRONG
3522 016032 000426 426
3523
3524 016034 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
3525 016040 001403 BEQ .+10
3526 016042 004767 000372 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3527 ;QUOTIENT IS WRONG
3528 016046 000427 427
3529
3530 016050 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
3531 016054 001403 BEQ .+10
3532 016056 004767 000356 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3533 ;WRONG REMAINDER
3534 016062 000430 430
3535 016064 021527 000304 CMP (R5),#304
3536 016070 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3537 016072 004767 000342 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3538 ;TEST IS IN WRONG SEQUENCE
3539 016076 000431 431
3540 016100 005215 INC (R5)
3541

```

```

3542 ;*****
3543 ;TEST:305 DIV 0 52525 / 0(4)+ = 25252 REM = 1 PS = 0
3544 ;*****
3545
3546 016102 010701 TST305: SCOPE
3547 016104 012700 000000 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
3548 016110 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
3549 016114 071034 000000 DIV 0(4)+,%0 ;DIVIDE BY 0(4)+
3550 016116 106737 000432 MFPS 0#PSWORD ;SAVE PS
3551
3552 016122 122737 000000 000432 CMPB #0,0#PSWORD ;IS PS = 0
3553 016130 001403 BEQ .+10
3554 016132 004767 000302 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3555 ;PS IS WRONG
3556 016136 000432 432
3557
3558 016140 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
3559 016144 001403 BEQ .+10
3560 016146 004767 000266 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3561 ;QUOTIENT IS WRONG
3562 016152 000433 433
3563
3564 016154 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
3565 016160 001403 BEQ .+10
3566 016162 004767 000252 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3567 ;WRONG REMAINDER
3568 016166 000434 434
3569 016170 021527 000305 CMP (R5),#305
3570 016174 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3571 016176 004767 000236 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3572 ;TEST IS IN WRONG SEQUENCE
3573 016202 000435 435
3574 016204 005215 INC (R5)
3575

```

```

3576 ;*****
3577 ;TEST:306 DIV 0 52525 / 0=(4) = 25252 REM = 1 PS = 0
3578 ;*****
3579
3580 016206 010701 TSTJ06: SCOPE
3581 016210 012700 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
3582 016214 012701 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
3583 016220 071054 DIV 0=(4),%0 ;DIVIDE BY 0=(4)
3584 016222 106737 MFPS 0#PSWORD ;SAVE PS
3585
3586 016226 122737 CMPB #0,%#PSWORD ;IS PS = 0
3587 016234 001463 BEQ ,+10
3588 016236 004767 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3589 ;PS IS WRONG
3590 016242 000436 436
3591
3592 016244 022700 CMP #25252,%0 ;IS QUOTIENT = 25252
3593 016250 001403 BEQ ,+10
3594 016252 004767 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3595 ;QUOTIENT IS WRONG
3596 016256 000437 437
3597
3598 016260 022701 CMP #1,%0+1 ;IS REMAINDER = 1
3599 016264 001403 BEQ ,+10
3600 016266 004767 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3601 ;WRONG REMAINDER
3602 016272 000440 440
3603 016274 021527 CMP (R5),#306
3604 016300 001403 BEQ ,+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
3605 016302 004767 JSR PC,$HLT ;SEEN AN ERROR, GO TO TH HALT ROUTINE
3606 ;TEST IS IN WRONG SEQUENCE
3607 016306 000441 441
3608 016310 005215 INC (R5)
3609
    
```

```

3610 ;*****
3611
3612 .SBTTL END OF PASS ROUTINE
3613
3614 ;*INCREMENT THE PASS NUMBER ($PASS)
3615 ;*TYPE "END PASS"
3616 ;*IF THERES A MONITOR GO TO IT
3617 ;*IF THERE ISN'T JUMP TO BEGIN
3618 ;*IF IT IS DESIRED TO HAVE A BELL INDICATE THE "END OF PASS" LOCATION
3619 ;*$ENDMG CAN BE CHANGED TO 7.
3620
3621 016312 SEOP:
3622 016312 010701 SCOPE
3623 016314 005267 INC $PASS ;INCREMENT THE PASS NUMBER
3624 016320 042767 BIC #100000,$PASS ;DON'T ALLOW A NEG. NUMBER
3625 016326 005327 DEC (PC)+ ;LOOP?
3626 016330 000001 SEOPCT: ,WORD 1
3627 016332 003015 BGT $DOAGN ;YES
3628 016334 012737 MOV (PC)+,%(PC)+ ;RESTORE COUNTER
3629 016336 000001 SENDCT: ,WORD 1
3630 016340 016330 SEOPCT
3631 016342 000004 016372 TYPE , $ENDMG ;TYPE "END PASS"
3632 016346 SGET42:
3633
3634 016346 013700 MOV #42,R0 ;GET MONITOR ADDRESS
3635 016352 001405 BEQ $DOAGN ;BRANCH IF NO MONITOR
3636 016354 000005 RESET ;CLEAR THE WORLD
3637 016356 004710 SENDAD: JSR PC,(R0) ;GO TO MONITOR
3638 016360 000240 NOP ;SAVE ROOM
3639 016362 000240 NOP ;FOR
3640 016364 000240 NOP ;ACT11
3641 016366
3642 016366 000137 000600 $DOAGN: JMP @#BEGIN ;RETURN
3643 016372 005015 047105 020104 SENDMG: .ASCII <15><12>/END PASS/
3644 016400 040520 051523
3645 016404 377 000 SNULL: ,BYTE =1,-1,0 ;NULL CHARACTER STRING
3646 016410 ,EVEN
3647
3648 016336 000400 ENDCT: 400
3649
3650 ;*****
3651
3652 .SBTTL POWER FAIL ROUTINE
3653
3654
3655 016410 012737 016420 000024 $PWRDN: MOV $PWRUP,%#24
3656 016416 000000 HALT
3657
3658 016420 012706 000600 $PWRUP: MOV #BEGIN,SP ;RESTORE THE SP
3659 016424 012737 016410 000024 MOV $PWRDN,%#24
3660 016432 000004 000522 TYPE ,POWER ;GO AND TYPE "POWER"
3661 016436 000753 BR $DOAGN
    
```

```

3662
3663          ;*      HALT ROUTINE
3664          ;*      -----
3665          ;*
3666          ;*
3667          ;*      PROGRAM COMES HERE ON ENCOUNTERING ANY ERROR
3668          ;*
3669
3670 016440 017637 000000 000402 8HLT:  MOV    0(SP),0##FATAL ;PLACE THE ERROR NUMBER AT LOCATION 0FATAL
3671 016446 032777 020000 162026      BIT    #20000,0$WR  ;HAS THE OPERATOR ASKED TO SUPPRESS ERROR TYPE OUTS
3672 016454 001046      BNE    6$
3673 016456 000004 000514      TYPE   ,%CRLF      ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
3674 016462 010046      MOV    R0,-(SP)    ;SAVE R0
3675 016464 112767 000002 161757      MOVB  #2,$TPCNT    ;ALLOW TYPE OUTS OF PC AND ERROR NUMBER
3676 016472 016000 000002      MOV    2(SP),R0    ;BRING THE RETURN PC IN R0
3677 016476 162700 000004      SUB    #4,R0
3678 016502 112737 000006 000450 2$:  MOVB  #6,0$TYPCNT  ;ALLOW TYPE OUT OF 6 DIGITS
3679 016510 005046      CLR    -(SP)
3680 016512 000241      4$:   CLC
3681 016514 006100      ROL    R0
3682 016516 006116      ROL    (SP)        ;BRING THE C BIT FROM R0 IN (SP)
3683 016520 052716 000060      BIS    #60,(SP)    ;PREPARE TO TYPE IT OUT
3684 016524 004767 000130      JSR    PC,$TPCHR   ;AND GO TO OUT PUT A CHARACTER
3685 016530 005016      CLR    (SP)
3686 016532 006100      ROL    R0
3687 016534 006116      ROL    (SP)
3688 016536 006100      ROL    R0
3689 016540 006116      ROL    (SP)
3690 016542 105307 161702      DECB  TYPCNT      ;HAS ALL THE SIX CHARACTERS BEEN TYPED ?
3691 016546 001361      BNE    4$
3692 016550 005726      TST   (SP)+
3693 016552 017000 000002      MOV    02(SP),R0   ;RESTORE STACK POINTER
3694 016556 000004 000516      TYPE   ,%CRLF+2   ;PREPARE TO OUT PUT THE ERROR NUMBER
3695 016562 105367 161663      DECB  $TPCNT      ;GO AND TYPE 3 SPACES
3696 016566 001345      BNE    2$
3697 016570 012600      MOV    (SP)+,R0    ;IF BOTH PC AND ERROR NUMBER HAS NOT BEEN
3698 016572 105767 161622      6$:   TSTB  $ENV        ;REPORTED THEN REPEAT FROM 2$
3699 016576 001403      BEQ   8$           ;RESTORE R0
3700 016600 005237 000400      INC   0##MSGTY    ;IF WE ARE NOT UNDER APT, THEN GO TO
3701 016604 000777      BR    8$           ;8$
3702 016606 005777 161670      8$:   TST   0$WR      ;OTHERWISE INFORM APT, ABOUT SEEING THE ERROR
3703 016612 100001      BPL   10$          ;AND LOOP
3704 016614 000000      HALT
3705 016616 062716 000002      10$:  ADD    #2,(SP)    ;IS IT REQUIRED TO HALT ON ERROR ?
3706 016622 000207      RTS   PC          ;IF NOT THEN GO TO 10$
                        ;ADJUST THE RETURN ADDRESS
                        ;AND RETURN
    
```

```

3707
3708          ;*      TYPE OUT ROUTINE
3709          ;*      -----
3710          ;*
3711          ;*
3712          ;*      THIS ROUTINE IS USED TO TYPE ASCIZ MESSAGES
3713          ;*
3714
3715 016624 010046 000002 8TYPE:  MOV    R0,-(SP)    ;SAVE R0
3716 016626 017600      MOV    02(SP),R0   ;GET THE ADDRESS OF THE ASSCIZ STRING
3717 016632 112046 2$:   MOVB  (R0)+,-(SP) ;PUSH THE CHARACTER TO BE TYPED ONTO STACK
3718 016634 001005      BNE    4$
3719 016636 005726      TST   (SP)+
3720 016640 012600      MOV    (SP)+,R0    ;BRANCH IF IT IS NOT THE TERMINATOR
3721 016642 062716 000002 3$:   ADD    #2,(SP)    ;OTHERWISE RESTORE THE STACK AND R0
3722 016646 000002      RTI                ;ADJUST THE RETURN PC
3723
3724 016650 004767 000004 4$:   JSR    PC,$TPCHR   ;AND RETURN
3725 016654 005726      TST   (SP)+
3726 016656 000765      BR    2$           ;GO TO TYPE A CHARACTER
3727
3728 016660 132737 000040 000421 8TPCHR: BITB  #40,0##ENV    ;RESTORE THE STACK POINTER
3729 016666 001006      BNE    4$         ;AND RETURN TO 2$
3730 016670 105777 161616 2$:   TSTB  0$TPS      ;HAS THE CONSOLE OUTPUTS BEEN SUPPRESSED?
3731 016674 100375      BPL   2$         ;IF SO THEN RETURN FROM THE SUBROUTINE VIA 4$
3732 016676 116677 000002 161604 4$:   MOVB  2(SP),0$TPB ;IS THE PRINTER AVAILABLE?
3733 016704 000207      4$:   RTS   PC        ;IF NOT THEN LOOP HERE
3734          000001      ,END           ;OUT PUT THE CHARACTER
    
```

A	= 016410	DISPRE	000174	TST163	004430	TST247	012120	TTYOUT	000506
ABASE	= 000000	DUMMY	= 000000	TST164	004472	TST250	012212	TYPCNT	000450
ACDW1	= 000000	ENDCT	016336	TST165	004530	TST251	012306	TYPE	= 000004
ACDW2	= 000000	ENT176	005070	TST166	004554	TST252	012402	APTHD	000430
ACPUOP	= 000000	ENT51	002462	TST167	004604	TST253	012474	SCPUOP	000426
ADDW0	= 000000	ERRNM	= 000442	TST170	004642	TST254	012566	CRLF	000514
ADDW1	= 000000	F	= 000063	TST171	004666	TST255	012674	DEVC	000410
ADDW10	= 000000	N	= 000307	TST172	004714	TST256	013002	SDAGN	016366
ADDW11	= 000000	NEGAT	002062	TST173	004736	TST257	013114	ENDAD	016366
ADDW12	= 000000	POWER	000522	TST174	004774	TST260	013222	ENDCT	016336
ADDW13	= 000000	PSWORD	000432	TST175	005030	TST261	013330	ENDMG	016372
ADDW14	= 000000	REG01	003504	TST176	005106	TST262	013442	ENULL	016404
ADDW15	= 000000	REG1	001102	TST177	005202	TST263	013550	ENV	000420
ADDW2	= 000000	REG2	001244	TST200	005276	TST264	013656	ENVM	000421
ADDW3	= 000000	REG23	003706	TST201	005376	TST265	013764	EOP	016312
ADDW4	= 000000	REG3	001406	TST202	005472	TST266	014076	EOPCT	016330
ADDW5	= 000000	REG4	001546	TST203	005566	TST267	014200	EOPCT	016330
ADDW6	= 000000	REG45	004104	TST204	005666	TST270	014312	EOPCT	016330
ADDW7	= 000000	REG5	001712	TST205	005762	TST271	014424	EOPCT	016330
ADDW8	= 000000	RESTR	000222	TST206	006106	TST272	014532	EOPCT	016330
ADDW9	= 000000	RITSH	004312	TST207	006210	TST273	014640	EOPCT	016330
ADEVCT	= 000000	SCOPE	= 010701	TST210	006312	TST274	014730	EOPCT	016330
ADEVH	= 000000	SCOPE1	= 010701	TST211	006414	TST275	015030	EOPCT	016330
AENV	= 000000	SCOPE3	= 010703	TST212	006514	TST276	015136	EOPCT	016330
AENVM	= 000000	START	000740	TST213	006614	TST277	015244	EOPCT	016330
AFATAL	= 000000	SWR	000502	TST214	006714	TST300	015352	EOPCT	016330
AMADR1	= 000000	SWREG	= 000176	TST215	007016	TST301	015456	EOPCT	016330
AMADR2	= 000000	SW09	= 001000	TST216	007120	TST302	015562	EOPCT	016330
AMADR3	= 000000	SW10	= 002000	TST217	007220	TST303	015666	EOPCT	016330
AMADR4	= 000000	SW11	= 004000	TST220	007320	TST304	015774	EOPCT	016330
AMAMS1	= 000000	SW12	= 010000	TST221	007414	TST305	016102	EOPCT	016330
AMAMS2	= 000000	S0	000452	TST222	007510	TST306	016206	EOPCT	016330
AMAMS3	= 000000	S1	000454	TST223	007604	TST37	002110	EOPCT	016330
AMAMS4	= 000000	S10	000476	TST224	007704	TST40	002144	EOPCT	016330
AMSGAD	= 000000	S11	000500	TST225	010000	TST41	002160	EOPCT	016330
AMSGLG	= 000000	S2	000456	TST226	010074	TST42	002176	EOPCT	016330
AMSGTY	= 000000	S3	000460	TST227	010174	TST43	002232	EOPCT	016330
AMTYP1	= 000000	S4	000462	TST230	010270	TST44	002264	EOPCT	016330
AMTYP2	= 000000	S5	000464	TST231	010364	TST45	002316	EOPCT	016330
AMTYP3	= 000000	S6	000466	TST232	010460	TST46	002350	EOPCT	016330
AMTYP4	= 000000	S7	000470	TST233	010560	TST47	002406	EOPCT	016330
APASS	= 000000	S8	000472	TST234	010654	TST50	002436	EOPCT	016330
APRIOR	= 000000	S9	000474	TST235	010750	TST51	002514	EOPCT	016330
ASWREG	= 000000	TEMP1	000434	TST236	011044	TST52	002574	EOPCT	016330
ATESTN	= 000000	TEMP2	000436	TST237	011144	TST53	002654	EOPCT	016330
AUNIT	= 000000	TEMP3	000440	TST240	011240	TST54	002734	EOPCT	016330
AUSWR	= 000000	TEMP4	000442	TST241	011334	TST55	003012	EOPCT	016330
AVECT1	= 000000	TEMP5	000444	TST242	011450	TST56	003070	EOPCT	016330
AVECT2	= 000000	TEMP6	000446	TST243	011544	TST57	003146	EOPCT	016330
BEGIN	000600	TST160	004346	TST244	011640	TST60	003226	EOPCT	016330
COUNT	000430	TST161	004374	TST245	011734	TST61	003306	EOPCT	016330
DISFLA	000504	TST162	004412	TST246	012026	TST62	003364	EOPCT	016330

, ABS. 016706 000

ERRORS DETECTED: 0

DFKACA,DFKACA=DFKACA,SRC/SOL
 RUN-TIME: 10 11 ,2 SECONDS
 RUN-TIME RATIO: 210/22=9,5
 CORE USED: 11K (21 PAGES)