

LISTING

096-000444-00

PROGRAM

MICRO NOVA ARITHMETIC TEST

TAPE

095-000444-00

ABSTRACT

ARITHMETIC TEST IS A MAINTENANCE PROGRAM DESIGNED TO EXERCISE THE ARITHMETIC AND LOGICAL INSTRUCTIONS OF THE NOVA COMPUTERS. THE PROGRAM ADJUSTS ITS PARAMETERS TO THE SIZE OF MEMORY AND RELUCATES ITSELF TO VARIOUS AREAS OF MEMORY.

0001 ARITH MACRO REV 04.00

12:29:55 12/03/76

```

01
02
03
04
05
06
07 ;*****
08 ;
09 ; NAME: ARITHST.SR          PART NUMBER: 094-000842
10 ;
11 ;
12 ; DESCRIPTION: ARITHMETIC TEST (MICRO NOVA)
13 ;
14 ;
15 ; REVISION HISTORY:
16 ;
17 ;     REV.      DATE
18 ;
19 ;     00       12/03/76
20 ;
21 ;
22 ; COPYRIGHT (C) DATA GENERAL CORPORATION, 1976
23 ; ALL RIGHTS RESERVED.
24 ;*****

```

10002 ARITH

```

01
02 ; ARITHMETIC TEST
03
04
05
06 ;1. ABSTRACT
07 ; ARITHMETIC TEST IS A MAINTENANCE PROGRAM
08 ; DESIGNED TO EXERCISE THE ARITHMETIC AND
09 ; LOGICAL INSTRUCTIONS OF THE NOVA COMPUTERS,
10 ; THE PROGRAM ADJUSTS ITS PARAMETERS TO THE
11 ; SIZE OF MEMORY AND RELOCATES ITSELF TO VAR-
12 ; IOUS AREAS OF MEMORY.
13
14 ;2. MACHINE REQUIREMENTS
15 ;2.1 NOVA FAMILY PROCESSOR
16 ;2.2 4K READ/WRITE MEMORY
17 ;2.3 TELETYPE
18
19 ;3. SWITCH SETTINGS
20 ;
21 ; LOCATION "SWREG" IS USED TO SELECT THE PROGRAM OPTIONS
22 ; (NOT SYSTEM CONFIGURATION). WHILE RUNNING UNDER DTOS,
23 ; THIS LOCATION WILL BE LOADED BY THE MONITOR,
24 ; HOWEVER UNDER STAND ALONE AND PROGRAM LOAD MODES THIS
25 ; LOCATION WILL BE SET ACCORDING TO THE ANSWERS SUPPLIED
26 ; BY THE OPERATOR. IN ANY CASE THE OPTIONS CAN BE CHANGED
27 ; OR VERIFIED BY USING ONE OF THE COMMANDS GIVEN IN SEC.
28 ; 3.2
29
30 ;3.1 SWITCH OPTIONS
31 ; DIFFERENT BITS AND THEIR INTERPRETATION AT LOCATION
32 ; "SWREG" IS AS FOLLOWS:
33 ;
34 ; BIT      OCTAL      BINARY      INTERPHETATION
35 ; VALUE    VALUE
36 ;
37 ; 1         40000      0           LOOP ON ERROR
38 ;          40000      1           SKIP LOOPING ON ERROR
39 ;
40 ; 2         20000      0           PRINT TO CONSOLE
41 ;          20000      1           ABORT PRINT OUT TO CONSOLE
42 ;
43 ; 3         10000      0           PRINT DETAILED ERROR ON THE
44 ;          10000      1           SELECTED DEVICE/DEVICES
45 ;                                     ONLY % FAILURE REQUIRED
46 ;
47 ; 4         04000      0           ALLOW END OF PASS PRINT OUT
48 ;          04000      1           SUPPRESS END OF PASS PRINT OUT
49 ;
50 ; 5         02000      0           DO NOT PRINT ON THE LINE PRINTER
51 ;          02000      1           PRINT ON THE LINE PRINTER
52 ;
53 ; 6         01000      0           DO NOT HALT ON ERROR
54 ;          01000      1           HALT ON ERROR
55 ;
56 ;
57 ;3.2 SWITCH COMMANDS
58 ; ONCE THE PROGRAM STARTS EXECUTING THE STATE OF ANY OF
59 ; THE BITS CAN BE CHANGED BY HITTING KEYS 1 THROUGH 6, THE
60 ; PROGRAM WILL CONTINUE RUNNING AFTER UPDATING THE OPTIONS

```

0003 AKITH

```
01 / EACH KEY WILL COMPLEMENT THE STATE OF THE BIT AFFILIAT-
02 / ED WITH IT, THUS BIT 4 CAN BE ALTERED BY HITTING KEY 4.
03 / SETTIN UP ANY BIT OF LOCATION "SWREG" WILL SET BIT 0.
04 / (DEFAULT MODE IS DEFINED AS ALL BITS OF SWREG SET TO 0)
05 / THE PROGRAM CAN BE LOCKED INTO SWITCH MODIFICATION MODE
06 / BY TYPING A 0, IN WHICH CASE MORE THAN ONE BITS CAN BE
07 / CHANGED BEFORE THE CONTROL IS ALLOWED TO RETURN TO THE
08 / MAIN PROGRAM.
09 /
10 /3.2.1 OTHER COMMANDS
11 /
12 / "CR" A "RETURN" CAN BE TYPED TO CONTINUE THE PROGRAM
13 / AFTER ITS LOCKED IN A SWITCH MODIFICATION MODE
14 /
15 / AD THIS COMMAND GIVEN AT ANY TIME WILL RESET "SWREG
16 / TO DEFAULT MODE AND RESTART THE PROGRAM.
17 /
18 / AR THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE
19 / PROGRAM, SWITCHES ARE LEFT WITH THE VALUES THEY
20 / HAD BEFORE THE COMMAND WAS ISSUED.
21 /
22 / AD THIS COMMAND GIVEN AT ANY TIME WILL CAUSE THE
23 / PROGRAM CONTROL TO GO TO ODT (SEE SEC. 6)
24 /
25 /
26 / M THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE
27 / CURRENT OPERATING MODES.
28 /
29 /
30 / OPERATING PROCEDURE
31 /4.1 LOAD THE PROGRAM VIA THE BINAKY LOADER
32 / AND STANT AT 200.
33 / SEE SECTION 0,0 FOR RESTART PROCEDURE.
34 /4.2 UPON STARTING, THE PROGRAM WILL PRINT
35 / THE MESSAGE "LAST LOCATION IN MEMORY IS XXXXX".
36 / THE OPERATOR SHOULD CHECK THE VALUE PRINTED
37 / WITH THE SYSTEM MEMORY SIZE. IF THEY DO NOT
38 / AGREE, A ERRON HAS OCCURED.
39 /4.3 THE PROGRAM WILL RUN UNTILL MANUALLY STOPPED.
40 / AT THE END OF EACH PROGRAM ITERATION THE WORD
41 / "PASS" WILL BE PRINTED AND THE PROGRAM WILL
42 / RELOCATE ITSELF TO ANOTHER AREA OF MEMORY.
43 /
44 /
45 /15. PROGRAM OUTPUT/ERROR DESCRIPTION
46 /15.1 WHEN A ERROR IS DETECTED BY THE PROGRAM THE
47 / FOLLOWING WILL BE PRINTED:
48 /15.1.1 ABS PC THE MEMORY LOCATION OF THE
49 / ERROR SUBROUTINE CALL,
50 /15.1.2 LIST PC WHERE TO LOOK IN THE LISTING
51 / TO FIND THE FAILING ROUTINE.
52 /15.1.3 ORIGINAL THE ACCUMULATOR VALUES AS
53 / AC0,AC1,AC2 DETERMINED VIA THE RANDOM
54 / NUMBER GENERATOR.
55 /15.1.4 RESULT THE RESULT IN THE ACCUMULATORS
56 / AC0,AC1,AC2 PRIOR TO THE ERROR CALL
57 /15.2 AFTER THE PRINTOUT THE PROGRAM WILL ITERATE THE
58 / FAILING ROUTINE WITH THE SAME NUMBERS AS PRINTED.
59 / CONSOLE SWITCH 3(1) MAY BE USED TO DETERMINE THE
60 / RATE OF FAILURE. SWITCH 1(1) WILL CAUSE THE
```

0004 ARITH

```
01 / PROGRAM TO PROCEED TO THE NEXT TEST.
02 /
03 /16. PROGRAM DESCRIPTION
04 /16.1 THIS PROGRAM CONSIST OF A NUMBER OF SUBROUTINES
05 / PERFORMING VARIOUS OPERATIONS ON SETS OF PSEUDO
06 / RANDOM NUMBERS, SOME ROUTINES PERFORM MORE THEN
07 / ONE OPERATION , SUCH AS TAKING THE SQUARE ROOT OF
08 / A SQUARED NUMBER, WHEN THE OPERATOR QUESTIONS
09 / WHICH RESULT IS CORRECT, THE OPERATIONS MAY BE
10 / PERFORMED WITH PENCIL AND PAPER ON THE ORIGINAL
11 / NUMBERS.
12 /16.2 AT THE END OF EACH PROGRAM ITERATION THE PROGRAM
13 / RELOCATES ITSELF TO ANOTHER AREA OF MEMORY FOR
14 / EXECUTION AT THAT SPOT. THIS RELOCATION PLACES
15 / DIFFERENT OPERATING CONDITIONS ON THE PROCESSOR
16 / REGISTERS SUCH AS, THE MEMORY ADDRESS REGISTER
17 / AND THE PROGRAM COUNTER.
18 /16.3 BECAUSE THIS PROGRAM IS USED AS A FINAL TEST OF
19 / THE ARITHMETIC AND LOGICAL INSTRUCTIONS IT IS
20 / COMPLEX AND DIFFICULT TO TROUBLE SHOOT, IT IS
21 / THEREFOR SUGGESTED THAT THIS PROGRAM BE RUN
22 / AFTER OTHER PROCESSOR TEST,
```

16065 ARITH

```

01 17. OCTAL DEBUG TOOL (ODT)
02 /
03 / THE DIAGNOSTIC IS EQUIPPED WITH A BUILT IN ODT WHICH CAN
04 / BE ACCESSED BY HITTING CONTROL O (AO) AT ANY TIME DURING
05 / THE EXECUTION OF THE PROGRAM (AFTER SETTING THE PARA-
06 / METERS).
07 / ON ENTERING ODT THE ADDRESS OF THE LOCATION HAVING THE
08 / NEXT INSTRUCTION TO BE EXECUTED WILL BE TYPED-OUT.
09 /
10 17.1 CONVENTIONS AND SYMBOLS
11 / THE FOLLOWING CONVENTIONS ARE USED BY THE ODT:
12 / ? PRESSING ANY ILLEGAL KEY CAUSES THE ODT TO RES-
13 / POND WITH A "?".
14 / * ODT IS READY AND AT YOUR SERVICE.
15 /
16 17.2 COMMAND STRUCTURE
17 / AN ODT COMMAND THE FOLLOWING FORMAT:
18 / [ARGUMENT][COMMAND]
19 / AN ARGUMENT MAY BE ONE OF THE FOLLOWING:
20 / EXP AN OCTAL EXPRESSION CONSISTING OF OCTAL NUMBERS
21 / SEPARATED BY PLUS (+) OR MINUS (-) SIGNS. LEAD-
22 / ING ZEROS NEED NOT BE TYPED.
23 / ADR AN ADDRESS IS THE SAME AS AN EXPRESSION EXCEPT
24 / THAT BIT 0 IS NEGLECTED.
25 / A COMMAND IS A SINGLE TELETYPE CHARACTER
26 /
27 17.3 ODT COMMANDS
28 / THE LOCATIONS THAT CAN BE EXAMINED AND MODIFIED BY THE
29 / USER ARE CALLED CELLS. THESE CELLS ARE OF TWO TYPE:
30 / INTERNAL CPU CELLS AND MEMORY LOCATIONS,
31 /
32 17.3.1 OPENING INTERNAL CELLS
33 / THE COMMAND TO OPEN ONE OF THE INTERNAL REGISTERS IS OF
34 / THE FORM "NA" WHERE N IS ANY OCTAL EXPRESSION BETWEEN
35 / 0 AND 7
36 / 0-3 FOR ACCUMULATORS 0-3
37 / 4 FOR PC OF THE NEXT INSTRUCTION TO BE EXECUTED IN
38 / THE EVENT OF A "P" COMMAND.
39 / 5 CPU AND I/O STATUS
40 / BIT INTERPRETATION
41 / 15 STATUS OF I/O DONE FLAG
42 / 14 STATUS OF INTERRUPTS
43 / 13 STATUS OF CARRY BIT
44 / 6 ADDRESS OF THE LOCATION HAVING BREAK POINT (IF
45 / ANY)
46 / 7 INSTRUCTION AT THE BREAK POINT LOCATION
47 /
48 / OTHER COMMANDS TO OPEN CELLS ARE:
49 /
50 / ADR/ OPEN THE CELL AND PRINT ITS CONTENTS
51 / ./ OPEN THE CELL CURRENTLY POINTED BY THE POINTER
52 / AND PRINT ITS CONTENTS.
53 / .+ADR/ ADD ADR TO THE POINTER, OPEN THE CELL AND PRINT
54 / ITS CONTENTS.
55 / .-ADR/ SUBTRACT ADR FROM THE POINTER, OPEN THE CELL AND
56 / PRINT ITS CONTENTS.
57 / "CR" THE RETURN KEY IS USED TO CLOSE THE OPEN CELL
58 / WITH OR WITHOUT MODIFICATION.
59 / "LF" LINE FEED IS USED TO CLOSE THE OPEN CELL WITH OR
60 / WITHOUT MODIFICATION AND TO OPEN THE SUCCEEDING

```

16066 ARITH

```

01 / CELL.
02 / / CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
03 / OPEN THE CELL POINTED BY ITS CONTENTS.
04 / +ADR/ CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
05 / OPEN THE CELL POINTED BY ITS CONTENTS + ADR.
06 / -ADR/ CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
07 / OPEN THE CELL POINTED BY ITS CONTENTS - ADR.
08 /
09 17.3.2 MODIFICATION OF A CELL
10 / ONCE A CELL HAS BEEN OPENED ITS CONTENTS CAN BE MODIFIED
11 / BY TYPING THE NEW VALUE THE CELL IS TO CONTAIN IN THE
12 / FORM OF AN OCTAL EXPRESSION FOLLOWED BY "CR" OR "LF".
13 / IF A + OR - IS TYPED AS THE FIRST CHARACTER OF THE EX-
14 / PRESSION THEN THE VALUE OF THE EXPRESSION IS ADDED TO OR
15 / SUBTRACTED FROM THE OLD CONTENTS OF THE CELL.
16 / ADDRESS ITSELF OR AN EXPRESSION RELATIVE TO THE ADDRESS
17 / CAN BE DEPOSITED BY TYPING A "." OR ".+/-OCTAL EXPRESS-
18 / ION". A RUBOUT COMMAND GIVEN RIGHT AFTER OPENING A CELL
19 / ALLOWS THE MODIFICATION OF ITS CONTENTS AS IF THEY WERE
20 / TYPED IN JUST BEFORE THE COMMAND WAS ISSUED.
21 /
22 17.3.3 OTHER ODT COMMANDS
23 / RUBOUT THIS KEY IS USED TO DELETE ERRONEOUSLY TYPED
24 / DIGITS. EACH TIME THE KEY IS PRESSED RIGHT MOST
25 / DIGIT IS DELETED AND ECHOED ON THE TERMINAL. IF
26 / THE RUBOUT KEY IS PRESSED RIGHT AFTER OPENING A
27 / CELL THEN IT ALLOWS THE MODIFICATION OF THE CELL
28 / AS IF ITS CONTENTS WERE TYPED IN JUST BEFORE THE
29 / KEY WAS PRESSED.
30 / ADRB INSERT A BREAK POINT AT LOCATION "ADR".
31 / ONLY ONE BREAK POINT CAN BE INSERTED AND ANY
32 / ENTRY TO ODT AFTER EXECUTING A BREAK POINT WILL
33 / CAUSE IT TO BE DELETED.
34 / D DELETE THE BREAK POINT IF ANY.
35 / P RESTART THE EXECUTION OF THE PROGRAM AT LOCATION
36 / POINTED BY 4A.
37 / ADRK START EXECUTING THE PROGRAM AT "ADR" AFTER AN
38 / I/O-RESET.
39 / K KILL THE STRING TYPED SO FAR. THE ODT RESPONDS
40 / WITH A "?" AND THE OPEN CELL IS CLOSED WITHOUT
41 / MODIFICATION.
42 /
43 17.3.4 MISC
44 / IF THIS PROGRAM IS MANUALLY STOPPED WHEN IN THE
45 / PROCESS OF RELOCATION, IT CAN BE RESTARTED AT LOCATI
46 /
47 /

```

```

0007 AKITH
01 000001 .NOCON 1
02 000000 .NOLOC 0
03
04 000000 .LOC 0
05 000000 004222 DIRT
06 000001 000000 0
07 000002 000200 JMP SIZIT
08 000003 000200 JMP SIZIT
09 001277 MNRST=DUAC 0,CPU
10 000000 .NOLOC 0
11 000200 RES?T=200
12 000300 SYNC=NIDP 0
13 000045 .LOC 45
14 000045 000105 EGG5: 0
15 000046 000000 0
16 000047 002725 IERR: EKR?
17 000050 000010 .BLK 0.
18 000060 002462 IMOVE: MOVE
19 000061 002424 PINTM: INTM
20 000062 003142 ICRLF?: CKL?F
21 000063 003514 INP?: INP?I
22 000064 003065 IMES?: MES?S
23 000065 003104 IPDC?: PDC?S
24 000066 003156 IPDC?: PDC?T
25 000067 003311 IYPT?: YPT?E
26 000070 003152 IZOC?: ZOC?T
27 000071 003756 IUOT?: OUT?I
28 000072 000000 INTS?: 0
29 000073 000000 INTS?: 0
30 000074 000017 CO: 17
31 000075 002447 ICAL: CAL
32 000076 002462 ICAT: MOVE
33 000077 004175 ISHIGH: LAST
34 001000 000500 ISLOW: FIRST
35 001001 002574 IENIER: ENT?R
36 001002 002604 ICYCLE: CYC?E
37 001003 002522 IKAND: RAND
38 001004 002317 ISEQEN: SEGER
39 001005 000000 EGG5: 0
40 001006 000000 0
41 001007 000000 0
42 001100 000000 0
43 001111 000000 0
44 001112 000000 SWREG: 0
45 001113 177770 M8: -10
46 001114 177760 M20: -20
47 001115 004175 SHIGH: LAST
48 001116 000500 SLOW: FIRST
49 001117 000000 CALRET: 0
50 001200 000000 CAL2: 0
51 00121 000047 UPREL: 47
52 00122 000000 RELOC: 0
53 00123 000000 MSIZE: 0
54 00124 000000 MOVTEM: 0
55 00125 000000 RANRET: 0
56 00126 000142 KDATA: ESWIT-1
57 00127 000144 ITR: 144
58 00130 174303 PSIZE: FIRST-LAST
59 00131 000402 K402: 402
60 00132 000377 K377: 377

```

```

JOTOS DIRT BLK PNTR
JPROGRAM STARTS HERE

```

```

JOTOS EGGS BLK PNTR
JRETURN FROM TRAP POINTER
JPOINTER TO ERROR HANDLER
JFOR DEBUGGER
JINDIRECT POINTERS TO

```

PDC?S

JPROGRAM SUBROUTINES

```

JPOINTER TO OUT OF SEQUENCE
JOTOS AUTO MODE SW
JDEVICE CODE
JCAT SWITCH
J# OF PASSES THIS RUN
JOTOS RTN ADDR
JSWITCH REGISTER

```

```

JWHERE LAST SUBROUTINE CALLED FROM
J(CAC2) AT THAT TIME
JADDRESS OF INDECT SUBROUTINE TABLE
JCURRENT RELOCATION
JMEMORY SIZE
JRETURN FOR RANDOM NUMBER
JROUTINE ITERATION CONSTANT
JPROGRAM SIZE

```

```

0008 AKITH
01 00133 000240 K24M: 240
02 00134 000060 K6P: 60
03 00135 000200 K20P: 200
04 00136 002000 K20MP: 2000
05 00137 052525 K525: 52525
06 00140 000000 CALIEM: 0
07 00141 004175 KENU: FINISH JPROGRAM END
08 00142 000007 K7: 7
09 00143 000000 ESWIT: 0
10 00144 000000 LSTPC: 0
11 00145 000000 OAC1: 0
12 00146 000000 OAC1: 0
13 00147 000000 OAC2: 0
14 00150 000000 SAV0: 0
15 00151 000000 SAV1: 0
16 00152 000000 SAV2: 0
17 00153 000000 RANCT: 0
18 00154 000000 TSTNM: 0
19
20 00155 000000 .UD01: 0
21 00156 000000 .UD02: 0
22 00157 000000 .UD03: 0
23
24 00160 000000 .UD10: 0
25
26 00161 033031 .UD20: 33031
27 00162 000010 .UD21: 10
28
29 00163 000000 HELP: 0
30 00164 000000 PASS: 0
31 00165 000000 AC3?: 0
32 00166 003514 INP?: INP?I

```

```

JERRON SWITCH/ABS PC OF ERROR.
JWHERE ERROR MAY BE FOUND IN LISTING.

```

```

JORIGINAL AND CURRENT AC VALUES.
JRANDOM LOOP LIMIT COUNTER
JTEST NUMBER STORAGE LOCATION

```

```

J *SAVE AC1
J *SAVE AC2
J SAVE RETURN
J ITERATION COUNT
J INCREMENT
J ITERATION COUNT, X

```

```

I0009 ARITH
01      000200 .LOC 200
02
03 00200 030001 SIZIT: LDA 2,PINTM IGET INTERRUPT POINTER
04 00201 050001 STA 2,1 ISAVE AT LOCATION 1
05 00202 060177 INTEN
06 00203 152400 SUB 2,2 ICLEAR AC2
07 00204 050122 STA 2,RELOC ICLEAR RELOCATION COUNT
08 00205 034136 LDA 3,K2000 ISTART SIZING AT 2000
09 00206 030137 LDA 2,K025 IGET TEST DATA
10 00207 025400 SIZI1: LDA 1,0,3 IGET THE MEMORY DATA
11 00210 051400 STA 2,0,3 ISTORE THE TEST DATA
12 00211 021400 LDA 0,0,3 IHEAD BACK THE TEST DATA
13 00212 142404 SUB 2,0,SZR IIS THE MEMORY LOCATION PRESENT
14 00213 000221 JMP SIZI3 INO. ITS THE END OF MEMORY
15 00214 045400 STA 1,0,3 IRESTORE THE MEM LOC
16 00215 175400 INC 3,3 IUPDATE MEM ADDRESS POINTER
17 00216 175132 MOVZL# 3,3,SZC IIS IT THE MAX MEM POSSIBLE
18 00217 000221 JMP SIZI3 IYES
19 00220 000207 JMP SIZI1 IEND CONTINUE SIZING
20 00221 054123 SIZI3: STA 3,MSIZE ISAVE TOP OF MEM
21 00222 014123 NSZ MSIZE IDECREASE BY ONE
22 00223 024105 LDA 1,EGBS
23 00224 125004 MOV 1,1,SZR
24 00225 000234 JMP .+7
25
26 00226 000075 JSR #ICAL
27 00227 003065 MESS
28 00230 004104 WORLD
29 00231 024123 LDA 1,MSIZF
30 CALL
31 00232 000075 JSR #ICAL
32 00233 003156 PUCTAL IPRINT HIGHEST MEMORY LOC.
33 00234 020135 LDA 0,K200
34 00235 034107 LDA 3,EGBS+2
35 00236 175004 MOV 3,3,SZR
36 00237 020136 LDA 0,K2000
37 00240 024123 LDA 1,MSIZE
38 00241 100400 SUB 0,1
39 00242 044123 STA 1,MSIZE
40 00243 020131 LDA 0,K002
41 00244 107000 ADD 0,1
42 00245 125400 INC 1,1
43 00246 175004 MOV 3,3,SZR
44 00247 044076 STA 1,ICAT
45 CALL
46 00250 005075 JSR #ICAL
47 00251 000500 FIRST

```

```

I0010 ARITH
01
02
03      000500 .LOC 500
04 00500 101000 FIRST: MOV 0,0
05 00501 102400 SUB 0,0 ICLEAR AC0
06 00502 040153 STA 0,RANDT ICLEAR RANDOM LOOP COUNTER
07 00503 040154 STA 0,TESTM ICLEAR TEST NUMBER COUNTER
08

```

```

0011 ARITH
01 *****
02 J1ST#1 MISC TEST OF NEG/COM
03 *****
04
05 00504 010154 T1: ISZ TSTNM JENTERING A NEW TEST
06
07 MS1: SETUP JMISC TEST OF NEG/COM
08 00505 000101 JSR @IENTER
09 RANDOM
10 00506 000103 JSR @IRAND
11 00507 104700 NEG# 0,1
12 00510 130342 COM# 1,2,SZC
13 00511 142014 ADC# 2,0,SZR
14 EKRRR
15 00512 000047 JSR @IEKR
16 LOOP
17 00513 000102 JSR @ICYCLE
18
19 00514 000402 JMP .+2
20 00515 000001 1
21 00516 024777 TN1: LDA 1,TN1
22 00517 020154 ET1: LDA 0,TSTNM JGET THE TEST NUMBER
23 00520 122454 SUB# 1,0,SZR JCHECK THE PROGRAM FLOW
24 00521 000104 JSR @ISEQER

```

```

0012 ARITH
01 *****
02 J1ST#2 MISC TEST OF INC SWAPPED
03 *****
04
05 00522 010154 T2: ISZ TSTNM JENTERING A NEW TEST
06
07 MS2: SETUP JMISC TEST OF INC SWAPPED.
08 00523 000101 JSR @IENTER
09 RANDOM
10 00524 000103 JSR @IRAND
11 00525 111700 INCS 0,2
12 00526 145323 MOVZS 2,1,SNC
13 00527 106314 AUCS# 0,1,SZR
14 EKRRR
15 00530 000047 JSR @IEKR
16 LOOP
17 00531 000102 JSR @ICYCLE
18
19 00532 000402 JMP .+2
20 00533 000002 TN2: 2
21 00534 024777 ET2: LDA 1,TN2
22 00535 020154 LDA 0,TSTNM JGET THE TEST NUMBER
23 00536 122454 SUB# 1,0,SZR JCHECK THE PROGRAM FLOW
24 00537 000104 JSR @ISEQER

```

0013 ARITH

```

01
02
03
04
05 00540 010154 T3: ISZ TSTNM JENTERING A NEW TEST
06
07 MS3: SETUP MISC NEG TEST
08 00541 000101 JSR #IENTER
09 00542 102400 SUB 0,P
10 00543 100644 NEGCM 0,0,SZR
11 ERROR
12 00544 000047 JSR #IEKR
13 00545 100664 NEGCR 0,0,SZR
14 ERROR
15 00546 000047 JSR #IEKR
16 00547 100664 NEGCR 0,0,SZR
17 ERROR
18 00550 000047 JSR #IEKR
19 LOOP
20 00551 000102 JSR #ICYCLE
21
22
23 00552 000402 JMP ,+2
24 00553 000003 TN3: 3
25 00554 024777 ET3: LDA 1,TN3
26 00555 020154 LDA 0,TSTNM JGET THE TEST NUMBER
27 00556 122454 SUBCM 1,0,SZR JCHECK THE PROGRAM FLOW
28 00557 000104 JSR #ISEQER

```

0014 ARITH

```

01
02
03
04
05 00560 010154 T4: ISZ TSTNM JENTERING A NEW TEST
06
07 ROT1: SETUP JTEST ROTATE LEFT
08 00561 000101 JSR #IENTER
09 RANDOM
10 00562 000103 JSR #IRAND
11 00563 105000 MOV 0,1
12 00564 101100 MOVL 0,0
13 00565 111100 MOVL 0,2
14 00566 151100 MOVL 2,2
15 00567 151100 MOVL 2,2
16 00570 155100 MOVL 2,3
17 00571 175100 MOVL 3,3
18 00572 175100 MOVL 3,3
19 00573 161100 MOVL 3,0
20 00574 101100 MOVL 0,0
21 00575 101100 MOVL 0,0
22 00576 101100 MOVL 0,0
23 00577 101100 MOVL 0,0
24 00600 101100 MOVL 0,0
25 00601 101100 MOVL 0,0
26 00602 101100 MOVL 0,0
27 00603 101100 MOVL 0,0
28 00604 101100 MOVL 0,0
29 00605 100414 SUBM 0,1,SZR
30 ERROR
31 00606 000047 JSR #IEKR
32 LOOP
33 00607 000102 JSR #ICYCLE

```


10015 ARITH

```
01
02
03 00010 000402      JMP      .+2
04 00011 000004 TN4:  4
05 00012 024777 ET4: LDA      1,TN4
06 00013 020154      LDA      0,TSTNM
07 00014 122454      SUBW   1,0,SZR
08 00015 000104      JSR      @ISEGER
```

IGET THE TEST NUMBER
ICHECK THE PROGRAM FLOW

0016 ARITH

```
01 *****
02 ITST#5 TEST ROTATE RIGHT *****
03
04
05 00016 010154 T5:   ISZ      TSTNM      ENTERING A NEW TEST
06
07 ROT2:  SETUP      TEST ROTATE RIGHT
08 00017 000101      JSR      @IENTER
09
10 00020 000103      JSR      @IRAND
11 00021 131000      MOV  1,2
12 00022 121200      MOVW 1,0
13 00023 101200      MOVW 0,0
14 00024 101200      MOVW 0,0
15 00025 101200      MOVW 0,0
16 00026 101200      MOVW 0,0
17 00027 101200      MOVW 0,0
18 00030 101200      MOVW 0,0
19 00031 101200      MOVW 0,0
20 00032 101200      MOVW 0,0
21 00033 101200      MOVW 0,0
22 00034 105200      MOVW 0,1
23 00035 125200      MOVW 1,1
24 00036 135200      MOVW 1,3
25 00037 175200      MOVW 3,3
26 00040 175200      MOVW 3,3
27 00041 151200      MOVW 3,0
28 00042 101200      MOVW 0,0
29 00043 112414      SUBW 0,2,SZR
30 ERROR
31 00044 000047      JSR      @IERR
32 LOOP
33 00045 000102      JSR      @ICYCLE
34
35 00046 000402      JHP      .+2
36 00047 000005 TN5:  5
37 00050 024777 ET5: LDA      1,TN5
38 00051 020154      LDA      0,TSTNM
39 00052 122454      SUBW   1,0,SZR
40 00053 000104      JSR      @ISEGER
```

IGET THE TEST NUMBER
ICHECK THE PROGRAM FLOW

0017 ARITH

```

01 *****
02 ;TST#6 TEST ANY NUMBER ANDED TO ITSELF DOES NOT CHANGE
03 *****
04
05 00654 010154 T6: ISZ TSTNM ;ENTERING A NEW TEST
06
07 AND#1: SETUP ;ANY NUMBER ANDED WITH
08 00655 006101 JSR #IENTER ;ITSELF SHOULD NOT
09 RANDOM ;
10 00656 006103 JSR #IRAND ;BE CHANGED.
11 00657 131000 MOV 1,2
12 00660 127700 ANDS 1,1
13 00661 127700 ANDS 1,1
14 00662 127700 ANDS 1,1
15 00663 127700 ANDS 1,1
16 00664 127700 ANDS 1,1
17 00665 135300 MOVS 1,3
18 00666 177700 ANDS 3,3
19 00667 177700 ANDS 3,3
20 00670 101300 MOVS 0,0
21 00671 103700 ANDS 0,0
22 00672 112414 SUB# 0,2,SZK
23 ERROR
24 00673 006047 JSR #IEKR
25 LOOP
26 00674 006102 JSR #ICYCLE
27
28 00675 000402 JMP +2
29 00676 000006 TN6: 6
30 00677 024777 ET6: LDA 1,TN6
31 00700 020154 LDA 0,TSTNM ;GET THE TEST NUMBER
32 00701 122454 SUB# 1,0,SZR ;CHECK THE PROGRAM FLOW
33 00702 006104 JSR #ISEQR

```

0018 ARITH

```

01 *****
02 ;TST#7 TEST A NUMBER ANDED WITH ITS COMPLIMENT EQUALS 0
03 *****
04
05 00700 010154 T7: ISZ TSTNM ;ENTERING A NEW TEST
06
07 AND#1: SETUP ;A NUMBER ANDED WITH ITS
08 00704 006101 JSR #IENTER ;COMPLIMENT SHOULD
09 RANDOM ;
10 00705 006103 JSR #IRAND
11 00706 104042 COM# 0,1,SZC ;PRODUCE ZERO RESULT.
12 00707 123704 ANDS 1,0,SZR
13 ERROR
14 00710 006047 JSR #IEKR
15 LOOP
16 00711 006102 JSR #ICYCLE
17

```

10019 ARITH

01			
02	0P712	000442	JMP
03	0P713	000007	TN7:
04	0P714	024777	ET7:
05	0P715	020154	LDA
06	0P716	122454	SUBUM
07	0P717	000104	JSR

1GET THE TEST NUMBER
1CHECK THE PROGRAM FLOW

0020 ARITH

31				
32				11ST#10 TEST THE AND FOR CORRECT OPERATION
33				
34				
05	0P720	010154	TIP:	ISZ
06				TSTNM
07				JENTERING A NEW TEST
08	0P721	000101	AND3:	SETUP
09				JPERFORM A AND INSTRUCTION
10	0P722	000103		JSR
11	0P723	170620		RANDOM
12	0P724	131000		JWITH THE RESULT IN AC2.
13	0P725	113400		JSR
14	0P726	101113		IRAND
15	0P727	000403		JSIMULATE THE AND VIA
16	0P730	120112		LOOKING FOR ADDEN CARRY.
17	0P731	101141		SUBZR 3,3
18	0P732	101121		MOV 1,2
19	0P733	120141		AND 0,2
20	0P734	120120		MOVL# 0,R,SNC
21	0P735	170224		JMP ,+3
22	0P736	000770		MOVL# 1,1,SZC
23	0P737	100415		MOVOL 0,0,SKP
24	0P740	130414		MOVZL 0,0,SKP
25				MOVZL 1,1,SKP
26	0P741	000047		MOVZL 1,1
27				MUVZR 3,3,SZR
28	0P742	000102		JMP AND3+5
29				SUB# 0,1,SNN
30				JCHECK IF AC0=1 ARE
31				JTHE SAME AND IF THEY
32				JARGE WITH INST.
33	0P743	000402		JSR
34	0P744	000010		LOOP
35	0P745	024777		JCYCLE
36	0P746	020154		
37	0P747	122454		
38	0P750	000104		

1TEST PROCESSOR VIA EXCLUSIVE OR ROUTINES.

1GET THE TEST NUMBER
1CHECK THE PROGRAM FLOW

```

#021 ARITH
#1 *****
#2 #TST#11 TEST PROCESSOR WITH XOR *****
#3 *****
#4 *****
#5 #0751 #1#154 T11: ISZ TSTNM #ENTERING A NEW TEST
#6
#7 X1: SETUP #IENTER #C(AC1) IS SAVED IN C(AC2).
#8 #0752 #0#01#1 JSR #RANDOM #AC0 IS EXCLUSIVE ORED
#9 #0753 #0#01#3 JSR #IRAND #WITH AC1 TWICE. THE SECOND
#10 #0754 131#0#0 MOV 1,2 #EXCLUSIVE OR SHOULD
#11 #0755 #0#0#75 JSR #ICAL #RESTORE AC1 TO ITS
#12 #0756 #0#2352 CALL #ORIGINAL CONTENTS.
#13 #0757 #0#0#75 JSR #ICAL
#14 #0758 #0#2352 XOR SUB# 1,2,SZ#
#15 #0759 132414 ERROR
#16 #0760 #0#0#47 JSR #IE#R
#17 #0761 #0#0#47 LOOP
#18 #0762 #0#0#47 JSR #ICYCLE
#19
#20 #0763 #0#0#42 JMP #+2
#21 #0764 #0#0#11 TN11: 11
#22 #0765 #24777 ET11: LDA 1,TN11
#23 #0766 #2#154 LDA #,TSTNM #GET THE TEST NUMBER
#24 #0767 122454 SUB# 1,#,SZ# #CHECK THE PROGRAM FLOW
#25 #0768 #0#0#4 JSR #ISE#ER

```

```

#022 ARITH
#1 *****
#2 #TST#12 XOR TEST (CONT.) *****
#3 *****
#4 *****
#5 #0772 #1#0154 T12: ISZ TSTNM #ENTERING A NEW TEST
#6
#7 X2: SETUP #IENTER #THE FIRST EXCLUSIVE OR
#8 #0773 #0#01#1 JSR #RANDOM #ROUTINE EXCHANGES THE
#9 #0774 #0#01#3 JSR #IRAND #CONTENTS
#10 #0775 #0#0#75 JSR #ICAL #OF AC0 AND AC1, IT ALSO
#11 #0776 #0#2352 CALL #FORMS THE EXCLUSIVE OR
#12 #0777 #0#0#75 JSR #ICAL
#13 #0778 #0#2357 XOR1 SUB# #,2,SZ# #IN AC2. THE SECOND EXCLUSIVE
#14 #0779 112414 ERROR #OR FORMS THE RESULT
#15 #0780 #0#0#47 JSR #IE#R #OF AC0-AC1 IN AC0.
#16 #0781 #0#0#47 LOOP
#17 #0782 #0#0#47 JSR #ICYCLE

```

10023 ARITH

```

01
02 01004 000402      JMP      .+2
03 01005 000012 TN12: LDA      12
04 01006 024777 ET12: LDA      1,TN12
05 01007 020154      LDA      0,TSTNM
06 01010 122454      SUBM#   1,0,SZR
07 01011 000104      JSR      @ISEQER
    
```

JGET THE TEST NUMBER
 JCHECK THE PROGRAM FLOW

0024 ARITH

```

01
02
03
04
05 01012 010154 T13:  ISZ      TSTNM      JENTERING A NEW TEST
06
07
08 01013 000101      X3:      SETUP      JSAVE C(AC1) NEGATED IN
09
10 01014 000103      JSR      @IENTER   JC(AC2), EXCLUSIVE OR C(AC0) TO
11 01015 130400      JSR      @IRAND    JC(AC1), EXCLUSIVE OR THE
12
13 01016 000075      JSR      @ICAL     JRESULT BACK TO AC0.
14 01017 002352      XUR      @ICAL     JCHECK VIA ADDITION TO
15
16 01020 000075      CALL    @ICAL     JCOMPLIMENT OF ORIGINAL
17 01021 002357      JSR      @ICAL     JNUMBER.
18 01022 113014      XUR1     ERROR
19
20 01023 000047      ADD#    0,2,SZR
21
22 01024 000102      JSR      @IERR
23
24 01025 000402      JSR      @ICYCLE
25 01026 000013 TN13:  JMP      .+2
26 01027 024777 E113: LDA      13
27 01030 020154      LDA      1,TN13
28 01031 122454      SUBM#   0,TSTNM   JGET THE TEST NUMBER
29 01032 000104      JSR      1,0,SZR  JCHECK THE PROGRAM FLOW
    JSR      @ISEQER
    
```

```

0025 ARITH
01 ;*****
02 ;TST#14 XOR TEST (CONT.)
03 ;*****
04
05 01033 010154 T14: ISZ TSTNM JENTERING A NEW TEST
06
07 X4: SETUP JEXCLUSIVE OR C(AC0) TO
08 01034 000101 JSR #IENTER
09 RANDOM JALL ZEROS IN C(AC1).
10 01035 000103 JSR #IRAND
11 01036 100400 SUB 1,1
12 CALL
13 01037 000075 JSR #ICAL
14 01040 002352 XOR
15 01041 100414 SUB# 0,1,SZR
16 ERROR
17 01042 000047 JSR #IERR
18 LOOP
19 01043 000102 JSR #ICYCLE
20
21 01044 000402 JMP .+2
22 01045 000014 TN14: 14
23 01046 024777 ET14: LDA 1,TN14
24 01047 020154 LDA 0,TSTNM JGET THE TEST NUMBER
25 01050 122454 SUB# 1,0,SZR JCHECK THE PROGRAM FLOW
26 01051 000104 JSR #ISEQER

```

```

0026 ARITH
01 ;*****
02 ;TST#15 XOR TEST (CONT.)
03 ;*****
04
05 01052 010154 T15: ISZ TSTNM JENTERING A NEW TEST
06
07 X5: SETUP JEXCLUSIVE OR C(AC1) TO
08 01053 000101 JSR #IFENTER
09 RANDOM JALL ZEROS IN C(AC0).
10 01054 000103 JSR #IRAND
11 01055 100400 SUB 0,0
12 CALL
13 01056 000075 JSR #ICAL
14 01057 002357 XOR1
15 01060 100414 SUB# 0,1,SZR
16 ERROR
17 01061 000047 JSR #IERR
18 LOOP
19 01062 000102 JSR #ICYCLE
20
21 01063 000402 JMP .+2
22 01064 000015 TN15: 15
23 01065 024777 ET15: LDA 1,TN15
24 01066 020154 LDA 0,TSTNM JGET THE TEST NUMBER
25 01067 122454 SUB# 1,0,SZR JCHECK THE PROGRAM FLOW
26 01070 000104 JSR #ISEQER

```

0027 ARITH

```

01
02
03
04
05 01071 010154 T16: ISZ TSTNM ENTERING A NEW TEST
06
07 X6: SETUP EXCLUSIVE OR C(AC1) TO
08 JSR #ENTER
09 RANDOM I(-1) IN C(AC0).
10 01073 000103 JSR #IRAND
11 01074 102000 ADC P,0 THE COMPLIMENT OF
12 CALL THIS RESULT SHOULD
13 01075 000075 JSR #ICAL
14 01076 002357 XOR1 EQUAL C(AC1).
15 01077 110000 COM 0,2
16 01100 140414 SUB# 2,1,SZK
17 ERROR
18 01101 000047 JSR #IEHR
19 LOOP
20 01102 000102 JSR #ICYCLE

```

10028 ARITH

```

01
02 01103 000402 JMP .+2
03 01104 000016 TN16: 16
04 01105 024777 ET16: LDA 1,TN16
05 01106 020104 LDA 0,TSTNM
06 01107 122404 SUB# 1,0,SZR
07 01110 000104 JSR #ISEQER

```

```

JGET THE TEST NUMBER
JCHECK THE PROGRAM FLOW

```

```

0029 ARITH
31 *****
32 ;TST#17 XOR TEST (CONT.)
33 *****
34
35 01111 010154 T17: ISZ TSTNM ENTERING A NEW TEST
36
37 X7: SETUP IC(AC1) IS SET EQUAL TO
38 01112 006101 JSR #IENTER
39 RANDOM IC(AC0), C(AC0) AND C(AC1)
40 01113 006103 JSR #IRAND
41 01114 105000 MOV 0,1 ARE EXCLUSIVE ORED WITH
42 CALL THE RESULT GOING TO AC2.
43 01115 006075 JSR #ICAL
44 01116 002306 XOR2
45 01117 151004 MOV 2,2,SZR
46 ERROR
47 01120 006047 JSR #IERR
48 LOOP
49 01121 006102 JSR #ICYCLE
50
51 01122 006402 JMP +2
52 01123 006017 TN17: 17
53 01124 024777 EI17: LDA 1,TN17
54 01125 020154 LDA 0,TSTNM GET THE TEST NUMBER
55 01126 122454 SUB# 1,0,SZR CHECK THE PROGRAM FLOW
56 01127 006104 JSR #ISEQER

```

```

0030 ARITH
31 *****
32 ;TST#20 XOR TEST (CONT.)
33 *****
34
35 01130 010154 T20: ISZ TSTNM ENTERING A NEW TEST
36
37 X8: SETUP IC(AC0) IS SET TO THE
38 01131 006101 JSR #IENTER
39 RANDOM IC(AC0), C(AC0) AND C(AC1)
40 01132 006103 JSR #IRAND
41 01133 120000 COM 1,0 COMPLEMENT OF C(AC1), THE
42 CALL RESULT OF A EXCLUSIVE OR
43 JSR #ICAL SHOULD BE ALL BITS
44 01135 002306 XOR2
45 01136 150014 COM# 2,2,SZR
46 ERROR
47 01137 006047 JSR #IERR
48 LOOP
49 01140 006102 JSR #ICYCLE
50
51 01141 006402 JMP +2
52 01142 006020 TN20: 20
53 01143 024777 EI20: LDA 1,TN20
54 01144 020154 LDA 0,TSTNM GET THE TEST NUMBER
55 01145 122454 SUB# 1,0,SZR CHECK THE PROGRAM FLOW
56 01146 006104 JSR #ISEQER

```


0031 ARITH

```

01 ;*****
02 ;TST#21 XOR TEST (CONT.)
03 ;*****
04
05 01147 010154 T21: ISZ TSTNM JENTERING A NEW TEST
06
07 X9: SETUP JEXCLUSIVE OR ALL ONES
08 JSR #IENTER
09 ACCS 0,0 JTO ALL ONES, THE
10 ADC 1,1 JRESULT SHOULD BE
11 CALL JALL ZEROS.
12 JSR #ICAL
13 XOR
14 ANDS 1,1,SZR
15 ERROR
16 JSR #IEHR
17 LOOP
18 JSR #ICYCLE
19
20 JMP .+2
21 01161 000021 TN21: 21
22 01162 024777 ET21: LDA 1,TN21
23 01163 020154 LDA 0,TSTNM JGET THE TEST NUMBER
24 01164 122454 SUBW 1,0,SZR JCHECK THE PROGRAM FLOW
25 01165 000104 JSR #ISEQER

```

0032 ARITH

```

01 ;*****
02 ;TST#22 XOR TEST (CONT.)
03 ;*****
04
05 01166 010154 T22: ISZ TSTNM JENTERING A NEW TEST
06
07 X10: SETUP JEXCLUSIVE OR ALL ZEROS
08 JSR #IENTER
09 SUBS 0,0 JTO ALL ZEROS, THE
10 SUB 1,1 JRESULT SHOULD BE
11 CALL JALL ZEROS IN C(AC1).
12 JSR #ICAL
13 XOR
14 ADUS 0,1,SZR
15 ERROR
16 JSR #IEHR
17 LOOP
18 JSR #ICYCLE

```

10033 ARITH

```

01
02 01177 000402      JMP      .+2
03 01200 000022      TN22:   22
04 01201 024777      ET22:   LDA      1,TN22
05 01202 020154      LDA      0,TSTNM
06 01203 122454      SUBRM   1,0,SZR
07 01204 006104      JSR      @ISEGER
    
```

IGET THE TEST NUMBER
ICHECK THE PROGRAM FLOW

0034 ARITH

```

01
02
03
04
05 01205 010154      T23:   ISZ      TSTNM           IENTERING A NEW TEST
06
07
08
09
10
11
12 01210 006075      X11:   SETUP      JFORM EXCLUSIVE OF FUNCTION
13 01211 002306      JSR      @IENTER
14
15 01212 006075      JSR      RANDOM      JIN C(AC2).
16 01213 002352      JSR      @IRAND      JCALL ANOTHER EXCLUSIVE
17
18 01214 006075      CALL     @ICAL      JFOR FUNCTION SEVEN TIMES.
19 01215 002352      JSR      @ICAL      JTHE RESULT SHOULD BE THE
20
21 01216 006075      JSR      @ICAL      JSAME AS THE FIRST XOR
22 01217 002352      CALL     @ICAL
23
24 01220 006075      JSR      @ICAL
25 01221 002352      XOR
26
27 01222 006075      CALL     @ICAL
28 01223 002352      JSR      @ICAL
29
30 01224 006075      JSR      @ICAL
31 01225 002352      XOR
32
33 01226 006075      CALL     @ICAL
34 01227 002352      JSR      @ICAL
35 01230 132414      SUBRM   1,2,SZR
36
37 01231 006047      ERKOR
38
39 01232 000102      JSR      @IERR
40
41 01233 000402      LOOP
42 01234 000023      JSR      @ICYCLE
43 01235 024777      JMP      .+2
44 01236 020154      TN23:   23
45 01237 122454      ET23:   LDA      1,TN23
46 01240 006104      LDA      0,TSTNM
         SUBRM   1,0,SZR
         JSR      @ISEGER
    
```

IGET THE TEST NUMBER
ICHECK THE PROGRAM FLOW

```

0035 AKITH
01
02 *****
03 /LIST#24 TEST PROCESSOR WITH ADD INSTRUCTIONS
04 *****
05 01241 010154 T24: ISZ TSTNM ENTERING A NEW TEST
06
07 A1: SETUP /SAVE C(AC2) IN C(AC1).
08 01242 000101 JSR #IENTER
09 RANDOM /ADD AND ADD C(AC0) TO
10 01243 000103 JSR #IRAND
11 01244 145000 MOV 2,1 /C(AC2). THE VALUE IN
12 01245 112400 SUB 0,2 /AC2 SHOULD NOT BE
13 01246 113000 ADD 0,2 /CHANGED.
14 01247 140414 SUB# 2,1,SZK
15 ERROR
16 01250 000047 JSR #IERR
17 LOOP
18 01251 000102 JSR #ICYCLE
19
20 01252 000402 JMP .+2
21 01253 000024 TN24: 24
22 01254 024777 ET24: LDA 1,TN24
23 01255 020154 LDA 0,TSTNM /GET THE TEST NUMBER
24 01256 122454 SUB# 1,0,SZR /CHECK THE PROGRAM FLOW
25 01257 000104 JSR #ISEGER

```

```

PR36 AKITH
01 *****
02 /LIST#25 ADD TEST (CONT.)
03 *****
04
05 01260 010154 T25: ISZ TSTNM ENTERING A NEW TEST
06
07 A2: SETUP
08 01261 000101 JSR #IENTER
09 RANDOM
10 01262 000103 JSR #IRAND
11 01263 102000 ADC 0,0
12 01264 123000 ADD 1,0
13 01265 111400 INC 0,2
14 01266 140414 SUB# 2,1,SZR
15 ERROR
16 01267 000047 JSR #IERR
17 LOOP
18 01270 000102 JSR #ICYCLE

```

10037 ARITH

```
01
02 01271 000402      JMP      .+2
03 01272 000025 TN25: LDA      25
04 01273 024777 ET25: LDA      1, TN25
05 01274 020154      LDA      0, TSTNM
06 01275 122454      SUBDM   1, 0, SZR
07 01276 000104      JSR      @ISEQER
```

IGET THE TEST NUMBER
ICHECK THE PROGRAM FLOW

10038 ARITH

```
01
02
03
04
05 01277 010154 T26: ISZ      TSTNM      ENTERING A NEW TEST
06
07
08 01300 000101      AJ:      SETUP   #INCREMENT THE VALUE IN
09
10 01301 000103      JSN      @IENTER #AC1 AND ADD THAT VALUE
11 01302 131460      RANDOM
12 01303 102360      JSR      @IRAND
13 01304 143060      INCC 1, 2      #TO (=1). THE RESULT
14 01305 100414      AUCC 0, 0     #SHOULD BE THE ORIGINAL
15
16 01306 000047      AUCC 2, 0     #NUMBER.
17
18 01307 000102      SUBM 0, 1, SZR
19
20 01308 000047      ENRR
21
22 01309 000047      JSR      @IENR
23
24 01310 000102      LOOP
25 01311 000102      JSR      @ICYCLE
26
27
28 01312 000402      JMP      .+2
29
30 01313 000025 TN26: LDA      25
31 01314 024777 ET26: LDA      1, TN26
32 01315 020154      LDA      0, TSTNM      IGET THE TEST NUMBER
33 01316 122454      SUBDM   1, 0, SZR      ICHECK THE PROGRAM FLOW
34 01317 000104      JSR      @ISEQER
```

```

0039 ANITH
01 *****
02 JTST#27 ADD TEST (CONT.)
03 *****
04
05 01316 010154 T27: JSZ TSTNM JENTERING A NEW TEST
06
07 A4: SETUP JSR #IENTER JSAVE THE C(AC1) IN C(AC0).
08 01317 000101 RANDOM JSR #IRAND JA "ADDR" INSTRUCTION SHOULD
09 01320 000103 MOVZ 1,0 JNOT CHANGE THE VALUE OF
10 01321 121020 ADDR 1,1 JTHE AC.
11 01322 127200 ADDR 1,1
12 01323 127200 ADDR 1,1
13 01324 127200 ADDR 1,1
14 01325 127200 ADDR 1,1
15 01326 127200 ADDR 1,1
16 01327 127200 ADDR 1,1
17 01330 127200 ADDR 1,1
18 01331 127200 ADDR 1,1
19 01332 127200 ADDR 1,1
20 01333 127200 ADDR 1,1
21 01334 127200 ADDR 1,1
22 01335 127200 ADDR 1,1
23 01336 127200 ADDR 1,1
24 01337 127200 ADDR 1,1
25 01340 100454 SUBOM 0,1,SZR
26 01341 000047 ENDR
27 JSR #IERR
28 01342 000102 LOOP
29 JSR #ICYCLE
30
31 JMP .+2
32 01343 000402 TN27: LDA 1,TN27
33 01344 000027 ET27: LDA 0,TSTNM JGET THE TEST NUMBER
34 01345 024777 SUBOM 1,0,SZR JCHECK THE PROGRAM FLOW
35 01346 020154 JSR #ISEGER
36 01347 122454
37 01350 000104

```

```

0040 ANITH
01 *****
02 JTST#30 ADD TEST (CONT.)
03 *****
04
05 01351 010154 T30: JSZ TSTNM JENTERING A NEW TEST
06
07 A5: SETUP JSR #IENTER JTHE RANDOM NUMBER IN
08 01352 000101 RANDOM JSR #IRAND JC(AC0) IS INCREMENTED VIA "INC"
09 01353 000103 INC 0,3 JAND DECREMENTED VIA "ADD".
10 01354 115400 ADC 1,1 JTHE FINAL RESULT IN C(AC2)
11 01355 126000 ADD 1,3 JSHOULD BE EQUAL TO THE
12 01356 137000 INC 3,3 JORIGINAL NUMBER IN C(AC0).
13 01357 175400 ADD 1,3
14 01358 137000 INC 3,2
15 01361 171400 ADD 1,2
16 01362 133000 INC 2,2
17 01363 151400 ADD 1,2
18 01364 133000 SUBM 2,0,SZR
19 01365 142414 ENDR
20 01366 000047 JSR #IERR
21 01367 000102 LOOP
22
23 JSR #ICYCLE
24

```

10041 ARITH

```

01
02 01370 000402 JMP     .+2
03 01371 000030 TN30: 30
04 01372 024777 ET30: LDA     1,TN30
05 01373 020154 LDA     0,TSTNM
06 01374 122454 SUBO#  1,0,SZR
07 01375 000104 JSR     @ISEQER
    
```

```

;GET THE TEST NUMBER
;CHECK THE PROGRAM FLOW
    
```

0042 ARITH

```

01 ;*****
02 ;TSI#31 ADD TEST (CNT.)
03 ;*****
04
05 01370 010154 T31: ISZ     TSTNM      ;ENTERING A NEW TEST
06
07 ;A0:
08 01377 000101 JSR     @IENTER ;THE SUM OF AC0-1 IS
09 RANDOM ;CHECKED WITH THE SIMULATED
10 01400 000103 JSR     @IRAND
11 01401 135000 MOV    1,3      ;SUM.
12 01402 117000 ADD    0,3
13 01403 054430 STA    3,ADDTEM ;SUM VIA ADD INSTRUCTION.
14 01404 131000 MOV    1,2
15 CALL
16 01405 000075 JSR     @ICAL
17 01406 002352 XOR
18 01407 143524 ANDZL 2,0,SZR ;SIMULATE THE ADD VIA
19 01410 000774 JMP    .-4      ;EXCLUSIVE OR, C(AC2)=
20 01411 020422 LDA    0,ADDTEM ;RIPPLE CARRY,C(AC1)=RESULT.
21 01412 122414 SUB#   1,0,SZR
22 ERROR
23 01413 000047 JSR     @IERR
24 LOOP
25 01414 000102 JSR     @ICYCLE
26
27 01415 000402 JMP     .+2
28
29 01416 000031 TN31: 31
30 01417 024777 ET31: LDA     1,TN31
31 01421 122454 SUBO#  1,0,SZR ;GET THE TEST NUMBER
32 01422 000104 JSR     @ISEQER ;CHECK THE PROGRAM FLOW
    
```

U043 ARITH

```

01 *****
02 ;TST#32 ADD TEST (CONT.)
03 *****
04
05 01420 010154 T32: ISZ TSTNM ENTERING A NEW TEST
06
07 A7: SETUP ADDITION OF NEGATED
08 01424 000101 JSR #IENTER
09 RANDOM INUMBERS SHOULD PRODUCE
10 01425 000103 JSR #IRAND
11 01426 110440 NEG0 2,2 IZERO AND A CARRY.
12 01427 143204 ADDM 2,0,SZR
13 ERROR
14 01430 000447 JSR #IERR
15 LOOP
16 01431 000102 JSR #ICYCLE
17
18 01432 101001 MOV 0,0,SKP
19 01433 177777 ADDTEM: -1
20
21 01434 000402 JMP .+2
22 01435 000032 TN32: 32
23 01436 024777 ET32: LDA 1,TN32
24 01437 020154 LDA 0,TSTNM IGET THE TEST NUMBER
25 01440 122454 SUBM# 1,0,SZR ICHECK THE PROGRAM FLOW
26 01441 000104 JSR #ISEQER

```

U044 ARITH

```

01 *****
02 ;TST#33 ADD TEST (CONT.)
03 *****
04
05 01442 010154 T33: ISZ TSTNM ENTERING A NEW TEST
06
07 A8: SETUP IADDU TEST.
08 01443 000101 JSR #IENTER
09 RANDOM
10 01444 000103 JSR #IRAND
11 01445 152520 SUBZL 2,2 I#(+1)
12 01446 151300 MOV5 2,2 I#(400)
13 01447 150400 NEG 2,2 I#(177400)
14 01450 143700 ANDS 2,0 ISAVE HIGH ORDER 8 BITS,
15 01451 105320 MOVZ5 0,1 ISAME 8 BITS TO C(AC)IL.
16 01452 103100 ADDL 0,0 IMOVE C(AC0) LEFT VIA
17 01453 103100 ADDL 0,0 IADD SHIFT.
18 01454 103100 ADDL 0,0
19 01455 103100 ADDL 0,0
20 01456 100414 SUB# 0,1,SZR
21 ERROR
22 01457 000447 JSR #IERR
23 LOOP
24 01460 000102 JSR #ICYCLE
25

```

10045 ARITH

```

01
02
03 01461 101001      MOV 0,0,SKP
04 01462 177400 M400: -400
05 01463 000400      JMP      .+2
06 01464 000033 TN33: 33
07 01465 024777 ET33: LDA      1,TN33
08 01466 020154      LDA      0,TSTNM
09 01467 122454      SUBU#   1,0,SZR
10 01470 000104      JSR      #ISEQER
    
```

;GET THE TEST NUMBER
 ;CHECK THE PROGRAM FLOW

00046 ARITH

```

01
02
03
04
05 01471 010154 T34: ISZ      TSTNM      ENTERING A NEW TEST
06
07
08
09
10 01472 000101 AK1:  SETUP      ;THE ORIGINAL CONTENTS OF
11 01473 000103      JSR      #IENTER
12 01474 030766      RANDOM   ;AC1, BITS 0-7 ARE SQUARED
13 01475 133700      JSR      #IRAND
14 01476 145000      LDA 2,M400 ;VIA MULTIPLY. THE SQUARE
15 01477 000075      ANDS 1,2  ;ROOT OF THE PRODUCT SHOULD
16 01500 002323      MOV 2,1  ;EQUAL THE ORIGINAL.
17 01501 121000      CALL
18 01502 000075      JSR      #ICAL
19 01503 002401      ;SEE SYSTEM REFERENCE
20 01504 112414      ;MANUAL FOR FURTHER INFORMATION
21 01505 000047      ;ON MULTIPLY/SQ ROOT
22 01506 000102      JSR      #ICAL
23 01507 000402      ;PROGRAMS.
24 01508 000034 TN34: SUB#   0,2,SZR
25 01509 000034      ERROR
26 01510 000034      JSR      #IEER
27 01511 024777 ET34: LOOP
28 01512 020154      JSR      #ICYCLE
29 01513 122454      JMP      .+2
30 01514 000104      ;34
31
32
    
```

;GET THE TEST NUMBER
 ;CHECK THE PROGRAM FLOW


```

0047 ARITH
01
02 *****
03 ITST#35 SQUARE TEST (CONT.)
04 *****
05
06
07
08 01515 000154 T35: ISZ TSTNM ENTERING A NEW TEST
09
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

```

06	01515	000154	T35:	ISZ	TSTNM	ENTERING A NEW TEST
07			AK2:	SETUP		
08	01516	000101		JSR	#IENTER	TAKE THE SQUARE ROOT
09				RANUOM		OF A NUMBER, THE SQUARE
10	01517	000103		JSR	#IRAND	
11				CALL		ROOT OF THE RESULT SQUARED
12	01520	000075		JSR	#ICAL	
13	01521	002401		SGRT		SHOULD BE THE SAME AS THE
14	01522	105000		MOVL 0,1		FORIGIONAL ROOT.
15	01523	131000		MOVL 1,2		
16				CALL		
17	01524	000075		JSR	#ICAL	
18	01525	002323		.MPYU		
19	01526	121000		MOV 1,0		
20				CALL		SQUARED NOW TAKE ROOT.
21	01527	000075		JSR	#ICAL	
22	01530	002410		SW		
23	01531	112714		SUBSR 0,2,SZR		
24				ERROR		
25	01532	000047		JSR	#IERR	
26				LOOP		
27	01533	000102		JSR	#ICYCLE	
28						
29	01534	000402		JMP	.*2	
30	01535	000035	TN35:	JS		
31	01536	024777	ET35:	LDA	1,TN35	
32	01537	022154		LDA	0,TSTNM	GET THE TEST NUMBER
33	01540	122454		SUBSR 1,0,SZR		CHECK THE PROGRAM FLUM
34	01541	000104		JSR	#ISEQR	

```

0048 ARITH
01
02 *****
03 ITST#36 SQUARE TEST (CONT.)
04 *****
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

```

08	01542	010154	T36:	ISZ	TSTNM	ENTERING A NEW TEST
09			AK3:	SETUP		
10	01543	000101		JSR	#IENTER	FIND SQUARE ROOT VIA
11	01544	000103		RANUOM		DIFFIRENT SUBROUTINES.
12	01545	141000		JSR	#IRAND	
13				MOV 2,0		
14	01546	000075		CALL		
15	01547	002401		JSR	#ICAL	
16	01550	105000		SGRT		SAVE FIRST RESULT IN AC1
17	01551	141000		MOVL 0,1		
18				MOV 2,0		
19	01552	000075		CALL		
20	01553	002410		JSR	#ICAL	RESULT IN AC0.
21	01554	106714		SW		
22				SUBSR 0,1,SZR		
23	01555	000047		ERROR		
24	01556	000102		JSR	#IERR	
				LOOP		
				JSR	#ICYCLE	

10049 ARITH

```

31
32 01557 000402      JMP      .+2
33 01560 000036 TN36: 36
34 01561 024777 ET36: LDA      1, TN36
35 01562 020154      LDA      0, TSTNM
36 01563 122454      SUBUM#  1, 0, SZR
37 01564 000104      JSR      @ISEQER
    
```

```

;GET THE TEST NUMBER
;CHECK THE PROGRAM FLOW
    
```

0050 ARITH

```

31
32
33
34
35 01565 010154 T37:  ISZ      TSTNM      ;ENTERING A NEW TEST
36
37
38
39 01566 000101      AR4:  SETUP   ;THE CONTENTS OF AC2 IS
40
41
42
43
44
45 01567 000103      JSR      @IENTER ;DIVIDED INTO AC0-1,
46 01570 142432      RANDOM
47 01571 000776      JSR      @IRAND  ;THIS RESULT MULTIPLIED
48
49
50 01572 000075      SUBR#  2, 0, SZC ;BY AC2 SHOULD PRODUCE
51 01573 002347      JMP      .-2     ;THE ORIGINAL NUMBERS.
52
53
54
55 01574 000075      CALL
56 01575 002324      JSR      @ICAL
57 01576 034147      .DIVU
58 01577 156414      CALL
59
60
61
62
63
64
65 01578 000047      JSR      @ICAL
66 01579 034146      .MPYA
67 01580 136714      LDA  3, 0AC2
68
69
70
71
72
73
74
75 01581 034146      SUBR#  2, 3, SZR ;AC2 CHANGED?
76 01582 136714      ERROR
77
78
79
80
81
82
83
84
85 01583 000047      JSR      @IERR
86 01584 034145      LDA  3, 0AC1
87 01585 116714      SUBR#  1, 3, SZR ;AC1 CHANGED.
88
89
90
91
92
93
94
95 01586 000047      ERROR
96 01587 000102      JSR      @IERR
97
98
99
100
101
102
103
104
105 01588 000402      JSR      @ICYCLE
106
107
108
109
110
111
112
113
114
115 01589 000402      JMP      .+2
116 01590 000037 TN37: 37
117 01591 024777 ET37: LDA      1, TN37
118 01592 020154      LDA      0, TSTNM ;GET THE TEST NUMBER
119 01593 122454      SUBUM#  1, 0, SZR ;CHECK THE PROGRAM FLOW
120 01594 000104      JSR      @ISEQER
    
```

0051 ARITH

```

01 *****
02 ITST#40 SQUARE TEST (CONT.)
03 *****
04
05 01010 010154 T40: ISZ TSTNM JENTERING A NEW TEST
06
07 AK5: SETUP JUSE INC TO FORM
08 01017 000101 JSR #IENTER
09 01020 152440 SUBO 2,2 JTHE NUMBER 177400
10 01021 151504 INCL 2,2,SZR JIN AC1 AND 400 IN AC2.
11 01022 151504 INCL 2,2,SZR
12 01023 151504 INCL 2,2,SZR
13 01024 151504 INCL 2,2,SZR
14 01025 151507 INCL 2,2,SBN JTHE "SBN/SZR" SHOULD
15 01026 151507 INCL 2,2,SBN JNOT CAUSE A SKIP.
16 01027 151507 INCL 2,2,SBN
17 01030 145707 INCS 2,1,SBN
18 01031 151407 INC 2,2,SBN
19 01032 151400 INC 2,2
20 01033 133014 ADD# 1,2,SZH
21 ERROR
22 01034 000047 JSR #IFHR
23 LOOP
24 01035 000102 JSR #ICYCLE
25
26 01030 000402 JMP ,+2
27 01037 000040 TN40:
28 01040 024777 ET40: LDA 1,TN40
29 01041 020154 LDA 0,TSTNM JGET THE TEST NUMBER
30 01042 122454 SUBO# 1,0,SZR JCHECK THE PROGRAM FLOW
31 01043 000104 JSR #ISEGER

```

0052 ARITH

```

01 *****
02 ITST#41 SQUARE TEST (CONT.)
03 *****
04
05 01044 010154 T41: ISZ TSTNM JENTERING A NEW TEST
06
07 AK6: SETUP JUSE THE INCR INSTRUCTION
08 01045 000101 JSR #IENTER
09 01046 126427 SUBZ 1,1 JTO FORM THE NUMBER
10 01047 125606 INCR 1,1,SEZ J177400. THIS NUMBER
11 01050 125606 INCR 1,1,SEZ JCOMPLIMENTED AND SWAPPED
12 01051 125606 INCR 1,1,SEZ JSHOULD BE THE SAME
13 01052 125606 INCR 1,1,SEZ JNUMBER.
14 01053 125606 INCR 1,1,SEZ JTHE "SEZ" SHOULD NEVER
15 01054 125606 INCR 1,1,SEZ JCAUSE A SKIP.
16 01055 125606 INCR 1,1,SEZ
17 01056 125606 INCR 1,1,SEZ
18 01057 130304 COMS# 1,2,SZH
19 01060 132714 SUBS# 1,2,SZR
20 ERROR
21 01061 000047 JSR #IERR
22 LOOP
23 01062 000102 JSR #ICYCLE

```

10053 ARITH

```

01
02 01663 000402      JMP      .+2
03 01664 000041 TN41: 41
04 01665 024777 ET41: LDA      1,TN41
05 01666 020154      LDA      0,TSTNM
06 01667 122454      SUBD#   1,0,SZR
07 01670 000104      JSR      @ISEDER

```

```

08 GET THE TEST NUMBER
09 CHECK THE PROGRAM FLOW

```

0054 ARITH

```

01
02
03
04
05 01671 010154 T42:  ISZ      TSTNM      ENTERING A NEW TEST
06
07
08 01672 005101      AK7:   SETUP      1100000 NEGATED IS
09 01673 102025      JSR      @IENTER
10 01674 000047      SUBZR 0,0,SNR      1STILL 100000.
11 01675 104406      ENRR
12 01676 000047      JSR      @IEKR
13 01677 124406      NEG 0,1,SEZ      1C(CARRY)=0
14 01678 000047      ENRR
15 01679 124402      JSR      @IEHR
16 01680 000047      NEG 1,1,SZC
17 01681 000047      ENRR
18 01682 124403      JSR      @IFRR
19 01683 124403      NEG 1,1,SNC      1C(CARRY)=1
20 01684 000047      ENRR
21 01685 124407      JSR      @IEHR
22 01686 124407      NEG 1,1,SBN
23 01687 000047      ENRR
24 01688 124405      JSR      @IEHR
25 01689 124405      NEG 1,1,SNR      1C(1)=100000
26 01690 000047      ENRR
27 01691 124405      JSR      @IERR
28 01692 124405      NEG 1,1,SNK
29 01693 000047      ENRR
30 01694 124407      JSR      @IERR
31 01695 124407      NEG 1,1,SNB
32 01696 000047      ENRR
33 01697 124407      JSR      @IERR
34 01698 124407      NEG 1,1,SNB
35 01699 000047      ENRR
36 01700 106414      JSR      @IERR
37 01701 106414      SUB# 0,1,SZK
38 01702 000047      ENRR
39 01703 000047      JSR      @IERR
40 01704 000102      LUOP
41 01705 000102      JSR      @ICYCLE
42 01720 000402      JMP      .+2
43 01721 000042 TN42: 42
44 01722 024777 ET42: LDA      1,TN42
45 01723 020154      LDA      0,TSTNM
46 01724 122454      SUBD#   1,0,SZR
47 01725 000104      JSR      @ISEDER

```

```

08 GET THE TEST NUMBER
09 CHECK THE PROGRAM FLOW

```

0055 ARITH

```

01 *****
02 TST#43 SQUARE TEST (CONT.)
03 *****
04
05 01720 010104 T43: ISZ TSTNM ENTERING A NEW TEST
06
07 AMB: SETUP INEGATION OF ZERO
08 01727 006101 JSR #ENTER
09 01730 102440 SUBO #,0 SHOULD PRODUCE ZERO
10 01731 100702 NEGCS 0,0,SZC HAND A CARRY.
11 ERROR
12 01732 006047 JSR #IEHR
13 01733 100762 NEGLS 0,0,SZC
14 ERROR
15 01734 006047 JSR #IEHR
16 01735 100706 NEGS 0,0,SEZ
17 ERROR
18 01736 006047 JSR #IEHR
19 01737 100706 NEGS 0,0,SEZ
20 ERROR
21 01740 006047 JSR #IEHR
22 01741 100544 NEGOL 0,0,SZM
23 ERROR
24 01742 006047 JSR #IEHR
25 01743 100403 NEG 0,0,SNC
26 ERROR
27 01744 006047 JSR #IEHR
28 01745 100644 NEGOR 0,0,SZR
29 ERROR
30 01746 006047 JSR #IEHR
31 01747 104704 NEGS 0,1,SZH
32 ERROR
33 01750 006047 JSR #IEHR
34 01751 130704 NEGS 1,2,SZR
35 ERROR
36 01752 006047 JSR #IEHR
37 LOOP
38 01753 006102 JSR #ICYCLE

```

10056 ARITH

```

01
02 01754 000402 JMP .+2
03 01755 000043 TN43: 43
04 01756 024777 ET43: LDA 1,TN43
05 01757 020154 LDA 0,TSTNM
06 01760 122454 SUBUM 1,0,SZR
07 01761 000104 JSR #ISEQER

```

```

IGET THE TEST NUMBER
ICHECK THE PROGRAM FLOW

```

```

0057 ARITH
01 *****
02 )TST#44 SQUARE TEST (CONT.)
03 *****
04
05 01702 010154 T44: ISZ TSTNM )ENTERING A NEW TEST
06
07 AK9: SETUP )C(AC1)*0+C(AC0) SHOULD
08 01763 006101 JSR #IENTER
09 RANDOM )PLACE AC0 IN AC1. SEE
10 01764 006103 JSR #IRAND
11 01765 152400 SUB# 2,2 )SYSTEM REFERANCE MANUAL
12 CALL )FOR FURTHER INFORMATION.
13 01766 006075 JSR #ICAL
14 01767 002324 .MPYA
15 01770 034145 LDA 3,0AC0
16 01771 166414 SUB# 3,1,SZM
17 ERROR
18 01772 006047 JSR #IEHR
19 LOOP
20 01773 006102 JSR #ICYCLE
21
22 01774 000402 JMP .+2
23 01775 000044 TN44: 44
24 01776 024777 ET44: LDA 1,TN44
25 01777 020154 LDA 0,TSTNM )GET THE TEST NUMBER
26 02000 122454 SUBM# 1,0,SZR )CHECK THE PROGRAM FLOW
27 02001 006104 JSR #ISEQER

```

```

0058 ARITH
01 *****
02 )TST#45 SQUARE TEST (CONT.)
03 *****
04
05 02002 010154 T45: ISZ TSTNM )ENTERING A NEW TEST
06
07 AN10: SETUP )TEST "COM/INC" SWAPPED.
08 02003 006101 JSR #IENTER
09 02004 102300 ADCS 0,0
10 02005 105705 INCS 0,1,SNR
11 02006 130304 CUMS 1,2,SZM
12 02007 145705 INCS 2,1,SNR
13 02010 124346 COMOS 1,1,SEZ
14 02011 131707 INCS 1,2,SBM
15 02012 150304 COMS 2,2,SZM
16 02013 151704 INCS 2,2,SZM
17 ERROR
18 02014 006247 JSR #IEHR
19 LOOP
20 02015 006102 JSR #ICYCLE
21
22 02016 000402 JMP .+2
23 02017 000045 TN45: 45
24 02020 024777 ET45: LDA 1,TN45
25 02021 020154 LDA 0,TSTNM )GET THE TEST NUMBER
26 02022 122454 SUBM# 1,0,SZR )CHECK THE PROGRAM FLOW
27 02023 006104 JSR #ISEQER

```

```

0059 AKITH
01 *****
02 ;TST#46 SQUARE TEST (CONT.)
03 *****
04
05 02024 010154 T46: ISZ TSTNM ;ENTERING A NEW TEST
06
07 AK11: SETUP ;COMPLIMENT AND INCREMENT
08 02025 006101 JSR @IENTER
09 RANDOM ;SHOULD BE THE SAME AS
10 02026 006103 JSR @IRAND
11 02027 110400 NEG 0,2 ;NEGATE
12 02030 104000 COM 0,1
13 02031 125400 INC 1,1
14 02032 132414 SUB# 1,2,SZR
15 ERROR
16 02033 006047 JSR @IEKR
17 LOOP
18 02034 006102 JSR @ICYCLE
19
20 02035 000402 JMP .+2
21 02036 000046 TN46: 46
22 02037 024777 ET46: LVA 1,TN46
23 02040 020154 LVA 0,TSTNM ;GET THE TEST NUMBER
24 02041 122454 SUB# 1,0,SZR ;CHECK THE PROGRAM FLOW
25 02042 006104 JSR @ISEWER

```

```

0060 AKITH
01 *****
02 ;TST#47 SQUARE TEST (CONT.)
03 *****
04
05 02043 010154 T47: ISZ TSTNM ;ENTERING A NEW TEST
06
07 AK12: SETUP ;TEST COM LEFT AND RIGHT.
08 02044 006101 JSR @IENTER
09 RANDOM
10 02045 006103 JSR @IRAND
11 02046 110100 COML 0,2
12 02047 144200 COMCR 2,1
13 02050 106714 SUBSW 0,1,SZR
14 ERROR
15 02051 006047 JSR @IEKR
16 LOOP
17 02052 006102 JSR @ICYCLE

```

10061 ARITH

```

31
32 02053 000402 JMP .+2
33 02054 000047 TN47: 47
34 02055 024777 ET47: LDA 1,TN47
35 02056 020154 LDA 0,TSTNM
36 02057 122454 SUBO# 1,0,SZR
37 02060 006104 JSR #ISEQER

```

```

)GET THE TEST NUMBER
)CHECK THE PROGRAM FLOW

```

10062 ARITH

```

41 *****
42 )TST#5 SQUARE TEST (CONT.)
43 *****
44
45 02061 010104 T50: ISZ TSTNM )ENTERING A NEW TEST
46
47 AR13: SETUP )FORM THE PARITY OF
48 02062 006101 JSR #IENTER
49 02063 105020 MOVZ 0,1 )C(AC0) IN DIFFIRENT
50 02064 176000 ADC 3,3 )ROUTINES. CHECK THAT
51 02065 117000 ADD 0,3 )THE RESULTS ARE EQUAL.
52 02066 163704 ANDS 3,0,SZR
53 02067 000775 JMP .-3
54 02070 102060 SUBCR 0,0 )SAVE PARITY IN BIT 0.
55 02071 170620 SUBZR 3,3
56 02072 125102 MUVL 1,1,SZC
57 02073 101400 INC 0,0 )BIT 15 WILL CONTAIN
58 02074 175224 MUVZR 3,3,SZR )THE PARITY.
59 02075 000775 JMP .-3
60 02076 115200 MUVR 0,3 )CHECK TO SEE IF BITS
61 02077 103012 ADD# 0,0,SZC )ARE LIKE.
62 ERROR
63 JSK #IEKR
64 LOOP
65 02101 006102 JSR #ICYCLE
66
67 02102 000402 JMP .+2
68 02103 000050 TN50: 50
69 02104 024777 ET50: LDA 1,TN50
70 02105 020154 LDA 0,TSTNM
71 02106 122454 SUBO# 1,0,SZR
72 02107 006104 JSR #ISEQER

```

```

)GET THE TEST NUMBER
)CHECK THE PROGRAM FLOW

```



```

0063 ARITH
01 *****
02 ;TST#51 SQUARE TEST (CONT.)
03 *****
04
05 02110 010154 T51: ISZ TSTNM JENTERING A NEW TEST
06
07 AR15: SETUP JMSC TEST OF SUB LEFT
08 02111 006101 JSR #IENTER
09 RANDOM JAND RIGHT.
10 JSR #IRAND
11 02113 131000 MOV 1,2
12 02114 106500 SUHL 0,1
13 02115 107600 ANDK 1,1
14 02116 107000 ADD 0,1
15 02117 106500 SUBL 0,1
16 02120 127600 ANDK 1,1
17 02121 107000 ADD 0,1
18 02122 132414 SUB# 1,2,SZM
19 ERROR
20 02123 006047 JSR #IEKR
21 LOOP
22 02124 006102 JSR #ICYCLE
23
24 02125 006402 JMP .+2
25 02126 000051 TN51: 51
26 02127 024777 ET51: LDA 1,TN51
27 02130 020154 LDA 0,TSTNM
28 02131 122454 SUB# 1,0,SZR
29 02132 006104 JSR #ISEDER
;GET THE TEST NUMBER
;CHECK THE PROGRAM FLOW

```

```

0064 ARITH
01 *****
02 ;TST#52 SQUARE TEST (CONT.)
03 *****
04
05 02133 010154 T52: ISZ TSTNM JENTERING A NEW TEST
06
07 AR16: SETUP JMSC TEST OF AND RIGHT.
08 02134 006101 JSR #IENTER
09 RANDOM
10 JSR #IRAND
11 02136 131040 MOVU 1,2
12 02137 110600 ANDK 0,2
13 02140 107400 AND 0,1
14 02141 151100 MOVL 2,2
15 02142 146414 SUB# 2,1,SZR
16 ERROR
17 02143 006047 JSR #IEKR
18 LOOP
19 02144 006102 JSR #ICYCLE

```

1005 ARITH

```

01
02 02145 000402      JMP      .+2
03 02146 000052 TN52: LDA      52
04 02147 024777 ET52: LDA      1, TN52
05 02150 020154      LDA      0, TSTNM
06 02151 122454      SUBO#   1, 0, SZR
07 02152 000104      JSR      @ISEQER
    
```

```

JGET THE TEST NUMBER
JCHECK THE PRUGHAM FLOW
    
```

0006 ARITH

```

01 *****
02 JTST#53 SQUARE TEST (CONT.)
03 *****
04
05 02153 010154 T53:  ISZ      TSTNM      JENTERING A NEW TEST
06
07 AK17:  SETUP
08 02154 000101      JSR      @IENTER      JMISC ADD SWAPPED TEST.
09      RANDOM
10 02155 000103      JSR      @IRAND
11 02156 131000      MOV  1,2
12 02157 113000      ADD  0,2
13 02160 107300      ALDS 0,1
14 02161 125300      MOVS 1,1
15 02162 132414      SUB#  1,2, SZ#
16      ENDR
17 02163 000047      JSR      @IEHR
18      LOOP
19 02164 000102      JSR      @ICYCLE
20
21 02165 000402      JMP      .+2
22 02166 000053 TN53:  53
23 02167 024777 ET53:  LDA      1, TN53
24 02170 020154      LDA      0, TSTNM
25 02171 122454      SUBO#   1, 0, SZR
26 02172 000104      JSR      @ISEQER
    
```

```

JGET THE TEST NUMBER
JCHECK THE PROGRAM FLOW
    
```

```

0067 ARITH
01 *****
02 IIST#54 SQUARE TEST (CONT.)
03 *****
04
05 02173 010154 T54: ISZ TSTNM JENTERING A NEW TEST
06
07 AR18: SETUP JCHECK ADC LEFT,
08 JSR #IENTER
09 RANDOM
10 JSR #IRAND
11 02176 131000 MOV 1,2
12 02177 112120 ADDCL 0,2
13 02200 100000 CUM 0,0
14 02201 107120 ADDCL 0,1
15 02202 132414 SUB# 1,2,SZK
16 ERROR
17 02203 006047 JSR #IERR
18 LOOP
19 02204 006102 JSR #ICYCLE
20
21 02205 000402 JMP .+2
22 02206 000054 TN54: 54
23 02207 024777 ET54: LDA 1,TN54
24 02210 020154 LDA 0,TSTNM JGET THE TEST NUMBER
25 02211 122454 SUB# 1,0,SZR JCHECK THE PROGRAM FLOW
26 02212 000104 JSR #ISEQER

```

```

0068 ARITH
01 *****
02 IIST#55 SQUARE TEST (CONT.)
03 *****
04
05 02213 010154 T55: ISZ TSTNM JENTERING A NEW TEST
06
07 AR19: SETUP JTEST ADC RIGHT,
08 JSR #IENTER
09 RANDOM
10 JSR #IRAND
11 02216 131000 MOV 1,2
12 02217 112220 ADDZR 0,2
13 02220 100000 CUM 0,0
14 02221 107220 ADDZR 0,1
15 02222 132414 SUB# 1,2,SZK
16 ENKOR
17 02223 006047 JSR #IERR
18 LOOP
19 02224 006102 JSR #ICYCLE
20
21 02225 000402 JMP .+2
22 02226 000055 TN55: 55
23 02227 024777 ET55: LDA 1,TN55
24 02230 020154 LDA 0,TSTNM JGET THE TEST NUMBER
25 02231 122454 SUB# 1,0,SZR JCHECK THE PROGRAM FLOW
26 02232 000104 JSR #ISEQER

```

0009 ARITH

01 ;*****
02 ;TST#56 SQUARE TEST (CONT.)
03 ;*****
04

05 02233 010154 T56: ISZ TSTNM ENTERING A NEW TEST
06
07 AM20: SETUP TEST SUB RIGHT,
08 02234 000101 JSR #ENTER
09 RANDUM
10 02235 000103 JSR #IRAND
11 02236 131000 MOV 1,2
12 02237 100020 SUBZR P,1
13 02240 100420 NEGZ 0,0
14 02241 113200 ADDR 0,2
15 02242 132414 SUB# 1,2,SZH
16 ERROR
17 02243 000047 JSR #IERR
18 LOOP
19 02244 000102 JSR #ICYCLE

10070 ARITH

01
02
03
04
05 02245 000402 JMP .+2
06 02246 000056 TN56: 56
07 02247 024777 ET56: LDA 1,TN56
08 02250 020154 LDA 0,TSTNM
09 02251 122454 SUB# 1,0,SZR
10 02252 000104 JSR #ISEQR

IGET THE TEST NUMBER
ICHECK THE PROGRAM FLOW

```

0071 ARITH
01
02 *****
03 *****
04 *****
05 02253 010154 T57: ISZ T57NM JENTERING A NEW TEST
06
07 ARJSR: SETUP JTEST THAT INDEX WITH
08 02254 006101 JSR #IENTER
09 02255 004401 JSR ,+1 JSIGN BIT SET DOES
10 02256 171122 MOVZL 3,2,SZC JNOT LOAD INTO PC ON JSR,
11 ENRRUR
12 02257 006047 JSR #IERR
13 02260 151240 MOVUR 2,2
14 02261 005004 JSR 4,2 JGO TO NEXT LOCATION
15 02262 004401 JSR ,+1
16 02263 165000 MOV 3,1 JJSR SHOULD NEVER
17 02264 125112 MOVLM 1,1,SZC JSTORE THE SIGN BIT,
18 ENRRUR
19 02265 006047 JSR #IERR
20 LOOP
21 02266 006102 JSR #ICYCLE
22
23 02267 010104 ARENDI: ISZ PASS JEND OF TEST
24 02270 020104 LDA 0,PASS
25 02271 024142 LDA 1,K7
26 02272 034045 LDA 3,IEGGS
27 02273 107414 AND# 0,1,SZM
28 02274 000415 JMP FNDE1
29 CALL
30 02275 006075 JSR #ICAL
31 02276 003065 MESS
32 02277 004170 PHES
33
34 02300 034045 TREM: LDA 3,IEGGS
35 02301 025400 LDA 1,0,3
36 02302 125004 MOV 1,1,SZR
37 02303 015403 OSZ 3,3
38 02304 000405 JMP ENDE1
39 02305 021403 LDA 0,3,3
40 02306 031404 RTRN: LDA 2,4,3
41 02307 041376 STA 0,-2,2
42 02310 001000 JMP 0,2
43 02311 126400 ENDE1: SUB 1,1 JCLEAR AC1
44 02312 044153 STA 1,MANCT JCLEAR RANDOM LOOP COUNTER
45 02313 011402 ISZ 2,3
46 02314 002060 JMP #IMOVE
47 CALL
48 02315 006075 JSR #ICAL
49 02310 002461 MOVE-1
50 SEDERI: CALL JPRINT MESSAGE THAT TESTS ARE RUNNING OU
51 02317 006075 JSR #ICAL
52 02320 003065 MESS
53 02321 004143 OUSEG
54 02322 002100 JMP #ISLOW JRESTART PROGRAM

```

```

10072 ARITH
01
02 02323 102400 ,MPYU: SUHC 0,0 JC(AC1)+C(AC2)
03 02324 054411 ,MPYA: STA 3,,CB03 JRESULT IN AC0,AC1.
04 02325 034411 LDA 3,,CB20 JSEE SYSTEM REFERENCE
05 02326 125203 ,CB99: MOVK 1,1,SNC JMANUAL FOR FURTHER
06 02327 101201 MOVUR 0,0,SKP JINFORMATION.
07 02330 143200 ADDZM 2,0
08 02331 175404 INC 3,3,SZR
09 02332 000774 JMP ,CB99
10 02333 125200 MOVUR 1,1
11 02334 002401 JMP #,CB93
12 02335 000700 ,CB03: 0
13 02336 177760 ,CB20: -20
14
15 02337 102400 ,DIVI: SUB 0,0 JC(AC0),C(AC1)/C(AC2).
16 02340 054775 ,DIVU: STA 3,,CB03 JAC0=REMAINDER
17 02341 034775 LDA 3,,CB20 JAC1=QUOTIENT
18 02342 125120 MOVZL 1,1 JSEE SYSTEM REFERENCE
19 02343 101100 ,CC98: MOVL 0,0 JMANUAL.
20 02344 142412 SUB# 2,0,SZC
21 02345 142400 SUB 2,0
22 02346 125100 MOVL 1,1
23 02347 175404 INC 3,3,SZR
24 02350 000775 JMP ,CC98
25 02351 002764 JMP #,CB23
26
27 02352 135000 XORI: MOV 1,3 JEXCLUSIVE OR C(AC0),C(AC1).
28 02353 117520 ANDZL 0,3 JRESULT IS IN C(AC1).
29 02354 100000 ADD 0,1
30 02355 166400 SUB 3,1
31 RETURN
32 02356 002117 JMP #CALRET
33
34
35 02357 135000 XORI: MOV 1,3 JEXCLUSIVE OR C(AC0),C(AC1).
36 02360 117400 AND 0,3 JRESULT IS IN C(AC0).
37 02361 174000 CUM 3,3
38 02362 163400 AND 3,0
39 02363 123000 ADD 1,0
40 02364 163400 AND 3,0
41 RETURN
42 02365 002117 JMP #CALRET
43
44
45 02366 034114 XOR2: LDA 3,M20 JEXCLUSIVE OR C(AC0),C(AC1).
46 02367 054434 STA 3,XURTEM JRESULT IN C(AC2).
47 02370 115000 MOV 0,3 JTHE CONTENTS OF AC0 AND
48 02371 137200 ADDM 1,3 JAC1 ARE EXCHANGED.
49 02372 151200 MOVK 2,2
50 02373 101220 MOVZR 0,0
51 02374 125200 MOVK 1,1
52 02375 103200 ADDM 0,0
53 02376 010425 ISZ XURTEM
54 02377 000771 JMP XUR2+2
55 RETURN
56 02400 002117 JMP #CALRET
57

```

10073 ARITH

```

01
02 02401 126520 SQRT: SUBZL 1,1      IFIND SQ ROOT OF C(AC0).
03 02402 135120      MOVZL 1,3      ISEE THE SYSTEM REFERENCE
04 02403 122422      SUBZ 1,0,SZC   I MANUAL.
05 02404 167001      ADD 3,1,SKP    I RESULT IN AC0.
06 02405 121221      MOVZR 1,0,SKP I AC1 DESTROYED.
07 02406 000775      JMP ,=3
08
09 02407 002117      RETURN
10
11
12 02410 176400 SW:   SUB 3,3      IFIND SQ ROOT OF C(AC0).
13 02411 054411      STA 3,SWTEM   ISAME RESULT AS PREVIOUS
14 02412 162023      ADCL 3,0,SNC  I TEST BUT CODE IS
15 02413 000405      JMP SW1       I DIFFERENT.
16 02414 010406      ISZ SWTEM
17 02415 010406      ISZ SWTEM
18 02416 034404      LDA 3,SWTEM   I RESULT WILL BE IN AC0.
19 02417 000773      JMP ,=5
20 02420 161200 SW1: MOVK 3,0
21
22 02421 002117      RETURN
23
24
25 02422 000000 SWTEM: 0
26 02423 000000 SWRTM: 0
27

```

```

28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

```

```

32 02424 053610 INTM: SKPUN TTI          I WAS TELETYPE INTERRUPTING
33 02425 000415      JMP INTM2     I ENABLE INTERRUPTS AND CONTINUE
34 02426 054072      STA 3,INTS3   I SAVE AC3
35 02427 000603      JSR @INP?     I GO HANDLE TELETYPE
36 02430 040073      STA 0,INTS0   I SAVE AC0
37 02431 020074      LDA 0,CO      I GET A CONTROL 0
38 02432 116415      SUB# 0,3,SNR  I SEE IF CONTROL 0
39 02433 000405      JMP INTM1     I GO TO UOI
40 02434 020073      LDA 0,INTS0   I RESTORE AC0
41 02435 034072      LDA 3,INTS3   I RESTORE AC3
42 02436 050177      INTEN        I ENABLE INTERRUPTS
43 02437 002000      JMP @0        I RETURN TO PROGRAM
44 02440 034072 INTM1: LDA 3,INTS3     I RESTORE AC3
45 02441 002071      JMP @IODT?    I GO TO UOI ROUTINE
46 02442 000904 INTM2: JSR @IMES?S I REPORT ILLEGAL INTERRUPT
47 02443 024123      ILIT
48 02444 050477      INTA 1       I GET DEVICES #
49 02445 000070      JSR @IZOC?T  I PRINT THE DEVICES #
50 02446 022111      JMP @EGGS+4  I RETURN TO MONITOR
51

```

10074 ARITH

```

41
42
43
44
45
46
47
48
49
50
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
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
160
161
162
163
164
165
166
167
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
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
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
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
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

W275 ARITH

```

01 W2527 W00146      OAC1
02 W2530 W40145      STA 0,OAC0
03 W2531 W04413      JSR ,RAND
04 W2532 W00145      OAC0
05 W2533 W24146      LWA 1,OAC1
06 W2534 130700      NEGS 1,2
07 W2535 112000      ADC 0,2
08 W2536 W50147      STA 2,OAC2
09 W2537 W02125      JMP @KANRET
10
11 W2540 W20145 RAND1: LWA 0,OAC0
12 W2541 W24146      LWA 1,OAC1
13 W2542 W30147      LWA 2,OAC2
14 W2543 W00773      JMP RAND1-2
15
16          ; RANDOM NUMBER GENERATOR
17          ; GENERATES A (PSEUDO) RANDOM SEQUENCE OF INTEGERS
18          ; IN THE RANGE 0<= N <= 2**16-1
19
20          ; INPUT:      ADDRESS OF OLD VALUE POINTED TO BY WORD
21          ;              AFTER JSR
22
23          ; OUTPUT:     16-BIT NEW RANDOM NUMBER IN AC0
24          ;              AND IN STORAGE REPLACING OLD VALUE
25
26          ; CALLING SEQUENCE:
27          ;             JSR ,RAND
28          ;             ADDRESS OF OLD VALUE
29          ;             RETURN
30
31          ; METHOD:      GENERATES A LINEAR CONGRUENTIAL
32          ;              SEQUENCE OF THE FORM:
33          ;              X(N+1) = (X(N)*A+C)MOD 2**16
34
35          ; CAUTION:    IF A K-BIT RANDOM NUMBER (AS OPPOSED
36          ;              TO A 16- BIT NUMBER) IS NEEDED,
37          ;              USE THE MOST SIGNIFIGANT
38          ;              K-BITS (THE LEAST SIGNIFIGANT K-BITS
39          ;              ARE NOT AS RANDOM),
40          ;              FOR EXAMPLE, TO OBTAIN RANDOM N MOD 2,
41          ;              USE THE SIGN BIT OF THE RESULT
42
43          ; UNCHANGED:  AC1, AC2
44          ; DESTROYED:  AC0, AC3, CARRY
45
46
47
48 W2544 W54157 ,RAND: STA 3,.UD03      ; SAVE RETURN
49 W2545 W10157      ISZ ,UD03      ; BUMP PAST ADDRESS CONSTANT
50 W2546 W44155      STA 1,.UD01      ; *SAVE AC1
51 W2547 W50156      STA 2,.UD02      ; *SAVE AC2
52 W2550 W31400      LWA 2,0,3        ; GET ADDRESS OF OLD VALUE
53 W2551 W21000      LWA 0,0,2        ; OLD VALUE TO AC0
54 W2552 W04407      JSR ,UD50        ; N=A
55 W2553 W34161      LWA 3,.UD20      ; GET INCREMENT, C
56 W2554 W63000      ADD 3,0          ; (N+A+C) MOD 2**16
57 W2555 W41000      STA 0,0,2        ; STORE I1
58 W2556 W24155      LWA 1,.UD01      ; *RESTORE AC1
59 W2557 W30156      LWA 2,.UD02      ; *RESTORE AC2
60 W2558 W02157      JMP ,UD03        ; RETURN

```

W276 ARITH

```

01
02          ; COMPUTE N*(2**11+2**2+1)
03
04 W2561 W24102 ,UD50: LWA 1,.UD21      ; GET COUNT, X
05 W2562 W44100      STA 1,.UD10      ; FOR ITERATION
06 W2563 W05120      MOVZL 0,1        ; N*2**(X+1)
07 W2564 W25120      MOVZL 1,1
08 W2565 W14150      DSZ ,UD10
09 W2566 W00776      JMP ,=2
10 W2567 W07000      ADD 0,1
11 W2570 W25120      MOVZL 1,1
12 W2571 W25120      MOVZL 1,1        ; 2**2*(N*2**(X+1)+N)
13 W2572 W23000      ADD 1,0
14 W2573 W01400      JMP 0,3        ; N+N*2**2+N*2**(X+3)
15
16
17
18

```

10077 ARITH

```

02 .TITL  DIAGS
03
04 /FILENAME=  DIAGSUP
05
06 /THIS PACKAGE IS USED TO SETUP THE SUBTESTS FOR THE PROGRAMS
07 /THAT RUN ON NOVA,ECLIPSE AND MICKUNOVA TYPE COMPUTERS.
08
09 /"ENT?R" ENTRY IS USED TO BEGIN A NEW TEST.
10 /"CYC?E" ENTRY IS USED TO END A TEST.
11 /"EKR?" ENTRY IS USED ON SEEING AN ERROR, WHEN THE FIRST ERROR
12 /OF A TEST IS SEEN A MESSAGE IS OUTPUT. THE PRECENT ERROR RATE
13 /CAN BE OBTAINED BY SETTING BIT 3 OF SWREG.
14
15 /NOTE: THIS PACKAGE REQUIRES THE TTYIO AND SWPAK PAKAGES
16 /IN ORDER TO RUN, IT ALSO NEEDS THE FOLLOWING VARIABLES IN
17 /PAGE ZERO:
18
19 /AT LOC 201      HELP:  0      TEST STARTING ADDRESS
20 /AT LOC 203      PASS:  0      PROGRAM PASS COUNTER
21 /ALSO           AC3?:  0      STORAGE FOR AC3
22 /              SWREG:  0      NEEDED BY SWPAK
23 /              IINP?:  INP?J   POINTERS
24 /              /          OR
25 /              /          INP?I
26 /              ICRL?F:  CRL?F   TTYIO
27 /              IPOC?S:  PDC?S   PACKAGE
28 /              IPOC?T:  PDC?T   ROUTINES
29 /              ILOC?T:  ZDC?T
30 /              IENT?R:  ENT?R   POINTERS TO ROUTINES IN
31 /              ICYC?E:  CYC?E   THIS PACKAGE
32 /              IERM?:  EHR?
33
34 /THE ENT?R ROUTINE STARTS EACH SUBTEST BY STORING ITS
35 /STARTING ADDRESS IN PAGE ZERO, SETTING ITS ITERATION
36 /COUNTER (ON THE FIRST PASS OF THE PROGRAM THE ITERATION
37 /COUNTER IS ALWAYS FORCED TO 1). ENT?R THEN CLEARS
38 /THE ERROR COUNT AND STARTS THE TEST.
39
40 /THE CALL TO ENT?R MUST BE FOLLOWED BY A WORD CONTAINING
41 /THE ITERATION COUNT FOR THIS TEST.
42
43 /  EXAMPLE:
44
45 /              JSR    #IENT?R
46
47 /THIS WILL START A TEST WITH ITS ITERATION COUNT SET TO 10.
48 /IENT?R IS A PAGE ZERO POINTER TO ENT?R
49
50 02074 054502 ENT?R: STA 3,LOP?E      /STORE IN RETURN LOCATION
51 02075 014503 STA 3,HELP      /STORE IN PAGE ZERO
52 02076 002500 LDA 3,ITR?      /LOAD IT
53 02077 034502 STA 3,ITR?T     /SET TEST ITERATION COUNTER
54 02078 175075 SUBC 3,3
55 02079 002502 STA 3,ITR?N     /CLEAN ERROR SWITCH
56 02080 034475 STA 3,ITR?C     /CLEAR ERROR COUNTER
57 02081 177100 JMP  #LOU?R      /EXIT TO TEST

```

10078 ARITH

```

01 /THE CYC?E ROUTINE IS USED TO END A SUBTEST IT HANDLES
02 /THE ITERATION OF THE SUBTEST AND THE PRINTING OF THE
03 /% ERROR RATE IF BIT 3 OF SWREG IS SET.
04
05 /THE CALL TO CYC?E IS:
06
07 /              JSR    #ICYC?E
08
09 /WHEN ICYC?E IS A PAGE ZERO POINTER TO CYC?E
10
11 02074 054502 CYC?E: STA 3,LOP?E      /STORE THE END OF TEST ADDRESS
12 02075 014503 DSZ ITR?T      /ALL ITERATIONS DONE?
13 02076 002500 JMP  #LOU?R     /NO, LOOP AGAIN
14 02077 034502 LDA 3,ITR?R     /ANY ERRORS?
15 02078 175075 MUV 3,3,SNR    /NO GO TO NEXT TEST
16 02079 002502 JMP  #LOP?E     /YES RESET THE ITERATION COUNTER
17 02080 034475 LDA 3,ITR?
18 02081 054475 STA 3,ITR?T
19 02082 034112 LDA 3,SWREG
20 02083 177100 ADDL 3,3
21 02084 177100 ADDL 3,3,SNR
22 02085 000424 JMP  CYC?E
23 02086 040474 STA  #,AC0?
24 02087 044474 STA 1,AC1?
25 02088 050474 STA 2,AC2?
26 02089 102500 SURCL 0,0
27 02090 140473 STA 0,CAR?
28 02091 000062 JSR  #ICRL?F   /PRINT CARRAGE RETURN
29 02092 102400 SUB 0,0
30 02093 024463 LDA 1,ITR?C
31 02094 030472 LDA 2,K10?R
32 02095 004442 JSR  MUL?
33 02096 030455 LDA 2,ITR?
34 02097 004424 JSR  DIV?D
35 02098 006065 JSR  #IPDC?S   /PRINT FAILURE RATE
36 02099 000045 "x
37 02100 020461 LDA  #,CAR?
38 02101 101200 MUVH 0,0
39 02102 020454 LDA  #,AC0?
40 02103 024454 LDA 1,AC1?
41 02104 030454 LDA 2,AC2?
42 02105 170400 CYC?E: SUB 3,3
43 02106 054446 STA 3,ITR?C
44 02107 034444 LDA 3,ITR?R
45 02108 175004 MUV 3,3,SZR
46 02109 034112 LDA 3,SWREG
47 02110 177113 ADDL# 3,3,SNR
48 02111 022435 JMP  #LOU?R
49 02112 102441 JMP  #LOP?E
50
51 /AC1#REM  AC0#(AC0,AC1)/AC2
52 02113 102400 DIV?1: SUB 0,0
53 02114 054444 DIV?2: STA 3,MOV?S
54 02115 142432 SUB# 2,0,SZC
55 02116 000413 JMP  MOV?3
56 02117 054441 DIV?D: STA 3,MOV?S
57 02118 034443 LDA 3,MOV?1
58 02119 125120 MUV?2: MUVL 1,1
59 02120 101100 MUV?1: MUVL 0,0
60 02121 142412 SUB# 2,0,SZC
61 02122 142400 SUB 2,0

```



```

0079 ARITH
01 02665 125100   MOVL 1,1
02 02666 175404   INC 3,3,SZR
03 02667 000773   JMP MDV?2
04 02670 170441   SUB0 3,3,SKP
05 02671 176420 MDV?3: SUBZ 3,3
06 02672 002426   JMP 0MDV?5
07
08 02673 102400   J(AC0,AC1)=AC1+AC2
09 02674 054424 MDL?A: STA 3,MDV?5
10 02675 034426   LVA 3,MDV?1
11 02676 125203 MDV?4: MOVH 1,1,SNC
12 02677 101201   MOVH 0,0,SKP
13 02700 143220   ADDZH 2,0
14 02701 175404   INC 3,3,SZR
15 02702 000774   JMP MDV?4
16 02703 125200   MOVCR 1,1
17 02704 002414   JMP 0MDV?5

```

```

10000 ARITH
01 02705 002306 RIR?N: RTRN
02 02706 000000 L00?R: 0
03 02707 000144 ITR?I: 100.
04 02710 000000 ITR?T: 0
05 02711 000000 ITR?R: 0
06 02712 000000 ITR?C: 0
07 02713 000000 LUP?FT: 0
08 02714 000000 AC0?I: 0
09 02715 000000 AC1?I: 0
10 02716 000000 AC2?I: 0
11 02717 000000 CAR?I: 0
12 02720 000000 MDV?5: 0
13 02721 000000 .HT?N: 0
14 02722 000144 K10?0: 100.
15 02723 177760 MDV?1: -20
16 02724 001000 C10?6: 001000
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

```

;THE ERR? ROUTINE IS USED IN CASES WHEN A SUBTEST FAILS
 ;THIS ROUTINE KEEPS A COUNT OF THE NUMBER OF TIMES
 ;THE TEST FAILED AND FOR THE FIRST ERROR IN A TEST IT WILL
 ;PRINT THE CONTENTS OF THE ACCUMULATORS, THE CARRY
 ;AND THE PROGRAM COUNTER. THE PROGRAM THEN DOES ONE OF THREE
 ;THINGS DEPENDING ON THE BIT SETTINGS OF SWREG IT WILL
 ;PRINT A MESSAGE SAYING IT HALTED AND STOP EXECUTION,
 ;PRINT A MESSAGE SAYING IT IS LOOPING ON THE ERROR AND RESTART
 ;THE TEST OR IT WILL CONTINUE TO THE NEXT TEST PRINTING NO
 ;MESSAGE. IF BIT 0 OF SWREG IS SET THE ROUTINE WILL HALT ON
 ;ANY ERROR ENCOUNTERED.

```

30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

```

;THE CALL TO ERR? SHOULD BE EXECUTED ONLY IF THE SUBTEST
 ;FAILS AND SHOULD BE PRECEDED BY AN INSTRUCTION WHICH STORES
 ;AC3 IN A PAGE ZERO LOCATION "AC3?" IF THIS IS NOT DONE THE
 ;PRINTOUT FOR AC3 WILL BE ERRONEOUS.

```

35
36
37
38
39
40
41
42
43
44

```

;AN EXAMPLE OF THE CALL IS:

```

37
38
39
40
41
42
43
44

```

;	XXX	;	INST SKPS IF TEST PASSED
;	JMP .+2	;	TEST FAILED
;	JMP .+3	;	TEST PASSED
;	STA 3,AC3?	;	STORE AC3 IN AC3?
;	JSR 0IERR?		

```

43
44

```

;WHERE IENP? IS A PAGE ZERO POINTER TO ERR?

```

45 02725 054774 ERR?: STA 3,.RT?N ;SAVE RETURN ADDRESS
46 02726 010764 ISZ ITR?C ;INCREMENT ERROR COUNT
47 02727 000401 JMP .+1 ;NO OP
48 02730 040764 STA 0,AC0? ;SAVE THE WORLD
49 02731 044764 STA 1,AC1?
50 02732 050764 STA 2,AC2?
51 02733 102560 SUBLL 0,0
52 02734 040763 STA 0,CAR?
53 02735 034754 LVA 3,ITR?R ;IS THIS THE FIRST ERROR
54 02736 175004 MOV 3,3,SZR ;YES SKIP
55 02737 000452 JMP ERR?3 ;NO RETURN
56 02740 006064 JSR 0IMEST?
57 02741 003020 MEA?DH
58 02742 024755 LVA 1,CAR?
59 02743 006070 JSR 0IZOC?T
60 02744 024750 LVA 1,AC0? ;PRINT THE AC'S, CARRY AND PC

```

```

0001 ARITH
41 02745 000066 JSR #IPOC?T
42 02746 024747 LDA 1,AC1?
43 02747 000066 JSR #IPOC?T
44 02750 024746 LDA 1,AC2?
45 02751 000066 JSR #IPOC?T
46 02752 024165 LDA 1,AC3?
47 02753 000066 JSR #IPOC?T
48 02754 024745 LDA 1,,RT?N
49 02755 044734 STA 1,I?R?R
50 02756 000066 JSR #IPOC?T
51 02757 020742 LDA 0,,RT?N
52 02760 034045 LDA 3,I?E?G?G?S
53 02761 025402 LDA 1,0,3
54 02762 125004 MOV 1,1,S?Z?R
55 02763 002722 JMP @R?R?N
56 02764 034112 LDA 3,S?W?R?E?G
57 02765 030737 LDA 2,C?I?B?7?6
58 02766 157400 AND 2,3
59 02767 024732 LDA 1,C?A?R?
60 02770 125200 MOVN 1,1
61 02771 020723 LDA 0,AC0?
62 02772 024723 LDA 1,AC1?
63 02773 030723 LDA 2,AC2?
64 02774 175004 MOV 3,3,S?Z?R
65 02775 000007 JMP @R?R?N
66 02776 034112 LDA 3,S?W?R?E?G
67 02777 177132 ADDZL= 3,3,S?Z?C
68 03000 002721 JMP @R?R?N
69 03001 000004 JSR #I?M?E?S?7?S
70 03002 003040 LUO?T
71 03003 002716 JMP @R?R?N
72 03004 000064 JSR #I?M?E?S?7?S
73 03005 003053 STO?P
74 03006 063077 MALT
75 03007 034112 LDA 3,S?W?R?E?G
76 03010 000766 JMP @R?R?N
77 03011 034112 LDA 3,S?W?R?E?G
78 03012 030712 LDA 2,C?I?B?7?6
79 03013 157400 AND 2,3
80 03014 030702 LDA 1,AC2?
81 03015 175004 MOV 3,3,S?Z?R
82 03016 063077 MALT
83 03017 002702 JMP @R?R?N

```

ISET ERROR SWITCH

RESTORE THE WORLD

HALT ON ERROR? NO SKIP
YES GO TO HALT ROUTINE

LOOP ON ERROR?
NO RETURN
YES SAY SO AND RETURN

PRINT HALTED AND STOP

RESTORE AC3 WITH SWREG
CONTINUE FROM THE ERROR HALT
HALT ON ERROR?

YES STOP EXECUTION
NO RETURN TO TEST

```

0002 ARITH
01 03020 005215 HEADR: .TXTE 1<15><12><15><12><15>
02 03021 005215
03 03022 141015 CHY AC0 AC1 AC2 AC3 PC <15><12><15>
04 03023 054722
05 040411
06 030303
07 040411
08 130703
09 040411
10 131303
11 040411
12 031703
13 050011
14 120303
15 005215
16 000215
17 03040 005215 LUO?T: .TXTE 1<15><12>LOOPING ON ERROR<15><12>
18 147714
19 050317
20 040311
21 120107
22 040317
23 142640
24 151322
25 151317
26 005215
27 000000
28 03053 005215 STO?P: .TXTE 1<15><12>HALTED ON ERROR<15><12>
29 040510
30 152314
31 042305
32 147640
33 120110
34 151305
35 147722
36 106722
37 000012

```

10083 ARITH

```
01
02
03      .TIIL      TTYIO
04      IFILENAME*TTYIO
05
06      ITELETYPE NON INTERRUPT PACKAGE
07      ICARRY,AC0,AC1,AC2 SAVED
08
09      I"MESS" PRINTS ASCII MESSAGES AS SPECIFIED BY ASSEMBLER
10
11      I"CKLYF" PRINTS A CARRIAGE RETURN
12
13      I"PUCT" PRINTS C(1) IN OCTAL
14      I"ZUC?T" PRINTS C(1) IN OCTAL, LEADING ZEROS SUPPRESSED
15      I"PU?C" PRINTS C(1) IN DECIMAL, LEADING ZEROS SUPPRESSED,
16      I THE ABOVE THREE ARE FOLLOWED BY A TAB
17      I"PU?S" PRINTS C(1) IN DECIMAL, LEADING ZEROS SUPPRESSED,
18      IFOLLOWED BY THE CHARACTER STORED AT CALLING LOCATION +1.
19      IPROGRAM RETURNS TO CALLING LOCATION +2.
20
21      I"TI?0" ACCEPTS OCTAL, AND
22      I"TI?D" ACCEPTS DECIMAL SINGLE PRECISION SIGNED INTEGERS
23      I INTO AC1 FROM THE TTY. LEADING NULLS, TABS,
24      I AND SPACES ARE IGNORED. A 16 BIT UNSIGNED INTEGER IS
25      I FORMED, THEN NEGATED IF A MINUS SIGN IS TYPED.
26      I EXIT AT CALL+1 IF INPUT ERROR WITH AC0=BAD CHARACTER.
27      I (NOT A LEGAL DIGIT OR TERMINATING CHARACTER)
28      I EXIT AT CALL+2 UPON TERMINATING CHARACTER
29      I WITH AC0=0, 0, 40, 12, 15, 25
30      I FOR NULL, SPACE, LINE-FEED, CARRIAGE RETURN, COMMA
31      I THE ABOVE WAIT FOR TTY DONE, THEN TTY IS CLEARED.
32      I RUBOUT WILL DELETE THE LAST DIGIT TYPED IN 'TI?D' AND
33      I 'TI?D'
34
35      I"CHC?T" PRINTS ASCII CHARACTER IN C(0)N; C(0)L MUST BE 0.
36      I EXITS CALL +2 IF C(0)R=0; SIMULATES TAB
37
38      I"TY?E" PRINTS C(0)R TO THE TTY OR LPT OR BOTH AS PER THE
39      I SWITCH SELECTION REGISTER 'S#REG'.
40      I EXITS AT CALL+1. REPLACE "TY?E" WITH
41      I INTERRUPT 'TY?E' IF DESIRED.
42
43      I"TPS?P" PRINTS A SPACE AND EXITS AT CALL+1 WITH AC0 = 40
44
45
```

10084 ARITH

```
01
02      I
03      I MESS?S ROUTINE
04      I
05      I THE CALLING SEQUENCE IS:
06      I
07      I      JSR      @MESS?S
08      I      I
09      I      PUNTER TO MESSAGE TO BE PRINTED
10
11      I
12      I
13      I
14      I
15      I
16      I
17      I
18      I
19      I
20      I
21      I
22      I
23      I
24      I
25      I
26      I
27      I
28      I
29      I
30      I
31      I
32      I
33      I
34      I
35      I
36      I
37      I
38      I
39      I
40      I
41      I
42      I
43      I
44      I
45      I
46      I
47      I
48      I
49      I
50      I
51      I
52      I
53      I
54      I
55      I
56      I
57      I
58      I
59      I
60      I
61      I
62      I
63      I
64      I
65      I
66      I
67      I
68      I
69      I
70      I
71      I
72      I
73      I
74      I
75      I
76      I
77      I
78      I
79      I
80      I
81      I
82      I
83      I
84      I
85      I
86      I
87      I
88      I
89      I
90      I
91      I
92      I
93      I
94      I
95      I
96      I
97      I
98      I
99      I
100     I
101     I
102     I
103     I
104     I
105     I
106     I
107     I
108     I
109     I
110     I
111     I
112     I
113     I
114     I
115     I
116     I
117     I
118     I
119     I
120     I
121     I
122     I
123     I
124     I
125     I
126     I
127     I
128     I
129     I
130     I
131     I
132     I
133     I
134     I
135     I
136     I
137     I
138     I
139     I
140     I
141     I
142     I
143     I
144     I
145     I
146     I
147     I
148     I
149     I
150     I
151     I
152     I
153     I
154     I
155     I
156     I
157     I
158     I
159     I
160     I
161     I
162     I
163     I
164     I
165     I
166     I
167     I
168     I
169     I
170     I
171     I
172     I
173     I
174     I
175     I
176     I
177     I
178     I
179     I
180     I
181     I
182     I
183     I
184     I
185     I
186     I
187     I
188     I
189     I
190     I
191     I
192     I
193     I
194     I
195     I
196     I
197     I
198     I
199     I
200     I
201     I
202     I
203     I
204     I
205     I
206     I
207     I
208     I
209     I
210     I
211     I
212     I
213     I
214     I
215     I
216     I
217     I
218     I
219     I
220     I
221     I
222     I
223     I
224     I
225     I
226     I
227     I
228     I
229     I
230     I
231     I
232     I
233     I
234     I
235     I
236     I
237     I
238     I
239     I
240     I
241     I
242     I
243     I
244     I
245     I
246     I
247     I
248     I
249     I
250     I
251     I
252     I
253     I
254     I
255     I
256     I
257     I
258     I
259     I
260     I
261     I
262     I
263     I
264     I
265     I
266     I
267     I
268     I
269     I
270     I
271     I
272     I
273     I
274     I
275     I
276     I
277     I
278     I
279     I
280     I
281     I
282     I
283     I
284     I
285     I
286     I
287     I
288     I
289     I
290     I
291     I
292     I
293     I
294     I
295     I
296     I
297     I
298     I
299     I
300     I
301     I
302     I
303     I
304     I
305     I
306     I
307     I
308     I
309     I
310     I
311     I
312     I
313     I
314     I
315     I
316     I
317     I
318     I
319     I
320     I
321     I
322     I
323     I
324     I
325     I
326     I
327     I
328     I
329     I
330     I
331     I
332     I
333     I
334     I
335     I
336     I
337     I
338     I
339     I
340     I
341     I
342     I
343     I
344     I
345     I
346     I
347     I
348     I
349     I
350     I
351     I
352     I
353     I
354     I
355     I
356     I
357     I
358     I
359     I
360     I
361     I
362     I
363     I
364     I
365     I
366     I
367     I
368     I
369     I
370     I
371     I
372     I
373     I
374     I
375     I
376     I
377     I
378     I
379     I
380     I
381     I
382     I
383     I
384     I
385     I
386     I
387     I
388     I
389     I
390     I
391     I
392     I
393     I
394     I
395     I
396     I
397     I
398     I
399     I
400     I
401     I
402     I
403     I
404     I
405     I
406     I
407     I
408     I
409     I
410     I
411     I
412     I
413     I
414     I
415     I
416     I
417     I
418     I
419     I
420     I
421     I
422     I
423     I
424     I
425     I
426     I
427     I
428     I
429     I
430     I
431     I
432     I
433     I
434     I
435     I
436     I
437     I
438     I
439     I
440     I
441     I
442     I
443     I
444     I
445     I
446     I
447     I
448     I
449     I
450     I
451     I
452     I
453     I
454     I
455     I
456     I
457     I
458     I
459     I
460     I
461     I
462     I
463     I
464     I
465     I
466     I
467     I
468     I
469     I
470     I
471     I
472     I
473     I
474     I
475     I
476     I
477     I
478     I
479     I
480     I
481     I
482     I
483     I
484     I
485     I
486     I
487     I
488     I
489     I
490     I
491     I
492     I
493     I
494     I
495     I
496     I
497     I
498     I
499     I
500     I
501     I
502     I
503     I
504     I
505     I
506     I
507     I
508     I
509     I
510     I
511     I
512     I
513     I
514     I
515     I
516     I
517     I
518     I
519     I
520     I
521     I
522     I
523     I
524     I
525     I
526     I
527     I
528     I
529     I
530     I
531     I
532     I
533     I
534     I
535     I
536     I
537     I
538     I
539     I
540     I
541     I
542     I
543     I
544     I
545     I
546     I
547     I
548     I
549     I
550     I
551     I
552     I
553     I
554     I
555     I
556     I
557     I
558     I
559     I
560     I
561     I
562     I
563     I
564     I
565     I
566     I
567     I
568     I
569     I
570     I
571     I
572     I
573     I
574     I
575     I
576     I
577     I
578     I
579     I
580     I
581     I
582     I
583     I
584     I
585     I
586     I
587     I
588     I
589     I
590     I
591     I
592     I
593     I
594     I
595     I
596     I
597     I
598     I
599     I
600     I
601     I
602     I
603     I
604     I
605     I
606     I
607     I
608     I
609     I
610     I
611     I
612     I
613     I
614     I
615     I
616     I
617     I
618     I
619     I
620     I
621     I
622     I
623     I
624     I
625     I
626     I
627     I
628     I
629     I
630     I
631     I
632     I
633     I
634     I
635     I
636     I
637     I
638     I
639     I
640     I
641     I
642     I
643     I
644     I
645     I
646     I
647     I
648     I
649     I
650     I
651     I
652     I
653     I
654     I
655     I
656     I
657     I
658     I
659     I
660     I
661     I
662     I
663     I
664     I
665     I
666     I
667     I
668     I
669     I
670     I
671     I
672     I
673     I
674     I
675     I
676     I
677     I
678     I
679     I
680     I
681     I
682     I
683     I
684     I
685     I
686     I
687     I
688     I
689     I
690     I
691     I
692     I
693     I
694     I
695     I
696     I
697     I
698     I
699     I
700     I
701     I
702     I
703     I
704     I
705     I
706     I
707     I
708     I
709     I
710     I
711     I
712     I
713     I
714     I
715     I
716     I
717     I
718     I
719     I
720     I
721     I
722     I
723     I
724     I
725     I
726     I
727     I
728     I
729     I
730     I
731     I
732     I
733     I
734     I
735     I
736     I
737     I
738     I
739     I
740     I
741     I
742     I
743     I
744     I
745     I
746     I
747     I
748     I
749     I
750     I
751     I
752     I
753     I
754     I
755     I
756     I
757     I
758     I
759     I
760     I
761     I
762     I
763     I
764     I
765     I
766     I
767     I
768     I
769     I
770     I
771     I
772     I
773     I
774     I
775     I
776     I
777     I
778     I
779     I
780     I
781     I
782     I
783     I
784     I
785     I
786     I
787     I
788     I
789     I
790     I
791     I
792     I
793     I
794     I
795     I
796     I
797     I
798     I
799     I
800     I
801     I
802     I
803     I
804     I
805     I
806     I
807     I
808     I
809     I
810     I
811     I
812     I
813     I
814     I
815     I
816     I
817     I
818     I
819     I
820     I
821     I
822     I
823     I
824     I
825     I
826     I
827     I
828     I
829     I
830     I
831     I
832     I
833     I
834     I
835     I
836     I
837     I
838     I
839     I
840     I
841     I
842     I
843     I
844     I
845     I
846     I
847     I
848     I
849     I
850     I
851     I
852     I
853     I
854     I
855     I
856     I
857     I
858     I
859     I
860     I
861     I
862     I
863     I
864     I
865     I
866     I
867     I
868     I
869     I
870     I
871     I
872     I
873     I
874     I
875     I
876     I
877     I
878     I
879     I
880     I
881     I
882     I
883     I
884     I
885     I
886     I
887     I
888     I
889     I
890     I
891     I
892     I
893     I
894     I
895     I
896     I
897     I
898     I
899     I
900     I
901     I
902     I
903     I
904     I
905     I
906     I
907     I
908     I
909     I
910     I
911     I
912     I
913     I
914     I
915     I
916     I
917     I
918     I
919     I
920     I
921     I
922     I
923     I
924     I
925     I
926     I
927     I
928     I
929     I
930     I
931     I
932     I
933     I
934     I
935     I
936     I
937     I
938     I
939     I
940     I
941     I
942     I
943     I
944     I
945     I
946     I
947     I
948     I
949     I
950     I
951     I
952     I
953     I
954     I
955     I
956     I
957     I
958     I
959     I
960     I
961     I
962     I
963     I
964     I
965     I
966     I
967     I
968     I
969     I
970     I
971     I
972     I
973     I
974     I
975     I
976     I
977     I
978     I
979     I
980     I
981     I
982     I
983     I
984     I
985     I
986     I
987     I
988     I
989     I
990     I
991     I
992     I
993     I
994     I
995     I
996     I
997     I
998     I
999     I
1000    I
```

16085 ARITH

```

01      ;
02      ;CHC?T ROUTINE
03      ;
04      ;THE CALLING SEQUENCE IS:
05      ;
06      ;      LDA      0,CHARACTER TO BE PRINTED (RIGHT BYTE)
07      ;      JSR      @CHC?T
08      ;
09
10 03107 000000 PSP?: 0
11 03110 000000 SPT?: 0
12 03111 000377 P3?: 377
13 03112 000011 PC1?: 11
14 03113 000000 CHR?: 0
15 03114 000000 CAC?: 0
16
17 03115 040777 CHC?T: STA      0,CAC?0
18 03116 101315      MOVSW 0,0,SNR
19 03117 001401      JMP      1,3
20 03120 175100      MOVVL 3,3
21 03121 054772      STA      3,CHR?E
22 03122 034770      LWA      3,PC1?1
23 03123 116415      SUBW# 0,3,SNR
24
25 03124 000403      JMP      CHA?3
26 03125 004564      JSR      TYP?E
27 03126 000407      JMP      CHE?X
28 03127 004557 CHA?3: JSR      TFS?P
29 03130 020514      LDA      0,CHR?Z
30 03131 034410      LDA      3,PC?Z
31 03132 163404      AND      3,0,SZR
32 03133 000774      JMP      CHA?3
33
34 03134 040510      STA      0,CHR?Z
35 03135 020757 CHE?X: LWA      0,CAC?0
36 03136 034755      LDA      3,CHR?E
37 03137 175200      MOVW# 3,3
38 03140 001400      JMP      0,3
39
40 03141 000007 PC??: 7
41
42

```

```

;AC0 SAVE IN CHC?T ROUTINE
;SAVE AC0
;RETURN +2 IF NULL
;FOR CARRY SAVE
;PRINT C(0) RIGHT
;ACS = 11
;SKIP IF A TAB IS NOT TO
;BE SIMULATED
;PRINT IT
;EXIT
;PRINT A SPACE
;ACS = 7
;SIMULATE A TAB WITH 1
;TO 7 SPACES
;RESTORE AC0
;RESTORE CRY
;
;EXIT

```

16086 ARITH

```

01      ;
02      ;CRL?F ROUTINE
03      ;
04      ;THE CALLING SEQUENCE IS:
05      ;
06      ;      JSR      @CRL?F
07      ;
08
09
10 03142 054474 CRL?F: STA      3,RTN?A
11 03143 004507      JSR      SAV?E
12 03144 000405      LDA      0,K15?
13 03145 004544      JSR      TYP?E
14 03146 020402      LDA      0,K12?
15 03147 000735      JMP      PLS?T
16 03150 000012 K12?: 12
17 03151 000015 K15?: 15
18
19

```

```

;SAVE RETURN
;SAVE THE WORLD
;PRINT CARRIAGE AND LF
;GO TO RESTORE THE WORLD

```

10007 AMI1H

```

01 ;
02 ;ZOC?T, POC?T, POC?S AND PDE?C ROUTINES,
03 ;
04 ;THE CALLING SEQUENCE IS:
05 ;
06 ; LDA 1,OCTAL NUMBER TO BE PRINTED
07 ; (LEADING ZEROES SUPPRESSED)
08 ; JSR #IZOC?T
09 ;
10 ;THE CALLING SEQUENCE IS:
11 ;
12 ; LDA 1,OCTAL NUMBER TO BE PRINTED
13 ; (LEADING ZEROES NOT SUPPRESSED)
14 ; JSR #IPOC?T
15 ;
16 ;THE CALLING SEQUENCE IS:
17 ;
18 ; LDA 1,DECIMAL NUMBER TO BE PRINTED
19 ; (LEADING ZEROES SUPPRESSED)
20 ; JSR #IPDE?C
21 ;
22 ;THE CALLING SEQUENCE IS:
23 ;
24 ; LDA 1,DECIMAL NUMBER TO BE PRINTED
25 ; (LEADING ZEROES SUPPRESSED)
26 ; JSR #IPDC?S
27 ; ALPHA WHERE ALPHA IS THE CHARACTER PRINTED
28 ; AFTER THE DECIMAL NUMBER
29 ;
30 ;
31 03152 054404 ZOC?T: STA 3,RTN?A ;SAVE THE RTN ADDRESS
32 03153 004477 JSR SAV?E ;SAVE THE WORLD
33 03154 102400 SUR 0,0 ;
34 03155 000404 JMP ZPO?T ;
35 03156 054400 POC?T: STA 3,RTN?A ;SAVE THE RTN ADDRESS
36 03157 004473 JSR SAV?E ;SAVE THE WORLD
37 03160 020405 LDA 0,PC6?0 ;
38 03161 152020 ZPO?T: SUB?R 2,2 ;PRINT C(1) IN OCTAL
39 03162 034404 LDA 3,PC1?0 ;C(2)=100000, C(3)=10
40 03163 000416 JMP PDC?1 ;
41 03164 175400 PDC?S: INC 3,3 ;UPDATE THE RTN ADDR PNTR
42 03165 054401 STA 3,RTN?A ;
43 03166 004404 JSR SAV?E ;SAVE THE WORLD
44 03167 034447 LDA 3,RTN?A ;
45 03170 021777 LDA 0,-1,0 ;HEAD THE CHARACTER TO BE
46 ;PRINTED AFTER THE DECIMAL
47 ;NUMBER
48 03171 040716 STA 0,PSP? ;SAVE THE SPECIAL CHAR.
49 03172 102000 AUC 0,0 ;AC0 = -1
50 03173 000404 JMP PDC?2 ;
51 03174 054442 PDE?C: STA 3,RTN?A ;SAVE THE RTN ADDRESS
52 03175 004405 JSR SAV?E ;SAVE THE WORLD
53 03176 102400 SUB 0,0 ;
54 03177 034751 PDC?2: LDA 3,K12? ;C(3)=12
55 03200 030447 LDA 2,DET?0 ;PRINT C(1) IN DECIMAL
56 03201 040707 PDC?1: STA 0,SPT?G ;ACTIVATE/DEACTIVATE THE TAG FOR
57 ;SPECIAL CHARACTER
58 ;BOTH ENTRIES PRINT NUMBER
59 03202 101415 INC# 0,0,SNR ;SKIP IF AC0 IS NOT -1
60 03203 101400 INC 0,0

```

0008 AMI1H

```

01 03204 040551 STA 0,ZSU?P ;THEN TAB TO NEXT POSITION
02 03205 054443 S1A 3,TMP? ;SAVE AC3
03 03206 034547 DCO?T: LDA 3,ZSU?P ;ZEROS SUPPRESS STUF
04 03207 102001 DEC?T: AUC 0,0,SKP ;SKIP FIRST TIME HERE PER DIGIT
05 03210 140400 SUB 2,1 ;DIVIDE C(AC1) BY C(AC2)
06 03211 101405 INC 0,0,SNR ;
07 03212 151235 MOV?R# 2,2,SNR ;FOR ZERO SUPPRESS
08 03213 034432 LDA 3,PC6?0 ;
09 03214 140453 SUB# 2,1,SNR ;SUBTRACT MORE?
10 03215 000773 JMP .-5 ;YES,GO BACK
11 03216 054537 STA 3,ZSU?P ;NO,SAVE ZERO SUPPRESS FLAG
12 ;
13 03217 163004 ADD 3,0,SZR ;C(0)=DIGIT
14 03220 004675 JSH CMC?T ;MAKE ASCII
15 03221 034427 LDA 3,TMP? ;PRINT
16 03222 102400 SUB 0,0 ;RESTORE AC3
17 03223 172423 SUBZ 3,2,SNR ;
18 ;DIVIDE C(AC2) BY C(AC3)
19 03224 000403 JMP .+3 ;SKIP IF AC3 > AC2
20 03225 101400 INC 3,0 ;AC3 < AC2
21 03226 000775 JMP .-3 ;SUBTRACT MORE
22 03227 111004 MOV 0,2,SZR ;WAS IT LAST DIGIT?
23 03230 000756 JMP DCO?T ;NO,GET NEXT DIGIT
24 03231 034657 LDA 3,SPT?G ;YES,CHECK THE SPECIAL CHAR FLAG
25 03232 020660 LDA 0,PC1?1 ;FOLLOW THE PRINTOUT WITH
26 03233 175405 INC 3,3,SNR ;TAB IF NOT SPCL CHAR FLAG
27 03234 020553 LDA 0,PSP? ;OTHERWISE FOLLOW WITH THE CHAR
28 03235 000647 JMP PLS?T ;TO EXIT
29 ;
30 03236 000000 RTN?A: 0 ;
31 03237 000000 PCR?Y: 0 ;CRY SAVE LOCATION
32 03240 000000 PAC?0: 0 ;AC0 SAVE LOCATION
33 03241 000000 PAC?1: 0 ;AC1 SAVE LOCATION
34 03242 000000 PAC?2: 0 ;AC2 SAVE LOCATION
35 03243 000000 SAV?F: 0 ;INTERUPT ENABE FLAG
36 03244 000000 CHR?Z: 0 ;
37 03245 000050 PC6?0: 60 ;
38 03246 000010 PC1?0: 10 ;
39 03247 023420 DET?0: 10000 ;
40 03250 000000 TMP?1: 0 ;
41 03251 000013 MSK?1: 13 ;MASK FOR TTI,TTO,LPT

```

10089 ARITH

```

01
02 ;
03 ; SAV?E , SAVE THE WORLD ROUTINE
04 ;
05 ; THIS ROUTINE SAVES AC0,AC1,AC2 AND CRY
06 ;
07
08 03252 040706 SAV?E: STA 0,PAC?0 ;
09 03253 044706 STA 1,PAC?1 ;
10 03254 050706 STA 2,PAC?2 ;
11 03255 102500 SUBCL 0,0 ;
12 03256 040701 STA 0,PCR?Y ;
13 03257 020772 LDA 0,MSK? ;GET MASK FOR TTI, TTD ,LPT
14 03258 062077 MSKU 0 ;
15 03261 001400 JMP 0,3 ;
16
17
18 ;
19 ; RST?R , RESTORE THE WORLD ROUTINE
20 ;
21 ; THIS ROUTINE RESTORES THE AC0,AC1,AC2 AND CRY
22 ;
23
24 03262 020755 RST?R: LDA 0,PCR?Y ;
25 03263 101200 MOVH 0,0 ;
26 03264 020754 LDA 0,PAC?0 ;
27 03265 024754 LDA 1,PAC?1 ;
28 03266 152400 SUB 2,2 ;
29 03267 072077 MSKU 2 ;
30 03270 030752 LDA 2,PAC?2 ;
31 03271 001400 JMP 0,3 ;
32

```

10090 ARITH

```

01 ;
02 ;TYPE AND TPS?P ROUTINES.
03 ;
04 ;THE CALLING SEQUENCE IS:
05 ;
06 ; LDA 0,CHARACTER TO BE PRINTED (RIGHT BYTE)
07 ; JSR 0,TYPE
08 ;
09 ;THE CALLING SEQUENCE IS:
10 ;
11 ; JSR 0,ITPS?P
12 ; NORMAL RETURN WITH AC0 = 00
13 ;
14
15
16 03272 130005 RUB?I: MOV 1,3,SNR ;CAN'T RUB-OUT IF AC1 = 0
17 03273 000467 JMP TIN?R ;RETURN WITH ILLEGAL CHARACTER
18 03274 120400 SUB 1,1 ;
19 03275 150422 SUBZ 2,3,SZC ;SKIP IF AC3 IS LESS THAN AC2
20 03276 125401 INC 1,1,SKP ;
21 03277 157001 ADD 2,3,SKP ;
22 03300 000775 JMP ,=3 ;
23 03301 054501 STA 3,FST?0 ;"FST?0" IS NON -1
24 03302 020743 LDA 0,PC0?0 ;AC0 = 00
25 03303 163000 ADD 3,0 ;
26 03304 004405 JSR TYPE ;ECHO AND DELETE THE DIGIT
27 03305 000524 JMP TIN?N ;
28
29 03306 040445 TPS?P: STA 0,TAC?0 ;SAVE AC0
30 03307 020447 LDA 0,PC4?0 ;
31 03310 101001 MOV 0,0,SKP ;SKIP OVER AC0 SAVE
32 03311 040442 TYP?E: STA 0,TAC?0 ;SAVE AC0
33 03312 175100 MOVL 3,3 ;SAVE CRY AND RTN ADDR
34 03313 054441 STA 3,TYP?R ;TYPE THE RIGHT BYTE OF AC0
35 03314 034567 LDA 3,P17?7 ;STRIP THE PARITY BIT
36 03315 163400 AND 3,0 ;
37 03316 034564 LDA 3,INT? ;
38 ;
39 ;IF IT IS HERE DUE TO SWITCH
40 ;SETTING ROUTINE THEN THE TYPE
41 ;ROUTS TO THE TTY WILL BE ENABLED
42 ;SKIP IF INT? IS =1
43 ;READ THE SWITCHES
44 ;SHIFT AC3 BY 2 PLACES
45 ;SKIP IF TYPEOUTS ARE NOT
46 ;SUPPRESSED
47
48 03317 175404 INC 3,3,SZR ;
49 03320 034112 LDA 3,SWREG ;
50 03321 177100 ADDL 3,3 ;
51 03322 175112 MOVLM 3,3,SZC ;
52
53 03323 000405 JMP PLP?T ;
54 03324 061111 TTY?: O0AS 0,TTD ;
55 03325 063511 SKPBZ TTD ;
56 03326 000777 JMP ,=1 ;
57 03327 060211 NI0C 1TD ;
58 03330 177100 PLP?T: ADDL 3,3 ;
59 03331 177103 ADDL 3,3,SNC ;REQUIRED ON THE LPT
60
61 03332 000405 JMP TPR?T ;
62 03333 061117 O0AS 0,LPT ;OUTPUT THE CHARACTER TO LPT
63 03334 063517 SKPBZ LPT ;WAIT FOR LPT
64 03335 000777 JMP ,=1 ;
65 03336 060217 NI0L LPT ;CLEAR THE DONE FLAG FOR LPT
66 03337 034544 TPR?T: LDA 3,P17?7 ;
67 03340 110043 A0CU 0,3,SNC ;SKIP IF IT WAS RUBOUT
68 03341 034415 LDA 3,PC4?0 ;AC3 = 00

```

```

0091 ARITH
01 03342 162432 SUB4# 3,0,S4C      ;SKIP FOR NON PRINTING CHR.
02 03343 010701 ISZ   CHR?Z
03 03344 034541 LDA   3,PC175      ;AC3 = 15
04 03345 110445 SUBU  0,3,SNR      ;SKIP IF IT WAS NOT A "CR"
05 03346 054676 STA   3,CHR?Z      ;CLEAN THE HORZ POS
06 03347 020404 LDA   0,TAC?0      ;RESTORE AC0
07 03350 034404 LDA   3,TYP?R      ;RESTORE CRY AND RTN ADDR
08 03351 175200 MOVH  3,3
09 03352 001400 JMP   0,3          ;RETURN
10 03353 000000 TAC?0: 0
11 03354 000000 TYP?R: 0
12
13 03355 000000 ZSU?P: 0
14 03356 000040 PC4?0: 40
15
16

```

```

1092 ARITH
21 ;
22 ;TIN?0 AND TIN?D ROUTINES.
23 ;
24 ;THE CALLING SEQUENCE IS:
25 ;
26 ; JSR @TIN?0 ;ACCEPT IN OCTAL
27 ; ERROR RETURN WITH BAD CHARACTER IN AC0
28 ; NORMAL RETURN WITH TERMINATING
29 ; CHARACTER IN AC0
30 ; 0,40,12,15,55 FOR
31 ; NULL,SPACE,L/F,C/M AND
32 ; COMMA RESPECTIVELY.
33 ;
34 ;
35 ; NOTE:
36 ; THE NUMBER IS ACCEPTED IN AC1 FROM TTI BY TIN?0 AND
37 ; BAD CHARACTER IS ANY CHARACTER THAT IS NOT A LEGAL
38 ; DIGIT OR A TERMINATING CHARACTER
39 ;
40 ;THE CALLING SEQUENCE IS:
41 ;
42 ; JSR @TIN?0 ;ACCEPT IN DECIMAL
43 ; ERROR RETURN ;SAME FORMAT AS TIN?0
44 ; NORMAL RETURN
45 ;
46
47 03357 020525 TIN?C: LDA  0,PC172
48 03358 004731 JSR  TYP?E
49 03359 010655 TIN?X: ISZ  RTN?A
50 03360 040420 TIN?R: STA  0,FST?D      ;"FST?D" IS NON -1
51 03361 152000 AUC  2,2          ;AC2 = -1
52 03362 020771 TSI? : LDA  0,ZSU?P
53 03363 175620 INC4# 3,3          ;AC3 IS 1 IF THE CHARACTER
54 ; TYPED WAS A + AND A 100000
55 ; IF IT WAS A -.
56
57 03366 054767 STA  3,ZSU?P
58 03367 101112 MOV4# 0,0,S4C      ;SKIP IF THE PREVIOUS SIGN
59 ; WAS A PLUS
60 03370 124400 NEG  1,1          ;TAKE TWO'S COMPLEMENT IF
61 ; THE PREVIOUS SIGN WAS "-"
62
63 03371 034650 LDA  3,PAC?1
64 03372 167000 ADD  3,1
65 03373 044646 STA  1,PAC?1      ;PAC?1 HAS THE INTERMEDIATE
66 ; RESULT
67
68 03374 126400 SUB  1,1
69 03375 151113 MOV4# 2,2,SNC      ;SKIP IF EXIT IS REQUIRED
70 03376 000433 JMP  TIN?M
71 03377 004603 JSR  RST?R      ;RESTORE THE WORLD
72 03380 020402 LDA  0,FST?D      ;RESTORE AC0
73 03401 002635 JMP  0,RTN?A      ;RETURN
74
75 03402 000000 FST?D: 0
76
77 03403 054633 TOD?T: STA  3,RTN?A      ;SAVE THE RTN ADDR
78 03404 004646 JSR  SAV?E      ;SAVE THE WORLD
79 03405 192000 AUC  0,0          ;AC1 = -1 (ENTRY FOR 00T)
80 03406 040774 STA  0,FST?D      ;LOOK FOR FIRST DIGIT
81 03407 101120 MOVZL 0,0          ;AC1 = -2
82 03410 000411 JMP  TIN?Z
83 03411 054625 TIN?D: STA  3,RTN?A      ;OCTAL ENTRY,SAVE RTN ADDR

```

0093 ARITH

```

01 03412 004640 JSR SAV?E
02 03413 102120 ADCZL 0,0
03 03414 000404 JMP TIN?Q
04 03415 054621 TIN?D: STA 3,RTN?A
05 03416 004634 JSR SAV?E
06 03417 102440 SUBD 0,0
07 03420 120400 TIN?U: SUB 1,1
08 03421 030463 TIN?Z: LDA 2,PC1?2
09 03422 113000 ADD 0,2
10
11 03423 102440 SUBD 0,0
12 03424 140731 STA 0,ZSU?P
13 03425 034730 TIN?S: LDA 3,ZSU?P
14 03426 175014 MOVW 3,3,SZR
15 03427 000732 JMP TIN?X
16 03430 054611 STA 3,PAC?1
17 03431 063610 TIN?W: SAKPDN TTI
18 03432 000777 JMP =-1
19 03433 062610 DIAL 0,TTI
20 03434 004655 JSR TYP?E
21 03435 034446 LDA 3,P1?7?7
22 03436 163400 AND 3,0
23 03437 110415 SUB# 0,3,SNR
24 03440 000632 JMP RUB?
25 03441 034715 LDA 3,PC4?0
26 03442 110414 SUB# 0,3,SZR
27 03443 101015 MOVW 0,0,SNR
28 03444 000761 JMP TIN?S
29 03445 034442 LDA 3,TIN?2
30 03446 110405 SUB 0,3,SNR
31 03447 000712 JMP TIN?X
32 03450 175414 INC# 3,3,SZR
33 03451 175235 MOVZ# 3,3,SNR
34 03452 000712 JMP TSI?
35 03453 034432 TIN?M: LDA 3,PC1?5
36 03454 110415 SUB# 0,3,SNR
37 03455 000702 JMP TIN?C
38 03456 034426 LDA 3,PC1?2
39 03457 110404 SUB 0,3,SZR
40 03460 000403 JMP TIN?N
41 03461 020424 LDA 0,PC1?5
42 03462 000676 JMP TIN?C+1
43 03463 034423 TIN?N: LDA 3,TIN?1
44 03464 117022 ADCZ 0,3,SZC
45 03465 150513 SUBL# 2,3,SNR
46 03466 000674 JMP TIN?R
47 03467 110666 ISZ ZSU?P
48 03470 102400 SUB 0,0
49 03471 110711 ISZ FST?D
50
51 03472 121120 MOVZL 1,0
52 03473 105120 MOVZL 0,1
53 03474 125120 MOVZL 1,1
54 03475 167000 ADD 3,1
55 03476 155220 MOVZ# 2,3
56 03477 175232 MOVZ# 3,3,SZC
57 03500 107000 ADD 0,1
58 03501 000730 JMP TIN?W
59
60 03502 000000 INT?: 0

```

0094 ARITH

```

01
02 03503 000177 P1?7?7: 177
03 03504 000012 PC1?2: 12
04 03505 000015 PC1?5: 15
05 03506 177720 TIN?1: -00
06 03507 000054 TIN?2: 54
07 03510 000100 C10?0: 100

```

TTTT BY PLACING -1 IN THIS LOC.

```

ISAVE THE WORLD
IOCTAL ENTRY SWITCH
I
IDECIMAL ENTRY,RTN SAVED
ISAVE THE WORLD
IDECIMAL ENTRY SWITCH

IAC2 IS 10 FOR OCTAL AND 12
IFOR DECIMAL NUMBERS

ISIGN AND LEADING SPACES FLAG

ISKIP FOR LEADING SPACES

ISTRIP THE PARITY BIT
ISKIP IF NOT A RUB-OUT

ISPACE, OR NULL

ICOMMA

IMINUS
IUR PLUS ?
IMODIFY THE SIGN
IAC3 = 15
IS IS IT A CARRIAGE RETURN?
IIF CR THEN GO TO TIN?C
IAC3 = 12
ISKIP FOR LINE FEED

IAC0 = 15

ISKIP IF NOT A DIGIT
ISKIP IF DIGIT

IOUT OF LEADING SPACES
IAC0 = 0
ISKIP IF IT WAS FIRST DIGIT
IFOR ODT

IAC1 IS SHIFTED BY 3 PLACES
I0 OLD PAC?1'S + NEW DIGIT

ISKIP IF OCTAL MODE
IADD 2 OLD PAC?1'S

ITYPE OUTSS CAN BE FORCED TO

```


10095 ARITH

```

02 .TITL SWPAK
03 ;
04 ;FILENAME = SWPAK
05 ;
06 ;1. SWITCH SETTINGS
07 ;
08 ; LOCATION "SWREG" IS USED TO SELECT THE PROGRAM OPTIONS
09 ; (NOT SYSTEM CONFIGURATION), WHILE RUNNING UNDER DTOS,
10 ; THIS LOCATION WILL BE LOADED BY THE MONITOR,
11 ; HOWEVER UNDER STAND ALONE AND PROGRAM LOAD MODES THIS
12 ; LOCATION WILL BE SET ACCORDING TO THE ANSWERS SUPPLIED
13 ; BY THE OPERATOR, IN ANY CASE THE OPTIONS CAN BE CHANGED
14 ; OR VERIFIED BY USING ONE OF THE COMMANDS GIVEN IN SEC.
15 ; 1.2
16 ;
17 ;
18 ;1.1 SWITCH OPTIONS
19 ; DIFFERENT BITS AND THEIR INTERPRETATION AT LOCATION
20 ; "SWREG" IS AS FOLLOWS:
21 ;
22 ; BIT OCTAL BINARY INTERPRETATION
23 ; VALUE VALUE
24 ;
25 ; 1 0 LOOP ON ERROR
26 ; 40000 1 SKIP LOOPING ON ERROR
27 ;
28 ; 2 0 PRINT TO CONSOLE
29 ; 20000 1 ABORT PRINT OUT TO CONSOLE
30 ;
31 ; 3 0 PRINT DETAILED ERROR ON THE
32 ; SELECTED DEVICE/DEVICES
33 ; 10000 1 ONLY % FAILURE REQUIRED
34 ;
35 ; 4 0 ALLOW END OF PASS PRINT OUT
36 ; 04000 1 SUPPRESS END OF PASS PRINT OUT
37 ;
38 ; 5 0 DO NOT PRINT ON THE LINE PRINTER
39 ; 02000 1 PRINT ON THE LINE PRINTER
40 ;
41 ; 6 0 DO NOT HALT ON ERROR
42 ; 01000 1 HALT ON ERROR
43 ;
44 ;
45 ;1.2 SWITCH COMMANDS
46 ; ONCE THE PROGRAM STARTS EXECUTING THE STATE OF ANY OF
47 ; THE BITS CAN BE CHANGED BY HITTING KEYS 1 THROUGH 6, THE
48 ; PROGRAM WILL CONTINUE RUNNING AFTER UPDATING THE OPTIONS
49 ; EACH KEY WILL COMPLEMENT THE STATE OF THE BIT AFFILIAT-
50 ; ED WITH IT, THUS BIT 4 CAN BE ALTERED BY HITTING KEY 4.
51 ; SETTING OF ANY BIT OF LOCATION "SWREG" WILL SET BIT 0.
52 ; (DEFAULT MODE IS DEFINED AS ALL BITS OF SWREG SET TO 0)
53 ; THE PROGRAM CAN BE LOCKED INTO SWITCH MODIFICATION MODE
54 ; BY TYPING A 0, IN WHICH CASE MORE THAN ONE BITS CAN BE
55 ; CHANGED BEFORE THE CONTROL IS ALLOWED TO RETURN TO THE
56 ; MAIN PROGRAM.
57 ;
58 ;1.2.1 OTHER COMMANDS
59 ;
60 ; "CR" A "RETURN" CAN BE TYPED TO CONTINUE THE PROGRAM

```

0096 ARITH

```

01 ; AFTER ITS LOCKED IN A SWITCH MODIFICATION MODE
02 ;
03 ; AU THIS COMMAND GIVEN AT ANY TIME WILL RESET "SWREG
04 ; TO DEFAULT MODE AND RESTART THE PROGRAM,
05 ;
06 ; AR THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE
07 ; PROGRAM, SWITCHES ARE LEFT WITH THE VALUES THEY
08 ; HAD BEFORE THE COMMAND WAS ISSUED,
09 ;
10 ; AD THIS COMMAND GIVEN AT ANY TIME WILL CAUSE THE
11 ; PROGRAM CONTROL TO GO TO ODT (SEE SEC. 6)
12 ;
13 ; M THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE
14 ; CURRENT OPERATING MODES.
15 ;
16 ;

```

10097 ANITH

```

01
02
03
04
05
06
07
08
09
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
51
52
53
54
55
56
57
58
59
60

```

THIS PACKAGE IS USED TO CHANGE THE SETTINGS OF LOCATION
 "SWREG" OF PAGE 0. THE PROGRAM CONTROL SHOULD ENTER "INP?I OR
 "INP?J" WITH AC3 HAVING THE RTN ADDR, IF THE TTI INPUT IS
 ON INTERRUPT BASIS, ENTER AT "INP?I" OTHERWISE ENTER AT "INP?J"
 THE INPUT IS ECHOED AFTER A "CR". IF THE COMMAND IS
 NOT A LEGAL ONE THEN THE CONTROL IS RETURNED WITHOUT DOING
 ANY THING, OTHERWISE ONE OF THE FOLLOWING COMMANDS IS
 EXECUTED:
 KEYS 1-9 AND A-F ARE USED TO COMPLEMENT THE CURRENT VALUE
 OF BITS 1-15 OF "SWREG". IF ONE OF THESE KEYS IS HIT THE
 CORRESPONDING BIT OF "SWREG" IS COMPLEMENTED AND THE CONTROL
 IS RETURNED TO THE STATE PROGRAM HAD BEFORE HITTING THE KEY
 TYPING OF A "0" WILL LOCK THE PROGRAM IN A SWITCH MODIFICATION
 MODE IN WHICH CASE MORE THAN ONE BITS CAN BE CHANGED BEFORE
 THE CONTROL IS ALLOWED TO RETURN TO THE MAIN PROGRAM. HITTING
 THE "CR" KEY WILL UNLOCK THE PROGRAM FROM THIS MODE.
 "AD" THIS COMMAND GIVEN AT ANY TIME WILL RESET THE "SWREG"
 TO DEFAULT MODE (ALL ZEROS) AND RESTART THE PROGRAM AT ADD.
 STORED IN LOCATION "INS?"
 "AR" THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE PROG.
 AT ADDRESS STORED IN LOCATION "INS?"
 "M" THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE CURRENT
 OPERATING MODES.
 BEFORE THE CONTROL IS RETURNED TO THE MAIN PROGRAM BIT 0 WILL
 BE SET IF ANY OF THE OTHER BITS OF "SWREG" IS SET, OTHERWISE
 IT WILL BE CLEARED
 "ODT" ROUTINE USES ONE LOCATION IN PAGE 0 FOR BREAK POINT.
 ADDRESS OF THIS LOCATION SHOULD BE PLACED AT "ODO?F"
 THE CALLING SEQUENCE IS:
 JSP @INP?, WHERE IINP? POINTS TO
 INP?I OR INP?J
 ENTER AT INP?I FOR INTERRUPT
 DRIVEN PROGRAMS ,OTHERWISE AT INP?J

```

41 03511 000033 INJ?3: 33
42 03512 000136 INI?36: 136
43 03513 000104 INI?04: 104
44
45 03514 054574 INP?I: STA 3,INP?R
46 03515 170121 ADC#L 3,3,SKP
47 03516 054572 INP?J: STA 3,INP?R ;SAVE THE RETURN ADDRESS
48 03517 054406 STA 3,INT?E
49 03520 004543 JSR INS?V ;SAVE THE STATUS
50 03521 040574 STA 0,INL?K ;"INL?K" IS NOT -1
51 03522 060610 INO?: DIAL 0,TTI ;READ THE INPUT
52 03523 034760 LDA 3,P1?7?7 ;AC3 = 177
53 03524 163407 AND 3,0 ;GET MID OF THE PARITY BIT
54 03525 034760 LDA 3,PC1?5 ;AC1 = 15
55 03526 110415 SUB# 0,3,SNR ;SKIP IF THE CHARACTER TYPED
56 ;WAS NOT "CR"
57 03527 000426 JMP INR?
58 03530 040557 STA 0,INS?3 ;SAVE THE CHARACTER
59 03531 024757 LDA 1,C10?0 ;AC1 = 100
60 03532 034752 LDA 3,PC1?2

```

0098 ANITH

```

01 03533 116414 SUB# 0,3,SZR
02 03534 034755 LDA 3,IN3?3
03 03535 162453 SUB# 3,0,SNR
04
05 03536 000451 JMP IN1?
06 03537 107000 INO?: ADD 0,1
07
08 03540 020752 LUA 0,IN1?36
09 03541 000067 JSR 0ITYP?E
10 03542 121000 MUV 1,0
11 03543 000067 JSR 0ITYP?E
12 03544 034562 LDA 3,I12?1
13 03545 162015 ADC# 3,0,SNR
14 03546 000405 JMP IN0?
15 03547 034744 LDA 3,IN1?04
16 03550 116404 SUB 0,3,SZR
17 03551 000504 JMP IN4?
18 03552 054112 STA 3,SWREG
19 03553 034541 INO?: LDA 3,INS?
20
21 03554 054534 STA 3,INP?R
22 03555 170400 INR?: SUB 3,3
23 03556 163000 MUV 3,1
24 03557 000062 JSR 0ICHL?F
25 03560 054722 STA 3,INT?
26 03561 030112 LDA 2,SWREG
27 03562 170220 ADC#R 3,3
28 03563 170404 AND 3,2,SZR
29
30 03564 172000 ADC 3,2
31 03565 050112 STA 2,SWREG
32 03566 034523 LDA 3,STA?T
33 03567 170223 MOV#R 3,3,SNR
34
35 03570 060211 NIOC TTO
36 03571 175220 MOV#R 3,3
37 03572 175204 MOV# 3,3,SZR
38
39
40 03573 125004 MOV 1,1,SZR
41
42 03574 176000 ADC 3,3
43
44 03575 054514 STA 3,STA?T
45
46 03576 020506 LDA 0,INS?0
47 03577 024506 LDA 1,INS?1
48 03600 030506 LDA 2,INS?2
49 03601 034506 LDA 3,INS?3
50 03602 010507 ISZ STA?T
51 03603 060177 INTEN
52 03604 002504 JMP @INP?K
53
54
55
56 03605 000000 INT?E: 0
57 03606 003306 I1P5?4: TPS?P

```

SKIP IF IT IS A LINE FEED
 AC3 = 33
 DON'T SKIP IF AC0 IS 33
 FOR MORE
 AC1 = 100+ ASCII VALUE OF
 CONTROL CHARACTER
 AC0 = 136
 TYPE A
 AC3 = 121
 SKIP IF IT IS NOT AR
 AC3 = 104
 SKIP IF IT WAS A AD
 LOAD "SWREG" WITH 0
 AC3 = ADDRESS OF THE LOCATION
 WHERE THE PROGRAM WILL START
 AC3 = 7777
 SKIP IF THE SWITCHES ARE SET
 TO ALL ZERO'S
 SKIP IF DONE BIT ON TTO IS TO
 BE LEFT SET
 LOAD THE CARRY BIT AND SKIP
 IF THE INTERRUPTS ARE NOT TO
 BE ENABLED
 SKIP IF THE INSTRUCTION BEING
 EXECUTED IS A "P"
 INTERRUPTS ARE TO BE LEFT DIS-
 ABLED
 "STA?T" IS 0 IF INTERRUPTS ARE
 TO BE ENABLED AND +1 OTHERWISE
 RESTORE THE ACCUMULATORS
 START EXECUTING THE USER'S
 PROGRAM

10099 ARITH				610P ARITH			
01 03607 006067 IN1?:	JSR	#ITYP?E	JECHO THE CHARACTER	01 03670 06J577	SKPBZ	CPU	JSKIP IF INTERRUPTS ARE NOT
02 03610 034517	LDA	3,IN070	FAC3 = 00	02			ENABLED
03 03611 152620	SUBLR	2,2	FAC2 = 100000	03 03671 101141	MOVOL	0,0,SKP	
04 03612 110405	SUB	0,3,SNR	JSKIP IF THE DIGIT TYPED WAS	04 03672 101120	MOVZL	0,0	
05			INOT 0	05 03673 060277	INTDS		
06 03613 000503	JMP	IN3?		06 03674 060611	SKPDN	TTO	JSAVE THE TTO STATUS
07				07 03675 101121	MOVZL	0,0,SKP	
08 03614 151221 IN2?:	MOVZR	2,2,SKP	JSKIP AC2 TO RIGHT	08 03676 101140	MOVOL	0,0	
09 03615 120520	SUBZL	1,1	FAC1 = 1	09 03677 040412	STA	0,STA?T	
10 03616 175405	INC	3,3,SNR		10 03700 152000	ADC	2,2	
11 03617 000501	JMP	IN3?+2		11 03701 050601	STA	2,INT?	IFORCE THE TYPE OUTS ON
12 03620 147415	AND#	2,1,SNR	JSKIP IF THE DIGIT TYPED WAS	12			THE TELETYPE
13			INOT 0	13 03702 002062	JMP	#ICRL?F	
14 03621 000773	JMP	IN2?		14			
15 03622 106400	SUB	0,1	JSKIP IF THE DIGIT TYPED WAS	15 03703 005400 INS?A:	JSR	0,3	
16			INOT 0	16 03704 000000 INS?0:			
17 03623 135000	MOV	1,3	JSKIP IF THE DIGIT TYPED WAS	17 03705 000000 INS?1:			
18 03624 151225	MOVZR	2,2,SNR	INOT 0	18 03706 000000 INS?2:			
19 03625 000430	JMP	IN4?		19 03707 000000 INS?3:			
20 03626 024503	LDA	1,IN175		20 03710 000000 INP?R:			
21 03627 167004	AUD	3,1,SZR	FAC1 = 15	21 03711 000000 STA?T:			
22 03630 000765	JMP	IN2?+1	JSKIP IF THE COMMAND WAS "M"	22 03712 177777 INB?A:	-1		
23				23 03713 000000 INB?I:	0		
24 03631 000062 INM?:	JSR	#ICRL?F	JSKIP IF THE COMMAND WAS "M"	24 03714 000200 INS?:	RES?T		
25 03632 152520	SUBZL	2,2		25 03715 000000 INL?K:	0		
26 03633 000065	JSR	#IPUC?S	JSKIP IF THE COMMAND WAS "M"	26			
27 03634 000040	40			27 03716 170000 IN3?:	ADC	3,3	FAC3 = -1
28 03635 125400	INC	1,1		28 03717 054776	STA	3,INL?K	IFLOCK IN SWITCH INPUT MODE
29 03636 034472	LDA	3,IN172		29 03720 024112	LDA	1,SNREG	IFREAD THE CURRRENT VALUE OF
30 03637 166492	SUB0#	3,1,SZC	FAC3 = 12	30			"SNREG"
31			JSKIP IF AC1 IS GREATER OR EQUAL	31 03721 133414	AND#	1,2,SZR	IFTAKE XOR OF AC1 AND AC2
32 03640 000746	JSR	#ITPS?P	TO AC3	32 03722 146401	SUB	2,1,SKP	
33 03641 151124	MOVZL	2,2,SZR	PRINT A SPACE	33 03723 147000	ADD	2,1	
34 03642 000771	JMP	INM?+2	JSKIP AFTER TYPING # 15	34 03724 044112	STA	1,SNREG	JSAVE THE NEW VALUE OF "SNREG"
35 03643 000062	JSR	#ICRL?F		35 03725 000730	JMP	IN4?	
36 03644 030112	LDA	2,SNREG		36			
37 03645 151140	MOVOL	2,2	FAC2 HAD SWITCH SETTINGS	37			
38 03646 126500	SUBCL	1,1	BRING THE CARRY BIT IN AC1	38 03726 000121 I12?1:	121		
39 03647 000065	JSR	#IPUC?S	TYPE THE CONTENTS OF AC1	39 03727 000060 IN6?0:	60		
40 03650 000040	40			40 03730 000012 IN1?2:	12		
41 03651 0006735	JSR	#ITPS?P	TYPE A SPACE	41 03731 000015 IN1?5:	15		
42 03652 151124	MOVZL	2,2,SZR	SKIP AFTER TYPING ALL THE 16	42			
43			BITS	43			
44 03653 000773	JMP	=5					
45 03654 000062	JSR	#ICRL?F	SKIP IF THE PROGRAM IS LOCKED				
46 03655 010440 IN4?:	ISZ	INL?K	IN SWITCH INPUT MODE				
47							
48 03656 000677	JMP	INR?	NEVER SKIP				
49 03657 014436	OSZ	INL?K	WAIT FOR OPERATOR INPUT				
50 03660 063610	SKPDN	TTI					
51 03661 000777	JMP	=-1					
52 03662 000640	JMP	IN0?					
53							
54							
55 03663 040421 INS?V:	STA	0,INS?0	SAVE THE ACC,				
56 03664 044421	STA	1,IN3?1					
57 03665 050421	STA	2,IN9?2					
58 03666 102560	SUBCL	0,0	SAVE THE CARRY				
59 03667 010716	ISZ	INT?E	SKIP IF DDT IS ENTERED THRU				
60			INTERRUPT HANDLER				

10101 ARITH

01
02
03
04
05
06
07
08
09
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
51
52
53
54
55
56
57
58
59
60

.TITLE OUTPK

FILENAME = OUTPK

THE CALLING SEQUENCE IS:

JSR #10DT?, WHERE 10DT? POINTS TO 0DT? OR
0DT?J
ENTER AT 0DT? IF RUNNING ON
INTERRUPT BASIS, OTHERWISE
ENTER AT 0DT?J

0DA?C: JSR #IPUC?S ;TYPE THE ACCUMULATOR NUMBER TO
;BE OPENED
0DA?: JSR #ITPS?P ;TYPE A SPACE
JSR INS?A ;LOAD AC3 WITH THE ADDRESS OF
;INS?B
MOV 3,2
ADD 1,2
LDA 3,0D?7? ;AC3 = 7
SUBR# 1,3,SNC ;SKIP IF ACC. NUMBER IS NOT OK
JMP 00A?L

0DP?C: AUC 0,3,SNR
0DR?T: JMP INR?+1 ;IF IT IS A "P" THEN PROCEED
LDA 3,0D6?M ;AC3 = 0
AUC 0,3,SNR ;SKIP IF AC0 IS NOT "/"
JMP 0DL?C
MOV 2,1
MOVZR 3,3,SNR ;SKIP IF IT IS NOT A "."
JMP 0DWT+4
JSR #IMES?S ;TYPE A "?"
TP??
JMP 0DWT

0DT?I: STA 3,INS?3 ;NORMALLY THE PROGRAM SHOULD
;ENTER 0DT AT THIS LOCATION
;THROUGH AN INTERRUPT HANDLER
;AC3 = RETURN ADDRESS

0DT?J: STA 3,INS?3 ;0DT SHOULD BE ENTERED AT THIS
;POINT THRU A JSR
; "INP?R" HAS THE RETURN ADD.
;SAVE THE STATUS

STA 3,INP?K
JSP IN?7V+3
SUB 2,2
LDA 1,INP?R
0DT?I: STA 1,INP?R
JSR #IZUC?T ;TYPE THE HALT LOCATION+1
INC 2,2,SZR ;SKIP IF THE BREAK POINT HAS
;TO BE DELETED

JMP 0DWT-1
0DD?B: LDA 0,INB?I ;LOAD AC0 IN CASE THE BREAK
;POINT IS TO BE REMOVED
LDA 3,INB?A
AUC 1,1,SKP

0102 ARITH

01 03775 050000 ODB?P: STA 3,0,2
02 03776 175414 INC# 3,3,SZR
03
04 03777 042713 STA 0,INB?A
05 04000 044712 STA 1,INB?A
06 04001 060210 NIOL TTI
07 04002 005002 0DWT: JSR #ICRL?F
08 04003 054474 STA 3,0DL?T
09
10 04004 000004 JSR #IMES?S
11 04005 004103 0DWT?P
12 04006 006472 JSR #ITUO?T
13
14 04007 060077 NIOL CPU
15 04010 034720 LDA 3,0D1?2
16 04011 110414 SUB# 0,3,SZR
17 04012 034717 LDA 3,0D1?5
18 04013 110414 SUB# 0,3,SZR
19 04014 000426 JMP 0DD?R
20 04015 010402 ISZ 0DL?T
21 04016 000402 JMP +2
22 04017 040000 STA 1,0,2
23 04020 101233 MOVZR# 0,0,SNC
24 04021 000701 JMP 0DWT
25 04022 140400 INC 2,1
26
27 04023 152220 0DL?C: AUCZR 2,2
28 04024 147400 AND 2,1
29 04025 000450 LDA 0,0D?7
30 04026 000455 JSR INS?A
31
32 04027 160422 SUBZ 3,1,SZC
33
34 04030 122433 SUBZ# 1,0,SNC
35 04031 167001 ADD 3,1,SKP
36 04032 000700 JMP 0DA?C
37 04033 000070 JSR #IZOC?T
38 04034 131000 MOV 1,2
39 04035 120000 00A?L: ADC 1,1
40 04036 044441 STA 1,0DL?T
41 04037 020000 LDA 1,0,2
42 04040 000000 JSR #IPDC?T
43 04041 000745 JMP 0DWT+4
44
45 04042 034671 0DD?R: LDA 3,N10?1
46 04043 110405 SUB 0,3,SNR
47 04044 000070 JMP 0DA?C
48
49 04045 175655 INCUR# 3,3,SNR
50 04046 034644 LDA 3,INB?A
51 04047 175645 INCUR 3,3,SNR
52 04050 120112 MOVLR 1,1,SZC
53
54
55 04051 000414 JMP 0DD?C
56 04052 040400 STA 1,INB?A
57 04053 030637 LDA 3,INB?A
58 04054 054637 STA 3,INB?I
59 04055 030421 LDA 2,0D0?F
60

JSKIP IF THERE IS NO BREAK
POINT
ENABLE BREAK POINTS
CLEAN THE TTY INPUT DONE SIGNAL
MAKE THE TAG FOR OPEN LOCATION
INON = 1 (NO LOCATION IS OPEN)
ACCEPT OCTAL NUMBERS SEPERATED
BY + OR - SIGNS
NO-OP
CHECK FOR A LINE FEED OR RETURN
DECODE REST OF THE COMMANDS
SKIP IF A LOCATION IS OPEN
RESTORE THE OPEN LOCATION
SKIP IF IT WAS NOT A "CR"
IF IT IS A LINE FEED THEN ADD
1 TO AC2
AC2 = 77777
NEGLECT ADDRESS BIT 0
AC0 = 7
LOAD AC3 WITH THE ADDRESS
OF INS?B
SKIP IF THE ADDRESS IN AC1 IS
LESS THAN THE ADDRESS OF INS?B
SKIP IF AC1 IS 7 OR LESS
OTHERWISE IT MUST BE AN ACC,
TYPE AC1
OPEN A LOCATION
TYPE THE OPEN LOCATION
AC3=101
IF AN "A" HAS BEEN TYPED
THEN GO TO SERVICE ACCUMULATOR
ROUTINE
SKIP IF IT WAS NOT A "B"
MAKE SURE THAT THIS IS THE
FIRST BREAK POINT
SKIP IF BIT 0 OF THE ADDRESS
WHERE THE BREAK POINT WILL
BE PLACED IS 0
DECODE OTHER COMMANDS
SAVE THE BREAK POINT ADD.
SAVE THE BREAK POINT INS?
BRING THE ADDRESS OF LOC.
IN PAGE 0 TO BE USED BY THE

```

0103 ARITH
01
02 04050 020423      LDA      0,ODI?N
03 04057 143000      ADD      2,0
04 04060 004715      JSR      ODB?P
05 04061 054026 OUB?E: STA      3,INS?3
06 04062 004001      JSR      INS?V
07 04063 024627      LDA      1,INB?A
08 04064 000702      JMP      ODT?1
09
10 04065 175065 ODD?C: INCCR   3,3,SNR
11
12 04066 000704      JMP      ODD?B
13 04067 034637      LDA      3,I12?1
14 04070 162014      AUC#    3,0,SZR
15
16 04071 000652      JMP      ODP?C
17 04072 062677      IORST
18 04073 044615      STA      1,INP?R
19 04074 000650      JMP      ODR?T
20
21      0003730 ODI?2# IN122
22      0003731 ODI?5# IN175
23      0003727 ODB?0# IN6?0
24 04075 000007 ODT?1: 7
25 04076 000377 ODD?F: 377
26
27 04077 000000 ODL?T: 0
28 04100 003403 ITOD?T: TOD?T
29 04101 002000 ODI?N: JMP      #0
30 04102 000077 TP??: 77
31 04103 000100 ODT?P: "0
32

```

```

BREAK POINT INSTRUCTION
JAC# = BREAK POINT INST.
THIS IS THE BREAK POINT ENTRY
SAVE EVERY THING

SKIP IF THE COMMAND IS NOT
"D" TO DELETE BREAK POINT

JACS#121
IF IT WAS A "R" THEN START THE
USERS PROGRAM

IORST

ADDRESS IN PAGE 0 TO BE USED
BY THE BREAK POINT

```

```

0104 ARITH
01 04104 105015 WORLD: .TXTE I<215><212>
02 04105 040714 LAST LOCATION IN MEMORY IS I
03      152123
04      140240
05      141717
06      152101
07      147711
08      120116
09      047311
10      046640
11      046705
12      151317
13      120131
14      051711
15      000240
16
17
18
19
20 04125 000215 ILTI: .TXTE I<15><12>ILLEGAL INTERRUPT BY DEVICE I
21      146311
22      142714
23      043501
24      142714
25      144640
26      152116
27      151305
28      050125
29      120324
30      054502
31      042240
32      053305
33      141711
34      120305
35      000000
36 04143 005215 OUSEQ: .TXTE I<15><12>PROGRAMS ARE RUNNING OUT OF SEQUENCE<15><
37      151120
38      043717
39      040722
40      051515
41      040640
42      142722
43      151240
44      047125
45      144516
46      043516
47      147640
48      152125
49      147640
50      120306
51      142523
52      052721
53      047305
54      142703
55      005215
56      000000
57 04170 105015 PMS: .TXTE I<215><212> PASS I
58      050240
59      051501
60      120123

```

0105 ARITH

```

01 000000
02
03 04175 000000 LAST= 0
04 004175 FINISH= LAST
05
06 04176 047503 .TXT /COPYRIGHT(C)DGC,1976
07 054520
08 044522
09 044107
10 024124
11 024503
12 043504
13 020103
14 034401
15 033067
16 04210 040101 ALL RIGHTS RESERVED/
17 020114
18 044522
19 044107
20 051524
21 051040
22 051505
23 051105
24 042526
25 000104
26
27 04222 151101 DIRT: .TXTE /ARITH TST001
28 152311
29 120110
30 051724
31 030324
32 000000
33 04230 000000 000000
34 04231 000200 000200
35 04232 002002 2002
36 04233 000000 000000
37 04234 000000 000000
38 04235 000000 000000
39 04236 000000 000000
40 04237 000000 000000
41
42 003065 MESS= MESS?
43 003174 PDECI= PDETC
44 003156 POCTA= POCPT
45 003152 ZOCTA= ZOCPT
47 .END .TITL ARITH
**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS

```

0100 ARITH

```

A1 001242 35/37
A2 001261 36/37
A3 001300 38/37
A4 001317 39/37
A5 001352 40/37
A6 001377 42/37
A7 001424 43/37
A8 001443 44/37
AC0? 002714 78/23 78/39 80/08 80/48 80/60 81/21
AC1? 002715 78/24 78/40 80/09 80/49 81/02 81/22
AC2? 002716 78/25 78/41 80/10 80/50 81/04 81/23 81/40
AC3? 000165 8/31 81/06
ADDE 001433 42/13 42/20 43/19
AND0 000655 17/37
AND1 000704 18/37
AND3 000121 20/37 20/22
AK1 001472 46/37
AK10 002003 58/37
AK11 002025 59/37
AK12 002044 60/37
AK13 002062 62/37
AK15 002111 63/37
AK16 002134 64/37
AK17 002154 66/37
AK18 002174 67/37
AK19 002214 68/37
AK2 001516 47/37
AK20 002234 69/37
AK3 001543 48/37
AK4 001566 50/37
AK5 001617 51/37
AK6 001645 52/37
AK7 001672 54/37
AK8 001727 55/37
AK9 001763 57/37
AKEND 002267 71/23
AKJSR 002254 71/37
C10? 003510 94/37 97/59
C1B? 002724 80/16 81/17 81/38
CAC? 003114 85/15 85/17 85/35
CAL 002447 7/31 74/05
CAL2 000120 7/50 74/06 74/11
CALL 000240 MC 7/10 9/25 9/30 9/45 21/12 21/15 22/11
22/14 24/12 24/15 25/12 26/12 27/12 29/12
30/12 31/11 32/11 34/11 34/14 34/17 34/20
34/23 34/26 34/29 34/32 42/15 46/14 46/18
47/11 47/16 47/20 48/12 48/17 50/13 50/16
57/12 71/29 71/47 71/50 74/52
7/49 72/32 72/42 72/56 73/09 73/22 74/05
74/08 74/09 74/13
8/06 74/12 74/14
78/27 78/37 80/11 80/52 80/58 81/19
85/25 85/28 85/32
84/22 84/25 85/17 88/14
85/27 85/35
85/14 85/21 85/36
85/29 85/34 88/36 91/02 91/05
7/30 73/37
003142 7/20 86/10

```

0107 ARITH

CYC22	002643	78/22	78/42						
CYC7E	002604	7/36	78/11						
DCD7T	003206	88/03	88/23						
DEC7T	003207	88/04							
DET7B	003247	87/55	88/34						
DIRT	004222	7/05	105/27						
DIV9	002653	78/51							
DIV70	002654	78/52							
DIV7D	002657	78/34	78/55						
EGGS	000105	7/14	7/39	9/22	9/34	73/50			
ENDE1	002311	71/28	71/36	71/43					
ENDTS	000232	MC	7/02						
ENT2R	002574	7/35	77/50						
ERR0R	000242	MC	7/10	11/14	12/14	13/11	13/14	13/17	14/30
			16/30	17/23	18/13	20/25	21/19	22/18	24/19
			25/16	26/16	27/17	29/16	30/18	31/15	32/15
			34/36	35/15	36/15	38/15	39/27	40/21	42/22
			43/15	44/21	46/22	47/24	48/21	50/21	50/25
			50/24	51/21	52/20	54/10	54/13	54/16	54/19
			54/22	54/25	54/28	54/31	54/34	54/37	55/11
			55/14	55/17	55/20	55/23	55/26	55/29	55/32
			55/35	57/17	58/17	59/15	60/14	62/22	63/19
			64/16	66/16	67/16	68/16	69/16	71/11	71/18
ERR7	002725	7/16	80/45						
ERR71	003004	81/25	81/32						
ERR72	002776	81/26	81/36						
ERR73	003011	80/55	81/37						
ESWIT	000143	7/50	8/09	74/57					
ET1	002516	11/21	21/25	31/21	42/28	53/03	63/25		
ET10	000745	20/34							
ET11	000766	21/26							
ET12	001006	23/04							
ET13	001027	24/26							
ET14	001046	25/23							
ET15	001065	26/23							
ET16	001105	28/04							
ET17	001124	29/23							
ET2	000534	12/21	23/03	33/03	43/22	54/43	65/03		
ET20	001143	30/23							
ET21	001162	31/22							
ET22	001201	33/04							
ET23	001235	34/43							
ET24	001254	35/22							
ET25	001273	37/04							
ET26	001312	38/22							
ET27	001345	39/34							
ET3	000554	13/25	24/25	34/42	45/06	56/03	66/22		
ET30	001372	41/04							
ET31	001417	42/29							
ET32	001436	43/23							
ET33	001465	45/07							
ET34	001511	46/29							
ET35	001536	47/31							
ET36	001561	49/04							
ET37	001612	50/36							
E14	000612	15/05	25/22	35/21	46/28	57/23	67/22		
E140	001040	51/28							
E141	001665	53/04							

0108 ARITH

ET42	001722	54/44							
ET43	001750	56/04							
ET44	001776	57/24							
ET45	002020	58/24							
ET46	002037	59/22							
ET47	002055	61/04							
ET5	000650	16/37	26/22	37/03	47/30	58/23	68/22		
ET50	002104	62/24							
ET51	002127	63/26							
ET52	002147	65/04							
ET53	002167	66/23							
ET54	002207	67/23							
ET55	002227	68/23							
ET56	002247	70/05							
ET6	000677	17/30	28/03	38/21	49/03	59/21	70/04		
ET7	000714	18/09	19/04	29/22	39/33	50/35	61/03		
ETEST	000171	MC	7/02						
FINIS	004175	8/07	105/04						
FIRST	000500	7/34	7/48	7/58	9/47	10/04	74/54		
FST70	003402	90/23	92/30	92/49	92/52	92/57	93/49		
HEA70	003020	80/57	82/01						
HELP	000163	8/29	77/51						
I1271	003726	98/12	100/38	103/13					
ICAL	000075	7/31	9/26	9/31	9/46	21/13	21/16	22/12	
		22/15	24/13	24/16	25/13	25/13	27/13	29/13	
		30/13	31/12	32/12	34/12	34/15	34/18	34/21	
		34/24	34/27	34/30	34/33	42/16	46/15	46/19	
		47/12	47/17	47/21	48/13	48/18	50/14	50/17	
		57/13	71/30	71/40	71/51	74/53			
ICAT	000076	7/32	9/44	74/15					
ICRL7	000062	7/20	78/28	98/24	99/24	99/35	99/45	100/13	
		102/07							
ICYCL	000102	7/36	11/17	12/17	13/20	14/33	16/33	17/26	
		18/16	20/20	21/22	22/21	24/22	25/19	26/19	
		27/20	29/19	30/19	31/16	32/18	34/39	35/18	
		36/18	38/18	39/30	40/24	42/25	43/16	44/24	
		46/25	47/27	48/24	50/32	51/24	52/23	54/40	
		55/38	57/20	58/20	59/18	60/17	62/25	63/22	
		64/19	66/19	67/19	68/19	69/19	71/21		
IEGGS	000045	7/14	71/26	71/34	81/12				
IENTE	000101	7/35	11/08	12/08	13/08	14/08	16/08	17/08	
		18/08	20/08	21/08	22/08	24/08	25/08	26/08	
		27/08	29/08	30/08	31/08	32/08	34/08	35/08	
		36/08	38/08	39/08	40/08	42/08	43/08	44/08	
		46/08	47/08	48/08	50/08	51/08	52/08	54/08	
		55/08	57/08	58/08	59/08	60/08	62/08	63/08	
		64/08	66/08	67/08	68/08	69/08	71/08		
IERN	000047	7/16	11/15	12/15	13/12	13/15	13/18	14/31	
		16/31	17/24	18/14	20/26	21/20	22/19	24/20	
		25/17	26/17	27/18	29/17	30/17	31/16	32/16	
		34/37	35/16	36/16	38/16	39/20	40/22	42/23	
		43/14	44/22	46/23	47/25	48/22	50/22	50/26	
		50/30	51/22	52/21	54/11	54/14	54/17	54/20	
		54/23	54/26	54/29	54/32	54/35	54/38	55/12	
		55/15	55/18	55/21	55/24	55/27	55/30	55/33	
		55/36	57/18	58/18	59/16	60/15	62/23	63/20	
		64/17	66/17	67/17	68/17	69/17	71/12	71/19	
IINP7	000063	7/21	73/35						

0109 ARITH

ILIT	004123	73/47	104/20						
IMES?	000064	7/22	73/46	80/56	81/29	81/32	101/37	102/10	
IMOVE	000060	7/18	71/46						
INM?	003522	97/51	99/52						
IN1?	003607	98/05	99/01						
IN10?	003513	97/43	98/15						
IN12?	003730	99/29	100/40	103/21					
IN13?	003512	97/42	98/08						
IN15?	003731	99/20	100/41	103/22					
IN2?	003014	99/00	99/14	99/22					
IN3?	003710	99/06	99/11	100/27					
IN32?	003511	97/41	98/02						
IN4?	003655	98/17	99/19	99/46	100/35				
IN5?	003537	98/06							
IN6?	003553	98/14	98/19						
IN62?	003727	99/02	100/39	103/23					
IN87A	003712	100/22	101/59	102/04	102/05	102/50	102/50	102/57	
		103/07							
IN87I	003713	100/23	101/57	102/58					
INL7K	003715	97/50	99/46	99/49	100/25	100/28			
INM?	003031	99/24	99/34						
INP?	000166	8/32							
INP7I	003514	7/21	8/32	97/45					
INP7J	003516	97/47							
INP7R	003710	97/45	97/47	98/21	98/52	100/20	101/48	101/51	
		101/52	103/18						
INR?	003555	97/57	98/22	99/46	101/30				
INS?	003714	98/19	100/24						
INS20	003704	98/46	99/55	100/16					
INS21	003705	98/47	99/56	100/17					
INS22	003706	98/48	99/57	100/18					
INS23	003707	97/58	98/49	100/19	101/41	101/46	103/05		
INS7A	003703	100/15	101/21	102/30					
INS7V	003663	97/49	99/55	101/49	103/06				
INTM	002424	7/19	73/32						
INTM1	002440	73/39	73/44						
INTM2	002442	73/33	73/46						
INTS0	000073	7/29	73/36	73/40					
INTS3	000072	7/28	73/34	73/41	73/44				
INT?	003502	90/37	93/60	98/25	100/11				
INT7E	003605	97/48	98/56	99/59					
IODT?	000071	7/27	73/45						
IPOC?	000065	7/23	78/35	99/26	99/39	101/17			
IPOC?	000066	7/24	81/01	81/03	81/05	81/07	81/10	102/42	
IRAND	000103	7/37	11/10	12/10	14/10	16/10	17/10	18/10	
		20/10	21/10	22/10	24/10	25/10	26/10	27/10	
		29/10	30/10	34/10	35/10	36/10	38/10	39/10	
		40/10	42/10	43/10	44/10	46/10	47/10	48/10	
		50/10	57/10	59/10	60/10	63/10	64/10	66/10	
		67/10	68/10	69/10	74/17				
ISEWE	000104	7/38	10/09	11/24	12/24	13/28	15/08	16/40	
		17/33	19/07	20/37	21/29	23/07	24/29	25/26	
		26/26	28/07	29/26	30/26	31/25	33/07	34/46	
		35/25	37/07	38/25	39/37	41/07	42/32	43/26	
		45/10	46/32	47/34	49/07	50/39	51/31	53/07	
		54/47	56/07	57/27	58/27	59/25	61/07	62/32	
		63/29	65/07	66/26	67/26	68/26	70/08		
ISHIG	000077	7/33							

0110 ARITH

ISLOW	000100	7/34	71/54	74/21					
IT00?	004100	102/12	103/28						
ITPS?	003006	98/57	99/32	99/41	101/20				
ITR	000127	7/57							
ITR?	002707	77/52	78/17	78/33	80/03				
ITR7C	002712	77/56	78/30	78/43	80/06	80/46			
ITR7R	002711	77/55	78/14	78/44	80/05	80/53	81/09		
ITR7T	002710	77/53	78/12	78/18	80/04				
ITYP?	000067	7/25	98/09	98/11	99/01				
I0L7	000070	7/26	73/49	80/59	101/53	102/37			
K107A	002722	78/31	80/14						
K12?	003150	86/14	86/16	87/54					
K15?	003151	86/12	86/17						
K200	000135	8/33	9/33						
K2000	000130	8/34	9/08	9/36					
K240	000133	8/31							
K377	000132	7/50							
K402	000131	7/59	9/40						
K525	000137	8/35	9/09						
K60	000134	8/32							
K7	000142	8/36	71/25						
KDATA	000126	7/56							
KEND	000141	8/37	74/32						
LAST	004175	7/33	7/47	7/58	105/03	105/04			
LUOP	000240	MC	7/10	11/10	12/16	13/19	14/32	16/32	17/25
			18/15	20/27	21/21	22/20	24/21	25/18	26/18
			27/19	29/18	30/18	31/17	32/17	34/38	35/17
			36/17	38/17	39/29	40/23	42/24	43/15	44/23
			46/24	47/26	48/23	50/31	51/23	52/22	54/39
			55/37	57/19	58/19	59/17	60/16	62/24	63/21
			64/18	66/18	67/18	68/18	69/18	71/20	
L007R	002706	77/50	77/57	78/13	78/48	80/02			
L007T	003040	81/30	82/17						
L0P7E	002713	78/31	78/16	78/49	80/07				
LSTPC	000144	8/30							
M20	000114	7/46	72/45						
M400	001462	45/04	46/11						
M8	000113	7/45							
M0V21	002723	78/56	79/11	80/15					
M0V22	002062	78/56	79/03						
M0V23	002671	78/54	79/05						
M0V24	002676	79/12	79/16						
M0V25	002720	78/52	78/55	79/06	79/10	79/18	80/12		
MES5	003065	9/27	71/31	71/52	105/42				
MES7M	003073	84/16	84/23						
MES7S	003065	7/22	84/10	105/42					
MNRST	001277	7/09							
MOVE	002462	7/18	7/32	71/49	74/16	74/34			
MOVE1	002473	74/28							
MOVE2	002502	74/36							
MOVE3	002504	74/38							
MOVE5	002515	74/46	74/49						
MOVE6	002460	74/19	74/22						
MUVE	000124	7/54	74/30	74/44					
MS1	000505	11/07							
MS2	000523	12/07							
MS3	000541	13/07							
MS14E	000123	7/53	9/20	9/21	9/29	9/37	9/39	74/23	

0113 ARITH

SURT	002401	46/20	47/13	48/14	73/02				
SQTEM	002422	73/10	73/16	73/17	73/18	73/25			
STAT	003711	98/32	98/44	98/50	100/09	100/21			
ST07P	003053	81/33	82/28						
SWREG	000112	7/44	78/19	78/46	81/16	81/26	81/35	81/37	
		90/41	98/18	98/26	98/31	99/36	100/29	100/34	
SYNC	000300	7/12							
T1	000504	11/05	21/02	31/02	42/02	52/02	63/02		
T10	000720	20/05							
T11	000751	21/05							
T12	000772	22/05							
T13	001012	24/05							
T14	001033	25/05							
T15	001052	26/05							
T16	001071	27/05							
T17	001111	29/05							
T2	000522	12/05	22/02	32/02	43/02	54/02	64/02		
T20	001130	30/05							
T21	001147	31/05							
T22	001166	32/05							
T23	001205	34/05							
T24	001241	35/05							
T25	001260	36/05							
T26	001277	38/05							
T27	001316	39/05							
T3	000540	13/05	24/02	34/02	44/02	55/02	66/02		
T30	001351	40/05							
T31	001376	42/05							
T32	001423	43/05							
T33	001442	44/05							
T34	001471	46/05							
T35	001515	47/05							
T36	001542	48/05							
T37	001563	50/05							
T4	000560	14/05	25/02	35/02	46/02	57/02	67/02		
T40	001616	51/05							
T41	001644	52/05							
T42	001671	54/05							
T43	001726	55/05							
T44	001762	57/05							
T45	002002	58/05							
T46	002024	59/05							
T47	002043	60/05							
T5	000616	16/05	26/02	36/02	47/02	58/02	68/02		
T50	002061	62/05							
T51	002110	63/05							
T52	002133	64/05							
T53	002153	66/05							
T54	002173	67/05							
T55	002213	68/05							
T56	002233	69/05							
T57	002253	71/05							
T6	000654	17/05	27/02	38/02	48/02	59/02	69/02		
T7	000703	18/05	29/02	39/02	50/02	60/02	71/02		
TACTP	003353	90/29	90/32	91/06	91/10				
TIN21	003500	93/43	94/05						
TIN22	003507	93/29	94/06						
TIN2C	003357	92/27	93/37	93/42					

0114 ARITH

TIN2D	003415	93/44							
TIN2M	003453	93/35							
TIN2N	003463	93/40	93/43						
TIN2O	003411	92/50							
TIN2Q	003420	93/43	93/07						
TIN2P	003362	90/17	92/30	93/40					
TIN2S	003425	93/13	93/20						
TIN2W	003431	90/27	92/47	93/17	93/58				
TIN2X	003361	92/29	93/15	93/31					
TIN2Z	003421	92/59	93/00						
TMP?	003250	88/02	88/15	88/40					
TN1	000515	11/20	11/21	21/25	31/21	42/28	53/03	63/25	
TN10	000744	20/33	20/34						
TN11	000765	21/25	21/26						
TN12	001005	23/03	23/04						
TN13	001026	24/25	24/26						
TN14	001045	25/22	25/23						
TN15	001064	26/22	26/23						
TN16	001104	28/03	28/04						
TN17	001123	29/22	29/23						
TN2	000533	12/20	12/21	23/03	33/03	43/22	54/43	65/03	
TN20	001142	30/22	30/23						
TN21	001161	31/21	31/22						
TN22	001200	33/03	33/04						
TN23	001234	34/42	34/43						
TN24	001253	35/21	35/22						
TN25	001272	37/03	37/04						
TN26	001311	38/21	38/22						
TN27	001344	39/33	39/34						
TN3	000553	13/24	13/25	24/25	34/42	45/06	56/03	66/22	
TN30	001371	41/03	41/04						
TN31	001410	42/28	42/29						
TN32	001435	43/22	43/23						
TN33	001464	45/06	45/07						
TN34	001510	46/28	46/29						
TN35	001535	47/30	47/31						
TN36	001560	49/03	49/04						
TN37	001611	50/35	50/36						
TN4	000611	15/04	15/05	25/22	35/21	46/28	57/23	67/22	
TN40	001637	51/27	51/28						
TN41	001664	53/03	53/04						
TN42	001721	54/43	54/44						
TN43	001755	56/03	56/04						
TN44	001775	57/23	57/24						
TN45	002017	58/23	58/24						
TN46	002036	59/21	59/22						
TN47	002054	61/03	61/04						
TN5	000647	16/36	16/37	26/22	37/03	47/30	58/23	68/22	
TN50	002103	62/20	62/20						
TN51	002126	63/25	63/26						
TN52	002146	65/03	65/04						
TN53	002166	66/22	66/23						
TN54	002206	67/22	67/23						
TN55	002226	68/22	68/23						
TN56	002245	70/04	70/05						
TN6	000670	17/29	17/30	28/03	38/21	49/03	59/21	70/04	
TN7	000713	18/09	18/03	19/04	29/22	39/33	50/35	61/03	
T002T	003403	92/54	103/20						

W115 ARITH

TPR2T	003337	90/53	90/58						
TPS7P	003306	85/28	90/29	98/57					
TP77	004102	101/30	103/30						
TREM	002300	71/34							
TSI?	003364	92/32	93/34						
TSTNM	000154	8/18	10/07	10/09	11/05	11/06	11/22	12/05	
		12/06	12/22	13/05	13/06	13/26	14/05	14/06	
		15/06	16/05	16/06	16/38	17/05	17/06	17/31	
		18/05	18/06	19/05	20/02	20/05	20/06	20/35	
		21/02	21/05	21/06	21/27	22/02	22/05	22/06	
		23/05	24/02	24/05	24/06	24/27	25/02	25/05	
		25/06	25/24	26/02	26/05	26/06	26/24	27/02	
		27/05	27/06	28/05	29/02	29/05	29/06	29/24	
		30/02	30/05	30/06	30/24	31/02	31/05	31/06	
		31/23	32/02	32/05	32/06	33/05	34/02	34/05	
		34/06	34/44	35/02	35/05	35/06	35/23	36/02	
		36/05	36/06	37/05	38/02	38/05	38/06	38/23	
		39/02	39/05	39/06	39/35	40/02	40/05	40/06	
		41/05	42/02	42/05	42/06	42/30	43/02	43/05	
		43/06	43/24	44/02	44/05	44/06	45/08	46/02	
		46/05	46/06	46/30	47/02	47/05	47/06	47/32	
		48/02	48/05	48/06	49/05	50/02	50/05	50/06	
		50/37	51/02	51/05	51/06	51/29	52/02	52/05	
		52/06	53/05	54/02	54/05	54/06	54/45	55/02	
		55/05	55/06	56/05	57/02	57/05	57/06	57/25	
		58/02	58/05	58/06	58/25	59/02	59/05	59/06	
		59/23	60/02	60/05	60/06	61/05	62/02	62/05	
		62/06	62/30	63/02	63/05	63/06	63/27	64/02	
		64/05	64/06	65/05	66/02	66/05	66/06	66/24	
		67/02	67/05	67/06	67/24	68/02	68/05	68/06	
		68/24	69/02	69/05	69/06	70/06	71/02	71/05	
		71/06							
TIY?	003324	90/46							
TYP?E	003311	7/25	85/26	86/13	90/26	90/32	92/28	93/20	
TYP?R	003354	90/34	91/07	91/11					
UPREL	000121	7/51							
WORLD	004104	9/28	104/01						
X1	000752	21/07							
X10	001167	32/07							
X11	001206	34/07							
X2	000773	22/07							
X3	001013	24/07							
X4	001034	25/07							
X5	001053	26/07							
X6	001072	27/07							
X7	001112	29/07							
X8	001131	30/07							
X9	001150	31/07							
XUM	002352	21/14	21/17	24/14	25/14	31/13	32/13	34/16	
		34/19	34/22	34/25	34/28	34/31	34/34	42/17	
		72/27							
XUR1	002357	22/16	24/17	26/14	27/14	72/35			
XUR2	002366	22/13	29/14	30/14	34/13	72/45	72/54		
XURTE	002423	72/46	72/53	73/26					
ZDCTA	003152	105/45							
ZGCT?T	003152	7/26	87/31	105/45					
ZPD?T	003161	87/34	87/38						
ZSU?P	003355	88/01	88/03	88/11	91/13	92/32	92/36	93/12	

W116 ARITH

.CB03	002335	93/13	93/47						
.CB20	002335	72/03	72/11	72/12	72/16	72/25			
.CB99	002320	72/04	72/13	72/17					
.CC98	002343	72/05	72/09						
.D1V1	002337	72/19	72/24						
.D1VU	002340	72/15							
.DPYA	002324	58/15	72/16						
.MPYU	002323	58/18	57/14	72/03					
.RAND	002544	46/10	47/16	72/02					
.KT?N	002721	74/60	75/03	75/48					
.UD01	000155	80/13	80/45	81/08	81/11	81/28	81/31	81/43	
.UD02	000156	8/20	75/50	75/58					
.UD03	000157	8/21	75/51	75/59					
.UD10	000160	8/22	75/48	75/49	75/60				
.UD20	000161	8/24	76/05	76/06					
.UD21	000162	8/26	75/55						
.UD50	002561	8/27	76/04						
?A	000057	75/54	76/04						
		7/01	10/09	11/06	11/19	11/20	12/06	12/19	
		12/20	13/06	13/23	13/24	14/06	15/03	15/04	
		15/06	16/35	16/36	17/06	17/20	17/29	18/06	
		19/02	19/03	20/06	20/32	20/33	21/06	21/24	
		21/25	22/06	23/02	23/03	24/06	24/24	24/25	
		25/06	25/21	25/22	26/06	26/21	26/22	27/06	
		28/02	28/03	29/06	29/21	29/22	30/06	30/21	
		30/22	31/06	31/20	31/21	32/06	33/02	33/03	
		34/06	34/41	34/42	35/06	35/20	35/21	36/06	
		37/02	37/03	38/06	38/20	38/21	39/06	39/32	
		39/33	40/06	41/02	41/03	42/06	42/27	42/28	
		43/06	43/21	43/22	44/06	45/05	45/06	46/06	
		46/27	46/28	47/06	47/29	47/30	48/06	49/02	
		49/03	50/06	50/34	50/35	51/06	51/26	51/27	
		52/06	53/02	53/03	54/06	54/42	54/43	55/06	
		56/02	56/03	57/06	57/22	57/23	58/06	58/22	
		58/23	59/06	59/20	59/21	60/06	61/02	61/03	
		62/06	62/27	62/28	63/06	63/24	63/25	64/06	
		65/02	65/03	66/06	66/21	66/22	67/06	67/21	
		67/22	68/06	68/21	68/22	69/06	70/03	70/04	
		71/06							
?B	000260	7/01	11/02	11/06	12/02	12/06	13/02	13/06	
		14/02	14/06	16/02	16/06	17/02	17/06	18/02	
		18/06	20/02	20/06	21/02	21/06	22/02	22/06	
		24/02	24/06	25/02	25/06	26/02	26/06	27/02	
		27/06	29/02	29/06	30/02	30/06	31/02	31/06	
		32/02	32/06	34/02	34/06	35/02	35/06	36/02	
		36/06	38/02	38/06	39/02	39/06	40/02	40/06	
		42/02	42/06	43/02	43/06	44/02	44/06	46/02	
		46/06	47/02	47/06	48/02	48/06	50/02	50/06	
		51/02	51/06	52/02	52/06	54/02	54/06	55/02	
		55/06	57/02	57/06	58/02	58/06	59/02	59/06	
		60/02	60/06	62/02	62/06	63/02	63/06	64/02	
		64/06	66/02	66/06	67/02	67/06	68/02	68/06	
		69/02	69/06	71/02	71/06				
?X	000002	10/09	11/02	11/06	11/20	11/22	12/06	12/06	
		12/20	12/22	13/02	13/06	13/24	14/02	14/02	
		14/06	15/04	15/06	16/02	16/06	16/36	16/38	
		17/02	17/06	17/31	18/02	18/06	18/06	19/03	
		19/05	20/02	20/06	20/33	20/35	21/02	21/06	

0117 ARITH

21/25	21/27	22/02	22/06	23/03	23/05	24/02
24/06	24/25	24/27	25/02	25/06	25/22	25/24
26/02	26/06	26/22	26/24	27/02	27/06	28/03
28/05	29/02	29/06	29/22	29/24	30/02	30/06
30/22	30/24	31/02	31/06	31/21	31/23	32/02
32/06	33/03	33/05	34/02	34/06	34/42	34/44
35/02	35/06	35/21	35/23	36/02	36/06	37/03
37/05	38/02	38/06	38/21	38/23	39/02	39/06
39/33	39/35	40/02	40/06	41/03	41/05	42/02
42/06	42/28	42/30	43/02	43/06	43/22	43/24
44/02	44/06	45/06	45/08	46/02	46/06	46/28
46/36	47/02	47/06	47/30	47/32	48/02	48/06
49/03	49/05	50/02	50/06	50/35	50/37	51/02
51/06	51/27	51/29	52/02	52/06	53/03	53/05
54/02	54/06	54/43	54/45	55/02	55/06	56/03
56/05	57/02	57/06	57/23	57/25	58/02	58/06
58/23	58/25	59/02	59/06	59/21	59/23	00/02
60/06	61/03	61/05	62/02	62/06	62/28	62/30
63/02	63/06	63/25	63/27	64/02	64/06	65/03
65/05	66/02	66/06	66/22	66/24	67/02	67/06
67/22	67/24	68/02	68/06	68/22	68/24	69/02
69/05	70/04	70/06	71/02	71/06		
70/09	11/02	11/06	11/20	11/22	12/02	12/06
12/20	12/22	13/02	13/06	13/24	13/26	14/02
14/06	15/04	15/06	16/02	16/06	16/36	16/38
17/02	17/06	17/29	17/31	18/02	18/06	19/03
19/05	20/02	20/33	21/02	21/25	22/02	23/03
24/02	24/25	25/02	25/22	26/02	26/22	27/02
28/03	29/02	29/22	30/02	30/22	31/02	31/21
32/02	33/03	34/02	34/42	35/02	35/21	36/02
37/03	38/02	38/21	39/02	39/33	40/02	41/03
42/02	42/28	43/02	43/22	44/02	45/06	46/02
46/28	47/02	47/36	48/02	49/03	50/02	50/35
51/02	51/27	52/02	53/03	54/02	54/43	55/02
56/03	57/02	57/23	58/02	58/23	59/02	59/21
60/02	61/03	62/02	62/28	63/02	63/25	64/02
65/03	66/02	66/22	67/02	67/22	68/02	68/22
69/02	70/04	71/02				
10/09	11/02	11/20	12/02	12/20	13/02	13/24
14/02	15/04	16/02	16/36	17/02	17/29	18/02
19/03	20/02	20/33	21/02	21/25	22/02	23/03
24/02	24/25	25/02	25/22	26/02	26/22	27/02
28/03	29/02	29/22	30/02	30/22	31/02	31/21
32/02	33/03	34/02	34/42	35/02	35/21	36/02
37/03	38/02	38/21	39/02	39/33	40/02	41/03
42/02	42/28	43/02	43/22	44/02	45/06	46/02
46/28	47/02	47/36	48/02	49/03	50/02	50/35
51/02	51/27	52/02	53/03	54/02	54/43	55/02
56/03	57/02	57/23	58/02	58/23	59/02	59/21
60/02	61/03	62/02	62/28	63/02	63/25	64/02
65/03	66/02	66/22	67/02	67/22	68/02	68/22
69/02	70/04	71/02				

7Y 000007

7Z 000011