IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZMMI-A-D
(SUPERSEDES MAINDEC-11-D1IC)

PRODUCT NAME: RANDAT

DATE REVISED: MAY 15,1972

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: JOHN RODENHISER/ JIM LACEY

1, ABSTRACT

THIS TEST COMBINES A RANDOM NUMBER GENERATOR WITH A RANDOM
LOCATION) EXERCISER TO DEMONSTRATE THAT THE MEMORY TEST
ZONE IS CAPABLE OF STORING RANDOM DATA. THE RANDOM EXERCISER
PROVIDES A READ LOOP DETERMINED BY THE LEAST SIGNIFICANT
OCTAL BIT OF THE DATA IN THE LOCATION UNDER TEST TO DEMON-
STRATE COMPATIBILITY BETWEEN LOCATIONS HOLDING RANDOM DATA.
EACH ERROR PRINTOUT LISTS THE ERROR ADDRESS, ERROR CONTENTS,
AND CORRECT RANDOM DATA.

2, REQUIREMENTS

2,1 EQUIPMENT

PDP-11 WITH A MINIMUM OF 4K MEMORY

2,2,1 PROGRAM STORAGE - THE ROUTINE USES MEMORY
FROM 200 TO 2624.

3, LOADING PROCEDURE

3,1 METHOD

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

1. ABSOLUTE LOADER MUST BE IN MEMOTY
2. PLACE BINARY TAPE IN READER
3. LOAD ADDRESS *7500. (* DETERMINED BY ADDRESS OF LOADER)
4. PRESS "START" (PROGRAM WILL LOAD).

4, STARTING PROCEDURE

4,1 CONTROL SWITCH SETTING

STARTING AT SA 200 ALL SWITCHES SHOULD BE DOWN OR ZERO.

4,2 STARTING ADDRESS OR ADDRESSES

200 = START WITH AUTOMATIC TEST LIMITS
202 = START WITH SELECTED TEST LIMITS

4,3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY.
SET SWITCH REGISTER TO STARTING ADDRESS.
LOAD ADDRESS 200.
PRESS START.
THE PROGRAM WILL RUN THROUGH A TEST OF THE SELECTED
ADDRESS FIELD AND LOOP.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

5.1.1 WITH ALL SWITCHES DOWN, THE PROGRAM WILL PRINT OUT ERRORS AND CONTINUE IN TEST.

5.1.2 SWITCH SETTINGS ARE

SW15 = 1 OR UP,,,HALT ON ERROR
SW13 = 1 OR UP,,,INHIBIT PRINTOUT
SW12 = 1 OR UP,,,HALT ON END OF PROGRAM

5.1.3 AUTOMATIC TEST LIMITS

IF THE PROGRAM IS STARTED AT ADDRESS 200 THE PROGRAM WILL TEST ALL AVAILABLE MEMORY, CARE SHOULD BE TAKEN TO SELECT THE PROPER OPERATIONAL SWITCH SETTING (REFER TO 5.1.2) BEFORE STARTING.

5.1.4 SELECTED TEST LIMITS

IF THE PROGRAM IS STARTED AT ADDRESS 202 A MESSAGE WILL BE PRINTED ON THE TELETYPE INSTRUCTING THE OPERATOR ON THE METHOD OF SELECTING OTHER MEMORY TEST AREAS AND THEN THE PROGRAM WILL STOP AT THE FIRST OF THREE HALTS.

A. SET THE LOW TEST LIMIT IN SWITCH REGISTER AND PRESS CONTINUE.

B. THEN SET THE HIGH TEST LIMIT IN THE SWITCH REGISTER AND PRESS CONTINUE.

C. THEN SET THE OPERATIONAL SWITCH SETTINGS (REF 5.1.2) AND PRESS CONTINUE.

THE PROGRAM WILL NOW BEGIN TESTING THE SELECTED AREA. THE PROGRAM WILL NOT ALLOW LIMITS TO BE SELECTED THAT WOULD CAUSE ITSELF TO BE DESTROYED. AN EXCEPTION TO THIS WOULD BE THE INSTRUCTIONAL TEXT STORAGE AREA. STARTING THE PROGRAM AT 200 OR SELECTING LIMITS WHICH OVERLAP THIS AREA WILL CAUSE THE TEXT MESSAGE TO BE DESTROYED.

5,2 SUBROUTINE ABSTRACTS

5,2,1 DAYGEN

SUBROUTINE DAYGEN IS A CALL TO THE RANDOM DATA GENERATOR, IT IS THE FUNCTION OF THIS SUBROUTINE TO GENERATE TWO RANDOM 16 BIT NUMBERS ON EACH PASS, THE ALGORITHM FOR THIS TYPE OF NUMBER GENERATOR MAY BE SUMMARIZED AS FOLLOWS: NX129+60474. 01001057 (OCTAL) THE CONSTANT IS BROKEN INTO A TRIPLE PRECISION WORD AS FOLLOWS: 000006, 047401, 001057, INITIALLY N IS PROVIDED BY THE MAIN PROGRAM, WITH THE TWO NUMBERS PRODUCED ON EACH GENERATOR PASS SAVED TO PRIME THE GENERATOR FOR THE NEXT PASS, THE HIGH ORDER PORTION OF THE TRIPLE PRECISION ARITHMETIC IS END AROUND CARRIED INTO THE LOW ORDER NUMBER, REGISTERS USED DURING THE SUBROUTINE ARE SAVED ON ENTERING THE SUBROUTINE AND RESTORED ON EXIT.

5,2,2 ERROR

SUBROUTINE ERROR IS CALLED WHENEVER THE PROGRAM DETECTS A LOCATION WHICH CONTAINS INCORRECT RANDOM DATA, UPON ENTERING THE SUBROUTINE THE INHIBIT PRINT SWITCH (SW13) IS TESTED TO DETERMINE IF THE ADDRESS AND CONTENTS OF THE ERROR LOCATION WILL BE PRINTED, AN EXAMPLE OF A RANDOM DATA ERROR PRINTOUT IS AS FOLLOWS:

INCORRECT RANDOM DATA AT THE FOLLOWING LOCATION:
002400 012304
012305

WHERE 2400 IS THE ADDRESS, 12304 ARE THE CONTENTS, AND 12305 IS THE CORRECT RANDOM DATA, THE SUBROUTINE UPON THE END OF PRINTING, OR IN THE EVENT SW13 IS PRESENT, CHECKS THE HALT SWITCH (SW15) IF THE SWITCH IS SET THE PROGRAM IMMEDIATELY RETURNS TO THE MAIN TEST PROGRAM.

6, ERRORS

6,1 ERROR PRINTOUT

PRINTS ALL ERRORS UNLESS INHIBITED BY SW13,

6,2 ERROR RECOVERY

DEPRESS CONTINUE TO RESTART SECTION OR RELOAD STARTING ADDRESS AND START,

7, RESTRICTIONS

7,1 STARTING RESTRICTION

NONE

7,2 OPERATIONAL RESTRICTION

NONE

8, MISCELLANEOUS

8,1 EXECUTION TIME

THE PROGRAM RINGS THE TELETYPE BELL AFTER EVERY 56 PASSES THRU THE PROGRAM WHICH IS APPROXIMATELY ONCE PER MINUTE WITH 4K OF MEMORY,

9, PROGRAM DESCRIPTION

THE RANDOM DATA GENERATOR TEST (RANDAT) IS DESIGNED
TO VERIFY THE ABILITY OF MEMORY TO STORE RANDOM DATA
CONFIGURATIONS WITHOUT MALFUNCTION. THE TEST OERATES
IN THREE PHASES. THE FIRST PHASE LOADS THE MEMORY TEST
ZONE WITH RANDOM NUMBERS. THE SECOND PHASE READ-EXERCISES
THESE NUMBERS WITH ADDRESS SELECTION DETERMINED BY THE
LEAST SIGNIFICANT OCTAL BIT OF THE PREVIOUS NUMBER EXERCISED.
THE THIRD PHASE REINITIALIZES AND RERUNS THE RANDOM
NUMBER GENERATOR, SEQUENTIALLY COMPARING THE CONTENTS OF THE
TEST LOCATIONS WITH EACH NUMBER GENERATED. THE RANDOM
NUMBER IS TURNED ON BY A CALL TO SUBROUTINE DAYGEN. IN THE
EVENT OF AN ERROR THE PROGRAM WILL JUMP TO THE SUBROUTINE
ERROR FOR THE OUTPUT PRINTING AND HALT OPTIONS.

TO INDICATE THAT THE PROGRAM IS OPERATING THE
TELETYPE BELL WILL RING APPROXIMATELY EVERY
MINUTE. THE PROGRAM WILL HALT AT THE END OF
EACH PASS IF SWITCH 12 IS PRESENT. A CONTINUE
FROM THE HALT (OR IF SWITCH 12 IS NOT PRESENT)
WILL JUMP TO THE NEXT PASS.

10, LISTING

DZMMIA.P11

```
                                        .NLIST  SEQ

                                        .TITLE  PROGRAM RANDAT  MAINDEC-11-DZMMI-A
                                        ;COPYRIGHT 1970, 1971,1972 DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

            000240                      NOP=240
                                                .ENABL  ABS
            000000                              .=0

                                        ;TRAP CATCHER 0-176

            000004                              .=4
  000004    000234                              TLG
            000200                              .=200
  000200    000407                              BR      START
  000202    000443                              BR      START1
  000204    000000                    LOLMT:    0
  000206    000000                    HILMT:    0
  000210    177564                    TCSR:     177564
  000212    177566                    TDBR:     177566
  000214    177570                    SWREG:    177570
  000216    000000                    SUMMIT:   0
```

```
000220  012706  002262          START:  MOV     #BUFFER,%6
000224  010603                          MOV     %6,%3           ;TEST POINT IN EACH BANK
000226  005723                  SEE:    TST     (%3)+           ;TEST
000230  000240                          NOP                     ;PRECAUTIONARY DELAY
000232  000775                          BR      SEE             ;NO TRAP, CONTINUE
000234  162703  000004          TLG:    SUB     #4,%3           ;TRAPPED,
000240  005737  000042                  TST     @#42
000244  001407                          BEQ     $1
000246  023727  000042  000754          CMP     @#42,#ENDADR
000254  001405                          BEQ     $2
000256  162703  002734                  SUB     #1500.,%3
000262  000402                          BR      $2
000264  162703  000300          $1:     SUB     #300,%3
000270  010367  177712          $2:     MOV     %3,HILMT
000274  012767  177710  000662          MOV     #-70,BELLCT
000302  012767  002264  177674          MOV     #QEFA,LOLMT     ;SETUP END PGM MARKER
000310  000446                          BR      AUTOST          ;SKIP OVER CONTROL OPTIONS
000312  012767  177710  000644  START1: MOV     #-70,BELLCT
000320  012767  002264  177656          MOV     #QEFA,LOLMT     ;GET END OF PGM MARKER
000326  012767  017470  177652          MOV     #17470,HILMT    ;SET HILMT
000334  012706  002262                  MOV     #BUFFER,%6      ;SET LP
000340  012702  002266                  MOV     #MSG1,%2
000344  004767  001370                  JSR     %7,TOP
000350  000000                          HALT                    ;PUT LOLMT IN SWREG AND CONTINUE
000352  005777  177636                  TST     @SWREG          ;LOOK FOR LOLMT
000356  001407                          BEQ     HISET           ;BRANCH IF DEFERRED
000360  027767  177630  177616          CMP     @SWREG,LOLMT    ;CHK LOLMT>LOADPT
000366  103403                          BLO     HISET           ;LOLMT<LOADPT, USE LMT DEFINED
000370  017767  177620  177606          MOV     @SWREG,LOLMT    ;LOLMT>LOADPT, STORE INPUT
000376  000000                  HISET:  HALT
000400  005777  177610                  TST     @SWREG          ;LOOK FOR HILMT
000404  001407                          BEQ     CTLSET          ;BRANCH IF DEFERRED
000406  027767  177602  177570          CMP     @SWREG,LOLMT    ;CHK HILMT>LOLMT
000414  103403                          BLO     CTLSET          ;BRANCH IF HILMT<LOLMT
000416  017767  177572  177562          MOV     @SWREG,HILMT    ;STORE SWREG
000424  000000                  CTLSET: HALT                    ;WAIT FOR, PGM CONTROL SELECTIONS
000426  012767  123456  000532  AUTOST: MOV     #123456,LONUM   ;INITIALIZE LONUM LOCATION
000434  012767  176543  000526          MOV     #176543,HINUM   ;DITTO FOR HINUM
000442  016767  000520  000522  NXPASS: MOV     LONUM,SAVLO     ;SAVE FOR READCK INITIALIZATION
000450  016767  000514  000516          MOV     HINUM,SAVHI     ;SAVE FOR READCK INITIALIZATION
000456  016700  177522                  MOV     LOLMT,%0        ;GET LOLMT FOR R0
000462  004767  000314          REWRYT: JSR     %7,DAYGEN       ;GENERATE TWO RANDOM NUMBERS
000466  014420                          MOV     -(4),(0)+       ;WRITE 1ST NUMBER
000470  020067  177512                  CMP     %0,HILMT        ;CKN COUNT UP TO HILMT
000474  101005                          BHI     MEMCZN          ;IF SO GO EXERCIZE TEST FIELD
000476  014420                          MOV     -(4),(0)+       ;WRITE 2ND NUMBER
000500  020067  177502                  CMP     %0,HILMT        ;CKN COUNT UP TO HILMT
000504  101001                          BHI     MEMCZN          ;IF SO GO EXERCIZE TEST FIELD
000506  000765                          BR      REWRYT          ;LOOP AND CONTINUE WRITING
000510  016767  177470  000472  MEMCZN: MOV     LOLMT,EXRADD    ;GET LOLMT
```

```
000516  017767  000466  000460  NXERCZ: MOV     @EXRADD,MASK     ;SET UP LSOB IN MASK
000524  042767  177770  000460          BIC     #177770,MASK     ;AND IN DATA
000532  005767  000454                  TST     MASK             ;CKN FOR 0 OCTAL BIT
000536  001430                          BEQ     ZIPBIT           ;BRANCH IF 0
000540  016767  000446  000440          MOV     MASK,SVMASK      ;SAVE MASK CONTENTS
000546  027777  000436  000434  XRCZON: CMP     @EXRADD,@EXRADD  ;EXERCISE LOCATION THE NUMBER
000554  005367  000432                  DEC     MASK             ;OF TIMES SPECIFIED BY THE
000560  001401                          BEQ     .+4              ;LS OCTAL BIT OF RANDOM NUMBER
000562  000771                          BR      XRCZON
000564  032767  000001  000422          BIT     #1,SVMASK        ;SETUP MASK TO CK FOR ODD NO.
000572  001402                          BEQ     .+6
000574  005267  000414                  INC     SVMASK           ;INCREMENT ODD NO, TO STAY ON EVEN
000600  066767  000410  000402          ADD     SVMASK,EXRADD    ;BOUNDS, ADD TO ADDRESS POINTER
000606  026767  000376  177372  XRC1:   CMP     EXRADD,HILMT     ;CKN IF ADDRESS UP TO HILMT
000614  101740                          BLOS    NXERCZ           ;CONTINUE EXERCIZE IF <= HILMT
000616  000404                          BR      READCK           ;EXERCZ COMPLETE, GO CK MEMORY
000620  062767  000002  000362  ZIPBIT: ADD     #2,EXRADD        ;INCREMENT ADDRESS POINTER
000626  000767                          BR      XRC1             ;LOOP AND TRY AGAIN
000630  016700  177350          READCK: MOV     LOLMT,R0         ;PUT IN LOLMT IN R0
000634  016767  000332  000324          MOV     SAVLO,LONUM      ;REINITIALIZE LONUM
000642  016767  000326  000320          MOV     SAVHI,HINUM      ;DITTO FOR HINUM
000650  004767  000126          READON: JSR     %7,DAYGEN        ;GO GENERATE NUMBERS
000654  024410                          CMP     -(4),@%0         ;CK AGAINST MEMORY
000656  001402                          BEQ     .+6              ;SKIP NX IF OK
000660  004767  000334                  JSR     %7,ERROR         ;TO ERROR ROUTINE
000664  020067  177316                  CMP     %0,HILMT         ;CKN FOR COUNT UP TO HILMT
000670  103012                          BHIS    ENDPASS          ;BRANCH IF >=
000672  005720                          TST     (0)+             ;JUST INCREMENTING
000674  024410                          CMP     -(4),@%0         ;CKN 2ND WORD AGAINST MEMORY
000676  001402                          BEQ     .+6              ;SKIP NX IF OK
000700  004767  000314                  JSR     %7,ERROR         ;TO ERROR ROUTINE
000704  020067  177276                  CMP     %0,HILMT         ;CKN HILMT AGAIN
000710  103002                          BHIS    ENDPASS          ;BRANCH IF >=
000712  005720                          TST     (0)+             ;INCREMENTING AGAIN
000714  000755                          BR      READON           ;LOOP TO CONTINUE READ
000716  005267  000242          ENDPASS: INC     BELLCT           ;INCREMENTING PASS ITERATION COUNTER
000722  001020                          BNE      CONTCK          ;GO CK HALT-CONTINUE SWITCH
000724  012767  177710  000232          MOV     #-70,BELLCT      ;REINITIALIZE COUNTER
000732  105777  177252                  TSTB    @TCSR            ;CKN TTY AVAILABILITY
000736  100375                          BPL     .-4              ;WAIT LOOP
000740  012777  000207  177244          MOV     #207,@TDBR       ;RING A DING
000746  013702  000042                  MOV     @#42,%2
000752  001404                          BEQ     CONTCK
000754  004712                  ENDADR: JSR     %7,(2)
000756  000240                          NOP
000760  000240                          NOP
000762  000240                          NOP
000764  032777  010000  177222  CONTCK: BIT     #10000,@SWREG    ;CKN HALT-CONTINUE SWITCH
000772  001401                          BEQ     .+4              ;SKIP NX IF NO SWITCH
000774  000000                          HALT
000776  000167  177440                  JMP     NXPASS           ;JUMP FOR NEXT PASS
```

```
                                ;RANDOM DATA GENERATOR SUBROUTINE

001002  010067  000170  DAYGEN: MOV     %0,SVZERO       ;SAVN (C) R0
001006  010167  000166          MOV     %1,SVONE        ;SAVN (C) R1
001012  010267  000164          MOV     %2,SVTWO        ;SAVN (C) R2
001016  010367  000162          MOV     %3,SVTRIP       ;SAVN (C) R3
001022  012704  001156          MOV     #ENDGEN,%4      ;GET END OF SUBRTN MARKER
001026  016700  000134          MOV     LONUM,%0        ;SET UP R0 WITH 5 DIGITS LOW
001032  016701  000132          MOV     HINUM,%1        ;SET UP R1 WITH 5 DIGITS HIGH
001036  012703  000007          MOV     #7,%3           ;SET UP SHIFT COUNT
001042  005002                  CLR     %2              ;CLEAR R2
001044  006300          SHIFT:  ASL     %0              ;SHIFT R0 LEFT AND
001046  006101                  ROL     %1              ;ROTATE CARRY INTO LSB OF R1 AND
001050  006102                  ROL     %2              ;ROTATE CARRY OUT OF R1 INTO R2
001052  005303                  DEC     %3              ;DECREMENT R3
001054  001373                  BNE     SHIFT           ;CONTINUE SHIFT LOOP
001056  066700  000104          ADD     LONUM,%0        ;ADDN IN NUMBER TO MAKE X 129
001062  005501                  ADC     %1              ;PROPOGATE CARRY
001064  066701  000100          ADD     HINUM,%1        ;ADDN IN NUMBER TO MAKE X 129
001070  005502                  ADC     %2              ;PROPOGATE CARRY
001072  062700  001057          ADD     #1057,%0        ;ADDN LOW CONSTANT
001076  005501                  ADC     %1              ;PROPOGATE CARRIES
001100  005502                  ADC     %2              ;PROPOGATE AGAIN
001102  062701  047401          ADD     #47401,%1       ;ADDN HIGH CONSTANT
001106  005502                  ADC     %2              ;PROPOGATE CARRY
001110  062702  000006          ADD     #6,%2           ;ADDN HIGHEST CONSTANT
001114  060200                  ADD     %2,%0           ;REPRIME R0 WITH HIGH DIGIT
001116  005501                  ADC     %1              ;PROPOGATE CARRY
001120  010067  000042          MOV     %0,LONUM        ;PUT R0 BACKIN LONUM
001124  010024                  MOV     %0,(4)+         ;HOLD LONUM FOR PROGRAM
001126  010167  000036          MOV     %1,HINUM        ;PUT R1 BACK IN HINUM
001132  010124                  MOV     %1,(4)+         ;HOLD HINUM FOR PROGRAM
001134  016700  000036          MOV     SVZERO,%0       ;RETURN DATA TO R0,R1,R2
001140  016701  000034          MOV     SVONE,%1        ;AND R3
001144  016702  000032          MOV     SVTWO,%2
001150  016703  000030          MOV     SVTRIP,%3
001154  000207                  RTS     %7              ;RETURN TO PROGRAM
001156  000000          ENDGEN: 0
001160  000000                  0
001162  000000                  0
001164  000000          BELLCT: 0
001166  000000          LONUM:  0
001170  000000          HINUM:  0
001172  000000          SAVLO:  0
001174  000000          SAVHI:  0
001176  000000          SVZERO: 0
001200  000000          SVONE:  0
001202  000000          SVTWO:  0
001204  000000          SVTRIP: 0
001206  000000          SVFOUR: 0
```

```
001210  000000                  EXRADD: 0
001212  000000                  MASK:   0
001214  000000                  SVMASK: 0
001216  000000                  SPOCK:  0
001220  010467  177762          ERROR:  MOV     %4,SVFOUR       ;SAVN R0,R1, AND R4
001224  010067  177746                  MOV     %0,SVZERO
001230  010167  177744                  MOV     %1,SVONE
001234  032777  020000  176752          BIT     #20000,@SWREG   ;SETUP INHIBIT PRINT MASK
001242  001064                          BNE     SWCHK           ;SW UP,SKIP PRINTING
001244  012702  002100                  MOV     #MSG2,%2        ;SEND "INCORRECT RANDOM DATA" MESSAGE
001250  004767  000464                  JSR     %7,TOP
001254  010002                          MOV     %0,%2           ;GET (C) R0
001256  004767  000162                  JSR     %7,PRTAB        ;GO PRINT ADDRESS
001262  105777  176722          SPACE:  TSTB    @TCSR           ;CK TTY AVAIL
001266  100375                          BPL     SPACE
001270  012777  000240  176714          MOV     #240,@TDBR      ;SEND SPACE
001276  005267  177714                  INC     SPOCK
001302  026727  177710  000006          CMP     SPOCK,#6        ;CK 6 SPACES
001310  001364                          BNE     SPACE
001312  011002                          MOV     @%0,%2          ;GET ERROR DATA
001314  005067  177676                  CLR     SPOCK
001320  004767  000120                  JSR     %7,PRTAB        ;GO TO PRINT ROUTINE
001324  105777  176660                  TSTB    @TCSR           ;CK TTY
001330  100375                          BPL     .-4
001332  012777  000215  176652          MOV     #215,@TDBR      ;SEND CR
001340  105777  176644                  TSTB    @TCSR           ;CK TTY
001344  100375                          BPL     .-4
001346  012777  000212  176636          MOV     #212,@TDBR      ;SEND LF
001354  017702  177624                  MOV     @SVFOUR,%2      ;GET CORRECT DATA FOR PRINTER
001360  004767  000060                  JSR     %7,PRTAB        ;GO PRINT (C) OF R4
001364  105777  176620                  TSTB    @TCSR           ;CKN TTY
001370  100375                          BPL     .-4
001372  012777  000215  176612          MOV     #215,@TDBR      ;SEND CR
001400  105777  176604                  TSTB    @TCSR           ;CKN TTY
001404  100375                          BPL     .-4
001406  012777  000212  176576          MOV     #212,@TDBR      ;SEND LF
001414  032777  100000  176572  SWCHK:  BIT     #100000,@SWREG  ;SETUP HALT MASK
001422  001401                          BEQ     .+4             ;SKIP HALT IF NOT HALT ON ERROR
001424  000000                          HALT
001426  016704  177554                  MOV     SVFOUR,%4       ;RETURNING DATA TO R0,R1, & R4
001432  016700  177540                  MOV     SVZERO,%0
001436  016701  177536                  MOV     SVONE,%1
001442  000207                          RTS     %7              ;RETURN TO PROGRAM
```

```
001444  005067  000252          PRTABI: CLR     BINCT
001450  005067  000244                  CLR     WGTCT
001454  012704  001726                  MOV     #LIST,X4        ;GET LIST ADDRESS
001460  142777  000177  176522          BICB    #177,@TCSR      ;CLR, INT FLAG
001466  012767  000005  000230          MOV     #5,ASCNT
001474  012767  000007  000210          MOV     #7,SEVEN
001502  012767  000001  000206          MOV     #1,DECML
001510  105777  176474          WAIT1:  TSTB    @TCSR
001514  100375                  BPL     WAIT1
001516  005702                  TST     X2
001520  100424                  BMI     MINUS           ;NEG SIGN PRINT 1
001522  012777  000260  176462          MOV     #260,@TDBR      ;POS SIGN PRINT 0
001530  000403                  BR      STARTX
001532  012777  000261  176452  MINUS:  MOV     #261,@TDBR
001540  016703  000146          STARTX:  MOV     SEVEN,X3        ;PUT MASK IN R3
001544  010267  000144                  MOV     X2,TOODLE       ;GET READY TO DOODLE NUMBER IN TOODLE
001550  005167  000140                  COM     TOODLE          ;COMPENSATES FOR COMPLEMENT DURING BIC
001554  046703  000134                  BIC     TOODLE,X3       ;AND IN OCTAL CHARACTER
001560  001410                  BEQ     WRTOC           ;ZERO, WRITE 0 IN LIST
001562  066767  000130  000130  MKNUM:  ADD     DECML,WGTCT     ;COUNT UP TO
001570  005267  000126                  INC     BINCT           ;AND RECORD
001574  026703  000120                  CMP     WGTCT,X3        ;SAME BINARY WEIGHT
001600  001370                  BNE     MKNUM           ;KEEP COUNTN
001602  062767  000260  000112  WRTOC:  ADD     #260,BINCT      ;ADD ASCII PREFIX
001610  016724  000106                  MOV     BINCT,(4)+      ;WRITE ASCII CHAR IN LIST
001614  066767  000072  000074          ADD     SEVEN,DECML     ;EXPAND BINARY WEIGHT
001622  005067  000072                  CLR     WGTCT
001626  005067  000070                  CLR     BINCT
001632  005367  000066                  DEC     ASCNT
001636  001410                  BEQ     XLIST           ;5 CHAR IN LIST
001640  012703  000003                  MOV     #3,X3           ;SET X3 FOR ADD LOOP
001644  066767  000042  000040  MOADD:  ADD     SEVEN,SEVEN     ;MAKING SEVENTY BE SEVEN
001652  005303                  DEC     X3
001654  001373                  BNE     MOADD
001656  000730                  BR      STARTX          ;NX SEVEN SET GET NX OCTAL
001660  012767  000005  000036  XLIST:  MOV     #5,ASCNT        ;SEND 5 CHAR TO TTY
001666  105777  176316          WAIT2:  TSTB    @TCSR
001672  100375                  BPL     WAIT2
001674  014477  176312                  MOV     -(4),@TDBR
001700  005367  000020                  DEC     ASCNT
001704  001401                  BEQ     HOFHM           ;FINISH PRINTING GET NXT NUM
001706  000767                  BR      WAIT2
001710  000207          HOFHM:  RTS     X7              ;HEAD FOR HOME
001712  000000          SEVEN:  0
001714  000000          TOODLE: 0
001716  000000          DECML:  0
001720  000000          WGTCT:  0
001722  000000          BINCT:  0
001724  000000          ASCNT:  0
001726  000000          LIST:   0
001730  000000                  0
001732  000000                  0
001734  000000                  0
001736  000000                  0
```

```
001740  142777  200177  176242  TOP:    BICB    #177,@TCSR      ;CLR INT FLAG
001746  105777  176236                  TSTB    @TCSR
001752  100375                          BPL     .-4
001754  112777  000215  176230          MOVB    #215,@TDBR      ;SEND CARRIAGE RETURN
001762  105777  176222                  TSTB    @TCSR
001766  100375                          BPL     .-4
001770  112777  000212  176214          MOVB    #212,@TDBR      ;SEND LINE FEED
001776  112267  300074                  MOVB    (2)+,EOMK       ;MOVE IN EOM MARKER
002002  121267  000070          TOP1:   CMPB    @X2,EOMK        ;COMPARE FOR EOM
002006  001411                          BEQ     ATRATE
002010  121227  000100                  CMPB    @X2,#'@
002014  001406                          BEQ     ATRATE
002016  105777  176166                  TSTB    @TCSR           ;CK TTY
002022  100375                          BPL     .-4             ;WAIT FOR DONE
002024  112277  176162                  MOVB    (2)+,@TDBR      ;MOVE CHARACTER
002030  000764                          BR      TOP1            ;BRANCH BACK
002032  105777  176152          ATRATE: TSTB    @TCSR
002036  100375                          BPL     .-4
002040  112777  000215  176144          MOVB    #215,@TDBR      ;SEND CARRIAGE RETURN
002046  105777  176136                  TSTB    @TCSR
002052  100375                          BPL     .-4
002054  112777  000212  176130          MOVB    #212,@TDBR      ;SEND LINE FEED
002062  121267  000010                  CMPB    @X2,EOMK        ;CKN END MESSAGE MARK
002066  001402                          BEQ     .+6             ;SKP NX2 IF EOMK
002070  105202                          INCB    X2              ;INCRMTN R2
002072  000743                          BR      TOP1            ;NO EOM, SO LOOP
002074  000207                          RTS     X7
002076     000  EOMK:   .BYTE   0
            002100                      .EVEN
002100  044454  041516  051117  MSG2:   .ASCII  /,INCORRECT RANDOM DATA AT THE FOLLOWING LOCATION:,/
002106  042522  052103  051040
002114  047101  047504  020115
002122  040504  040524  040440
002130  020124  044124  020105
002136  047506  046114  053517
002144  047111  020107  047514
002152  040503  044524  047117
002160  026072
                                    .EVEN
002162  000000  QEF:    0
            002262                  .=QEF+100
002262  000000  BUFFER: 0
002264  000000  QEFA:   0
002266  051457  052105  046440  MSG1:   .ASCII/SET MEMORY ADDRESS LIMITS VIA SWITCH REGISTER@/
002274  046505  051117  020131
002302  042101  051104  051505
002310  020123  044514  044515
002316  051524  053040  040511
002324  051440  044527  041524
002332  020110  042522  044507
002340  052123  051105     100
002345     123  052105  046040          .ASCII/SET LOWER LIMIT IN SW-REG AND PRESS CONTINUE@/
002352  053517  051105  046040
002360  046511  052111  044440
```

```
002366  020116  053523  051055
002374  043505  040440  042116
002402  050040  042522  051523
002410  041440  047117  044524
002416  052516  040105
002422  042523  020124  050125          ,ASCII!SET UPPER LIMIT IN SW-REG AND PRESS CONTINUE@!
002430  042520  020122  044514
002436  044515  020124  047111
002444  051440  026527  042522
002452  020107  047101  020104
002460  051120  051505  020123
002466  047503  052116  047111
002474  042525     100
002477     125  042116  043105          ,ASCII!UNDEFINED LIMITS WILL BE SET TO 2264-17470@!
002504  047111  042105  046040
002512  046511  052111  020123
002520  044527  046114  041040
002526  020105  042523  020124
002534  047524  031040  033062
002542  026464  033461  033464
002550  040060
002552  044514  044515  051524          ,ASCII!LIMITS BELOW THIS RANGE WILL BE DEFAULTED/!
002560  041040  046105  053517
002566  052040  044510  020123
002574  040522  043516  020105
002602  044527  046114  041040
002610  020105  042504  040506
002616  046125  042524  027504
        000001                          ,END
```

```
ASCNT   001724      ATRATE  002032      AUTOST  000426      BELLCT  001164
BINCT   001722      BUFFER  002262      CONTCK  000764      CTLSET  000424
DAYGEN  001002      DECML   001716      ENDADR  000754      ENDGEN  001156
ENDPAS  000710      EOMK    002076      ERROR   001220      EXRADD  001210
HOFHM   001710      HILMT   000206      HINUM   001170      HISET   000376
LIST    001720      LOLMT   000204      LONUM   001166      MASK    001212
MEMCEN  000510      MINUS   001532      MKNUM   001562      MOADD   001644
MSG1    002260      MSG2    002100      NOP     = 000240    NXERCZ  000516
NXPASS  000442      PRTAB   001444      QEF     002162      QEFA    002264
READCK  000630      READON  000650      REWRYT  000462      SAVHI   001174
SAVLO   001172      SEE     000226      SEVEN   001712      SHIFT   001044
SPACE   001262      SPOCK   001216      START   000220      STARTX  001540
START1  000312      SUMMIT  000214      SVFOUR  001206      SVMASK  001214
SVONE   001200      SVTRIP  001204      SVTWO   001202      SVZERO  001176
SWCHK   001414      SWREG   000214      TCSR    000210      TOBR    000212
TLG     000234      TOODLE  001714      TOP     001740      TOP1    002002
WAIT1   001510      WAIT2   001666      WGTCT   001720      WRTOC   001602
XLIST   001660      XRCZON  000546      XRC1    000606      ZIPBIT  000620
$1      000264      $2      000270      .       = 002624
```

ERRORS DETECTED: 0

DZMMIA,P11

```
*DZMMIA,DZMMIA=DZMMIA/SOL
RUN-TIME:  2  4  0 SECONDS
CORE USED:  3K
```