

LISTING

096-000146-05

PROGRAM

ARITHMETIC TEST

TAPE

095-000037-05

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 RELOCATES ITSELF TO VARIOUS AREAS OF MEMORY.



01 0001 .MAIN

02 0002 .MAIN

03 ARITHMETIC TEST

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

 NAME: ARITHMETIC TEST
 PART NUMBER: 094-000007
 DESCRIPTION: ARITHMETIC TEST
 REVISION HISTORY:
 REV. DATE
 00 11/30/70
 01 01/25/74
 02 06/07/74
 03 04/18/75

 COPYRIGHT (C) DATA GENERAL CORPORATION, 1970, 1971, 1973,
 1974, 1975
 ALL RIGHTS RESERVED.

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 RELOCATES ITSELF TO VARIOUS
 AREAS OF MEMORY.
 MACHINE REQUIREMENTS
 NOVA FAMILY PROCESSOR
 2K READ/WRITE MEMORY
 TELETYPE
 SWITCH SETTINGS #000002
 STARTING ADDRESS #PROCEED FROM ERROR
 SWITCH 0(1) #INHIBIT ERROR PRINT/OUT
 SWITCH 1(1) #PRINT FAILURE RATE
 SWITCH 2(1)

OPERATING PROCEDURE
 LOAD THE PROGRAM VIA THE BINARY LOADER
 SET THE SWITCHES TO 000002
 PRESS START. CAUTION!
 SEE SECTION 7.0 FOR RESTART PROCEDURE.
 UPON STARTING, THE PROGRAM WILL PRINT
 THE MESSAGE "LAST LOCATION IN MEMORY IS XXXXX".
 THE OPERATOR SHOULD CHECK THE VALUE PRINTED
 WITH THE SYSTEM MEMORY SIZE. IF THEY DO NOT
 AGREE, A ERROR HAS OCCURED.
 THE PROGRAM WILL RUN UNTILL MANUALLY STOPPED.
 AT THE END OF EACH PROGRAM ITERATION THE WORD
 "PASS" WILL BE PRINTED AND THE PROGRAM WILL
 RELOCATE ITSELF TO ANOTHER AREA OF MEMORY.

A 0003 MAIN

```

01 0004 MAIN
02 0000V .LUC 0
03 00000 002100
04 00001 000000
05 00002 000117
06 00003 000117
07 00004 177770 N6:
08 00005 177750 M20:
09 00006 002120 SM10M: LAST
10 00007 000000 SLUM: FIKSI
11 00008 000045 .LUL 45
12 00009 000040
13 00010 000000 EGUS:
14 00011 000000
15 00012 000000
16 00013 000000
17 00014 000000
18 00015 000000 CALMET: 0
19 00016 000000 CAL2: 0
20 00017 000000 RELOC: 0
21 00018 000000 ERRET: 0
22 00019 000000 ERTEM: 0
23 00020 000000 LOOPK: 0
24 00021 000000 ERCT: 0
25 00022 000000 CYCNET: 0
26 00023 000000 ITRCTR: 0
27 00024 000000 MSIZE: 0
28 00025 000000 MVTIM: 0
29 00026 000000 RANPET: 0
30 00027 000233 KUATA: ES=IT-1
31 00028 000144 ITR: 144
32 00029 170300 PSIZE: FIRST-LAST
33 00030 000400 K400: 400
34 00031 000377 K377: 377
35 00032 000240 K240: 240
36 00033 000050 K05: 00
37 00034 000200 K200: 200
38 00035 000200 K200: 200
39 00036 000500 CALTEM: 0
40 00037 000007 K7: 7
41 00038 000000 NORES: 0
42 00039 000000 PASS: 0
43 00040 000000 CALL:
44 00041 000000 STA 3,CALRET
45 00042 000000 STA 2,CAL2
46 00043 000000 STA 1,RELOC
47 00044 000000 LIA 3,CALMET
48 00045 000000 ISZ CALMET
49 00046 000000 ADU 2,3
50 00047 000000 LIA 2,CAL2
51 00048 000000 STA 3,CALTEM
52 00049 000000 LIA 3,CALRET
53 00050 000000 JMP #CALTEM
54 00051 000000 RETURNEJMP #CALMET
55 00052 000000 CALL=JSK CAL

```

```

PROGRAM OUTPUT/ERROR DESCRIPTION
WHEN AN ERROR IS DETECTED BY THE PROGRAM THE
FOLLOWING WILL BE PRINTED:
AMS PC THE MEMORY LOCATION OF THE
ERRM SUBROUTINE CALL.
WHERE TO LOOK IN THE LISTING
TO FIND THE FAILING ROUTINE.
ORIGINAL THE ACCUMULATOR VALUES AS
ACW,AC1,AC2 DETERMINED VIA THE RANDOM
NUMBER GENERATOR.
RESULT THE RESULT IN THE ACCUMULATORS
ALW,AC1,AC2 PRIOR TO THE ERRM CALL
AFTER THE PRINTOUT THE PROGRAM WILL ITERATE THE
FAILING ROUTINE WITH THE SAME NUMBERS AS PRINTED.
CONSOLE SWITCH Z(1) MAY BE USED TO DETERMINE THE
RATE OF FAILURE. SWITCH W(1) WILL CAUSE THE
PROGRAM TO PROCEED TO THE NEXT TEST.
A IUPLS (A74) IS ISSUED ONE INSTRUCTION PRIOR
TO THE EXIT FROM THE RANDOM NUMBER GENERATOR
(MANDUM) FOR A SCOPE SYNC.

```

```

PROGRAM DESCRIPTION
THIS PROGRAM CONSIST OF A NUMBER OF SUBROUTINES
PERFORMING VARIOUS OPERATIONS ON SETS OF PSEUDOC
RANDOM NUMBERS. SOME ROUTINES PERFORM MORE THEN
ONE OPERATION, SUCH AS TAKING THE SQUARE ROOT OF
A SQUARED NUMBER. WHEN THE OPERATOR QUESTIONS
WHICH RESULT IS CORRECT, THE OPERATIONS MAY BE
PERFORMED WITH PENCIL AND PAPER ON THE ORIGINAL
NUMBERS.
AT THE END OF EACH PROGRAM ITERATION THE PROGRAM
RELOCATES ITSELF TO ANOTHER AREA OF MEMORY FOR
EXECUTION AT THAT SPOT. THIS RELOCATION PLACES
DIFFERENT OPERATING CONDITIONS ON THE PROCESSOR
REGISTERS SUCH AS, THE MEMORY ADDRESS REGISTER
AND THE PROGRAM COUNTER.
BECAUSE THIS PROGRAM IS USED AS A FINAL TEST OF
THE ARITHMETIC AND LOGICAL INSTRUCTIONS IT IS
COMPLEX AND DIFFICULT TO TROUBLE SHOOT. IT IS
THEREFOR SUGGESTED THAT THIS PROGRAM BE RUN
AFTER OTHER PROCESSOR TEST.

```

```

MISC
IF THIS PROGRAM IS MANUALLY STOPPED WHEN IN THE
PROCESS OF RELOCATION, IT CANNOT...BE RESTARTED.

```

```

PROGRAM STARTS HERE

```

```

;ERRM FLAG
;NOT USED
;DATA BLK ADDR
;# OF PASSES
;RETURN ADDRESS
;WHERE LAST SUBROUTINE CALLED FROM
;(AC2) AT THAT TIME
;CURRENT RELOCATION
;WHERE ERROR WAS ENTERED.
;ADDRESS OF LAST EXECUTED TEST
;ERROR COUNTER
;ADDRESS WHERE LAST "LOOP" EXECUTED
;ITERATION COUNTER
;MEMORY SIZE
;RETURN FOR RANDOM NUMBER
;ROUTINE ITERATION CONSTANT
;PROGRAM SIZE

```

```

A 0005 MAIN
01 02 0011/ 030077 SIZIT: LDA 2,K2000
03 00120 150001 MOV 2,K5,SKP
04 00121 150000 MOVZL4 3,K5,SZC
05 00122 175132 JMP *7
06 00123 000132 LDA 0,K0,J
07 00124 021400 STA 0,K0,J
08 00125 050400 LDA 1,K0,J
09 00126 020400 STA 0,K0,J
10 00127 041400 SUB# 1,K5,SNK
11 00130 130410 JMP SIZIT+2
12 00131 000121 ADD 1,K1
13 00132 120000 ADD 3,K1
14 00133 107000 STA 1,MSIZE
15 00134 044004 ISZ NUKES
16 00135 010103 JMP *3
17 00136 000141 HALT
18 00137 000077 JMP *-1
19 00140 000137 LDA 1,K6G5
20 00141 024040 MOV 1,K1,SZR
21 00142 125004 CALL *7
22 00143 000152 MESS
23 00144 004105 M0NLU
24 00145 001712 LDA 1,MSIZE
25 00146 002074 CALL
26 00147 024004 POCTAL
27 00150 004105 LDA 0,K2000
28 00151 001775 LDA 0,KEGGS*2
29 00152 020070 MOV 0,K3,SZR
30 00153 034050 LDA 0,K2000
31 00154 175004 LDA 1,MSIZE
32 00155 020077 SUB 0,K1
33 00156 024004 STA 1,MSIZE
34 00157 100400 LDA 0,K400
35 00160 044004 ADD 0,K1
36 00161 020072 INC 1,K1
37 00162 107000 MOV 0,K3,SZR
38 00163 125000 STA 1,ICAT
39 00164 175004 CALL
40 00165 044170 CALL
41 00166 004105 FIRST
42 00167 000500 MOVE
43
44 00170 000320 ICAT:
45
46 00171 054006 ERR: STA 0,KERRET
47 00172 010061 ISZ ERRET
48 00173 034234 LDA 0,ESKIT
49 00174 175004 MOV 0,K3,SZR
50 00175 002050 JMP 0,KERRET
51 00176 050243 STA 0,SAV2
52 00177 044242 STA 1,SAV1
53 00200 040241 STA 0,SAV0
54
55 00201 004105 EKRI: CALL
56 00202 001712 MESS
57 00203 002030 HEADK
58 00204 034050 LDA 0,KERRET
59 00205 054234 STA 0,ESKIT

```

```

0000 MAIN
01 00206 014234
02 00207 024055
03 00210 130000
04 00211 054235
05 00212 020004
06 00213 040057
07 00214 020067
08 00215 040022

```

```

USZ ESKIT
LDA 1,KELUC
ADC 1,K3
STA 0,1STPC
LDA 0,KMB
STA 0,ERTEM
LDA 0,KDATA
STA 0,1Z2

```

```

;FIND THE SIZE OF MEMORY.

```

```

;IF PROGRAM STOPPED IN
;MIDDLE OF MOVE,CANT
;BE RESTARTED... RELOAD IT.

```

```

;PRINT HIGHEST MEMORY LOC.

```

```

;ERROM SUBROUTINE.
;INCREMENT ERROR COUNTER.

```

```

;EXIT NOT FIRST ERROR.

```

```

;THE ABSOLUTE PC SETTING.
;PC FROM THE LISTING.

```

```

A 0007 .MAIN
41
42 00210 020022
43 00217 004105
44 00220 001775
45 00221 010057
46 00222 000210
47 00223 000241
48 00224 024242
49 00225 000243
50 00226 034040
51 00227 175005
52 00230 000205
53 00231 005404
54 00232 000207
55 00233 001400
56 00234 000000
57 00235 000000
58 00236 000000
59 00237 000000
60 00240 000000
61 00241 000000
62 00242 000000
63 00243 000000
64 00244 004000
65 00245 034070
66 00246 054063
67 00247 170400
68 00250 004234
69 00251 004061
70 00252 000000
71 00253 004002
72 00254 034234
73 00255 175004
74 00256 000202
75 00257 014063
76 00260 000313
77 00267 030070
78 00270 000063
79 00271 070477
80 00272 153120
81 00273 151113
82 00274 000300

LUA 1,PSZ
CALL
PUCLAL
LSZ ERLEN
JMP 0,4
LUA 0,SAV0
LUA 1,SAV1
LUA 2,SAV2
LUA 3,ELG5
MOV 0,3,SNR
JMP PERRET
LUA 0,4,0
LOKST
JMP 0,3

FERROR SWITCH/ABS PC OF ERR0R.
FMEKE ERR0R MAY BE FOUND IN LISTING.

FORIGIONAL AND CURRENT AL VALUES.

INITIALIZE EACH TEST.
FSET ITERATION COUNTER.
FRESET ERR0R SWITCH
FAND ERR0R COUNTER.
FRETURN.
FEND OF EACH ITERATION
F0F TEST PROG.
FND ERR0R.
FITERATE TEST
FEXIT TO NEXT TEST.
FERR0R PRESENT

FNOT 100 TIMES YET.
FRESET COUNT.
FLOOK AT SWITCH 2.

LUA 1,PSZ
CALL
MESS
CKLF
LUA 1,ERRCT
CALL
PDECI
CALL
MESS
PUENT
LUA 0,0
STA 0,ERRCT
LUA 0,SAV0
LUA 1,SAV1
LUA 2,SAV2
LUA 0,0
STA 0,ERRCT
FRESTORE ACS.
FMESSAGE "2"
FRESET ERR0R COUNTER
FRESTORE ACS.
IF SWITCH 0(1)
FPROCEED, OTHERWISE
FITERATE THIS TEST.
FREL0CATE THIS PROGRAM.
FAC0 = RANDOM LOCATI0N.
F(CMSIZE)=TOP OF USEABLE MEMORY.
F(CAC2) WILL BE WITHIN THE RANGE
FOF 0 TO TOP OF MEMORY.
F(AC2-AC3)=TRIAL TOP AND
FBOTTEM OF PROGRAM DESTINATION.
F(C(MOVTEM)=NUMBER OF WORDS
FIN THE PROGRAM.
FIF DESTINATION BOTTEM IS
F<5000, GET ANOTHER NUMBER.
FAC2=DESTINATION TOP
FAC3= DESTINATION BOTTEM.
FAC0=1 = LOCATI0N OF THE
FPROGRAM NOW.
FUPDATA FOR NEW DESTINATION.
MOV 2,3
LDA 1,PSIZE
STA 1,MOVTEM
ADD 1,3
LDA 0,K500
SUBL# 0,3,SZC
JMP MOVE
ADC 0,0
STA 0,NUKES
LDA 0,SHIGH
LDA 1,SLOW
STA 2,SHIGH
STA 3,SLOW
SUBZ# 2,0,SNR
JMP MOVE4
CYC2:
01 00275 004105
02 00276 001712
03 00277 002070
04 00278 002070
05 00300 024001
06 00301 004105
07 00302 001745
08 00303 004105
09 00304 001712
10 00305 002072
11
12 00300 102400
13 00307 040001
14 00310 020241
15 00311 020242
16 00312 002074
17
18 00313 074477
19 00314 175113
20 00315 002000
21 00316 002002
22
23 00317 000170
24 00320 004375
25 00321 111250
26 00322 024004
27 00323 132423
28 00324 133001
29 00325 000323
30
31 00326 155000
32 00327 024071
33 00330 044065
34 00331 137000
35 00332 020101
36 00333 110512
37 00334 000320
38 00335 102000
39 00336 040103
40
41 00337 020006
42 00340 024007
43 00341 050006
44 00342 050007
45 00343 142433
46 00344 000300

FPRINT VALUES.
FERROR SWITCH/ABS PC OF ERR0R.
FMEKE ERR0R MAY BE FOUND IN LISTING.

FORIGIONAL AND CURRENT AL VALUES.

INITIALIZE EACH TEST.
FSET ITERATION COUNTER.
FRESET ERR0R SWITCH
FAND ERR0R COUNTER.
FRETURN.
FEND OF EACH ITERATION
F0F TEST PROG.
FND ERR0R.
FITERATE TEST
FEXIT TO NEXT TEST.
FERR0R PRESENT

FNOT 100 TIMES YET.
FRESET COUNT.
FLOOK AT SWITCH 2.

LUA 1,PSZ
CALL
MESS
CKLF
LUA 1,ERRCT
CALL
PDECI
CALL
MESS
PUENT
LUA 0,0
STA 0,ERRCT
LUA 0,SAV0
LUA 1,SAV1
LUA 2,SAV2
LUA 0,0
STA 0,ERRCT
FRESTORE ACS.
FMESSAGE "2"
FRESET ERR0R COUNTER
FRESTORE ACS.
IF SWITCH 0(1)
FPROCEED, OTHERWISE
FITERATE THIS TEST.
FREL0CATE THIS PROGRAM.
FAC0 = RANDOM LOCATI0N.
F(CMSIZE)=TOP OF USEABLE MEMORY.
F(CAC2) WILL BE WITHIN THE RANGE
FOF 0 TO TOP OF MEMORY.
F(AC2-AC3)=TRIAL TOP AND
FBOTTEM OF PROGRAM DESTINATION.
F(C(MOVTEM)=NUMBER OF WORDS
FIN THE PROGRAM.
FIF DESTINATION BOTTEM IS
F<5000, GET ANOTHER NUMBER.
FAC2=DESTINATION TOP
FAC3= DESTINATION BOTTEM.
FAC0=1 = LOCATI0N OF THE
FPROGRAM NOW.
FUPDATA FOR NEW DESTINATION.
MOV 2,3
LDA 1,PSIZE
STA 1,MOVTEM
ADD 1,3
LDA 0,K500
SUBL# 0,3,SZC
JMP MOVE
ADC 0,0
STA 0,NUKES
LDA 0,SHIGH
LDA 1,SLOW
STA 2,SHIGH
STA 3,SLOW
SUBZ# 2,0,SNR
JMP MOVE4
CYC2:
01 00275 004105
02 00276 001712
03 00277 002070
04 00278 002070
05 00300 024001
06 00301 004105
07 00302 001745
08 00303 004105
09 00304 001712
10 00305 002072
11
12 00300 102400
13 00307 040001
14 00310 020241
15 00311 020242
16 00312 002074
17
18 00313 074477
19 00314 175113
20 00315 002000
21 00316 002002
22
23 00317 000170
24 00320 004375
25 00321 111250
26 00322 024004
27 00323 132423
28 00324 133001
29 00325 000323
30
31 00326 155000
32 00327 024071
33 00330 044065
34 00331 137000
35 00332 020101
36 00333 110512
37 00334 000320
38 00335 102000
39 00336 040103
40
41 00337 020006
42 00340 024007
43 00341 050006
44 00342 050007
45 00343 142433
46 00344 000300

```

```

A 0009 *MAIN
01
02 00345 044020 MOVE3: STA 1,20
03 00346 014020 USZ 20
04 00347 054021 STA 3,21
05 00350 014021 USZ 21
06 00351 022021 LDA 0,020
07 00352 042021 STA 0,021
08 00353 010025 ISZ MOVTEM
09 00354 000351 JMP *-3
10 00355 000366 JMP MOVEM
11
12 00350 040030 MOVE4: STA 0,30
13 00357 010030 ISZ 30
14 00360 050031 STA 2,31
15 00361 010031 ISZ 31
16 00362 022030 LDA 0,030
17 00363 042031 STA 0,031
18 00364 010035 ISZ MOVTEM
19 00365 000362 JMP *-3
20
21 00360 130400 MOVE5: SUB 1,3
22 00367 030035 LDA 2,RELDC
23 00370 170000 ADD 3,2
24 00371 050035 STA 2,RELOC
25 00372 050103 STA 2,INDRES
26 00373 004100 CALL
27 00374 000301 FIRST*1
28
29 00375 054000 RAND: STA 3,RANDET
30 00376 032034 LDA 3,ESNIT
31 00377 170024 MOVZ 3,3,5ZK
32 00400 000414 JSK RANU
33 00401 004417 DAC1
34 00402 000237 STA 0,UAC0
35 00403 040236 JSR RANU
36 00404 004414 DAC0
37 00405 000236 LDA 1,UAC1
38 00406 024237 NEG8 1,2
39 00407 130700 AUC 0,2
40 00410 112000 STA 2,UAC2
41 00411 050240 NIOP 0
42 00412 060300 JMP RANDET
43 00413 002065
44
45 00414 020230 RAND1: LDA 0,UAC0
46 00415 024237 LDA 1,UAC1
47 00416 030240 LDA 2,UAC2
48 00417 000773 JMP RAND1-2
49
50 004171 ERROR=JSR ERR
51 004244 SETUP=JSR ENTER
52 004253 LOOP=JSK CYCLE
53 004375 RANLUM=JSR RANU
54 000300 SYNCENLUP 0

```

```

A 0010 *MAIN
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

```

```

; RANDUM NUMBER GENERATOR
; GENERATES A (PSEUDO) RANDUM SEQUENCE OF INTEGERS
; IN THE RANGE 0 <= N <= 2**10-1
; INPUT: ADDRESS OF OLD VALUE POINTED TO BY MORE
; AFTER JSR
; OUTPUT: 10-BIT NEW RANDUM NUMBER IN AC0
; AND IN STORAGE REPLACING OLD VALUE
; CALLING SEQUENCE:
; JSR *RAND
; ADDRESS OF OLD VALUE
; RETURN
; METHOD: GENERATES A LINEAR CONGRUENTIAL
; SEQUENCE OF THE FORM:
; X(N+1) = (X(N)*A+C)MOD 2**10
; CAUTION: IF A K-BIT RANDUM NUMBER (AS OPPOSED
; TO A 10-BIT NUMBER) IS NEEDED,
; USE THE MOST SIGNIFICANT
; K-BITS (THE LEAST SIGNIFICANT K-BITS
; ARE NOT AS RANDUM).
; FOR EXAMPLE, TO OBTAIN RANDUM N MOD 2,
; USE THE SIGN BIT OF THE RESULT
; UNCHANGED: AC1, AC2
; DESTROYED: AC0, ACS, CARRY
; RAND: STA 3,UD03
; SAVE RETURN
; ISZ UD03 ; BUMP PAST ADDRESS CONSTANT
; STA 1,UD01 ; *SAVE AC1
; STA 2,UD02 ; *SAVE AC2
; LDA 2,0,3 ; GET ADDRESS OF OLD VALUE
; LDA 0,0,2 ; OLD VALUE TO AC0
; JSR UD050 ; N=A
; LDA 3,UD020 ; GET INCREMENT, C
; ADD 3,0 ; (N*A+C) MOD 2**16
; STA 0,0,2 ; STORE IT
; LDA 1,UD01 ; *RESTORE AC1
; LDA 2,UD02 ; *RESTORE AC2
; JMP 0,UD03 ; RETURN

```

```

; DESTINATION ADDRESS < SOURCE
; ADDRESS.
; IC(20) BOTTOM OF SOURCE.
; IC(21) BOTTOM OF DESTINATION.
; MOVE THE PROGRAM FROM
; FROMTEM TO TOP.
; PROGRAM IS MOVED.
; DESTINATION ADDRESS > SOURCE
; ADDRESS.
; IC(30) TOP OF SOURCE.
; IC(31) TOP OF DESTINATION.
; MOVE FROM TOP TO BUTTEM.
; ADDRESS DIFFERENCE BETWEEN
; OLD AND NEW PROGRAMS.
; ADDRESS DIFFERENCE IS USED TO
; ADJUST RELOCATION.
; RESTART THE PROGRAM.
; THE RANDUM NUMBER.
; SYNC A SCOPE.

```

```

A 0011 .MAIN
01
02
03
04
05
06
07 00435 024420 .UD50: LIA I,0021 ; GET COUNT, X
08 00436 044415 STA I,0010 ; FOK ITERATION
09 00437 105120 MOVZL P,1 ; N*2*(X*1)
10 00440 120120 MOVZL I,1
11 00441 014412 USZ ,UD10
12 00442 006770 JMP ,*2
13 00443 107000 ADD P,1
14 00444 125120 MOVZL I,1
15 00445 125120 MOVZL I,1
16 00446 125000 ADD I,K
17 00447 001400 JMP P,3
18
19
20
21
22 00450 000000 .UD61: 0 ; *SAVE AC1
23 00451 000000 .UD62: 0 ; *SAVE AC2
24 00452 000000 .UD63: 0 ; SAVE RETURN
25
26 00453 000000 .UD10: 0 ; ITERATION COUNT
27
28 00454 033031 .UD20: 33031 ; INCREMENT
29 00455 000010 .UD21: 10 ; ITERATION COUNT, X
30
A 0012 .MAIN
01
02
03
04 00500 101000 .LUC Deb4 ;MOV E,P
05
06 00501 004244 MS1: SETUP
07 00502 004375 RANDOM
08 00503 104700 NEG8 0,1
09 00504 130342 CUMUS I,2,SZC
10 00505 142014 AUCR 2,0,SZR
11 00506 004171 ERROR
12 00507 004253 LOOP
13
14 00510 004244 MS2: SETUP
15 00511 004375 RANDOM
16 00512 111700 INCS 0,2
17 00513 145323 MOVZS 2,1,SNC
18 00514 106314 ADLS# 0,1,SZR
19 00515 004171 ERROR
20 00516 004253 LOOP
21
22 00517 004244 MS3: SETUP
23 00520 102400 SUB 0,0
24 00521 100044 NEGOR 0,0,SZR
25 00522 004171 ERROR
26 00523 100004 NEGCR 0,0,SZR
27 00524 004171 ERROR
28 00525 100064 NEGCR 0,0,SZR
29 00526 004171 ERROR
30 00527 004253 LOOP
31
32
33 00530 004244 ROT1: SETUP
34 00531 004375 RANDOM
35 00532 105000 MOV 0,1
36 00533 101100 MOVL 0,0
37 00534 111100 MOVL 0,2
38 00535 151100 MOVL 2,2
39 00536 151100 MOVL 2,2
40 00537 155100 MOVL 2,3
41 00540 175100 MOVL 3,3
42 00541 175100 MOVL 3,3
43 00542 101100 MOVL 3,0
44 00543 101100 MOVL 0,0
45 00544 101100 MOVL 0,0
46 00545 101100 MOVL 0,0
47 00546 101100 MOVL 0,0
48 00547 101100 MOVL 0,0
49 00550 101100 MOVL 0,0
50 00551 101100 MOVL 0,0
51 00552 101100 MOVL 0,0
52 00553 101100 MOVL 0,0
53 00554 100414 SUB# 0,1,SZR
54 00555 004171 ERROR
55 00556 004253 LOOP

```



```

A 0015 .MAIN
01
02 00701 004244 X3:
03 00702 004375
04 00703 130400
05 00704 004105
06 00705 001040
07 00706 004105
08 00707 001040
09 00710 110014
10 00711 004171
11 00712 004253
12
13 00713 004244 X4:
14 00714 004375
15 00715 126400
16 00716 004105
17 00717 001040
18 00720 106414
19 00721 004171
20 00722 004253
21
22 00723 004244 X5:
23 00724 004375
24 00725 102400
25 00726 004105
26 00727 001040
27 00730 106414
28 00731 004171
29 00732 004253
30
31 00733 004244 X6:
32 00734 004375
33 00735 102400
34 00736 004105
35 00737 001040
36 00740 110000
37 00741 106414
38 00742 004171
39 00743 004253

;SAVE C(AC1) NEGATED IN
;C(AC2). EXCLUSIVE OR L(AC0) TO
;C(AC1). EXCLUSIVE OR THE
;RESULT BACK TO AC0.
;CHECK VIA ADDITION TO
;COMPLIMENT OF ORIGINAL
;NUMBER.
SETUP
RANDOM
NEG 1,2
CALL
XOR
CALL
XOR1
ADDR 0,2,5ZR
ERRUK
LOOP

;EXCLUSIVE OR C(AC0) TO
;ALL ZEROS IN C(AC1).
SETUP
RANDOM
SUB 1,1
CALL
XOR
SUB# 0,1,5ZR
ERRUK
LOOP

;EXCLUSIVE OR C(AC1) TO
;ALL ZEROS IN C(AC0).
SETUP
RANDOM
SUB 0,0
CALL
XOR1
SUB# 0,1,5ZR
ERRUK
LOOP

;EXCLUSIVE OR C(AC1) TO
;(-1) IN C(AC0).
;THE COMPLIMENT OF
;THIS RESULT SHOULD
;EQUAL C(AC1).
SETUP
RANDOM
ADC 0,0
CALL
XOR1
COM 0,2
SUB# 2,1,5ZR
ERRUK
LOOP

A 0016 .MAIN
01
02 00744 004244 X7:
03 00745 004375
04 00746 105000
05 00747 004105
06 00750 001054
07 00751 151004
08 00752 004171
09 00753 004253
10
11 00754 004244 X8:
12 00755 004375
13 00756 120000
14 00757 004105
15 00760 001054
16 00761 150014
17 00762 004171
18 00763 004253
19
20 00764 004244 X9:
21 00765 102300
22 00766 120000
23 00767 004105
24 00770 001040
25 00771 127704
26 00772 004171
27 00773 004253
28
29 00774 004244 X10:
30 00775 102700
31 00776 126400
32 00777 004105
33 01000 001040
34 01001 107314
35 01002 004171
36 01003 004253

;C(AC1) IS SET EQUAL TO
;C(AC0). L(AC0) AND C(AC1)
;ARE EXCLUSIVE OREL WITH
;THE RESULT GOING TO AC2.
SETUP
RANDOM
MOV 0,1
CALL
XOR2
MOV 2,2,5ZR
ERRUK
LOOP

;C(AC0) IS SET TO THE
;COMPLIMENT OF C(AC1). THE
;RESULT OF A EXCLUSIVE OR
;SHOULD BE ALL BITS
;SET (-1).
SETUP
RANDOM
COM 1,0
CALL
XOR2
COM# 2,2,5ZR
ERRUK
LOOP

;EXCLUSIVE OR ALL ONES
;TO ALL ONES. THE
;RESULT SHOULD BE
;ALL ZEROS.
SETUP
ADC 0,0
ADC 1,1
CALL
XOR
ANDS 1,1,5ZR
ERRUK
LOOP

;EXCLUSIVE OR ALL ZEROS
;TO ALL ZEROS. THE
;RESULT SHOULD BE
;ALL ZEROS IN C(AC1).
SETUP
SUBS 0,0
SUB 1,1
CALL
XOR
ADD# 0,1,5ZR
ERRUK
LOOP

```

A 0017 .MAIN

```
01 02 01004 004244 X11:  
03 01005 004375  
04 01006 004100  
05 01007 001004  
06 01010 004105  
07 01011 001040  
08 01012 004105  
09 01013 001040  
10 01014 004105  
11 01015 001040  
12 01016 004105  
13 01017 001040  
14 01020 004105  
15 01021 001040  
16 01022 004105  
17 01023 001040  
18 01024 004105  
19 01025 001040  
20 01026 132414  
21 01027 004171  
22 01030 004253
```

```
SETUP  
RANDOM  
CALL  
XORZ  
CALL  
XOR  
CALL  
XOR  
CALL  
XOR  
SUB# 1,2,SZR  
ERROR  
LOOP
```

```
!FROM EXCLUSIVE OF FUNCTION  
!IN C(AC2).  
!CALL ANOTHER EXCLUSIVE  
!FOR FUNCTION SEVEN TIMES.  
!THE RESULT SHOULD BE THE  
!SAME AS THE FIRST XOR
```

A1:

```
24 01031 004244 A1:  
25 01032 004375  
26 01033 145000  
27 01034 112400  
28 01035 113000  
29 01036 140414  
30 01037 004171  
31 01040 004253
```

```
!SAVE C(AC2) IN C(AC1).  
!ADD AND ADD C(AC0) TO  
!C(AC2). THE VALUE IN  
!AC2 SHOULD NOT BE  
!CHANGED.
```

```
SETUP  
RANDOM  
MOV 2,1  
SUB 0,2  
ADD 0,2  
SUB# 2,1,SZR  
ERROR  
LOOP
```

A2:

```
33 01041 004244 A2:  
34 01042 004375  
35 01043 102000  
36 01044 123000  
37 01045 111400  
38 01046 140414  
39 01047 004171  
40 01050 004253
```

```
SETUP  
RANDOM  
ADC 0,0  
ADD 1,0  
INC 0,2  
SUB# 2,1,SZR  
ERROR  
LOOP
```

A 0018 .MAIN

```
01 02 01051 004244 A3:  
03 01052 004375  
04 01053 131400  
05 01054 102000  
06 01055 143000  
07 01056 106414  
08 01057 004171  
09 01058 004253  
10
```

```
!INCREMENT THE VALUE IN  
!AC1 AND ADD THAT VALUE  
!TO (-1). THE RESULT  
!SHOULD BE THE ORIGINAL  
!NUMBER.
```

A4:

```
11 01061 004244 A4:  
12 01062 004375  
13 01063 121020  
14 01064 127200  
15 01065 127200  
16 01066 127200  
17 01067 127200  
18 01068 127200  
19 01071 127200  
20 01072 127200  
21 01073 127200  
22 01074 127200  
23 01075 127200  
24 01076 127200  
25 01077 127200  
26 01100 127200  
27 01101 127200  
28 01102 106454  
29 01103 004171  
30 01104 004253
```

```
!SAVE THE C(AC1) IN C(AC0).  
!A "ADDR" INSTRUCTION SHOULD  
!NOT CHANGE THE VALUE OF  
!THE AC.
```

A5:

```
32 01105 004244 A5:  
33 01106 004375  
34 01107 115400  
35 01108 120000  
36 01111 137000  
37 01112 175400  
38 01113 137000  
39 01114 171400  
40 01115 133000  
41 01116 151400  
42 01117 133000  
43 01120 142414  
44 01121 004171  
45 01122 004253
```

```
SETUP  
RANDOM  
INC 0,3  
ADC 1,1  
ADD 1,3  
INC 3,3  
ADD 1,2  
INC 2,2  
SUB# 2,0,SZR  
ERROR  
LOOP
```

```
!THE RANDOM NUMBER IN  
!C(AC0) IS INCREMENTED VIA "INC"  
!AND DECREMENTED VIA "ADD".  
!THE FINAL RESULT IN C(AC2)  
!SHOULD BE EQUAL TO THE  
!ORIGINAL NUMBER IN C(AC0).
```

```

A 0019 .MAIN
01
02 01123 004244 AB:
03 01124 004375
04 01125 135000
05 01126 117000
06 01127 054421
07 01130 131000
08 01131 004105
09 01132 001546
10 01133 143524
11 01134 000774
12 01135 020413
13 01136 122414
14 01137 004171
15 01140 004253
16
17 01141 004244 A7:
18 01142 004375
19 01143 110440
20 01144 143204
21 01145 004171
22 01146 004253
23
24 01147 101001
25 01150 177777 ADDTEM: -1
26
27 01151 004244 AB:
28 01152 004375
29 01153 152520
30 01154 151300
31 01155 150400
32 01156 143700
33 01157 105320
34 01160 103100
35 01161 103100
36 01162 103100
37 01163 103100
38 01164 100414
39 01165 004171
40 01166 004253
41

A 0020 .MAIN
01
02 01167 101001
03 01170 177400 M400:
04 01171 004244 AK1:
05 01172 004375
06 01173 030775
07 01174 133700
08 01175 145000
09 01176 004105
10 01177 001011
11 01178 001011
12 01200 121000
13 01201 004105
14 01202 001067
15 01203 112414
16 01204 004171
17 01205 004253
18
19 01206 004244 AR2:
20 01207 004375
21 01210 004105
22 01211 001067
23 01212 105000
24 01213 131000
25 01214 004105
26 01215 001011
27 01216 121000
28 01217 004105
29 01220 001076
30 01221 112714
31 01222 004171
32 01223 004253
33
34 01224 004244 AR3:
35 01225 004375
36 01226 141000
37 01227 004105
38 01230 001067
39 01231 105000
40 01232 141000
41 01233 004105
42 01234 001076
43 01235 106714
44 01236 004171
45 01237 004253

;THE SUM OF AC0=1 IS
;CHECKED WITH THE SIMULATED
;SUM.
;SUM VIA ADD INSTRUCTION.
;SIMULATE THE ADD VIA
;EXCLUSIVE OR, C(AC2) =
;TRIPPLE CARRY,C(AC1)=RESULT.
;ADDITION OF NEGATED
;NUMBERS SHOULD PRODUCE
;ZERO AND A CARRY.
;ADD TEST.
; (= (+1)
; (= (400)
; (= (177400)
;SAVE HIGH ORDER 8 BITS.
;SAME 8 BITS TO C(AC1)L.
;MOVE C(AC0) LEFT VIA
;ADD SHIFT.

;TAKE THE SQUARE ROOT
;OF A NUMBER. THE SQUARE
;ROOT OF THE RESULT SQUARED
;SHOULD BE THE SAME AS THE
;ORIGINAL ROOT.
;SQUARED NOW TAKE ROOT.
;FIND SQUARE ROOT VIA
;DIFFERENT SUBROUTINES.
;SAVE FIRST RESULT IN AC1
;RESULT IN AC0.

MOV 0,0,SKP
-400
SETUP
RANDOM
LDA 2,M400
ANDS 1,2
MOV 2,1
CALL
.MPYU
MOV 1,0
CALL
SUB# 0,2,SRZ
ERROR
LOOP

SETUP
RANDOM
CALL
MOV# 0,1
MOV# 1,2
CALL
.MPYU
MOV 1,0
CALL
SUB# 0,2,SRZ
ERROR
LOOP

SETUP
RANDOM
MOV 2,0
CALL
MOV 0,1
MOV 2,0
CALL
SUBS# 0,1,SRZ
ERROR
LOOP

```

```

A 0021 .MAIN
01
02 01240 004244 AR4:
03 01241 004375
04 01242 142432
05 01243 000776
06 01244 004165
07 01245 001026
08 01246 004105
09 01247 001012
10 01250 034240
11 01251 156414
12 01252 004171
13 01253 034237
14 01254 130714
15 01255 004171
16 01256 034236
17 01257 116714
18 01258 004171
19 01259 004253
20
21 01262 004244 AR5:
22 01263 152440
23 01264 151504
24 01265 151504
25 01266 151304
26 01267 151504
27 01270 151507
28 01271 151507
29 01272 151507
30 01273 145707
31 01274 151407
32 01275 151400
33 01276 133014
34 01277 004171
35 01300 004253
36
37 01301 004244 AR6:
38 01302 120420
39 01303 125600
40 01304 125600
41 01305 125600
42 01306 125600
43 01307 125600
44 01310 125600
45 01311 125600
46 01312 125600
47 01313 130304
48 01314 132714
49 01315 004171
50 01316 004253

;THE CONTENTS OF AC2 IS
;DIVIDED INTO AC0-1.
;THIS RESULT MULTIPLIED
;BY AC2 SHOULD PRODUCE
;THE ORIGINAL NUMBERS.

;AC2 CHANGED?
;AC1 CHANGED.
;AC0 CHANGED.

;USE INC TO FORM
;THE NUMBER 177400
;IN AC1 AND 400 IN AC2.

;THE "SBN/SZR" SHOULD
;NOT CAUSE A SKIP.

;USE THE INCR INSTRUCTION
;TO FORM THE NUMBER
;1177400. THIS NUMBER
;IS COMPLETED AND SWAPPED
;SHOULD BE THE SAME
;NUMBER.
;THE "SEZ" SHOULD NEVER
;CAUSE A SKIP.

;NEGATIVE OF ZERO
;SHOULD PRODUCE ZERO
;AND A CARRY.

A 0022 .MAIN
01
02 01317 004244 AR7:
03 01320 102025
04 01321 004171
05 01322 104400
06 01323 004171
07 01324 124402
08 01325 004171
09 01326 124403
10 01327 004171
11 01330 124407
12 01331 004171
13 01332 124465
14 01333 004171
15 01334 124465
16 01335 004171
17 01336 124407
18 01337 004171
19 01340 124407
20 01341 004171
21 01342 106414
22 01343 004171
23 01344 004253
24
25 01345 004244 AR8:
26 01346 102440
27 01347 100762
28 01350 004171
29 01351 100762
30 01352 004171
31 01353 100706
32 01354 004171
33 01355 100706
34 01356 004171
35 01357 100544
36 01360 004171
37 01361 100403
38 01362 004171
39 01363 100644
40 01364 004171
41 01365 104704
42 01366 004171
43 01367 130704
44 01370 004171
45 01371 004253

;NEGATIVE OF ZERO
;SHOULD PRODUCE ZERO
;AND A CARRY.

```

```

A 0023 .MAIN
01
02 01372 004244 AK9:
03 01373 004375
04 01374 102400
05 01375 004100
06 01376 001612
07 01377 004236
08 01400 106414
09 01401 004171
10 01402 004253
11 01403 004244 AK10:
12 01404 102300
13 01405 105705
14 01406 130304
15 01407 145705
16 01408 124346
17 01411 131707
18 01412 150304
19 01413 131704
20 01414 004171
21 01415 004253
22 01416 004244 AK11:
23
24 01416 004244
25 01417 004375
26 01420 110400
27 01421 104000
28 01422 125400
29 01423 132414
30 01424 004171
31 01425 004253
32
33 01426 004244 AK12:
34 01427 004375
35 01430 110100
36 01431 144200
37 01432 106714
38 01433 004171
39 01434 004253

: C(AC1)*0+(ACC) SHOULD
: PLACE ACC IN AC1. SEE
: SYSTEM REFERENCE MANUAL
: FOR FURTHER INFORMATION.

*MPYA
LDA 3,AC1
SUB# 3,1,SZR
ERROR
LOOP

:TEST "COM/INC" SWAPPED.

SETUP
ADCS 0,0
INCS 0,1,SNR
COMS 1,2,SZR
INCS 2,1,SNR
COMOS 1,1,SEZ
INCS 1,2,SNR
COMS 2,2,SZR
INCS 2,2,SZR
ERROR
LOOP

SETUP
RANDOM
NEG 0,2
COM 0,1
INC 1,1
SUB# 1,2,SZR
ERROR
LOOP

: CUMPLIMENT AND INCREMENT
: SHOULD BE THE SAME AS
: NEGATE

SETUP
RANDOM
NEG 0,2
COM 0,1
INC 1,1
SUB# 1,2,SZR
ERROR
LOOP

:TEST COM LEFT AND RIGHT.

SETUP
RANDOM
COML 0,2
COMCK 2,1
SUBS# 0,1,SZR
ERROR
LOOP

A 0024 .MAIN
01
02 01435 004244 AK13:
03 01436 105820
04 01437 170600
05 01440 170000
06 01441 163704
07 01442 002775
08 01443 102000
09 01444 170620
10 01445 125102
11 01446 101400
12 01447 173224
13 01450 000775
14 01451 115200
15 01452 103012
16 01453 004171
17 01454 004253
18
19 01455 004244 AK15:
20 01456 004375
21 01457 131000
22 01460 105300
23 01461 127600
24 01462 107000
25 01463 105300
26 01464 127000
27 01465 107000
28 01466 132414
29 01467 004171
30 01470 004253
31
32 01471 004244 AK16:
33 01472 004375
34 01473 131000
35 01474 113000
36 01475 107400
37 01476 151100
38 01477 104414
39 01500 004171
40 01501 004253

:FROM THE PARITY OF
: C(ACC) IN DIFFERENT
: ROUTINES, CHECK THAT
: THE RESULTS ARE EQUAL.

:SAVE PARITY IN BIT 0.

BIT 15 WILL CONTAIN
:THE PARITY.

:CHECK TO SEE IF BITS
:ARE LIKE.

: MISC TEST OF SUB LEFT
: AND RIGHT.

: MISC TEST OF AND RIGHT.

SETUP
MOVZ 0,1
ADC 3,3
AND 0,3
ANDS 3,0,SZR
JMP *-3
SUBCH 0,0
SUBZK 3,3
MOVL 1,1,SZC
INC 0,0
MOVZK 3,3,SZK
JMP *-3
MOVZ 0,3
ADD# 0,0,SZC
ERROR
LOOP

SETUP
RANDOM
MOV 1,2
SUBL 0,1
ANDR 1,1
ADD 0,1
SUBL 0,1
ANDR 1,1
ADD 0,1
SUB# 1,2,SZR
ERROR
LOOP

SETUP
RANDOM
MOV 1,2
ANDR 0,2
AND 0,1
MOVL 2,2
SUB# 2,1,SZR
ERROR
LOOP

```

```

A 0025 .MAIN
01
02
03 01502 004244 AR17:
04 01503 004375
05 01504 131000
06 01505 113000
07 01506 107300
08 01507 125300
09 01510 132414
10 01511 004171
11 01512 004253
12 01513 004244 AR18:
13 01514 004375
14 01515 131000
15 01516 112120
16 01517 100000
17 01520 107120
18 01521 132414
19 01522 004171
20 01523 004253
21
22 01524 004244 AR19:
23 01525 004375
24 01526 131000
25 01527 112220
26 01530 100000
27 01531 107220
28 01532 132414
29 01533 004171
30 01534 004253
31
32 01535 004244 AR20:
33 01536 004375
34 01537 131000
35 01540 100000
36 01541 100000
37 01542 132000
38 01543 132414
39 01544 004171
40 01545 004253

MISC ADD SWAPPED TEST.
01
02
03 01546 004244 ARJSM:
04 01547 004401
05 01550 171122
06 01551 004171
07 01552 151240
08 01553 003004
09 01554 004401
10 01555 165000
11 01556 125112
12 01557 004171
13 01558 004253
14
15 01561 024846 AREND:
16 01562 125304
17 01563 000411
18 01564 010104
19 01565 020104
20 01566 024102
21 01567 107414
22 01570 000404
23 01571 004105
24 01572 001712
25 01573 002113
26
27 01574 024846 ENDER:
28 01575 125005
29 01576 000410
30 01577 014051
31 01580 000406
32 01581 034052
33 01582 020051
34 01583 062677
35 01584 041776
36 01585 001400
37 01586 010050
38 01587 000320
39 01588 000317

!TEST THAT INDEX WITH
!SIGN BIT SET DOES
!NOT LOAD INTO PC ON JSR.
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

!GO TO NEXT LOCATION
JSR *+1
MOV 3,1
MOV 3,2
MOV 3,2
JSR *+1
JSR *+1
MOV 3,1
MOVL# 1,1,SZC
ERROR
LOOP

!CHECK ADC LEFT.
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

!TEST ADC RIGHT.
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

!TEST SUB RIGHT.
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

LDA 1,EGGS
MOV 1,1,SNR
JMP *0
DSZ EGGS+J
JMP *0
LDA 3,EGGS+4
LDA 0,EGGS+3
IORST
STA 0,-2,3
JMP 0,3
ISZ EGGS+2
JMP MOVE
JMP MOVE-1

```

```

A 0027 .MAIN
01
02 01067 126020 SURT:
03 01070 135120
04 01071 122422
05 01072 157401
06 01073 121921
07 01074 000775
08 01075 002053
09
10 01076 176400 SG:
11 01077 054411
12 01700 102023
13 01701 000405
14 01702 010406
15 01703 010405
16 01704 034404
17 01705 000773
18 01706 161200 SG1:
19 01707 002053
20
21 01710 000000 SUTEM: 0
22 01711 000000 XORTEM: 0
23
24 01712 021400 MESS1
25 01713 036005
26 01714 113000
27 01715 024073
28 01716 021000 MESS1:
29 01717 125112
30 01720 123401
31 01721 123401
32 01722 151400
33 01723 124000
34 01724 004424
35 01725 000771
36 01726 010053
37 01727 002053
38
39 01730 001730 KTABL:
40 01731 100000
41 01732 010000
42 01733 001000
43 01734 000100
44 01735 000001
45 01736 000001
46 01737 001737 KTAB:
47 01740 023420
48 01741 001750
49 01742 000144
50 01743 000012
51 01744 000001
52
53 01745 020074 PDECI:
54 01746 030771
55 01747 000430
JMP PUCTAL+2

FC(AC1)+C(AC2)
RESULT IN AC1,AC1
SEE SYSTEM REFERENCE
MANUAL FOR FURTHER
INFORMATION.

FC(AC0),C(AC1)/C(AC2).
FAC0=REMAINDER
FAC1=QUOTIENT
SEE SYSTEM REFERENCE
MANUAL.

FEXCLUSIVE OR C(AC0),C(AC1).
RESULT IS IN C(AC1).

FEXCLUSIVE OR C(AC0),C(AC1).
RESULT IS IN C(AC0).

FEXCLUSIVE OR C(AC0),C(AC1).
RESULT IN C(AC2).
THE CONTENTS OF AC0 AND
AC1 ARE EXCHANGED.

SUBC 0,0
STA 3,CB03
LDA 3,CB20
MOVR 1,1,SNC
MOVR 0,0,SKP
ADDR 2,0
INC 3,3,SZR
JMP CB99
MOVR 1,1
JMP #CB03
0
-20
SUB 0,0
STA 3,CB03
LDA 3,CB20
MOVL 1,1
MOVL 0,0
SUB# 2,0,SZC
SUB 2,0
MOVL 1,1
INC 3,3,SZR
JMP #CB03
MOVR 1,3
ANDZL 0,3
ADD 0,1
SUB 3,1
RETURN
MOV 1,3
AND 0,3
CUM 3,3
AND 3,0
ADD 1,0
AND 3,0
RETURN
LDA 3,M20
STA 3,XURTEM
MOV 0,3
ADDR 1,3
MOVR 2,2
MOVR 0,0
MOVR 1,1
ADDR 0,0
ISZ XURTEM
JMP XUR2+2
RETURN
SUBZL 1,1
MOVL 1,3
SUBZL 1,0,SKP
MOVR 1,0,SKP
JMP #3
RETURN
SUB 3,3
STA 3,SUTEM
ADCC 3,0,SNC
JMP S01
ISZ SUTEM
ISZ SUTEM
LDA 3,SUTEM
JMP #5
MOVR 3,0
RETURN
LD 0,0,3
LDA 2,RELOC
ADD 0,2
LDA 1,K377
MOVL 1,1,SZC
ANDS 1,0,SKP
AND 1,0,SKP
INC 2,2
COM 1,1
JSR CMAR
JMP MESS1
ISZ CALRET
JMP #CALRET
DOCTAL TABLE
100000
10000
1000
100
10
1
DECIMAL TABLE
10000.
1000.
100.
10.
1.
DECIMAL PRINT C(1).
LDA 0,K240
LDA 2,KTAB
JMP PUCTAL+2

```



```

A 0029 .MAIN
01
02 01750 054417 CHAK: STA J,CHARM
03 01751 101325 MOVZS J,%,SNK
04 01752 001401 JMP J,3
05 01753 000415 STA J,CHARM
06 01754 170000 ADC J,3
07 01755 117000 ADD J,3
08 01756 103404 AND J,0,SR
09 01757 000775 JMP --J
10 01760 020410 LDA J,CHARM
11 01761 103200 ADDR J,0
12 01762 101300 MOVZS J,0
13 01763 001111 DNAS J,TIU
14 01764 000511 SAPDN TIU
15 01765 000777 JMP --1
16 01766 002401 JMP J,CHARM
17
18 01767 000000 CHARM: 0
19 01770 000000 CHARM: 0
20 01771 000000 ZSUPP: 0
21 01772 000000 OCTRET: 0
22
23 01773 020074 ZUCTAL: LDA J,K240
24 01774 101001 MOV J,%,SKP
25 01775 020075 PUCTAL: LDA J,K60
26 01776 030732 LDA J,KTABL
27 01777 000772 STA J,ZSUPP
28 02000 054772 STA J,OCTRET
29 02001 034055 LDA J,KELUC
30 02002 173000 ADD J,Z
31 02003 050027 STA Z,Z
32 02004 032027 LDA Z,027
33 02005 034764 LDA J,Z2SUPP
34 02006 102400 SUB J,0
35 02007 146512 DECOF: SUBL# 2,1,5ZC
36 02010 000405 JMP DECP
37 02011 100400 LDA J,K60
38 02012 034075 INC J,0
39 02013 101400 JMP DECOF
40 02014 000773 MOVZRH 2,2,SNR
41 02015 151235 DECP: LDA J,K60
42 02016 034075 STA J,0
43 02017 054752 ADD J,0
44 02020 100000 JSR CHAR
45 02021 004727 MOVZR 2,2,SR
46 02022 151224 JMP DECOF-3
47 02023 000751 LDA J,K240
48 02024 020074 JSR CHAR
49 02025 004723 JSR CHAR
50 02026 004722 JMP PUCTRET
51 02027 002743

```

```

A 0030 .MAIN
01
02 02030 005215 HEADM: .TXTE I<215><212>
03 02031 001101 AOS PC LIST PL ORIGINAL AC0,AC1,AL2 RESULT AL0,AC1,AL2 <215>
04 02032 120123
05 02033 141520
06 02034 120240
07 02035 144714
08 02036 152123
09 02037 050240
10 02040 120303
11 02041 151317
12 02042 003711
13 02043 007311
14 02044 140101
15 02045 120240
16 02046 040040
17 02047 030303
18 02050 040054
19 02051 130703
20 02052 040054
21 02053 151303
22 02054 120240
23 02055 142722
24 02056 052523
25 02057 152314
26 02060 040040
27 02061 030303
28 02062 040054
29 02063 150703
30 02064 040054
31 02065 151303
32 02066 100040
33 02067 000012
34
35 02070 005215 CRLF: .TXTE I<215><212>I
36 02071 000000
37
38 02072 122640 PCENT: .TXTE I X I
39 02073 000240
40
41 02074 005215 WORLD: .TXTE I<215><212>
42 02075 040714 LAST LOCATION IN MEMORY IS I
43 02076 152123
44 02077 140240
45 02100 141717
46 02101 152101
47 02102 147711
48 02103 120116
49 02104 047311
50 02105 040040
51 02106 040705
52 02107 151317
53 02110 120131
54 02111 051711
55 02112 000240

```


0035	.MAIN			
.C099	001014	27/03	27/09	
.CL98	001031	27/19	27/24	
.DIV1	001025	27/15		
.DIVU	001025	21/07	27/10	
.MPYA	001012	21/09	23/00	27/03
.MPYU	001011	20/11	20/20	27/02
.RANU	000420	9/33	9/30	10/34
.U001	000430	10/30	10/44	11/22
.U002	000431	10/37	10/45	11/23
.U003	000432	10/34	10/35	10/46
.U010	000433	11/06	11/11	11/26
.U020	000434	10/41	11/28	
.U021	000435	11/07	11/29	
.U050	000435	10/40	11/07	
				11/24