DataGeneral

TECHNICAL STATEMENT

TEXT LISTING

068-001068-01

PROGRAM

ARRAY PROCESSOR EXERCISER-D

TEXT TAPE

097-001068-01

ABSTRACT

THIS PROGRAM IS A COMBINATION DIAGNOSTIC AND EXERCISER FOR THE ARRAY PROCESSOR. THE PROGRAM IS EXECUTED BY A CENTRAL PROCESSOR (OR IOP) CONTROLLING THE ARRAY PROCESSOR (AP).

```
 0001 .MAIN    MACRO REV 06.30          10:10:55 08/29/79
01
02
03
04
05
06
07
08
09
10              ;*************************************************************
11              ;
12              ; NAME: APD.TX                              PART NUMBER: 097-1068
13              ;
14              ;
15              ; DESCRIPTION: ARRAY PROCESSOR EXERCISER-D.
16              ;
17              ;
18              ; REVISION HISTORY:
19              ;
20              ;       REV.            DATE
21              ;
22              ;       00              03/29/78
23              ;       01              12/29/78
24              ;
25              ;
26              ; COPYRIGHT (C) DATA GENERAL CORPORATION, 1978
27              ; ALL RIGHTS RESERVED.
28
29              ;*************************************************************
```

```
!0002 .MAIN
01
02              ;
03              ;               APD.TX          PART NUMBER:094-1469
04              ;
05              ;1.0    PROGRAM NAME
06              ;
07              ;       SOURCE FILE:    APD.SR
08              ;       DTOS FILE:      APD.EXEM
09              ;
10              ;2.0    REVISION HISTORY
11              ;
12              ;       DATE    REVISION
13              ;       ----    --------
14              ;       3/29/78    00
15              ;
16              ;3.0    MACHINE REQUIREMENTS
17              ;
18              ;       1. ECLIPSE FAMILY CENTRAL PROCESSOR (HOST) WITH
19              ;          AT LEAST 16.K READ/WRITE MEMORY
20              ;       2. AP1,AP2,AP3 PROCESSOR BOARDS
21              ;       3. BASIC I/O TELETYPE INTERFACE AND CONTROL
22              ;
23              ;4.0    TEST REQUIREMENTS
24              ;
25              ;       1. SAME AS MACHINE REQUIREMENTS
26              ;       2. I/O TESTER OR DISC
27              ;       3. MMPU1
28              ;
29              ;5.0    SUMMARY
30              ;
31              ;       THIS PROGRAM IS A COMBINATION DIAGNOSTIC AND EXERCISER
32              ;       FOR THE ARRAY PROCESSOR.  THE PROGRAM IS EXECUTED BY
33              ;       A CENTRAL PROCESSOR (OR IOP) CONTROLLING THE ARRAY
34              ;       PROCESSOR (AP).
35              ;
36              ;6.0    RESTRICTIONS
37              ;
38              ;       THE PROGRAM ASSUMES THAT THE HOST COMPUTER AND
39              ;       MEMORY/MAP SYSTEMS ARE WORKING PROPERLY.
40              ;
41              ;       ***THE PROGRAM ALSO ASSUMES THAT THE 2 AP MAINTENANCE
42              ;       DIAGNOSTICS USING THE MIS (MAINTENANCE INSTRUCTION
43              ;       SET) HAVE BEEN SUCCESSFULLY RUN.
44              ;
```

```
!0003 .MAIN
01              ;7.0    PROGRAM DESCRIPTION/THEORY OF OPERATION
02              ;
03              ;       7.1     APXD TEST
04              ;
05              ;               INSTRUCTIONS TESTED:
06              ;               LDR~    LOAD A VECTOR TO RAM
07              ;               MXR~    FIND MAX VECTOR ELEMENT
08              ;               MNR~    FIND MIN VECTOR ELEMENT
09              ;               SMA~    COMPLEX MAGNITUDE SQARED
10              ;               SER~    REAL SUM OF ELEMENTS
11              ;               PER~    REAL PRODUCT OF ELEMENTS
12              ;               PEC~    COMPLEX PRODUCT OF ELEMENTS
13              ;
14              ;
15              ;       7.2     TEST DESCRIPTION
16              ;
17              ;       7.2.1   MXR ~ MNR
18              ;               A REAL VECTOR WITH A KNOWN MAXIMUM VALUE IS LOADED
19              ;               INTO AP~RAM WITH AN LDR INSTRUCTION. THE MAXIMUM
20              ;               VALUE WHICH THE MXR CALCULATES IS THEN COMPARED
21              ;               WITH THE KNOWN MAX VALUE. THE VECTOR IS THEN RE~
22              ;               VERSED IN ORDER AND MXR IS RE~EXECUTED. SIMILAR
23              ;               TESTS ARE PERFORMED WITH DIFFERENT WORD COUNTS,
24              ;               STEP INDICES, AND RANDOM ADDRESSES IN RAM.
25              ;
26              ;               THE MNR TESTS DIFFER ONLY IN THAT THEY SEEK
27              ;               MINIMUM NUMBERS RATHER THAN MAXIMUM.
28              ;
29              ;       7.2.2   SMA ~ SER
30              ;               THE SMA INSTRUCTION IS TESTED BY FIRST LOADING
31              ;               AP~RAM (VIA LDR) WITH THE COMMPLEX NUMBERS TO BE
32              ;               SQUARED AND SUMMED. THE SMA IS THEN EXECUTED AND
33              ;               THE RESULTANT VECTOR IS COMPARED WITH THE PRE~
34              ;               VIOUSLY CALCULATED ANSWERS WHICH HAVE BEEN SET
35              ;               UP IN MAIN MEMORY. WORD COUNT VARIES FROM 1~25
36              ;               IN THE FIRST TEST, BUT OTHER TESTS USE RANDOM
37              ;               INDICES AND WORD COUNTS.
38              ;
39              ;               SER IS TESTED SIMILARLY EXCEPT THAT THE REAL
40              ;               VECTOR IS MERELY SUMMED.
41              ;
42              ;       7.2.3   PER ~ PEC
43              ;               THESE INSTRUCTIONS ARE TESTED BY SIMULATION
44              ;               ROUTINES WHICH PRODUCT REAL AND COMPLEX VECTORS
45              ;               RESPECTIVELY. THIS IS DONE FOR A WORD COUNT OF
46              ;               26 AND UNIT INDICES, INITIALLY, AND THEN RANDOM
47              ;               INDICES AND WORD COUNTS AND VECTOR ELEMENTS
48              ;               (WITH EXPONENT LIMITED +\~ 7)
49              ;
```

```
!0004 .MAIN
01                      S;WPD 8
02
03
04              ;8.     SWITCH SETTINGS
05              ;
06              ;       LOCATION "SWREG" IS USED TO SELECT THE PROGRAM OPTIONS
07              ;       (NOT SYSTEM CONFIGURATION). WHILE RUNNING UNDER DTOS,
08              ;       THIS LOCATION WILL BE LOADED BY THE MONITOR.
09              ;       HOWEVER UNDER STAND ALONE AND PROGRAM LOAD MODES THIS
10              ;       LOCATION WILL BE SET ACCORDING TO THE ANSWERS SUPPLIED
11              ;       BY THE OPERATOR. IN ANY CASE THE OPTIONS CAN BE CHANGED
12              ;       OR VERIFIED BY USING ONE OF THE COMMANDS GIVEN IN SEC.
13              ;       8.2
14              ;
15              ;
16              ;8.1    SWITCH OPTIONS
17              ;       DIFFERENT BITS AND THEIR INTERPRETATION AT LOCATION
18              ;       "SWREG" IS AS FOLLOWS:
19              ;
20              ;       BIT     OCTAL   BINARY  INERPRETATION
21              ;               VALUE   VALUE
22              ;
23              ;       1               0       LOOP ON ERROR
24              ;               40000   1       SKIP LOOPING ON ERROR
25              ;
26              ;       2               0       PRINT TO CONSOLE
27              ;               20000   1       ABORT PRINT OUT TO CONSOLE
28              ;
29              ;       3               0       DO NOT PRINT % FAILURE
30              ;               10000   1       PRINT % FAILURE
31              ;
32              ;       4               0       ALLOW END OF PASS PRINT OUT
33              ;               04000   1       SUPPRESS END OF PASS PRINT OUT
34              ;
35              ;       5               0       DO NOT PRINT ON THE LINE PRINTER
36              ;               02000   1       PRINT ON THE LINE PRINTER
37              ;
38              ;       6               0       DO NOT HALT ON ERROR
39              ;               01000   1       HALT ON ERROR
40              ;
41              ;       7               0       DO NOT PRINT SUMMARY AND/OR
42              ;                               PASSING OF EACH SUBTEST
43              ;               00400   1       PRINT SUMMARY AND/OR
44              ;                               PASSING OF EACH SUBTEST
45              ;       8               0       PRINT ONLY THE FIRST ERROR
46              ;               00200   1       PRINT EVERY ERROR
47
48              ;8.2    SWITCH COMMANDS
49              ;       ONCE THE PROGRAM STARTS EXECUTING THE STATE OF ANY OF
50              ;       THE BITS CAN BE CHANGED BY HITTING KEYS 1~9, A~F. THE
51              ;       PROGRAM WILL CONTINUE RUNNING AFTER UPDATING THE OPTIONS.
52              ;       EACH KEY WILL COMPLEMENT THE STATE OF THE BIT AFFILIAT~
53              ;       ED WITH IT, THUS BIT 4 CAN BE ALTERED BY HITTING KEY 4.
54              ;       SETTING OF ANY BIT OF LOCATION "SWREG" WILL SET BIT 0.
55              ;       (DEFAULT MODE IS DEFINED AS ALL BITS OF SWREG SET TO 0)
56              ;       THE PROGRAM CAN BE LOCKED INTO SWITCH MODIFICATION MODE
57              ;       BY TYPING A 0, IN WHICH CASE MORE THAN ONE BIT CAN BE
58              ;       CHANGED BEFORE  CONTROL IS ALLOWED TO RETURN TO THE
59              ;       MAIN PROGRAM.
60              ;
```

```
0005 .MAIN
01              ;8.2.1  OTHER COMMANDS
02              ;
03              ;       "CR"    A "RETURN" CAN BE TYPED TO CONTINUE THE PROGRAM
04              ;               AFTER ITS LOCKED IN A SWITCH MODIFICATION MODE
05              ;
06              ;       ^D      THIS COMMAND GIVEN AT ANY TIME WILL RESET "SWREG"
07              ;               TO DEFAULT MODE AND RESTART THE PROGRAM.
08              ;
09              ;       ^R      THIS COMMAND GIVEN AT ANY TIME WILL RESTART THE
10              ;               PROGRAM. SWITCHES ARE LEFT WITH THE VALUES THEY
11              ;               HAD BEFORE THE COMMAND WAS ISSUED.
12              ;
13              ;       ^O      THIS COMMAND GIVEN AT ANY TIME WILL CAUSE THE
14              ;               PROGRAM CONTROL TO GO TO ODT (NOTE: THIS IS AN
15              ;               OPTIONAL COMMAND AND IS AVAILBLE ONLY IF
16              ;               ODTPK IS PRESENT)
17              ;
18              ;       M       THIS COMMAND GIVEN AT ANY TIME WILL PRINT THE
19              ;               CURRENT OPERATING MODES.
20              ;
21
22              ;
```

```
10006 .MAIN
01              ;9.0    OPERATING PROCEDURE/OPERATOR INPUT
02              ;
03              ;       1. LOAD THE PROGRAM IN THE HOST COMPUTER
04              ;          USING THE BINARY LOADER OR DTOS TAPE.
05              ;       2. SET SWITCHES EQUAL TO ONE OF THE FOLLOWING
06              ;          STARTING ADDRESSES:
07              ;             200,500       RUNALL TESTS
08              ;             501           MXR INSTRUCTION
09              ;             502           MNR INSTRUCTION
10              ;             503           SMA INSTRUCTION
11              ;             505           SER INSTRUCTION
12              ;             505           PER INSTRUCTION
13              ;             506           PEC INSTRUCTION
14              ;             507           ENTER MEMORY DUMP (SEE 7.0)
15              ;       3. PRESS START.
16              ;       4. THE PROGRAM PRINTS "PASS" FOLLOWING ALL
17              ;          SUCCESSFUL COMPLETION OF TESTS.
18              ;
19              ;10.0   ERRORS
20              ;
21              ;       WHEN AN ERROR IS DETECTED A VALUE FOR AN
22              ;       ERROR PC WILL BE PRINTED ALONG WITH AC'S 0,1 AND 2
23              ;       OF THE HOST COMPUTER.
24              ;
25              ;       THE PC VALUE PRINTED MAY POINT TO ONE OF THE
26              ;       FOLLOWING:
27              ;
28              ;              1. A CALL TO EHALT
29              ;              2. A CALL TO THE ROUTINE WHICH DETECTED
30              ;                 THE ERROR AND SUBSEQUENTLY EXITED
31              ;                 TO EHALT SUCH AS THE VECTOR COMPARE
32              ;                 ROUTINES TCOMP,CCOMP,RCOMP, AND RCCMP.
33              ;
34              ;       SUBROUTINES OF THIS TYPE HAVE SPECIAL ERROR PRINTOUTS
35              ;       CALLED IN THE ERROR ROUTINE. IN THE CASE OF THE REAL
36              ;       VECTOR COMPARE ROUTINES, TCOMP AND RCOMP, UP TO 3
37              ;       PAIRS OF GOOD/BAD DATA IS PRINTED IN 2-16. BIT HEX
38              ;       OR OCTAL FORMAT (SEE SW10), ALONG WITH THE MM ADDRESS
39              ;       OF THE DATA AND THE RELATIVE INDEX WITHIN THE VECTOR.
40              ;       A TOTAL ERROR COUNT IS PRINTED ALONG WITH
41              ;       THE VECTOR/VECTOR N,IJ,IK DATA. IN THE CASE OF THE
42              ;       COMPLEX VECTOR COMPARE ROUTINES, CCOMP AND RCCMP, THE
43              ;       GOOD/BAD DATA IS PRINTED IN 4-16.  BIT HEX OR OCTAL
44              ;       FORMAT.
45              ;
46              ;       THE SIGNIFICANCE OF ERROR PRINTOUTS IS EXPLAINED IN THE
47              ;       LISTING AT THE SOURCE OF THE ERROR.  IN GENERAL
48              ;       AC0 CONTAINS THE CHECKWORD AND AC1 THE DATA IN
49              ;       QUESTION.
50              ;
51              ;       AFTER PRINTING THE ABOVE THE PROGRAM HALTS ALLOWING
52              ;       THE OPERATOR TO SET THE SWITCHES. UPON PRESSING
53              ;       CONTINUE THE PROGRAM MAY EITHER GO ON TO THE NEXT
54              ;       TEST OR GO INTO A SCOPE LOOP BETWEEN THE APPROPRIATE
55              ;       CALLS TO .SETUP AND .LOOP (SEE 4.1 SWITCH SETTINGS)
56              ;
```

```
10007 .MAIN
01              ;11.0   DEBUG HELP
02              ;
03              ;       11.1    MONITOR LOCATIONS
04              ;
05              ;               LOCATION        LABEL   MEANING
06              ;                  200                  DTOS START
07              ;                  201                  FIRST LOCATION OF CURRENT TEST
08              ;                  202                  START ADDRESS
09              ;                  203                  XXXX
10              ;                  204                  PASS COUNT
11              ;
12              ;       11.2    PROGRAMMING DESCRIPTION
13              ;
14              ;               SEE PROGRAMMERS REFERENCE
15              ;               FOR ARRAY PROCESSOR.
16              ;
17              ;       11.3    COMMON SUBROUTINES
18              ;
19              ;               A) ADDSET, SETP, LOOP -
20              ;               THESE ALL WORK IN CONJUNCTION TO
21              ;               SET UP THE TEST LOOP.
22              ;               ADDSET TELLS SETP WHERE THE LOOP ENDS.
23              ;               SETP MARKS THE START OF THE TEST LOOP.
24              ;               LOOP MARKS THE END OF THE TEST LOOP
25              ;               THE AP INSTRUCTION IS EXECUTED AND THE RESULTS ARE
26              ;               CHECKED FOR ERROR WITHIN THIS TEST LOOP. IF AN ERROR
27              ;               OCCURS, THEN THE PROGRAM CAN LOOP INDEFINITELY BETWEEN
28              ;               THE SETP AND THE LOOP. (SEE SECTION 8 ON VARIOUS SWITCH
29              ;               SETTINGS).
30              ;
31              ;               NORMALLY, WITH NO ERRORS, THE PROGRAM WILL GO THROUGH
32              ;               THIS LOOP ONCE ON THE FIRST PASS OF THE PROGRAM. ON
33              ;               SUBSEQUENT PASSES THE PROGRAM WILL STAY IN THE LOOP
34              ;               FOR 5 LOOPS. (OR 100. DEPENDING ON THE PROGRAM).
35              ;
36              ;               B) PARAMETER BLOCK LOADER ROUTINES -
37              ;               THESE CALLS ALL HAVE DIFFERENT NAMES DEPENDING ON THE
38              ;               INSTRUCTION. HOWEVER THEY OPERATE IN THE SAME MANNER.
39              ;               THE COMMENT FIELD ON THE CALL WILL ALWAYS INDICATE
40              ;               THAT THIS IS A PARAMETER BLOCK LOAD ROUTINE AND WILL
41              ;               NAME THE AP INSTRUCTION THAT THE LOAD IS COMPATIBLE WITH.
42              ;               THE ARGUMENTS OF THE CALL ARE THE PARAMETERS OF THE AP
43              ;               INSTRUCTION. THESE PARAMETERS ARE TO BE LOADED
44              ;               BY THESE ROUTINES INTO MAIN MEMORY AT A LOCATION POINTED
45              ;               TO BY AC2. AC2 MUST ALWAYS POINT TO THE PARAMETER BLOCK
46              ;               DURING AP INSTRUCTION EXECUTION.
47              ;
```

```
10008 .MAIN
01              ;
02              ;       11.4    SEQUENCE OF TESTING
03              ;
04              ;               IT IS IMPORTANT THAT THE PROGRAMS BE
05              ;               EXECUTED IN A SPECIFIC SEQUENCE:
06              ;
07              ;
08              ;               APIS DIAG (FOR CPU/AP ONLY)
09              ;               APFP DIAG (FOR CPU/AP ONLY)
10              ;               APA EXER
11              ;               APB EXER
12              ;               APC EXER
13              ;               APD EXER
14              ;               APE EXER
15              ;               APF EXER
16              ;               APG EXER
17              ;               APH EXER
18              ;               API EXER
19              ;               APJ EXER
20              ;               APK EXER
21              ;
22              ;
```

```
10009 .MAIN
01                      O?DTD 12
02
03              ;12.    OCTAL DEBUG TOOL (ODT)
04              ;
05              ;       THE DIAGNOSTIC IS EQUIPED WITH A BUILT IN ODT WHICH CAN
06              ;       BE ACCESSED BY HITTING CONTROL O (^O) AT ANY TIME DURING
07              ;       THE EXECUTION OF THE PROGRAM (AFTER SETTING THE PARA-
08              ;       METERS).
09              ;       ON ENTERING ODT THE ADDRESS OF THE LOCATION HAVING THE
10              ;       NEXT INSTRUCTION TO BE EXECUTED WILL BE TYPED-OUT.
11              ;
12              ;12.1   CONVENTIONS AND SYMBOLS
13              ;       THE FOLLOWING CONVENTIONS ARE USED BY THE ODT:
14              ;       ?       PRESSING ANY ILLEGAL KEY CAUSES THE ODT TO RES-
15              ;               POND WITH A "?".
16              ;       @       ODT IS READY AND AT YOUR SERVICE.
17              ;
18              ;12.2   COMMAND STRUCTURE
19              ;       AN ODT COMMAND HAS THE FOLLOWING FORMAT:
20              ;               [ARGUMENT][COMMAND]
21              ;       AN ARGUMENT MAY BE ONE OF THE FOLLOWING:
22              ;       "EXP"   AN OCTAL EXPRESSION CONSISTING OF OCTAL NUMBERS
23              ;               SEPARATED BY PLUS (+) OR MINUS (-) SIGNS. LEAD-
24              ;               ING ZEROS NEED NOT BE TYPED.
25              ;       "ADR"   AN ADDRESS IS THE SAME AS AN EXPRESSION EXCEPT
26              ;               THAT BIT 0 IS NEGLECTED.
27              ;       A COMMAND IS A SINGLE TELETYPE CHARACTER
28              ;
29              ;12.3   ODT COMMANDS
30              ;       THE LOCATIONS THAT CAN BE EXAMINED AND MODIFIED BY THE
31              ;       USER ARE CALLED CELLS. THESE CELLS ARE OF TWO TYPES:
32              ;       INTERNAL CPU CELLS AND MEMORY LOCATIONS.
33              ;
34              ;12.3.1 OPENING INTERNAL CELLS
35              ;       THE COMMAND TO OPEN ONE OF THE INTERNAL REGISTERS IS OF
36              ;       THE FORM "NA" WHERE N IS ANY OCTAL EXPRESSION BETWEEN
37              ;       0 AND 7
38              ;       0-3     FOR ACCUMULATORS 0-3
39              ;       4       FOR PC OF THE NEXT INSTRUCTION TO BE EXECUTED IN
40              ;               THE EVENT OF A "P" COMMAND.
41              ;       5       CPU AND TTO STATUS
42              ;               BIT     INTERPRETATION
43              ;               15      STATUS OF TTO DONE FLAG
44              ;               14      STATUS OF INTERRUPTS (ION FLAG)
45              ;               13      STATUS OF CARRY BIT
46              ;       6       ADDRESS OF THE LOCATION HAVING THE BREAK POINT (IF
47              ;               ANY)
48              ;       7       INSTRUCTION AT THE BREAK POINT LOCATION
49
50              ;       OTHER COMMANDS TO OPEN CELLS ARE:
51              ;
52              ;       "ADR"/  OPEN THE CELL AND PRINT ITS CONTENTS
53              ;       ./      OPEN THE CELL CURRENTLY POINTED TO BY THE POINTER
54              ;               AND PRINT ITS CONTENTS.
55              ;       .+"ADR"/        ADD "ADR" TO THE POINTER, OPEN THE CELL
56              ;               AND PRINT ITS CONTENTS.
57              ;       .-"ADR"/        SUBTRACT "ADR" FROM THE POINTER, OPEN
58              ;               THE CELL AND PRINT ITS CONTENTS.
59              ;       "CR"    THE RETURN KEY IS USED TO CLOSE THE OPEN CELL
60              ;               WITH OR WITHOUT MODIFICATION.
```

```
 0010 .MAIN
01              ;       "LF"    LINE FEED IS USED TO CLOSE THE OPEN CELL WITH OR
02              ;               WITHOUT MODIFICATION AND TO OPEN THE SUCCEEDING
03              ;               CELL.
04              ;       ^       CLOSE THE OPEN CELL WITH OR WITHOUT MODIFICATION
05              ;               AND OPEN THE PRECEDING CELL
06              ;       /       CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
07              ;               OPEN THE CELL POINTED TO BY ITS CONTENTS.
08              ;       +"ADR"/ CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
09              ;               OPEN THE CELL POINTED TO BY ITS CONTENTS + "ADDR".
10              ;       -"ADR"/ CLOSE THE OPEN CELL WITHOUT MODIFICATION, AND
11              ;               OPEN THE CELL POINTED TO BY ITS CONTENTS - "ADR".
12              ;
13              ;12.3.2 MODIFICATION OF A CELL
14              ;       ONCE A CELL HAS BEEN OPENED ITS CONTENTS CAN BE MODIFIED
15              ;       BY TYPING THE NEW VALUE THE CELL IS TO CONTAIN IN THE
16              ;       FORM OF AN OCTAL EXPRESSION FOLLOWED BY "CR" OR "LF".
17              ;       IF A + OR - IS TYPED AS THE FIRST CHARACTER OF THE EX-
18              ;       PRESSION THEN THE VALUE OF THE EXPRESSION IS ADDED TO OR
19              ;       SUBTRACTED FROM THE OLD CONTENTS OF THE CELL. THE
20              ;       ADDRESS ITSELF OR AN EXPRESION RELATIVE TO THE ADDRESS
21              ;       CAN BE DEPOSITED BY TYPING A "." OR ".+/-OCTAL EXPRESS-
22              ;       ION". A RUBOUT COMMAND GIVEN RIGHT AFTER OPENING A CELL
23              ;       ALLOWS THE MODIFICATION OF ITS CONTENTS AS IF THEY WERE
24              ;       TYPED IN JUST BEFORE THE COMMAND WAS ISSUED.
25              ;
26              ;12.3.3 OTHER ODT COMMANDS
27              ;       RUBOUT  THIS KEY IS USED TO DELETE ERRONEOUSLY TYPED
28              ;               DIGITS. EACH TIME THE KEY IS PRESSED THE RIGHT MOST
29              ;               DIGIT IS DELETED AND ECHOED ON THE TERMINAL. IF
30              ;               THE RUBOUT KEY IS PRESSED RIGHT AFTER OPENING A
31              ;               CELL THEN IT DELETES THE RIGHT MOST DIGIT OF THE CELLS
32              ;               CONTENTS. THIS ALLOWS THE MODIFICATION OF THE CELL
33              ;               AS IF ITS CONTENTS WERE TYPED IN JUST BEFORE THE
34              ;               KEY WAS PRESSED.
35              ;       "ADR"B  INSERT A BREAK POINT AT LOCATION "ADR".
36              ;               ONLY ONE BREAK POINT CAN BE INSERTED AND ANY
37              ;               ENTRY TO ODT AFTER EXECUTING A BREAK POINT WILL
38              ;               CAUSE IT TO BE DELETED.
39              ;       D       DELETE THE BREAK POINT IF ANY.
40              ;       P       RESTART THE EXECUTION OF THE PROGRAM AT LOCATION
41              ;               POINTED BY 4A.
42              ;       "ADR"R  START EXECUTING THE PROGRAM AT "ADR" AFTER AN
43              ;               IO-RESET.
44              ;       K       KILL THE STRING TYPED SO FAR. THE ODT RESPONDS
45              ;               WITH A "?" AND THE OPEN CELL IS CLOSED WITHOUT
46              ;               MODIFICATION.
47              ;       =       PRINT THE OCTAL VALUE OF THE INPUT ONLY.
48              ;               THIS WILL CLOSE ANY OPEN CELLS WITHOUT
49              ;               MODIFICATION AND WILL NOT OPEN A CELL
50              ;
51              ;
52              ;       NOTE:   IN PROGRAMS WHICH RELOCATE THEMSELVES THE
53              ;               THE USER SHOULD PLACE BREAK POINTS ONLY IN THE
54              ;               THE ORIGINAL PROGRAM AREA. IF A BREAK POINT IS
55              ;               PLACED OUTSIDE THIS AREA THE RESULTS WILL
56              ;               BE UNPREDICTABLE.
57              ;
58              ;
```

```
00011 .MAIN
01                ;
02                ;
03                ;13.0   SPECIAL NOTES/SPECIAL FEATURES
04                ;
05                ;       13.1    FOR A COMPLETE TEST ALL PROGRAMS
06                ;               SHOULD BE EXECUTED WITH CAT/KITTEN.
07                ;
08                ;       13.2    A NOTE ABOUT AP ADDRESSING
09                ;
10                ;               ADDRESSES IN THE AP CAN BE OF SEVERAL
11                ;               MODES:
12                ;
13                ;               1) ONE WORD MODE (1E) SAME AS STANDARD
14                ;                  ADDRESSING
15                ;               2) TWO WORD MODE  (2WM) - EACH 32 BITS IS
16                ;                  NOW ONE ADDRESS SPACE.  THIS IS USED
17                ;                  IN THE AP TO SIMPLIFY REAL NUMBER
18                ;                  ADDRESSING.
19                ;               3) FOUR WORD MODE (4WM) - EACH 64 BITS IS
20                ;                  NOW ONE ADDRESS SPACE.  THIS IS USED
21                ;                  IN THE AP TO SIMPLIFY COMPLEX NUMBER
22                ;                  ADDRESSING.
23                ;
24                ;               THE AP ACCESSES AN ADDRESS RELATIVE TO THE START OF
25                ;               THE AP RAM.  THUS AP RAM LOC 0 WOULD BE THE
26                ;               FIRST LOCATION THAT IS IN THE AP.  HOWEVER,
27                ;               AS FAR AS THE ECLIPSE CPU IS CONCERNED, AP
28                ;               LOC 0 IS CONTAINED AT LOCATION LABEL
29                ;               "RAMPT". (IN PAGE ZERO.  SO, IF RAMPT CON-
30                ;               TAINS 64000, THEN 2000 2WM (AP RAM) IS REALLY
31                ;               64000+2000+2000=70000.
32                ;
33                ;               NOTE: "STOP ON STORE" OR "STOP ON ADDRESS" IN AP RAM
34                ;               SPACE WILL NOT WORK IF THE AP IS USING
35                ;               THE INTERNAL AP ADDRESS LINES TO ACCESS AP RAM.
36                ;
37                ;14.0   RUN TIME
38                ;
39                ;       14.1    PASS 1          30 SEC
40                ;       14.2    SUBSEQUENT PASSES       30 SEC

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
0012 .MAIN

07?DTD 001551  MC     9/01
S?WPD 001075  MC     4/01
```