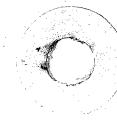
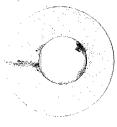
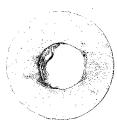


IDENTIFICATION

PRODUCT CODE: MAINDEC-08-D1EC-D
PRODUCT NAME: PDP-8, 8/I EXTENDED MEMORY
CHECKERBOARD
DATE CREATED: NOVEMBER 1, 1971
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: J. RICHARDSON - J. VROBEL

**COPYRIGHT© 1971
DIGITAL EQUIPMENT
CORPORATION**



1. ABSTRACT

The PDP-8, 8/I Extended Memory Checkerboard diagnostic is designed to provide worst case half-select noise conditions in order to determine the operational status of core memory. Four data patterns, and their complements, are written and checked for error. The patterns provided will generate the worst case noise conditions for a PDP-8 or 8/I equipped with standard or specially purchased core stacks, and will test systems equipped with from 8K to 32K words of core memory. Automatic program relocation is provided in order to test all memory stacks from each stack.

Teletype print-outs are provided for error identification. Also, the operator is given a degree of control over the program by various SR settings. These are explained in detail in Section 8.2.

2. REQUIREMENTS

2.1 Equipment

A standard PDP-8 or 8/I equipped with at least 8K words of core memory.

2.2 Storage

The program occupies locations 0010 to 3334.

2.3 Preliminary Programs

The Binary Loader must be in memory. Also, all diagnostics for a basic 4K PDP-8 must have been previously run successfully.

3. LOADING PROCEDURE

- a. Turn off the Teletype reader.
- b. Set the SR to 7777.
- c. Press LOAD ADDRESS; then START.
- d. Place the Binary tape in Teletype reader and turn on the reader.
- e. When the program has been loaded, stop the computer, turn off the reader, and remove the tape.

4. STARTING PROCEDURE

4.1 Starting Address

Start from address 200 to specify the amount of core memory to test; SR settings, and to receive a header print-out.

4.2 Restarting Address

Start from address 207 to change the test limits; SR settings, and to inhibit the header print-out.

4.3 Operator Action

Immediately after starting from address 200 or 207, the program will print TEST LIMITS. The operator must then specify, via the Teletype keyboard, the amount of core memory to test, followed by a carriage return.

The following rules govern the amount of memory to test:

- a. Type two octal numbers, separating the numbers with a comma. The first number signifies the lowest order 4K stack to test; the second signifies the highest order.
- b. The program expects the 4K stacks to be numbered sequentially starting with a stack 0.
- c. If the highest order stack to test is typed as the first stack, the program will interchange the two values so as to make the second value the first to test.
- d. After typing the second octal number, press the carriage return key to terminate the line.
- e. The program will test the lowest and highest order 4K stack specified, plus every stack between, starting with the lowest specified.
- f. Any single stack, or two or more sequential stacks may be specified.
- g. The stack containing the program may be included when specifying two or more stacks.

The stack containing the program will be tested after automatic program relocation takes place (see Section 5.3.1).

- h. If a typing error is made, press the RUB-OUT key. TEST LIMITS will be printed again.

All previous input is disregarded.

For the following examples assume the program to be located in stack 0, and the program has been started from address 200 or 207. The amount of core memory available is 32K.

Example A:

TEST LIMITS

0,7, (, denotes carriage return)

Example A indicates stacks 0, 1, 2, 3, 4, 5, 6 and 7 will be tested.

Example B:

TEST LIMITS

7,0,

The program will perform exactly as Example A.

Example C:

TEST LIMITS

4,5,

Only stacks 4 and 5 will be tested.

Example D:

TEST LIMITS

3,3,

Stack 3 alone will be tested.

Example E:

TEST LIMITS

0,0 PROGRAM IS LOCATED IN FIELD 0

TEST LIMITS

0,1,

Example E shows the message printed by the program when a single stack is selected which currently contains the program. TEST LIMITS is printed again, and the operator must then correct the test limits.

Operation of the program is unpredictable if the amount of memory selected for testing exceeds the actual amount available, i.e., selecting 32K for testing on a PDP-8 or 8/I equipped with a maximum of 28K.

4.3.1 Setup SR

After the test limit is specified, the program will print SETUP SR. For normal program operation, the SR must be set to equal 0000₈. Press the carriage return key after setting the SR to 0000. The program will then run until stopped by the operator. Normal program operation is defined as performing all four checkerboard patterns on all of available memory from every memory stack.

5. OPERATING PROCEDURE

5.1 Program and Operator Action

- a. Load the program into stack 0 using the procedure described in Section 3.
- b. Set the SR to 200; press LOAD ADDRESS, and then start.
- c. The message TEST LIMITS will be printed. Specify the limits, via keyboard, as described in Section 4.3.
- d. The message SETUP SR will be printed. Set the SR to 0000_8 , and press the carriage return key.
- e. The program will perform all four tests on all of core memory specified, after which, automatic program relocation takes place.

5.2 Operational Switch Settings

Normal operation of the program requires the SR set to 0000_8 . Refer to Section 8.2, applications, for switch settings provided for trouble-shooting.

5.3 Subroutine Abstracts

5.3.1 Program Relocation

Program relocation is governed entirely by the amount of core memory selected for testing. Under certain conditions the program will not relocate at all, but will remain in the current 4K stack to perform the tests (see below). The program first relocates to the highest order 4K stack under test. From there it relocates to the next lower stack (after performing all four tests). The program keeps relocating to the next lower stack until it reaches the lowest order stack under test. The testing and relocation cycle is then repeated.

The contents of the entire 4K stack are relocated. This enables the RIM Loader, and any other information to be carried with the program.

The program provides a degree of protection for itself by recording the first error encountered in any stack. When a faulty stack is next in sequence to contain the program, the program will skip the faulty stack and relocate to the first lower order stack which is error-free. If all lower order stacks are faulty, program relocation will not take place. The tests will be run again from the current stack. Relocation will resume when an error-free stack is found.

Also, the program will not relocate if any of the conditions described below exist.

- a. Only one 4K stack is selected for testing.
- b. SR 9 is on a 1 to inhibit relocation (see Section 8.2.6).

The INSTRUCTION FIELD indicators will indicate the current stack containing the program.

5.3.2 The Checkerboard Patterns

Four test patterns, and their complements, are used to test memory. All memory stacks, except the one with the program, are tested with one pattern before the next test is executed.

Any one, or any combination, of the four tests may be run by placing one, or any combination, of SR 3, 4, 5, or 6 on a 1 after the message SETUP SR is printed. The test specified by the most significant switch on a 1 will be executed first. SR 3, 4, 5 and 6 all on a 0 will enable all tests to be run. SR 3= test 1; 4= test 2; 5= test 3; 6= test 4.

The following steps are performed by each of the four tests:

- a. Write the pattern once in all stacks selected for testing; starting with the lowest order stack.
- b. Select the lowest order stack and perform a read, complement data, write sequence once on each location, until all 4K has been complemented.
- c. Repeat step b 31 more times. The stack will end up with the pattern originally loaded.
- No error checking has been performed as yet.
- d. Read 4-word segments and complement each segment 4 times; then read each of the 4 words and check for error.
- e. After checking the entire 4K stack for errors, repeat step d again. This time stall for a random period of time after reading and checking every 400_g word block. The maximum stall is 18.4 ms; the minimum is 3 μ s.
- f. Setup for the next sequential 4K stack and repeat steps b through f.

When all selected stacks have been checked the next test in sequence is executed, and steps a through f repeated. Program relocation takes place after the fourth test is executed in this manner.

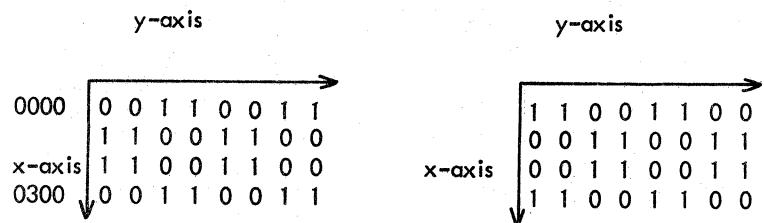
The patterns generated by each test are shown below. The matrices represent portions of one bit plane.

Test 1:

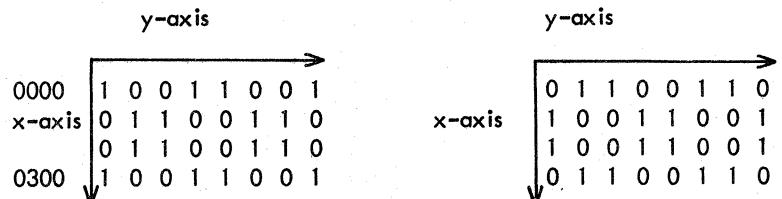
		y-axis				y-axis	
0000		0	0	1	1	0	0
x-axis		0	0	1	1	0	0
0300		1	1	0	0	1	1
		1	1	0	0	1	1

		y-axis				y-axis	
0000		1	1	0	0	1	1
x-axis		1	1	0	0	1	1
0300		0	0	1	1	0	0
		0	0	1	1	0	0

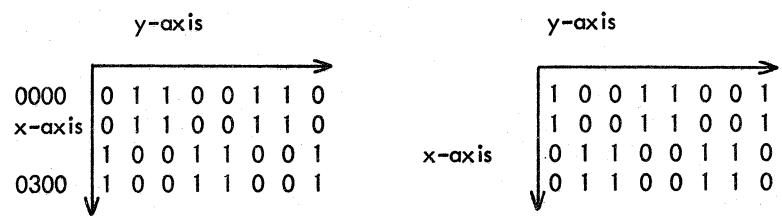
Test 2:



Test 3:



Test 4:



6.

ERRORS

Starting the program from address 200 will give a header print-out after the SR has been setup.

The header identifies the information printed when a data error is found. The header appears as:

	FIELD	OCTAL ADR.	GOOD	BAD	TEST
Where:	FIELD	= an octal number (0 to 7) indicating the 4K field containing the error.			
	OCTAL ADR.	= the memory address which contains the incorrect data.			
	GOOD	= what the data in octal, should have been. This will always equal 0000 or 7777.			
	BAD	= the data as read. This will equal the good data except for one or more bits complemented.			
	TEST	= the number (1 to 4) of the test which detected the error.			

After each error print-out the program continues on with the next sequential memory location.

6.1 Error Halts and Description

Placing SR 0 on a 1 during an error print-out will cause a halt at location 2641.
Press CONTINUE to resume testing.

7. RESTRICTIONS

7.1 Starting Restrictions

Start from address 200 to indicate the amount of core memory to test; to setup the SR and to receive a header print-out.

Starting from 207 requires the same operator action, but no header will be printed.

7.2 Operating Restrictions

None

8. MISCELLANEOUS

8.1 Execution Time

The time required to perform all four tests on one 4K memory stack is approximately 26 seconds.

8.2 Applications

For operating convenience, and as an aid to trouble-shooting, the SR may be used to control the program. The switch assignments and their effect on the program are described below. Please note that it is important that the program should be halted before changing the test selection switches. These switches are not sensed by the program during testing.

Halting the program with SR 0 is preferred, rather than with the STOP key. Using the STOP key may result in a halt while the program is in the process of relocating, which is disastrous.

8.2.1 Halt after Test or Error - SR 0

Placing SR 0 on a 1 at any time while the program is running will cause a halt after the current test is completed. The MB will equal 2461 in the current stack containing the program. Press CONTINUE to resume testing, or restart from 200 or 207 to enter new parameters.

Placing SR 0 on a 1 during an error type-out will also cause a halt at location 2461. Proceed exactly as described in the above paragraph.

8.2.2 Inhibit Error Print-out - SR 1

Placing SR 1 on a 1 causes all error print-outs to be inhibited. All other messages will not be inhibited. The program will continue to recognize errors, but will not print any information. SR 1 may be placed on a 1 or 0 while the program is running.

8.2.3 Bell on Error - SR 2

SR 2 on a 1 causes the program to ring the TTY BELL whenever an error is detected. This is convenient when testing with power supply margins. SR 2 has precedence over SR 1 if both should happen to be on a 1. SR 2 may be placed on a 1 or 0 while the program is running.

8.2.4 Test Selection SR 3 through 6

Any one, or any combination of tests may be executed by placing any one or any combination of SR 3 through 6 on a 1. Test selections may be made only when starting from 200 or 207. SR 3 specifies test 1; SR 4 test 2; SR 5 test 3; SR 6 test 4. The test specified by the most significant SR on a 1 will be executed first.

For most PDP-8s, SR 4 will provide the worst case pattern. For most PDP-8/Is, SR 5 will provide the worst case pattern.

If all four switches are on a 0, all four tests will be executed in order starting with test 1. Program relocation is not effected, regardless of the SR settings.

8.2.5 Inhibit Program Relocation - SR 7

The program normally relocates automatically as indicated by the INSTRUCTION FIELD indicators. To retain the program in its current 4K field, place SR 7 on a 1 at any time. Changing SR 7 to a 0 will permit relocation to resume.

8.2.6 SR 8, 9 and 10 - Not Used

8.2.7 Change TEST LIMITS and SR - SR 11

Placing SR 11 on a 1 will cause the program to automatically restart from address 207. The TEST LIMITS and SR may then be changed. SR 11 is sensed only after all specified tests have been completed on all of memory under test.

8.2.8 Loop on Address

A subroutine is provided which may be used to continuously loop on a single location, or a group of consecutive locations. No error checking is performed. The routine performs a read, and immediately follows with a write, on each location. The loop time between two reads, or two writes, is approximately 22.5 μ s.

Operating Procedure:

- a. Set the INSTRUCTION FIELD switches to the current field, and the SR to 1700.
- b. Set the DATA FIELD switches to equal the 4K field number to test.
- c. Press LOAD ADDRESS.
- d. Set the SR to equal the first address of the group.
- e. Press START. A halt will occur at 1703. Set the SR to equal the last address of the group.
- f. Press CONTINUE. The address(s) specified will be looped until stopped by the operator with STOP. SR 0 will not halt this routine.

To resume normal operation, restart the program from 200 or 207 of the current field.

9. PROGRAM DESCRIPTION

The PDP-8, 8/I Extended Memory Checkerboard diagnostic is designed to create worst case memory noise conditions on systems equipped with 8K to 32K words of memory. The program executes four checkerboard patterns, plus their complements, on each 4K memory field. In addition, the program automatically relocates from field to field in order to test all 4K fields from every 4K field. Under normal operation, the amount of core memory tested at one time is that specified by the operator minus the 4K field containing the program. A TTY keyboard input routine is provided to enable the operator to specify the exact number of 4K fields to be tested. A print-out is provided for each error detected by the program.

Further control of the program is given to the operator by means of the SR. The operator may halt the program, inhibit error print-outs, substitute the TTY BELL for error indication, halt after error print-out, select any one or a group of tests, inhibit program relocation, and create an automatic restart to change the amount of memory to test.

A small subroutine is provided which will continuously read and write any single, or a group of locations within any 4K field. The operator must specify the locations by means of the SR.

/PDP-8, 81, 88 EXTENDED MEMORY CHECKBOARD TEST.
 /COPYRIGHT 1971, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
 /START AT 200, RESTART AT 211 TO SKIP HEADER.
 /MIN. OF 8K OF CORE REQUIRED.

2001
 5000 0001 5001
 0 0002 0002
 0 0003 0003

/*1
 6201 COF=6201
 6202 CIF=6202
 6214 RDF=6214
 6224 RIF=6224
 0004 0000 0
 0005 0000 0
 0006 2000 0
 0007 2000 0
 45350010 0000 0
 45350011 0000 0
 24020012 0000 0
 0013 0000 0
 0014 0000 DAYFLD, 0
 0015 0000 FLAGS, 0
 0016 0000 INSLFD, 0
 0017 0000 LOAD, 0
 0020 0022 ERFBBL, ERWRD,
 0021 0032 ENPBL, ERWRD+10
 0022 0022 ERWRD,
 0023 7600 7600
 0024 7600 7600
 0025 7600 7600
 0026 7600 7600
 0027 7600 7600
 0030 7600 7600
 0031 7600 7600
 0032 7600 7600
 0033 0000 MCWA, 0
 0034 0010 K10, 10
 0035 0740 K740, 740
 0036 0400 K400, 400
 0037 0200 K200, 200
 0040 0100 K100, 100
 0041 0040 K40, 40
 0042 0020 K20, 20
 0043 2207 XLMTS, SLMTS
 0044 0300 XTST1, TST1
 0045 0400 XTST2, TST2
 0046 2452 XTST3, TST3
 0047 0600 XTST4, TST4
 0052 3000 XMOVE, CMOVE
 0051 2200 XSETU, SETU
 0052 2261 K261, 261

**

*

PAL12 V141 2=NOV=71 0119 PAGE 1-1

00053 02622 K2622, 262
00054 02632 K263, 263
00055 02642 K264, 264
00056 77602 K20, 7760

00057 77402 440, 7742
00060 77742 774, 774
00061 77732 775, 773
00062 00000 TNUM, CBANK,
00063 16072 XBANK,
00064 06522 X0011, W0011
00065 06672 X110, W110
00066 07042 X0110, W0110
00067 07212 X1001, W1001
00070 16002 XKBNK,
00071 16242 XTBNK,
00072 00000 COUNT, 0
00073 00000 FLCNT, 0
00074 00000 LOOP, 0
00075 07362 XRALL,
00076 10372 RCHK1, RCHK1C,
00077 10542 XCHHK1C,
0100 10712 XCHHK2, RCHK2C
0101 11062 XCHHK3, RCHK3C
0102 11232 XCHHK3C, RCHK3C
0103 11402 XCHHK4, RCHK4C
0104 12002 XCHHK4C, RCHK4C
0105 12172 XCHHK4C, RCHK4C
0106 10562 TOM20, TAD M20
0107 10572 TOM40, TAD M40
0110 45152 JMS1, JMS1 XRD1
0111 45162 JMS2, JMS1 XRD2
0112 45172 JMS3, JMS1 XRD3
0113 45202 JMS4, JMS1 XRD4
0114 45522 JMS5, JMS1 XSALL
0115 12452 XRD1, RD1
0116 13222 XRD2, RD2
0117 14002 XRD3, RD3
0120 14552 XRD4, RD4
0121 20002 XRROR, ERROR
0122 00000 MEMADR, 0
0123 00000 FIRST1, 0
0124 00000 LAST1, 0
0125 62012 KCDF, 6201
0126 62022 KCIF, 6202
0127 26412 XHLT, HALT
0130 02132 XRTN, RTN1
0131 16462 XFILD, FEILD
0132 21462 XPRER, PRERR
0133 00072 K7, 7
0134 00022 CHAR, 2
0135 24742 XHDR, PHDR
0136 21462 XPERR, PRERR
0137 21152 XPING, SPING

0 2 6

7752
4536

0 2 5

0140 7764 414, 7764
 0141 7770 412, 7770
 0142 0260 X260, 260
 0143 0215 K215, 215
 0144 0377 K377, 377
 0145 0370 K370, 370
 0146 0277 K277, 277
 0147 2154 XCRLF,
 0150 0001 K1, 1
 0151 0000 NXLOC, 0
 0152 2702 XSALL,
 0153 0000 EXIT, 0
 0154 2166 LASTX,
 LAST

0200 *200 /PI OFF
 0201 BEGIN, IOF
 0202 3015 CLA FLAGS
 0203 6224 RIF
 0204 3016 DCA INSLFD
 0205 4443 JMS I XLMTS
 0206 4677 JMS I XSTSR
 0207 4535 JMS I XHDR
 0210 5213 JMP RTN1

/RESTART HERE

0211 4443 RSTRT1, JMS I XLMTS /SET TEST LIMITS
 0212 4677 JMS I XSTSR /SETUP SR
 0213 6224 RIF /READ INSTRUCTION FIELD
 0214 3016 DCA INSLFD /CURRENT FIELD
 0215 4531 JMS I XFILD
 0216 1141 TAD M10 /=10
 0217 3074 DCA LOOP
 0220 7600 ALAW, 7600
 0221 1220 TAD ALAW
 0222 2022 ISZ ERWRD
 0223 3422 DCA I ERWRD
 0224 2074 ISZ LOOP
 0225 5220 JMP ALAW
 0226 1220 TAD ALAW
 0227 3554 DCA I LASTX
 0230 1020 TAD ERTBL
 0231 3022 DCA ERWRD

/EXAMINE SR
 0232 1033 TAD MCWA
 0233 0035 AND K740
 0234 7440 S2A /MADK 3,4,5 AND 6
 0235 5241 JMP EXAM1
 0236 1033 TAD MCWA
 0237 1035 TAD K740
 /SET ALL TEST BITS

/CLEAR PROGRAM FLAGS

/SETUP TEST LIMITS
 /SETUP SR
 /PRINT HEADER

PAL10 V141 2=NOV=71 0119 PAGE 1=3

0244 3033 DCA MCWA /SAVE
0241 7200 EXAM1, CLA
0242 1033 TAD MCWA
0243 0036 AND K422
0244 7440 SZA /TEST 1 IF NO SKIP
0245 5444 JMP I XTST1
0246 7200 EXAM2, CLA
0247 1033 TAD MCWA
0250 0037 AND K200 /TEST 2 IF NO SKIP
0251 7440 SZA
0252 5445 JMP I XTST2
0253 7200 EXAM3, CLA
0254 1033 TAD MCWA
0255 0040 AND K100 /TEST 3 IF NO SKIP
0256 7440 SZA
0257 5446 JMP I XTST3
0260 7200 EXAM4, CLA
0261 1033 TAD MCWA
0262 0041 AND K40 /TEST 4 IF NO SKIP
0263 7440 SZA
0264 5447 JMP I XTST4
0265 4531 JMS I XFILD /RESTORE DATA FIELD

0266 7604 LAS
0267 0150 AND K1 /CHECK SR 11
0270 7440 SZA
0271 5211 JMP RSTR1
0272 7604 LAS
0273 0042 AND K20 /INHIBIT MOVE IF A 1
0274 7440 SZA
0275 5213 JMP RTN1
0276 5450 JMP I XMOVE /GO RELOCATE
0277 2645 XSTSR, SETSR

/TEST 1. WRITE CHECKER PATTERN #1.
TST1, JMS I XSETU /SET OF TO 1ST FIELD
0300 4451 CLA
0301 7200 TAD K261 /TEST NUMBER
0302 1052 DCA TNUM
0303 3062 CMA
0304 7040 DCA 10 /SET ADDRESS COUNT TO 7777
0305 3010 JMS I XBANK /SEE IF FIELD HAS PROGRAM
0306 4463 SKP /NO. BEGIN WRITING
0307 7410 JMP EXIT /DONE ALL. NOW READ ALL
0310 5322 TAD KXT1
0311 1345 DCA EXIT
0312 3153 JMS I X0011 /WRITE 0011
0313 4464 JMS I X0011 /WRITE 0011 64 TIMES
0314 4464 JMS I X1100 /WRITE 1100 128 TIMES
0315 4465 JMS I X1100
0316 4465 JMS I X1100

HAL10

PAGE 1

V141 2*NOV*71

PAGE 1*4

0317 5313 JMP *4 /KEEP WRITING
JMS 1 XTBNK /SETUP FOR NEXT FIELD
JMP TST1+4
0320 4471 XIT1,
0321 5304 / EXT1,
0322 4476 JMS 1 XCHK1 /READ EACH FIELD AND CHECK
FOR ERRORS,
0323 5324 JMP TST1C /NOW WRITE COMPLEMENT

/ WRITE COMPLEMENT OF PATTERN 1
0324 4451 TST1C, JMS 1 XSETU /SEE DF TO 1ST FIELD.
0325 7240 CLA CMA
0326 3010 DCA 10 /SET ADDRESS COUNT TO 7777
0327 4463 JMS 1 XBANK /SEE IF FIELD HAS PROGRAM
0328 7410 SKP
0331 5343 JMP EXT1C /ALL DONE, READ ALL
0332 1346 TAD KXT1C
0333 3153 DCA EXIT
0334 4465 JMS 1 X1100 /WRITE 1100
0335 4465 JMS 1 X1100 /WRITE 1100 16 TIMES
0336 4464 JMS 1 X0011 /WRITE 0011 128 TIMES
0337 4464 JMS 1 X0011 /KEEP WRITING
0340 5334 JMP *4 /SETUP FOR NEXT FIELD
0341 4471 XIT1C,
0342 5325 JMP TST1C+1

0343 4477 EXT1C, JMS 1 XCHK1C /READ EACH BANK AND CHECK
FOR ERRORS!
0344 5246 JMP EXAM2 /SEE IF TEST 2 IS SELECTED

0345 0320 KXT1,
0346 0341 XIT1,
XIT1C,
XIT1C

/ TEST 2. WRITE CHECKER PATTERN #2
0400 4451 TST2,
0401 7200 JMS 1 XSETU /SET DF FOR 1ST FIELD
0402 1053 CLA TAD K262 /TEST #
0403 3062 DCA TNUM
0404 7240 CLA CMA
0405 3010 DCA 10 /SET ADDRESS COUNT TO 7777
0406 4463 JMS 1 XBANK /SEE IF FIELD HAS PROGRAM
0407 7410 SKP /NO. BEGIN WRITING
0410 5223 JMP EXT2 /DONE ALL, NOW READ ALL
0411 1250 TAD KXT2
0412 3153 DCA EXIT
0413 4464 JMS 1 X0011 /WRITE 0011
0414 4465 JMS 1 X1100 /WRITE 1100 128 TIMES
0415 4465 JMS 1 X1100 /WRITE 0011 128 TIMES
0416 4464 JMS 1 X0011 /WRITE 0011 128 TIMES
0417 4464 JMS 1 X0011

PAL12 V141 2-NOV-71 0119 PAGE 1*5
/ 0420 5214 JMP *4 JMS 1 XTBNK /SETUP FOR NEXT FIELD
0421 4471 X172, JMS 1 XTBNK
0422 5224 JMP TST2+4
0423 4522 EXT2, JMS 1 XCHK2 /READ EACH FIELD AND CHECK
0424 5225 JMP TST2C /NOW WRITE COMPLEMENT

/ WRITE COMPLEMENT OF PATTERN 2

/ TST2C, JMS 1 XSETU /SET UP FOR FIRST FIELD
0425 4451 CLA CMA
0426 7240 DCA 10 /SET ADR, COUNT TO 7777
0427 3010 JMS 1 XBANK /SEE IF FIELD HAS PROGRAM
0430 4463 SKP
0431 7410 JMP EXT2C /WRITE /GO READ
0432 5245 TAD KXT2C
0433 1251 DCA EXIT
0434 3153 JMS 1 X1100 /WRITE 1100
0435 4465 JMS 1 X0011 /WRITE 0011 128 TIMES
0436 4464 JMS 1 X0011
0437 4464 JMS 1 X0011
0440 4465 JMS 1 X1100 /WRITE 1100 128 TIMES
0441 4465 JMS 1 X1100
0442 5236 JMP *4
0443 4471 X172C, JMS 1 XTBNK /SETUP FOR NEXT FIELD
0444 5226 JMP TST2C+1
0445 4501 EXT2C, JMS 1 XCHK2 /READ EACH FIELD AND CHECK
0446 5647 JMP 1 .+1 /SEE IF TEST 3 IS SELECTED
0447 0293 EXAM3
0450 0421 KXT2, X172
0451 0443 KXT2C, X172C
/

/ TEST 3. WRITE CHECKER PATTERN #3

/ TST3, JMS 1 XSETU /SETUP FOR 1ST FIELD
0452 4451 CLA TNUM /TEST NUMBER
0453 7200 TAD K263
0454 1054 DCA CMA
0455 3062 DCA 10 /SET ADR, COUNT TO 7777
0456 7240 JMS 1 XBANK /SEE IF FIELD HAS PROGRAM
0457 3010 SKP
0460 4463 JMP EXT3 /GO WRITE /GO READ
0461 7410 TAD KXT3
0462 5275 DCA EXIT
0463 1322 JMS 1 X1001 /WRITE 1001
0464 3153 JMS 1 X0110 /WRITE 0110 128 TIMES
0465 4467 JMS 1 X0110
0466 4466 JMS 1 X0110 /WRITE 1001 128 TIMES
0467 4466 JMS 1 X1001
0470 4467 JMS 1 X0001 /WRITE 0001
0471 4467 JMP *4
0472 5266 JMS 1 XTBNK /SETUP FOR NEXT FIELD
0473 4471 X173,

PAL10 V141

2-Nov-71 2119 PAGE 1-6

2474 5256 / JMP TST3+4

2475 4502 EXT3, JMS I XCHK3 /READ EACH FIELD AND CHECK
2476 5277 JMP TST3C /WRITE COMPLEMENT

/ WRITE COMPLEMENT OF PATTERN 3

0477 4451 / TST3C, JMS I XSETU /SETUP DF FOR 1ST FIELD

0500 7240 CLA CMA
0501 3010 DCA 10 /SET ADR, COUNT TO 7777
0502 4463 JMS I XBANK /SEE IF FIELD HAS PROGRAM

0503 7410 SKP /WRITE
0504 5317 JMP EXT3C /READ ALL

0505 1323 TAD KXT3C
0506 3153 DCA EXIT

0507 4466 JMS I X0110 /WRITE 0110
0510 4467 JMS I X1001 /WRITE 1001 128 TIMES

0511 4467 JMS I X1001 /WRITE 0110 128 TIMES

0512 4466 JMS I X0110 /WRITE 0110 128 TIMES

0513 4466 JMS I X0110 /WRITE 0110 128 TIMES

0514 5310 JMP ! =4 /SETUP FOR NEXT FIELD

0515 4471 XIT3C, JMS I XTBNK
0516 5300 JMP TST3C+1 /SEE IF TEST 4 IS SELECTED

0517 4503 EXT3C, JMS I XCHK3C /READ EACH FIELD AND CHECK

0520 5721 JMP ! .+1 /SEE IF TEST 4 IS SELECTED

0521 0260 EXAM4 /TEST 4. WRITE PATTERN #4

0522 0473 KXT3C, XIT3
0523 0515 KXT3C, XIT3C

0600 *600 / TEST 4. WRITE PATTERN #4

0600 /

0601 4451 TST4, JMS I XSETU /SET UP FOR 1ST FIELD

0602 7200 CLA K264
0603 1055 TAD TNUM
0604 3062 DCA CMA /TEST NUMBER

0605 7240 DCA 10 /SET ADR, COUNT TO 7777
0606 4463 JMS I XBANK /SEE IF FIELD HAS PROGRAM

0607 7410 SKP /WRITE
0610 5223 JMP EXT4 /GO READ

0611 1245 TAD KXT4
0612 3153 DCA EXIT

0613 4466 JMS I X0110 /WRITE 0110
0614 4466 JMS I X0110 /WRITE 0110 64 TIMES

0615 4467 JMS I X1001 /WRITE 1001 128 TIMES

0616 4467 JMS I X1001 /WRITE 1001 128 TIMES

0617 4466 JMS I X2110 /WRITE 1210 64 TIMES

0620 5214 JMP ! =4 /SETUP FOR NEXT FIELD

0621 4471 XIT4, JMS I XTBNK

PAL10 V141 2•NOV•71 0119 PAGE 1•7

```
0622 5204 JHP TST4*4  
0623 4504 /  
0624 5225 EXT4, JMS I XCHK4 /READ EACH FIELD AND CHECK  
0625 4451 TST4C, JMS I XSETU /SET UP FOR FIRST  
0626 7240 CLA CMA /SET ADR, COUNT TO 7777  
0627 3010 DCA 10 /SEE IF FIELD HAS PROGRAM  
0630 4463 JMS I XBANK /WRITE  
0631 7410 SKP /READ  
0632 5247 JMP EXT4C  
0633 1246 TAD KXT4C  
0634 3153 DCA EXIT  
0635 4467 JMS I X1001 /WRITE 1001 64 TIMES  
0636 4466 JMS I X1001 /WRITE X0110 128 TIMES  
0637 4466 JMS I X1110  
0640 4466 JMS I X0110  
0641 4467 JMS I X1001  
0642 5236 JMP *4  
0643 4471 XIT4C, JMS I XTBNK /SETUP FOR NEXT FIELD  
0644 5226 /  
0645 0621 KXT4, XIT4  
0646 0643 KXT4C, XIT4C  
  
0647 4505 EXT4C, JMS I XCHK4C /READ EACH FIELD AND CHECK  
0650 5651 JMP I *4 /SEE IF READY TO MOVE  
0651 0265 EXAM4*5  
  
/ROUTINE TO WRITE 0011  
W0011, 0  
0652 0000 TAD M20  
0653 1096 DCA COUNT /0  
0654 3072 DCA 1 10 /0  
0655 3410 DCA 1 10 /0  
0656 3410 DCA 1 10 /0  
0657 7040 CMA /1  
0660 3410 DCA 1 10 /1  
0661 7040 CMA /1  
0662 3410 DCA 1 10 /1  
0663 2072 IS2 COUNT /COUNT = -16 OR -32  
0664 5255 JMP W0011+3 /LOOP  
0665 4470 JMS I XKBNK /SEE IF END OF FIELD  
0666 5652 JMP I W0011 /EXIT  
  
/ROUTINE TO WRITE 1100  
W1100, 0  
0667 0000 TAD M20  
0670 1056 DCA COUNT /1  
0671 3072 CMA /1  
0672 7040 DCA 1 10 /1  
0673 3410 CMA /1
```

PAL10 V141

2=NOV=71

0736

0737

0740

0741

0742

0743

0744

0745

0746

0747

0748

0749

0750

0751

0752

0753

0754

0755

0756

0757

0758

0759

0760

0761

0762

0763

0764

0765

0766

0767

0768

0769

0770

0771

0772

0773

0774

0775

0776

0777

0778

0779

0780

0781

0782

0783

0784

0785

0786

0787

0788

0789

0790

0791

0792

0793

0794

0795

0796

0797

0798

0799

0800

0801

0802

0803

0804

0805

0806

0807

0808

0809

0810

0811

0812

0813

0814

0815

0816

0817

0818

0819

0820

0821

0822

0823

0824

0825

0826

0827

0828

0829

0830

0831

0832

0833

0834

0835

0836

0837

0838

0839

0840

0841

0842

0843

0844

0845

0846

0847

0848

0849

0850

0851

0852

0853

0854

0855

0856

0857

0858

0859

0860

0861

0862

0863

0864

0865

0866

0867

0868

0869

0870

0871

0872

0873

0874

0875

0876

0877

0878

0879

0880

0881

0882

0883

0884

0885

0886

0887

0888

0889

0890

0891

0892

0893

0894

0895

0896

0897

0898

0899

0900

0901

0902

0903

0904

0905

0906

0907

0908

0909

0910

0911

0912

0913

0914

0915

0916

0917

0918

0919

0920

0921

0922

0923

0924

0925

0926

0927

0928

0929

0930

0931

0932

0933

0934

0935

0936

0937

0938

0939

0940

0941

0942

0943

0944

0945

0946

0947

0948

0949

0950

0951

0952

0953

0954

0955

0956

0957

0958

0959

0960

0961

0962

0963

0964

0965

0966

0967

0968

0969

0970

0971

0972

0973

0974

0975

0976

0977

0978

0979

0980

0981

0982

0983

0984

0985

0986

0987

0988

0989

0990

0991

0992

0993

0994

0995

0996

0997

0998

0999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

PAL10 V141 2-Nov-91 0119 PAGE 1-9

0745 3011 DCA 11
0746 3073 DCA FLCNT
0747 4463 JMS I XBANK
0750 7410 SKP /SEE IF FIELD HAS PROGRAM
0751 5360 JMP CDON1 /READ
0752 7200 CLA /DONE
0753 1410 TAD I 10 /READ ONE
0754 7040 CMA /WRITE BACK
0755 3411 DCA I 11 /DONE 1 FIELD WHEN SKIP
0756 2073 ISZ FLGNT /DONE 2 FIELD WHEN SKIP
0757 5353 JMP RDLOP /DONE 32 PASSES WHEN SKIP
0760 2072 CDON1: ISZ COUNT /DO ANOTHER PASS
0761 5342 JMP RDALL+4 /EXIT
0762 5736 JMP I RDALL /READ AND CHECK FOR ERROR ROUTINE
/ *1000
1000 /
1001 RCHKA: 0 JMS I XSETU /SET OF TO 1ST FIELD
1002 4451 JMS I XRALL /READ ALL, DON'T CHECK
1003 3122 DCA MEMADR /SET ADR, COUNT TO 0
1004 4463 JMS I XBANK /SEE IF FIELD HAS PROGRAM
1005 7410 SKP
1006 5600 JMP I RCHKA
1007 1235 TAD KRXT
1010 3153 DCA EXIT
1011 0000 RLOPA: 0 /WILL = JMS I XRD1, 2, 3, OR 4
1012 0000 0
1013 0000 0
1014 0000 0
1015 0000 0
1016 7000 NOP /WILL = NOP OR JMS STALL
1017 5212 JMP RLOPA+1
1020 1216 RXIT: TAD I^{*2}
1021 1236 TAD K1K /NOP IF 0
1022 7640 SZA CLA /NOP
1023 5227 JMP I^{*4} /JMS I XSALL
1024 1114 TAD JMSS /READ SLOW
1025 3216 DCA RXIT=2
1026 5203 JMP RCHKA+3 /SETUP FOR NEXT FIELD
1027 7000 NOP
1030 1227 TAD I^{*1}
1031 3216 DCA RXIT=2
1032 4471 JMS I XTBNK
1033 5202 JMP RCHKA+2 /EXIT
1034 5600 JMP I RCHKA
/ 1035 1020 KRXT, RXIT
1036 1000 K1K, 1000 /SETUP ROUTINES FOR RCHKA

PAL10 V141

2*NOV*71 0119 PAGE 1-10

1037 0000 RCHK1, 0 /JMS1 = JMS ! XRD1
1040 1110 TAD JMS1
1041 3211 DCA RLOPA
1042 1110 TAD JMS1
1043 3212 DCA RLOPA+1
1044 1111 TAD JMS2
1045 3213 DCA RLOPA+2
1046 1111 TAD JMS2
1047 3214 DCA RLOPA+3
1050 1110 TAD JMS1
1051 3215 DCA RLOPA+4
1052 4200 JMS RCHKA
1053 5637 JMP I RCHK1 /EXIT /GO READ

1054 0000 /RCHK1C, 0 /JMS2 = JMS ! XRD2
1055 1111 TAD JMS2
1056 3211 DCA RLOPA
1057 1111 TAD JMS2
1060 3212 DCA RLOPA+1
1061 1110 TAD JMS1
1062 3213 DCA RLOPA+2
1063 1110 TAD JMS1
1064 3214 DCA RLOPA+3
1065 1111 TAD JMS2
1066 3215 DCA RLOPA+4
1067 4200 JMS RCHKA
1070 5654 JMP I RCHK1C /EXIT /GO READ

1071 0000 /RCHK2, 0 /JMS1 = JMS ! XRD1
1072 1110 TAD JMS1
1073 3211 DCA RLOPA
1074 1111 TAD JMS2
1075 3212 DCA RLOPA+1
1076 1111 TAD JMS2
1077 3213 DCA RLOPA+2
1100 1110 TAD JMS1
1101 3214 DCA RLOPA+3
1102 1110 TAD JMS1
1103 3215 DCA RLOPA+4
1104 4200 JMS RCHKA
1105 5671 JMP I RCHK2 /EXIT /GO READ

1106 0000 /RCHK2C, 0 /JMS2 = JMS ! XRD1
1107 1111 TAD JMS2
1110 3211 DCA RLOPA
1111 1110 TAD JMS1
1112 3212 DCA RLOPA+1
1113 1110 TAD JMS1
1114 3213 DCA RLOPA+2
1115 1111 TAD JMS2
1116 3214 DCA RLOPA+3
1117 1111 TAD JMS2
1120 3215 DCA RLOPA+4
1121 4200 JMS RCHKA /GO READ

PAL10 V141 2-Nov-71 0119 PAGE 1-11
11222 5706 JMP I RCHK2C /EXIT

1123 0000 /RCHK3, 0 TAD JMS4 /JMS I XR04
1124 1113 DCA RLOPA /JMS I XR03
1125 3211 TAD JMS3 /JMS I XR03
1126 1112 DCA RLCPA+1
1127 3212 TAD JMS3 /JMS I XR03
1130 1112 TAD JMS3 /JMS I XR03
1131 3213 DCA RLOPA+2
1132 1113 TAD JMS4 /JMS I XR03
1133 3214 DCA RLOPA+3
1134 1113 TAD JMS4 /JMS I XR04
1135 3215 DCA RLOPA+4
1136 4200 JMS RCHKA /GO READ
1137 5723 JMP I RCHK3 /EXIT /GO READ

1140 0000 /RCHK3C, 0 TAD JMS3 /JMS I XR03
1141 1112 DCA RLOPA /JMS I XR04
1142 3211 TAD JMS4 /JMS I XR03
1143 1113 DCA RLOPA+1
1144 3212 TAD JMS4 /JMS I XR04
1145 1113 DCA RLOPA+2
1146 3213 TAD JMS3 /JMS I XR03
1147 1112 TAD JMS3 /JMS I XR04
1148 3214 DCA RLOPA+3
1150 1112 TAD JMS3 /JMS I XR04
1151 1112 DCA RLOPA+4
1152 3215 JMS RCHKA /GO READ
1153 4200 JMP I RCHK3C /EXIT
1154 5740 / *1200 /

1200 0000 /RCHK4, 0 JMS IXFIELD /JMS I XR03
1201 4531 TAD JMS3 /JMS I XR03
1202 1112 DCA IXLOPA /JMS I XR03
1203 3637 TAD JMS3 /JMS I XR03
1204 1112 DCA IXLOPB /JMS I XR03
1205 3640 TAD JMS4 /JMS I XR04
1206 1113 DCA IXLOPC /JMS I XR04
1207 3641 TAD JMS4 /JMS I XR04
1210 1113 DCA IXLOPD /JMS I XR04
1211 3642 TAD JMS3 /JMS I XR04
1212 1112 DCA IXLOPE /JMS I XR04
1213 3643 DCA IXCFL /JMS I XR04
1214 4644 JMS I XCHKA /GO READ
1215 4636 JMP I RCHK4 /EXIT
1216 5600 /RCHK4C, 0 JMS IXFIELD /JMS I XR04
1217 0000 TAD JMS4 /JMS I XR04
1220 4531 DCA IXLOPA /JMS I XR04
1221 1113 DCA IXLOPD /JMS I XR04
1222 3637 DCA IXCFL /JMS I XR04
1223 1113 TAD JMS4 /JMS I XR04

PAL10 V141 2*NOV-71 0119 PAGE 1*12

1224 3640 DCA 1 XLOPB
1225 1112 TAD JMS3
1226 3641 DCA 1 XLOPC
1227 1112 TAD JMS3
1230 3642 DCA 1 XLOPD
1231 1113 TAD JMS4
1232 3643 DCA 1 XLOPE
1233 4644 JMS 1 XCFL
1234 4636 JMS 1 XCHKA
1235 5617 JMP 1 RCHKA/C /EXIT /GO READ

PAUSE

1236 1000 XCHKA/RCHKA
1237 1011 XLOPA/RLOPA
1240 1012 XLOPB/RLOPA+1
1241 1013 XLOPC/RLOPA+2
1242 1014 XLOPD/RLOPA+3
1243 1015 XLOPE/RLOPA+4
1244 1722 CFLD

/81-88 EXTENDED CHECKERBOARD - TAPE 2
/READ ROUTINES FOR #0111: 01101 0110 AND 1001
/RD1, 0 TAD M20 //#16

1245 0000 RD1, 0 TAD M20 //#16
1246 1036 DCA COUNT //#4
1247 3072 TAD M4 //#4
1250 1069 DCA FLCNT //#8
1251 3073 TAD M10 //#8
1252 1141 CLOPI,
1253 3074 TAD M10
1254 1522 DCA LOOP
1255 7040 TAD 1 MEMADR
1256 3522 CMA
1257 2074 ISZ LOOP
1258 5254 JMP //#4
1261 2073 ISZ FLCNT
1262 7410 SKP
1263 5266 JMP //#3
1264 2122 ISZ MEMADR
1265 5252 JMP CLOP1

/COMPLEMENT 8 TIMES
/DONE 4 ADRS, WHEN SKIP

1266 1122 TAD MEMADR
1267 1060 TAD M4 //SUBTRACT 4
1270 3010 DCA 10 //NOW USE AUTO-INDEX
1271 7100 CLL
1272 1410 TAD 1 10
1273 7440 SZA //
1274 4521 JMS 1 XRROR //PRINT ERROR
1275 7100 CLL
1276 1410 TAD 1 10
1277 7442 SZA //
1300 4521 JMS 1 XRROR //PRINT ERROR

PAL10 V141 2•NOV•71 2119 PAGE 1•13

1301 7120 STL
1302 1410 TAD I 10 /1
1303 7040 CMA
1304 7440 SZA JMS I XRROR /PRINT ERROR
1305 4521 STL
1306 7120 TAD I 10 /1
1307 1410 CMA
1310 7040 SZA JMS I XRROR /PRINT ERROR
1311 7440 STL
1312 4521 TAD I 10 /1
1313 2072 COUNT ISZ
1314 5320 *4 JMP
1315 4470 JMS I XKBNK /SEE IF END OF FIELD
1316 2122 ISZ MEMADR
1317 5645 JMP I RD1 /KEEP READING
1320 2122 ISZ MEMADR
1321 5250 JMP RD1+3 /
1322 0000 RD2,
1323 1056 TAD M20 /#16
1324 3072 DCA COUNT /#4
1325 1060 TAD M4 /
1326 3073 DCA FLCNT /#8
1327 1141 CLOP2, TAD M10 /
1330 3074 DCA LOOP /READ
1331 1522 TAD I MEMADR /
1332 7040 CMA
1333 3522 DCA I MEMADR /COMPLEMENT 4 TIMES
1334 2074 ISZ LOOP /
1335 5331 *4 JMP
1336 2073 ISZ FLCNT /DONE 4 ADRS. WHEN SKIP
1337 7410 SKP
1340 5343 JMP *3 ISZ MEMADR
1341 2122 JMP CLOP2 /INCREMENT ADDRESS
1342 5327 /
1343 1122 TAD MEMADR
1344 1060 TAD M4 /NOW USE AUTO-INDEX
1345 3010 DCA 10 /1
1346 7120 STL
1347 1410 TAD I 10 /1
1350 7040 CMA
1351 7440 SZA JMS I XRROR /PRINT ERROR
1352 4521 STL
1353 7120 TAD I 10 /1
1354 1410 CMA
1355 7040 SZA JMS I XRROR /PRINT ERROR
1356 7440 CLL
1357 4521 CLL
1360 7100 TAD I 10 /1
1361 1410

PAL10 V141

2*NOV*71 0119 PAGE 1-14

1362	7440	SZA	/O
1363	4521	JMS I XRROR	/PRINT ERROR
1364	7100	CLL	
1365	1410	TAD I 10	/O
1366	7440	SZA	/PRINT ERROR
1367	4521	JMS I XRROR	/PRINT ERROR
1370	2072	1\$Z COUNT	
1371	5375	JMP ⁱ⁺⁴	
1372	4470	JMS I XKBNK	/SEE IF END OF FIELD
1373	2122	1\$Z MEMADR	
1374	5722	JMP I RD2	
1375	2122	1\$Z MEMADR	/KEEP READING
1376	5325	JMP RD2+3	
/ *1400			
1400	0000	/RDS,	
1401	1056	0 TAD M20	/=16
1402	3072	DCA COUNT	
1403	1060	TAD M4	/=4
1404	3073	DCA FLCNT	
1405	1141	CLOPS,	
1406	3074	TAD M16	/=8
1407	1322	DCA LOOP	
1410	7040	TAD I MEMADR	/READ
1411	3522	CMA	
1412	2074	DCA I MEMADR	/COMPLEMENT 8 TIMES
1413	5207	1\$Z LOOP	
1414	2073	JMP ⁱ⁺⁴	
1415	7410	1\$Z FLCNT	/DONE 4 IF 0
1416	5221	SKP ⁱ⁺³	
1417	2122	JMP ⁱ⁺³	
1420	5205	1\$Z MEMADR	/DO NEXT
1421	1122	JMP CLOPS	
1422	1060	TAD MEMADR	
1423	3010	TAD M4	/USE AUTO-INDEX
1424	7100	DCA 10	
1425	1410	CLL	
1426	7440	TAD I 10	/O
1427	4521	SZA	/PRINT ERROR
1430	7120	STL	
1431	1410	TAD I 10	/1
1432	7040	CMA	
1433	7440	SZA	/PRINT ERROR
1434	4521	JMS I XRROR	/PRINT ERROR
1435	7120	STL	
1436	1410	TAD I 10	/1
1437	7040	CMA	
1440	7440	SZA	/PRINT ERROR
1441	4521	JMS I XRROR	/PRINT ERROR
1442	7100	CLL	

PAL10 V141 2=NOV=71 2119 PAGE 1=15

1443 1410 TAD I 10
1444 7440 SZA /PRINT ERROR
1445 4521 JMS I XRROR
1446 2072 ISZ COUNT
1447 5253 JMP *4
1450 4470 JMS I XKBNK /SEE IF END OF FIELD
1451 2122 ISZ MEMADR
1452 5600 JMP I RD3
1453 2122 ISZ MEMADR
1454 5203 JMP RD3+3

1455 0000 /
1456 1055 RD4, 0 TAD M20 /#16
1457 3072 DCA COUNT
1460 1060 TAD M4 /#4
1461 3073 CLOP4, DCA FLCNT
1462 1141 TAD M10 /#8
1463 3074 DCA LOOP
1464 1522 TAD I MEMADR
1465 7040 CMA
1466 3522 DCA I MEMADR
1467 2074 ISZ LOOP
1470 5264 JMP *4
1471 2073 ISZ FLCNT
1472 7410 SKP /DONE 4 ADRS: WHEN SKIP
1473 5276 JMP *3 /INCREMENT ADDRESS
1474 2122 ISZ MEMADR
1475 5262 JMS CLOP4
1476 1122 TAD MEMADR
1477 1060 TAD M4
1500 3010 DCA 10 /USE AUTO=INDEX
1501 7120 STL
1502 1410 TAD I 10 /1
1503 7040 CMA
1504 7440 SZA /PRINT ERROR
1505 4521 JMS I XRROR
1506 7100 CLL
1507 1410 TAD I 10 /
1510 7440 SZA /PRINT ERROR
1511 4521 JMS I XRROR
1512 7100 CLL
1513 1410 TAD I 10 /1
1514 7440 SZA /PRINT ERROR
1515 4521 JMS I XRROR
1516 7120 STL
1517 1410 TAD I 10 /
1520 7040 CMA
1521 7440 SZA /PRINT ERROR
1522 4521 ISZ COUNT
1523 2072 JMP *4
1524 5330 JMS I XKBNK /SEE IF END OF FIELD
1525 4470

FAL1.8 V141 2=NOV-71 3119 PAGE 1-16

1526 2122 ISZ MENADR
1527 5655 JMP 1 RD4

/ ISZ MENADR
1530 2122 JMP RD4+3
1531 5262

/ ROUTINE TO CHECK FOR END OF FIELD

/ *1600 CKBNK, 0 CLA
1601 7200 TAD 10
1602 1010 CMA
1603 7040 SZA CLA
1604 7640 SZA CLA
1605 5602 JMP 1 CKBNK /NOT DONE
1606 5553 JMP 1 EXIT /DONE

/ ROUTINE TO SEE IF TESTED FIELD HAS PROGRAM

/ CBANK, 0 RIF /READ INST. FIELD
1610 6224 DCA SAVIF /SAVE
1611 3223 RDF /READ DATA FIELD
1612 6214 CIA
1613 7041 TAD SAVIF
1614 1223 SZA CLA /EQUAL IF AC=0
1615 7640 JMP 1 CBANK /DOESN'T HAVE PROGRAM
1616 5607 JMS 1 XTBNK /INCREMENT DATA FIELD
1617 4471 JMP 1 CBANK /TEST NEW FIELD
1620 5607 ISZ CBANK /DONE ALL CAUSE PROGRAM NOW
1621 2207 CIA
1622 5607 JMP 1 CBANK /IN HIGHEST FIELD
1623 2000 SAVIF, 0 /EXIT

/ ROUTINE TO SET DF FOR NEXT FIELD

/ NXTBNK, 0 CLA
1624 0000 RDF /READ DATA FIELD
1625 7200 CIA
1626 6214 TAD LAST1 /ALL DONE IF 0
1627 7041 SZA CLA
1630 1124 JMP 1*3 NXTBNK
1631 7640 JMP 1*6 /EXIT
1632 5235 ISZ XTBNK
1633 2224 JMP ADD ,6201 /INCREMENT DATA FIELD
1634 5242 TAD K12 /ADD ,6201
1635 6214 DCA *1 /CHANGE TO NEW DATA FIELD
1636 1034 CDF 00
1637 1125
1640 3241
1641 6201

/CHECK SWITCH REGISTER

1642 7634 LAS SPA CLA /CHEC HALT
 1643 7712 RDF JMS I XHLT /GO HALT, SR0=1
 1644 4527 JMP I NXTBNK /EXIT
 1645 5624

/RESTORE DATA FIELD AND CHECK SR

1646 2000 FEILD, 0 CLA
 1647 7200 RDF DATFLD /SAVE TESTED FIELD#
 1650 6214 DCA RIF TAD KCDF
 1651 3014 DCA *+1 CDF 00 /MAKE DATA AND INST FIELD EQUAL
 1652 6224 CLA JMP I FEILD

1653 1125 TAD HLT
 1654 3255 DCA 00
 1655 6201 CLA
 1656 7200 JMP I FEILD

1657 5646

1700 *1700
 /START HERE TO LOOP ON ADDRESS

1700 7200 CLA /READ LOWER LIMIT
 1701 7604 LAS DCA FIRST1 /NOW SETUP UPPER LIMIT
 1702 3123 HLT
 1703 7402 LAS
 1704 7604 DCA LAST1
 1705 3124 OVER, TAD FIRST1
 1706 1123 DCA MEMADR
 1707 3122 WRLOP, TAD I MEMADR /READ
 1710 1522 DCA I MEMADR /WRITE
 1711 3522 TAD MEMADR
 1712 1122 CIA
 1713 7041 TAD LAST1
 1714 1124 SNA CLA
 1715 7650 JMP OVER
 1716 5306 ISZ MEMADR
 1717 2122 JMP WRLOP
 1720 5310 HLT
 1721 7402
 1722 2000 CFLD, 0 CLA /TEST FIELD
 1723 7200 TAD DATFLD
 1724 1014 TAD KCDF
 1725 1125 DCA *+1 CDF 00 /RESTORE TEST FIELD
 1726 3227 CLA JMP I CFLD /EXIT
 1727 6201
 1730 7202
 1731 5722

PAL10 V141

21NOV-71 2149 PAGE 1-18

/PRINT ERROR ROUTINE

*2232
ERROR,
S2L
CMA
DCA BAD
S2L
CMA
DCA GOOD
TAD 10
DCA OCADR
JMS I XFLD
TAD DATFLD
CIA
TAD LAST
SNA CLA
JMP SW2
TAD DATFLD
DCA LAST
TAD ERWRD
CIA ENTBL
TAD ENTBL
S2A CLA
JMP *3
TAD ERTBL
DCA ERWRD
TAD DATFLD
ISZ ERWRD
DCA I ERWRD

/READING 1'S IF LINK = 1
/SAVE BAD DATA
/OCTAL ADDRESS
/RESTORE DATA FIELD
/DATA FIELD
/LAST = FIELD WITH LAST ERROR
/SAME IF 0
/DON'T STORE
/TABLE POINTER
/END OF TABLE IF * 0

2023 1021
2024 7640
2025 5230
2026 1020
2027 3022
2030 1014
2031 2022
2032 3422
2033 7604
2034 7006
2035 7700
2036 5242
2037 1367
2040 4346
2041 5311
2042 7604
2043 7004
2044 7700
2045 5254
2046 1014
2047 1125
2050 3251
2051 6201
2052 7200
2053 5600
SW2,
LAS
RTL
SMA CLA
JMP SW1
TAD K207
JMS PRERR
JMP SW0
LAS
RAL
SMA CLA
JMP EPRNT
TAD DATFLD
TAD KCDF
DCA *1
CDF 00
CLA
JMP 1 ERROR

/RING BELL
/RESTORE POINTER
/INCREMENT POINTER
/STORE IN TABLE
/SR2 ON A 1 = RING BELL
/RING BELL
/SR1 A 1 = NO PRINT
/SET TO TESTED FIELD
/EPRNT,
JMS CRLF
TAD DATFLD
RTR
RAR

2054 4354
2055 1014
2056 7012
2057 7010
/EPRNT,
JMS CRLF
TAD DATFLD
RTR
RAR

PAL12

2119 PAGE 1-19

2-Nov-71

TAD K260 /PRINT

JMS PRERR /*12 DECIMAL
TAD M14
DCA LOOP
JMS SPING
TAD OCADR
DCA CHAR
JMS PROCTL
TAD M10 /*8 DECIMAL
DCA LOOP
JMS SPING
TAD GOOD
DCA CHAR
JMS PROCTL
TAD M5 /*5
DCA LOOP
JMS SPING
TAD BAD
DCA CHAR
JMS PROCTL
TAD M5
DCA LOOP
JMS SPING
TAD TNUM
JMS PRERR

SWO, LAS SPA CLA /*CH CK SR0
JMS I XHLT /*GO HALT
JMP EREXT /*EXIT

/PRINT SPACES
SPING, 0 TAD K240 /PRINT /SPACE
TLS
TSF
JMP *1
ISE LOOP
JMP SPING*2
CLA I SPING /EXIT

/PRINT OCTAL
PROCTL, 2 TAD M4
DCA LOOP
POSIN, TAD CHAR
CLL RAL
RTL
DCA CHAR

2112 1142 2062 1142
2261 4346 2051 1142
2262 1142 2052 1142
2263 3274 2053 3274
2264 4313 2054 1365
2265 1365 2055 3134
2266 3134 2056 4325
2267 4325 2057 1141
2272 3074 2058 3074
2272 4315 2059 1364
2273 1364 2060 4315
2274 3134 2061 1061
2275 4326 2062 3134
2276 1061 2063 4326
2077 3074 2064 1061
2100 4315 2065 3074
2101 1363 2102 3134
2102 3134 2103 4326
2103 4326 2104 1062
2104 1062 2105 3074
2105 3074 2106 4315
2106 4315 2107 1062
2107 1062 2110 4346
2110 4346

2111 7604 2112 7710
2113 4527 2114 5246

2115 2000 2116 1370
2117 6046 2120 5041
2121 5320 2122 2074
2123 5317 2124 7200
2125 5715

2126 7000 2127 1060
2130 3274 2131 1134
2132 7104 2133 7026
2134 3134

PAL10

V141

28NOV#71

0119

PAGE 1-20

```
2135 1134 TAD CHAR
2136 7004 RAL
2137 2133 AND K7
2140 1142 TAD K260
2141 4346 JMS PRERR
2142 2074 ISZ LOOP
2143 5331 POSITN /DO NEXT
2144 7200 CLA
2145 5726 JMP ! PROCTL /EXIT

/PRINT A NUMBER
2146 0000 PRERR, 0
2147 6046 TLS
2150 6041 TSF
2151 5350 JMP , -1
2152 7200 CLA
2153 5746 JMP ! PRERR /EXIT

/CARRIAGE RETURN, LINE FEED
2154 0000 CRLF, 0
2155 7200 CLA
2156 1143 TAD K215
2157 4346 JMS PRERR
2160 1371 TAD K212
2161 4346 JMS PRERR
2162 5754 JMP ! CRLF
2163 0000 BAD, 0
2164 0000 GOOD, 0
2165 0000 OCADR, 0
2166 0000 LAST, 0
2167 0000 K207, 207
2170 0240 K240, 240
2171 0212 K212, 212

/ROUTINE TO SET DF TO FIRST TEST FIELD
2200 0000 SETU1, 0
2201 7200 CLA
2202 1123 TAD FIRST1 /FIRST TO TEST
2203 1125 TAD KCDF
2204 3205 DCA , +1
2205 6201 CDF @0
2206 5600 JMP ! SETU1 /CHANGE TO TEST FIELD
/ROUTINE TO ACCEPT TEST LIMITS FROM
/KEYBOARD INPUT
```

PAL10 V141 2-NOV-71 0119 PAGE 1-21

2207 0000 SLMTS, 2 JMS I XFILE /CR, LF
2210 4531 JMS I XCRLF /PRINT TEST LIMITS
2211 4547 JMS I XTLIM /CR, LF
2212 4751 JMS I XCRLF /GO ACCEPT INPUT
2213 4547 JMS KEYIN /SEE IF IT'S LEGAL
2214 4303 JMS LEGAL
2215 4314 TAD CHAR
2216 1134 AND K7 /MASK AC 9=11
2217 0133 CLL RAL
2220 7104 RTL /POSITION TO AC 6=8
2221 7006 DCA FIRST1 /FIRST TO TEST
2222 3123 JMS KEYIN /WAIT FOR COMMA
2223 4303 TAD CHAR
2224 1134 CIA
2225 7041 TAD K254 /GET INPUT
2226 1353 SNA /OK IF Ø
2227 7450 JMP !*3 /PRINT QUESTION MARK
2228 5233 JMS QUERY
2229 4344 JMP SLMTS+2
2230 5211 JMS KEYIN /WAIT FOR 2ND
2231 4303 JMS LEGAL /SEE IF IT'S LEGAL
2232 4314 TAD CHAR
2233 4334 AND K7 /MASK AC 9=11
2234 1134 CLL RAL
2235 0133 RTL /POSITION TO AC 6=8
2236 7104 DCA LAST1 /LAST TO TEST
2237 7104 TAD FIRST1
2238 7006 CIA
2239 3124 TAD LAST1
2240 7006 TAD FIRST1
2241 3124 CIA
2242 1123 TAD LAST1
2243 7041 CIA
2244 1124 TAD LAST1
2245 7500 SMA /1ST IS > LAST IF NEG
2246 5236 JMP OKAS
2247 7200 CLA

2250 1123 TAD FIRST1 /SEE IF EQUAL
2251 3134 DCA CHAR /YES IF Ø
2252 1124 TAD LAST1
2253 3123 DCA FIRST1 /LAST NOW IS FIRST
2254 1134 TAD CHAR
2255 3124 DCA LAST1 /FIRST IS NOW LAST
2256 7200 OKAS,
2257 1124 TAD LAST1
2258 7041 CIA
2259 1123 TAD FIRST1
2260 7440 SZA /CURRENT FIELD
2261 1123 TAD INSLD /NO IF A 1
2262 5273 SZA CLA /PRINT PROGRAM LOCATION
2263 5273 TAD ALOK /AND START OVER
2264 1123 TAD FIRST1 /PRINT KEYIN
2265 7041 JMS I XLCLAT
2266 1016 SZA CLA /WAIT FOR C.R.
2267 7640 TAD ALOK
2268 5273 JMS SLMTS+2
2269 4752 JMS KEYIN
2270 5211 ALOK,
2271 4303 2273 4303

PAL10 V141

2=NOV=91 0119 PAGE 1=22

2274 1134 TAD CHAR
2275 7041 CIA
2276 1143 TAD K215
2277 7450 SNA /NOT A C.R., IF A SKIP
2300 5607 JMP I SLMTS
2301 4344 JMS QUERY /PRINT QUESTION MARK
2302 5211 JMP SLMTS+2 /START OVER

2303 0000 /KEYIN, 2
2304 6032 KCC
2305 6031 KSF
2306 5305 JMP =1
2307 6036 KRB
2310 3134 DCA CHAR
2311 1134 TAD CHAR
2312 4536 JMS !XPERR
2313 5703 JMP ! KEYIN

2314 0000 /LEGAL, 2
2315 1134 TAD CHAR
2316 7041 CIA
2317 1144 TAD K377
2320 7650 SNA CLA /RUB=OUT IP 0
2321 5211 JMP SLMTS+2
2322 1134 TAD CHAR
2323 0145 AND K370
2324 7041 CIA
2325 1142 TAD K260
2326 7650 SNA CLA
2327 5714 JMP ! LEGAL
2330 1134 TAD CHAR
2331 7041 CIA
2332 1353 TAD K254
2333 7650 SNA CLA /A COMMA IP 0
2334 5714 JMP ! LEGAL
2335 1134 TAD CHAR
2336 7041 CIA
2337 1143 TAD K215
2340 7650 SNA CLA /A C.R., IF 0
2341 5714 JMP ! LEGAL
2342 4344 JMS QUERY /QUERY
2343 5211 JMP SLMTS+2 /START OVER
2344 0000 /QUERY, 2
2345 4547 JMS ! XCRLF
2346 1146 TAD K277
2347 4536 JMS ! XPERR
2350 5744 JMP ! QUERY
2351 2446 XTLM, TLIM
2352 2402 XLCAT, LOCAT
2353 0254 K254,

PAL10 V141 2-NQV-71 2119 PAGE 1-23

/PRINT FIELD PROGRAM IS IN
*2402
/LOCAT, 2 CLA INSFLD /CURRENT FIELD
2400 0000 TAD RTR
2401 7200 TAD RAR
2402 1016 AND K7
2403 7012 TAD K26B
2404 7010 DCA FLDN
2405 0133 AND PRGM
2406 1142 TAD 12
2407 3244 DCA 12
2410 1217 TAD PRGM
2411 3012 DCA 12
2412 1412 PLOCT, SNA TAD 1 12
2413 7450 JMP I LOCAT //DONE IF 0
2414 5600 JMS I XPERR //EXIT
2415 4536 JMS I PLOCT //PRINT
2416 5212 JMP PLOCT
/PRGAM,
2417 2417 //P
2420 0320 //R
2421 0322 //O
2422 0317 //G
2423 0307 //R
2424 0322 //A
2425 0301 //M
2426 0315 //W
2427 0240 //I
2428 0311 //S
2431 0323 //I
2432 0240 //N
2433 0311 //S
2434 0316 //I
2435 0240 //F
2436 0306 //I
2437 0311 //E
2440 0305 //L
2441 0314 //D
2442 0304 //O
2443 0240 //X
2444 0000 FLDN, 0 //TERMINATOR
2445 0000 0
/LIMIT, 0 CLA TSTL
2446 0000 TAD 12
2447 7200 DCA 12
2450 1257 TAD 1 12
2451 3012 PLIMT, SNA TLIMT
2452 1412 TAD 1 12
2453 7450 JMP I TLIMT
2454 5646 //DONE IF 0
//PRINT TEST LIMITS
/DONE IF 0
/DONE IF 0

*AL10 V141 2-Nov-71 Z119 PAGE 1-24

2455 4536 JMS I XPERR
2456 5252 JMP PLINT

2457 2457 /TSTL.
2460 0324 324 /T
2461 0305 305 /E
2462 0323 323 /S
2463 0324 324 /T
2464 0240 240 /L
2465 0314 314 /H
2466 0311 311 /I
2467 0315 315 /N
2470 0311 311 /I
2471 0324 324 /T
2472 0323 323 /S
2473 0000 0 /TERMINATOR

/HEADER ROUTINE

/PHDR, 0 JMS I XCRLF /CR, LF
2474 0000 TAD FILD
2475 4547 DCA 12 /PRINT FIELD
2476 1332 TAD FILD
2477 3012 DCA 12 /DONE IF 0
2500 1412 TAD 1 12 /DONE IF 0
2501 7450 SNA 143 /
2502 5305 JMP 143
2503 4536 JMS I XPERR
2504 5300 JMP PFILD
2505 1061 TAD MS
2506 3074 DCA LOOP
2507 4537 JMS I XPING /SPACE 5
2510 1341 TAD OTLD
2511 3012 DCA 12

2512 1412 POCOR, TAD 1 12 /PRINT OCTAL ADR
2513 7450 SNA /DONE IF 0
2514 5317 JMP 143
2515 4536 JMS I XPERR
2516 5312 JMP POCOR /
2517 1061 TAD MS
2520 3074 DCA LOOP
2521 4537 JMS I XPING /SPACE 5
2522 1355 TAD GOOD
2523 3012 DCA 12 /PRINT GOOD
2524 1412 TAD 1 12 /DONE IF 0
2525 7450 SNA 143 /
2526 5731 JMP 143
2527 4536 JMS I XPERR
2530 5324 JMP PGOOD
2531 2600 BSPCE /NEXT PAGE
2532 2532 /FILD, 306 /F
2533 0306

PAL10 V141 2=NOV=71 0119 PAGE 1=25

2534 0311 311
2535 0305 305
2536 2314 314
2537 0304 304
2540 0000 0

2541 2541 /OTLDR,
2542 0317 317
2543 0303 303
2544 0324 324
2545 0301 301
2546 0314 314
2547 0240 240
2548 0301 301
2550 0304 304
2551 0304 304
2552 0322 322
2553 0256 256
2554 0000 0

2555 2555 /GOOD,
2556 0307 307
2557 0317 317
2560 0317 317
2561 0304 304
2562 0000 0

2563 5674 /EXHDR,
2564 5674 JMP I PHDR

/ *2600

2600 1061 /SPCE,
2601 3074 TAD MS
2602 4537 DCA LOOP
2603 1234 JMS I XPING
2604 3012 TAD BADD
2605 1412 DCA 12
2606 7450 TAD I 12
2607 5212 SNA /DONE IF 0
2610 4536 JMP I XPERR
2611 5205 JMP PBAD
2612 1061 TAD MS
2613 3074 DCA LOOP
2614 4537 JMS I XPING
2615 1226 /
2616 3012 TAD TSTN
2617 1412 DCA 12
2620 7450 PTSTN,
2621 5224 TAD I 12
2622 4536 SNA /PRINT TEST
2623 5217 JMP PTSTN
2624 4547 JMS I XCRLF
2625 5644 JMP I XPHDR

```

/
2626 2626 TSTN, 324 /T
2627 0324 305 /E
2630 0305 323 /S
2631 0323 324 /T
2632 0324 0
2633 0000 /
2634 2634 BADD, 302 /B
2635 0302 301 /A
2636 0301 304 /D
2637 0304 0
2640 0000 /
2641 0000 HALT, 0
2642 7402 HALT /RESTART HERE OR RTRN1
2643 5641 JMP ! HALT
2644 2563 XPHDR, EXHDR
/
//WAIT HERE TO SETUP SR. TYPE CARRIAGE RETURN
//AFTER SETTING SR.
SETSR, 0 JMS ! XFILD /RESTORE DATA FIELD
2645 0000 4531 JMS ! XCRLF /CR, LF
2646 4547 1270 TAD STSR
2647 4547 3912 DCA 12
2648 1270 3912 TAD ! 12 /DONE IF 0
2649 3912 1412 SNA
2650 1412 7450 JMP !+3 /PRINT SETUP SR
2651 7450 5237 JMS ! XPERR
2652 5237 5236 JMP PSTSR
2653 5236 5252 KRB
2654 5252 6036 WTCR, KSF
2655 5252 6031 JMP !+4
2656 6036 5265 KRB
2657 6031 5265 JMP !+4
2658 5265 6036 JMS ! XPERR
2659 6036 4536 JMP ! SETSR
2660 4536 5645 LAS
2661 5645 7604 DCA MCWA
2662 7604 3033 JMP WTCR
2663 3033 5260 /
2664 5260 6031 STSR, /S
2665 6031 0323 323 /S
2666 0323 0305 305 /E
2667 0305 0324 324 /U
2668 0324 0325 325 /P
2669 0325 0320 320 /P
2670 0320 0240 240 /S
2671 0240 0323 323 /S
2672 0323 0324 324 /U
2673 0324 0325 325 /P
2674 0325 0320 320 /P
2675 0320 0240 240 /S
2676 0240 0323 323 /S
2677 0323

```

PAL12	V141	2eNOV=71	0119	PAGE 1=27
2700	2322	322		
2701	2000	0		/R
2702	2000	'STALL,	JMS I XFIELD	
2703	4531		JMS GENRAN	/GET ANOTHER
2704	4316		DCA LOOP	
2705	3074		ISZ LOOP	/18.5 MS MAX.
2706	2074		JMP *1	
2707	5306		TAD DATFLD	
2710	1014		TAD KCDF	
2711	1125		DCA *1	
2712	3313		CDF @	/RESTORE DATA FIELD
2713	6201		CLA	
2714	7200		JMP I STALL	/EXIT
2715	5702			
2716	0000	GENRAN, 0		
2717	1354	TAD RANTAB		
2720	7041	CIA		
2721	1342	TAD RANDEX		
2722	7640	SEA CLA		
2723	5333	JMP RANTAD+1		
2724	1355	TAD TBLRAN		
2725	3342	DCA RANDEX		
2726	1341	TAD RANCON		
2727	7104	CLL RAL		
2730	7430	SZL		
2731	1150	TAD K1		
2732	3341	DCA RANCON		
2733	1742	TAD I RANDEX		
2734	1341	RANTAD, TAD RANCON		
2735	3742	DCA I RANDEX		
2736	1742	TAD I RANDEX		
2737	2342	ISZ RANDEX		
2740	5716	JMP I GENRAN		
3412-2741	1234	RANCON, 1234		
2353-2742	2753	RANDEX, RANTBL+10		
7150-2743	4321	RANTBL, 4321		
3415-2744	1416	1416		
4112-2746	5363	5363		
1047-2747	6060	6060		
0624-2750	3035	3035		
1271-2751	2572	2572		
1246-2752	3237	3237		
2753	0214	0214		
2754	0000	0		
2755	2753	RANTAB, *1		
2756	2743	TBLRAN, RANTBL		
	0177	K177, 177		

/ROUTINE TO DETERMINE FIELD FOR RELOCATION

PAL10 V141 2=NOV=71 0119 PAGE 1=28

3020 *3000 /
3021 4531 CMOVE, JMS 1 XFLD /SET OF TO CURRENT FIELD
3021 7600 TAD ERTBL
3022 1020 DCA ERWD
3023 3022 TAD FIRST1 /SETUP ERROR TABLE POINTER
3024 1123 CIA
3025 7041 TAD LAST1 /FIRST TESTED FIELD
3026 1124 CIA
3027 7650 SNA CLA /LAST TESTED FIELD
3028 5530 JMP 1 XRTN
3029 1015 TAD FLAGS
3030 7430 RAR
3031 3013 S2L /FIRST MOVE IF A SKIP
3032 5725 JMP 1 XTMV /SETUP FOR NEXT MOVE
3033 7001 TAC
3034 3015 DCA FLAGS
3035 7001 TAD LAST1 /LAST TO TEST # 1ST MOVE
3036 3016 DCA INSLD /NEW CURRENT FIELD
3037 1124 TAD INSLD
3038 3016 TAD INSLD
3039 1016 TAD M10
3040 3021 TAD NXLOC
3041 1016 CIA
3042 3141 TAD RIF
3043 3151 TAD SNA
3044 3024 CIA
3045 6224 TAD CLA
3046 3025 CIA
3047 7041 TAD SNA
3048 1016 CLA
3049 7650 SUB1 /IS NEXT SAME AS CURRENT
3050 5266 JMP /YES, TRY NEXT LOWER FIELD
3051 3015 /SUBTRACT 1 FROM NEW CURRENT
3052 3151 /NXLOC>DOEST/N FOR NEXT TIME
3053 3024 /INITIALIZE MOVE
3054 3025 /POINTERT+1
3055 7650 /NO ERRORS RECORDED IF 0
3056 5266 /INITIALIZE MOVE
3057 3031 2022 CKERR, ISZ ERWD /CHECK FOR ERROR IN NEW FIELD
3058 1201 TAD GMOVE+1
3059 7041 CIA
3060 1422 TAD I ERWD
3061 7650 SNA CLA
3062 5310 JMP STMV
3063 3032 2022 TAD I ERWD
3064 1201 CIA
3065 7041 TAD INSLD
3066 1422 SNA CLA
3067 7650 JMP EQUAL
3068 5310 TAD ERWD
3069 3033 2022 CNXT, TAD I ERWD
3070 1201 CIA
3071 7041 TAD INSLD
3072 1422 SNA CLA
3073 7650 JMP EQUAL
3074 5310 TAD ERWD
3075 3034 2022 TAD ENtbl
3076 1201 CIA
3077 7041 TAD ENtbl
3078 1422 SNA CLA
3079 7650 JMP STMV
3080 5310 ISZ ERWD
3081 3035 2022 TAD ENtbl
3082 1201 CIA
3083 7041 TAD ENtbl
3084 1422 SNA CLA
3085 7650 JMP STMV
3086 5310 ISZ ERWD
3087 3036 2022 TAD ENtbl
3088 1201 CIA
3089 7041 TAD ENtbl
3090 1422 SNA CLA
3091 7650 JMP STMV
3092 5310 ISZ ERWD
3093 3037 2022 TAD ENtbl
3094 1201 CIA
3095 7041 TAD ENtbl
3096 1422 SNA CLA
3097 7650 JMP STMV
3098 5310 ISZ ERWD
3099 3038 2022 TAD ENtbl
3100 1201 CIA
3101 7041 TAD ENtbl
3102 1422 SNA CLA
3103 7650 JMP STMV
3104 5310 ISZ ERWD
3105 3039 2022 TAD ENtbl
3106 1201 CIA
3107 7041 TAD ENtbl
3108 1422 SNA CLA
3109 7650 JMP STMV
3110 5310 ISZ ERWD
3111 3040 2022 TAD ENtbl
3112 1201 CIA
3113 7041 TAD ENtbl
3114 1422 SNA CLA
3115 7650 JMP STMV
3116 5310 ISZ ERWD
3117 3041 2022 TAD ENtbl
3118 1201 CIA
3119 7041 TAD ENtbl
3120 1422 SNA CLA
3121 7650 JMP STMV
3122 5310 ISZ ERWD
3123 3042 2022 TAD ENtbl
3124 1201 CIA
3125 7041 TAD ENtbl
3126 1422 SNA CLA
3127 7650 JMP STMV
3128 5310 ISZ ERWD
3129 3043 2022 TAD ENtbl
3130 1201 CIA
3131 7041 TAD ENtbl
3132 1422 SNA CLA
3133 7650 JMP STMV
3134 5310 ISZ ERWD
3135 3044 2022 TAD ENtbl
3136 1201 CIA
3137 7041 TAD ENtbl
3138 1422 SNA CLA
3139 7650 JMP STMV
3140 5310 ISZ ERWD
3141 3045 2022 TAD ENtbl
3142 1201 CIA
3143 7041 TAD ENtbl
3144 1422 SNA CLA
3145 7650 JMP STMV
3146 5310 ISZ ERWD
3147 3046 2022 TAD ENtbl
3148 1201 CIA
3149 7041 TAD ENtbl
3150 1422 SNA CLA
3151 7650 JMP STMV
3152 5310 ISZ ERWD
3153 3047 2022 TAD ENtbl
3154 1201 CIA
3155 7041 TAD ENtbl
3156 1422 SNA CLA
3157 7650 JMP STMV
3158 5310 ISZ ERWD
3159 3048 2022 TAD ENtbl
3160 1201 CIA
3161 7041 TAD ENtbl
3162 1422 SNA CLA
3163 7650 JMP STMV
3164 5310 ISZ ERWD
3165 3049 2022 TAD ENtbl
3166 1201 CIA
3167 7041 TAD ENtbl
3168 1422 SNA CLA
3169 7650 JMP STMV
3170 5310 ISZ ERWD
3171 3050 2022 TAD ENtbl
3172 1201 CIA
3173 7041 TAD ENtbl
3174 1422 SNA CLA
3175 7650 JMP STMV
3176 5310 ISZ ERWD
3177 3051 2022 TAD ENtbl
3178 1201 CIA
3179 7041 TAD ENtbl
3180 1422 SNA CLA
3181 7650 JMP STMV
3182 5310 ISZ ERWD
3183 3052 2022 TAD ENtbl
3184 1201 CIA
3185 7041 TAD ENtbl
3186 1422 SNA CLA
3187 7650 JMP STMV
3188 5310 ISZ ERWD
3189 3053 2022 TAD ENtbl
3190 1201 CIA
3191 7041 TAD ENtbl
3192 1422 SNA CLA
3193 7650 JMP STMV
3194 5310 ISZ ERWD
3195 3054 2022 TAD ENtbl
3196 1201 CIA
3197 7041 TAD ENtbl
3198 1422 SNA CLA
3199 7650 JMP STMV
3200 5310 ISZ ERWD
3201 3055 2022 TAD ENtbl
3202 1201 CIA
3203 7041 TAD ENtbl
3204 1422 SNA CLA
3205 7650 JMP STMV
3206 5310 ISZ ERWD
3207 3056 2022 TAD ENtbl
3208 1201 CIA
3209 7041 TAD ENtbl
3210 1422 SNA CLA
3211 7650 JMP STMV
3212 5310 ISZ ERWD
3213 3057 2022 TAD ENtbl
3214 1201 CIA
3215 7041 TAD ENtbl
3216 1422 SNA CLA
3217 7650 JMP STMV
3218 5310 ISZ ERWD
3219 3058 2022 TAD ENtbl
3220 1201 CIA
3221 7041 TAD ENtbl
3222 1422 SNA CLA
3223 7650 JMP STMV
3224 5310 ISZ ERWD
3225 3059 2022 TAD ENtbl
3226 1201 CIA
3227 7041 TAD ENtbl
3228 1422 SNA CLA
3229 7650 JMP STMV
3230 5310 ISZ ERWD
3231 3060 2022 TAD ENtbl
3232 1201 CIA
3233 7041 TAD ENtbl
3234 1422 SNA CLA
3235 7650 JMP STMV
3236 5310 ISZ ERWD
3237 3061 2022 TAD ENtbl
3238 1201 CIA
3239 7041 TAD ENtbl
3240 1422 SNA CLA
3241 7650 JMP STMV
3242 5310 ISZ ERWD
3243 3062 2022 TAD ENtbl
3244 1201 CIA
3245 7041 TAD ENtbl
3246 1422 SNA CLA
3247 7650 JMP STMV
3248 5310 ISZ ERWD
3249 3063 2022 TAD ENtbl
3250 1201 CIA
3251 7041 TAD ENtbl
3252 1422 SNA CLA
3253 7650 JMP STMV
3254 5310 ISZ ERWD
3255 3064 2022 TAD ENtbl
3256 1201 CIA
3257 7041 TAD ENtbl
3258 1422 SNA CLA
3259 7650 JMP STMV
3260 5310 ISZ ERWD
3261 3065 2022 TAD ENtbl
3262 1201 CIA
3263 7041 TAD ENtbl
3264 1422 SNA CLA
3265 7650 JMP STMV
3266 5310 ISZ ERWD
3267 3066 2022 TAD ENtbl
3268 1201 CIA
3269 7041 TAD ENtbl
3270 1422 SNA CLA
3271 7650 JMP STMV
3272 5310 ISZ ERWD
3273 3067 2022 TAD ENtbl
3274 1201 CIA
3275 7041 TAD ENtbl
3276 1422 SNA CLA
3277 7650 JMP STMV
3278 5310 ISZ ERWD
3279 3068 2022 TAD ENtbl
3280 1201 CIA
3281 7041 TAD ENtbl
3282 1422 SNA CLA
3283 7650 JMP STMV
3284 5310 ISZ ERWD
3285 3069 2022 TAD ENtbl
3286 1201 CIA
3287 7041 TAD ENtbl
3288 1422 SNA CLA
3289 7650 JMP STMV
3290 5310 ISZ ERWD
3291 3070 2022 TAD ENtbl
3292 1201 CIA
3293 7041 TAD ENtbl
3294 1422 SNA CLA
3295 7650 JMP STMV
3296 5310 ISZ ERWD
3297 3071 2022 TAD ENtbl
3298 1201 CIA
3299 7041 TAD ENtbl
3300 1422 SNA CLA
3301 7650 JMP STMV
3302 5310 ISZ ERWD
3303 3072 2022 TAD ENtbl
3304 1201 CIA
3305 7041 TAD ENtbl
3306 1422 SNA CLA
3307 7650 JMP STMV
3308 5310 ISZ ERWD
3309 3073 2022 TAD ENtbl
3310 1201 CIA
3311 7041 TAD ENtbl
3312 1422 SNA CLA
3313 7650 JMP STMV
3314 5310 ISZ ERWD
3315 3074 2022 TAD ENtbl
3316 1201 CIA
3317 7041 TAD ENtbl
3318 1422 SNA CLA
3319 7650 JMP STMV
3320 5310 ISZ ERWD
3321 3075 2022 TAD ENtbl
3322 1201 CIA
3323 7041 TAD ENtbl
3324 1422 SNA CLA
3325 7650 JMP STMV
3326 5310 ISZ ERWD
3327 3076 2022 TAD ENtbl
3328 1201 CIA
3329 7041 TAD ENtbl
3330 1422 SNA CLA
3331 7650 JMP STMV
3332 5310 ISZ ERWD
3333 3077 2022 TAD ENtbl
3334 1201 CIA
3335 7041 TAD ENtbl
3336 1422 SNA CLA
3337 7650 JMP STMV
3338 5310 ISZ ERWD
3339 3078 2022 TAD ENtbl
3340 1201 CIA
3341 7041 TAD ENtbl
3342 1422 SNA CLA
3343 7650 JMP STMV
3344 5310 ISZ ERWD
3345 3079 2022 TAD ENtbl
3346 1201 CIA
3347 7041 TAD ENtbl
3348 1422 SNA CLA
3349 7650 JMP STMV
3350 5310 ISZ ERWD
3351 3080 2022 TAD ENtbl
3352 1201 CIA
3353 7041 TAD ENtbl
3354 1422 SNA CLA
3355 7650 JMP STMV
3356 5310 ISZ ERWD
3357 3081 2022 TAD ENtbl
3358 1201 CIA
3359 7041 TAD ENtbl
3360 1422 SNA CLA
3361 7650 JMP STMV
3362 5310 ISZ ERWD
3363 3082 2022 TAD ENtbl
3364 1201 CIA
3365 7041 TAD ENtbl
3366 1422 SNA CLA
3367 7650 JMP STMV
3368 5310 ISZ ERWD
3369 3083 2022 TAD ENtbl
3370 1201 CIA
3371 7041 TAD ENtbl
3372 1422 SNA CLA
3373 7650 JMP STMV
3374 5310 ISZ ERWD
3375 3084 2022 TAD ENtbl
3376 1201 CIA
3377 7041 TAD ENtbl
3378 1422 SNA CLA
3379 7650 JMP STMV
3380 5310 ISZ ERWD
3381 3085 2022 TAD ENtbl
3382 1201 CIA
3383 7041 TAD ENtbl
3384 1422 SNA CLA
3385 7650 JMP STMV
3386 5310 ISZ ERWD
3387 3086 2022 TAD ENtbl
3388 1201 CIA
3389 7041 TAD ENtbl
3390 1422 SNA CLA
3391 7650 JMP STMV
3392 5310 ISZ ERWD
3393 3087 2022 TAD ENtbl
3394 1201 CIA
3395 7041 TAD ENtbl
3396 1422 SNA CLA
3397 7650 JMP STMV
3398 5310 ISZ ERWD
3399 3088 2022 TAD ENtbl
3400 1201 CIA
3401 7041 TAD ENtbl
3402 1422 SNA CLA
3403 7650 JMP STMV
3404 5310 ISZ ERWD
3405 3089 2022 TAD ENtbl
3406 1201 CIA
3407 7041 TAD ENtbl
3408 1422 SNA CLA
3409 7650 JMP STMV
3410 5310 ISZ ERWD
3411 3090 2022 TAD ENtbl
3412 1201 CIA
3413 7041 TAD ENtbl
3414 1422 SNA CLA
3415 7650 JMP STMV
3416 5310 ISZ ERWD
3417 3091 2022 TAD ENtbl
3418 1201 CIA
3419 7041 TAD ENtbl
3420 1422 SNA CLA
3421 7650 JMP STMV
3422 5310 ISZ ERWD
3423 3092 2022 TAD ENtbl
3424 1201 CIA
3425 7041 TAD ENtbl
3426 1422 SNA CLA
3427 7650 JMP STMV
3428 5310 ISZ ERWD
3429 3093 2022 TAD ENtbl
3430 1201 CIA
3431 7041 TAD ENtbl
3432 1422 SNA CLA
3433 7650 JMP STMV
3434 5310 ISZ ERWD
3435 3094 2022 TAD ENtbl
3436 1201 CIA
3437 7041 TAD ENtbl
3438 1422 SNA CLA
3439 7650 JMP STMV
3440 5310 ISZ ERWD
3441 3095 2022 TAD ENtbl
3442 1201 CIA
3443 7041 TAD ENtbl
3444 1422 SNA CLA
3445 7650 JMP STMV
3446 5310 ISZ ERWD
3447 3096 2022 TAD ENtbl
3448 1201 CIA
3449 7041 TAD ENtbl
3450 1422 SNA CLA
3451 7650 JMP STMV
3452 5310 ISZ ERWD
3453 3097 2022 TAD ENtbl
3454 1201 CIA
3455 7041 TAD ENtbl
3456 1422 SNA CLA
3457 7650 JMP STMV
3458 5310 ISZ ERWD
3459 3098 2022 TAD ENtbl
3460 1201 CIA
3461 7041 TAD ENtbl
3462 1422 SNA CLA
3463 7650 JMP STMV
3464 5310 ISZ ERWD
3465 3099 2022 TAD ENtbl
3466 1201 CIA
3467 7041 TAD ENtbl
3468 1422 SNA CLA
3469 7650 JMP STMV
3470 5310 ISZ ERWD
3471 3100 2022 TAD ENtbl
3472 1201 CIA
3473 7041 TAD ENtbl
3474 1422 SNA CLA
3475 7650 JMP STMV
3476 5310 ISZ ERWD
3477 3101 2022 TAD ENtbl
3478 1201 CIA
3479 7041 TAD ENtbl
3480 1422 SNA CLA
3481 7650 JMP STMV
3482 5310 ISZ ERWD
3483 3102 2022 TAD ENtbl
3484 1201 CIA
3485 7041 TAD ENtbl
3486 1422 SNA CLA
3487 7650 JMP STMV
3488 5310 ISZ ERWD
3489 3103 2022 TAD ENtbl
3490 1201 CIA
3491 7041 TAD ENtbl
3492 1422 SNA CLA
3493 7650 JMP STMV
3494 5310 ISZ ERWD
3495 3104 2022 TAD ENtbl
3496 1201 CIA
3497 7041 TAD ENtbl
3498 1422 SNA CLA
3499 7650 JMP STMV
3500 5310 ISZ ERWD
3501 3105 2022 TAD ENtbl
3502 1201 CIA
3503 7041 TAD ENtbl
3504 1422 SNA CLA
3505 7650 JMP STMV
3506 5310 ISZ ERWD
3507 3106 2022 TAD ENtbl
3508 1201 CIA
3509 7041 TAD ENtbl
3510 1422 SNA CLA
3511 7650 JMP STMV
3512 5310 ISZ ERWD
3513 3107 2022 TAD ENtbl
3514 1201 CIA
3515 7041 TAD ENtbl
3516 1422 SNA CLA
3517 7650 JMP STMV
3518 5310 ISZ ERWD
3519 3108 2022 TAD ENtbl
3520 1201 CIA
3521 7041 TAD ENtbl
3522 1422 SNA CLA
3523 7650 JMP STMV
3524 5310 ISZ ERWD
3525 3109 2022 TAD ENtbl
3526 1201 CIA
3527 7041 TAD ENtbl
3528 1422 SNA CLA
3529 7650 JMP STMV
3530 5310 ISZ ERWD
3531 3110 2022 TAD ENtbl
3532 1201 CIA
3533 7041 TAD ENtbl
3534 1422 SNA CLA
3535 7650 JMP STMV
3536 5310 ISZ ERWD
3537 3111 2022 TAD ENtbl
3538 1201 CIA
3539 7041 TAD ENtbl
3540 1422 SNA CLA
3541 7650 JMP STMV
3542 5310 ISZ ERWD
3543 3112 2022 TAD ENtbl
3544 1201 CIA
3545 7041 TAD ENtbl
3546 1422 SNA CLA
3547 7650 JMP STMV
3548 5310 ISZ ERWD
3549 3113 2022 TAD ENtbl
3550 1201 CIA
3551 7041 TAD ENtbl
3552 1422 SNA CLA
3553 7650 JMP STMV
3554 5310 ISZ ERWD
3555 3114 2022 TAD ENtbl
3556 1201 CIA
3557 7041 TAD ENtbl
3558 1422 SNA CLA
3559 7650 JMP STMV
3560 5310 ISZ ERWD
3561 3115 2022 TAD ENtbl
3562 1201 CIA
3563 7041 TAD ENtbl
3564 1422 SNA CLA
3565 7650 JMP STMV
3566 5310 ISZ ERWD
3567 3116 2022 TAD ENtbl
3568 1201 CIA
3569 7041 TAD ENtbl
3570 1422 SNA CLA
3571 7650 JMP STMV
3572 5310 ISZ ERWD
3573 3117 2022 TAD ENtbl
3574 1201 CIA
3575 7041 TAD ENtbl
3576 1422 SNA CLA
3577 7650 JMP STMV
3578 5310 ISZ ERWD
3579 3118 2022 TAD ENtbl
3580 1201 CIA
3581 7041 TAD ENtbl
3582 1422 SNA CLA
3583 7650 JMP STMV
3584 5310 ISZ ERWD
3585 3119 2022 TAD ENtbl
3586 1201 CIA
3587 7041 TAD ENtbl
3588 1422 SNA CLA
3589 7650 JMP STMV
3590 5310 ISZ ERWD
3591 3120 2022 TAD ENtbl
3592 1201 CIA
3593 7041 TAD ENtbl
3594 1422 SNA CLA
3595 7650 JMP STMV
3596 5310 ISZ ERWD
3597 3121 2022 TAD ENtbl
3598 1201 CIA
3599 7041 TAD ENtbl
3600 1422 SNA CLA
3601 7650 JMP STMV
3602 5310 ISZ ERWD
3603 3122 2022 TAD ENtbl
3604 1201 CIA
3605 7041 TAD ENtbl
3606 1422 SNA CLA
3607 7650 JMP STMV
3608 5310 ISZ ERWD
3609 3123 2022 TAD ENtbl
3610 1201 CIA
3611 7041 TAD ENtbl
3612 1422 SNA CLA
3613 7650 JMP STMV
3614 5310 ISZ ERWD
3615 3124 2022 TAD ENtbl
3616 1201 CIA
3617 7041 TAD ENtbl
3618 1422 SNA CLA
3619 7650 JMP STMV
3620 5310 ISZ ERWD
3621 3125 2022 TAD ENtbl
3622 1201 CIA
3623 7041 TAD ENtbl
3624 1422 SNA CLA
3625 7650 JMP STMV
3626 5310 ISZ ERWD
3627 3126 2022 TAD ENtbl
3628 1201 CIA
3629 7041 TAD ENtbl
3630 1422 SNA CLA
3631 7650 JMP STMV
3632 5310 ISZ ERWD
3633 3127 2022 TAD ENtbl
3634 1201 CIA
3635 7041 TAD ENtbl
3636 1422 SNA CLA
3637 7650 JMP STMV
3638 5310 ISZ ERWD
3639 3128 2022 TAD ENtbl
3640 1201 CIA
3641 7041 TAD ENtbl
3642 1422 SNA CLA
3643 7650 JMP STMV
3644 5310 ISZ ERWD
3645 3129 2022 TAD ENtbl
3646 1201 CIA
3647 7041 TAD ENtbl
3648 1422 SNA CLA
3649 7650 JMP STMV
3650 5310 ISZ ERWD
3651 3130 2022 TAD ENtbl
3652 1201 CIA
3653 7041 TAD ENtbl
3654 1422 SNA CLA
3655 7650 JMP STMV
3656 5310 ISZ ERWD
3657 3131 2022 TAD ENtbl
3658 1201 CIA
3659 7041 TAD ENtbl
3660 1422 SNA CLA
3661 7650 JMP STMV
3662 5310 ISZ ERWD
3663 3132 2022 TAD ENtbl
3664 1201 CIA
3665 7041 TAD ENtbl
3666 1422 SNA CLA
3667 7650 JMP STMV
3668 5310 ISZ ERWD
3669 3133 2022 TAD ENtbl
3670 1201 CIA
3671 7041 TAD ENtbl
3672 1422 SNA CLA
3673 7650 JMP STMV
3674 5310 ISZ ERWD
3675 3134 2022 TAD ENtbl
3676 1201 CIA
3677 7041 TAD ENtbl
3678 1422 SNA CLA
3679 7650 JMP STMV
3680 5310 ISZ ERWD
3681 3135 2022 TAD ENtbl
3682 1201 CIA
3683 7041 TAD ENtbl
3684 1422 SNA CLA
3685 7650 JMP STMV
3686 5310 ISZ ERWD
3687 3136 2022 TAD ENtbl
3688 1201 CIA
3689 7041 TAD ENtbl
3690 1422 SNA CLA
3691 7650 JMP STMV
3692 5310 ISZ ERWD
3693 3137 2022 TAD ENtbl
3694 1201 CIA
3695 7041 TAD ENtbl
3696 1422 SNA CLA
3697 7650 JMP STMV
3698 5310 ISZ ERWD
3699 3138 2022 TAD ENtbl
3700 1201 CIA
3701 7041 TAD ENtbl
3702 1422 SNA CLA
3703 7650 JMP STMV
3704 5310 ISZ ERWD
3705 3139 2022 TAD ENtbl
3706 1201 CIA
3707 7041 TAD ENtbl
3708 1422 SNA CLA
3709 7650 JMP STMV
3710 5310 ISZ ERWD
3711 3140 2022 TAD ENtbl
3712 1201 CIA
3713 7041 TAD ENtbl
3714 1422 SNA CLA
3715 7650 JMP STMV
3716 5310 ISZ ERWD
3717 3141 2022 TAD ENtbl
3718 1201 CIA
3719 7041 TAD ENtbl
3720 1422 SNA CLA
3721 7650 JMP STMV
3722 5310 ISZ ERWD
3723 3142 2022 TAD ENtbl
3724 1201 CIA
3725 7041 TAD ENtbl
3726 1422 SNA CLA
3727 7650 JMP STMV
3728 5310 ISZ ERWD
3729 3143 2022 TAD ENtbl
3730 1201 CIA
3731 7041 TAD ENtbl
3732 1422 SNA CLA
3733 7650 JMP STMV
3734 5310 ISZ ERWD
3735 3144 2022 TAD ENtbl
3736 1201 CIA
3737 7041 TAD ENtbl
3738 1422 SNA CLA
3739 7650 JMP STMV
3740 5310 ISZ ERWD
3741 3145 2022 TAD ENtbl
3742 1201 CIA
3743 7041 TAD ENtbl
3744 1422 SNA CLA
3745 7650 JMP STMV
3746 5310 ISZ ERWD
3747 3146 2022 TAD ENtbl
3748 1201 CIA
3749 7041 TAD ENtbl
3750 1422 SNA CLA
3751 7650 JMP STMV
3752 5310 ISZ ERWD
3753 3147 2022 TAD ENtbl
3754 1201 CIA
3755 7041 TAD ENtbl
3756 1422 SNA CLA
3757 7650 JMP STMV
3758 5310 ISZ ERWD
3759 3148

PAL10 V141

2-Nov-71 0119 PAGE 1-29

3053 1422 EQUAL, TAD I ERWRD /GET ERROR FIELD
3054 7041 CIA FIRST1
3055 1123 TAD FIRST1
3056 7650 SNA CLA /DON'T MOVE IF = TO FIRST
3057 5530 JMP I XRTN /START OVER
3060 1422 TAD I ERWRD
3061 7650 SNA CLA /IS IT FIELD @?
3062 5266 JMP SUB1 /YES
3063 1016 TAD INSLD /CURRENT NEXT
3064 1141 TAD M10 /SUBTRACT 1 FROM OF
3065 3151 DCA NXLOC

3066 1020 /SUB1, TAD ERTBL /RESTORE TABLE POINTER
3067 3022 DCA ERWRD
3070 1151 TAD NXLOC
3071 7041 CIA
3072 1016 TAD INSLD /NEXT = CURRENT NEXT IF @
3073 7650 SNA CLA
3074 5253 JMP EQUAL
3075 1151 TAD NXLOC /NEW CURRENT FIELD
3076 3016 DCA INSLD
3077 1016 TAD INSLD
3100 7041 CIA FIRST1 /IS IT = LOWEST FIELD
3101 1123 TAD FIRST1
3102 7650 SNA CLA
3103 5251 JMP CKERR /YES
3104 1016 TAD INSLD /CURRENT NEW FIELD
3105 1141 TAD M10 /SUBTRACT 1 FROM OF
3106 3151 DCA NXLOC /NEXT FIELD LOWER
3107 5251 JMP CKERR

3110 7200 /STMV, CLA
3111 1020 TAD ERTBL
3112 3022 DCA ERWRD
3113 6224 RIF
3114 3723 DCA I XSRCE
3115 1723 TAD I XSRCE
3116 7041 CIA
3117 -1016 TAD INSLD
3120 7650 SNA CLA /DON'T MOVE IF EQUAL
3121 5530 JMP I XRTN /START OVER
3122 5724 JMP I XMVE /GO MOVE

3123 3323 XSRCE, SOURCE
3124 3307 XMVE, MOVE
3125 3200 XTMV, NXTMV,
/ *3200 7600
3200 7600 NXTMV, 7600

PAL10 V141

2=NOV=71 0119 PAGE 1=30

```

3201 6224 RIF      DCA SOURCE /CURRENT FIELD
3202 3323 ISZ ERWD /POINTER +1
3203 2022 CHNXT, TAD NXLOC
3204 1200 CIA
3205 7041 TAD I ERWD
3206 1422 SNA CLA /NO ERRORS RECORDED IF 0
3207 7650 JMP STNXT /INITIALIZE MOVE
3210 5225 TAD I ERWD
3211 1422 CIA
3212 7041 TAD NXLOC
3213 1151 SNA CLA /ERROR IN NEW FIELD IF 0
3214 7650 SNA CLA /TRY NEXT LOWER FIELD
3215 5255 JMP SUB2
3216 1022 TAD ERWD
3217 7041 CIA
3220 1021 TAD ENTBL
3221 7650 SNA CLA /DONE WITH TABLE IF 0
3222 5225 JMP SYNXT /INITIALIZE MOVE
3223 2022 TAD ERWD /POINTER +1
3224 5211 JMP CKNYT

3225 1020 STNXT, TAD ERTBL /RESTORE TABLE POINTER
3226 3022 DCA ERWD /NEXT LOWER FIELD
3227 1151 TAD NXLOC
3230 7041 CIA
3231 1016 TAD INSFLD
3232 7650 SNA CLA /NEXT=>CURRENT IF 0
3233 5242 JMP CKNT
3234 1151 TAD NXLOC
3235 7041 CIA
3236 1123 TAD FIRST1
3237 7640 SNA CLA /NEXT = LOWEST IF 0
3240 5247 JMP STNXT /MOVE TO LOWEST TEST FIELD
3241 5302 JMP MVBK
3242 1151 TAD NXLOC
3243 7041 CIA
3244 1123 TAD FIRST1
3245 7650 SNA CLA /NEXT = LOWEST IF 0
3246 5275 JMP NXTH1 /SETUP TO MOVE TO HIGHEST
3247 1151 TAD NXLOC /NEXT LOWER FIELD
3250 3016 DCA INSFLD /IS NOW CURRENT FIELD
3251 1016 TAD INSFLD
3252 1141 TAD M10
3253 3151 DCA NXLOC
3254 5307 JMP MOVE /SUBTRACT 1 FROM NEW
                                /NEW NEXT LOWER FIELD
                                /GO MOVE

3255 1020 SUB2, TAD ERTBL
3256 3022 DCA ERWD
3257 1151 TAD NXLOC
3260 7450 SNA
3261 5530 JMP IXRTN
3262 1141 TAD M10

```

/ RESTORE TABLE POINTER
 /NEXT LOWER FIELD
 /FIELD S IF 0
 /START OVER CAN'T MOVE
 /SUBTRACT 1

PAL10 V141 2-NOV-71 2119 PAGE 1-31

3263 3151 DCA NXLOC /NOW # 2 FIELDS LOWER
3264 1151 TAD NXLOC
3265 7041 CIA TAD INSLFD /CURRENT FIELD
3266 1016 CIA SZA CLA /ARE THEY EQUAL
3267 7640 JMP CHNXT /NO
3268 5203 TAD NXLOC /YES
3271 1151 SNA /DOES IT = FIELD 0
3272 7450 JMP CHNXT /YES
3273 5203 JMP SUB2±5 /NO
3274 5262 /
3275 1124 NXTHI, TAD LAST1 /VERY LAST TO TEST
3276 3151 DCA NXLOC /MAKE IT NEXT FIELD
3277 1124 TAD LAST1
3300 3016 DCA INSLFD
3301 5203 JMP CHNXT
3302 1151 TAD NXLOC
3303 3016 DCA INSLFD
3304 6224 RIF
3305 3323 DCA SOURCE
3306 3015 DCA FLAGS /CLEAR BIT 14

/ROUTINE TO RELOCATE 4K FIELDS

3307 1125 MOVE, TAD KCDF /6201
3310 1323 TAD SOURCE /CURRENT FIELD
3311 3323 DCA SOURCE /SOURCE NOW = CDF N
3312 1125 TAD KCDF /6201
3313 1616 TAD INSLFD /NEW FIELD
3314 3327 DCA DESTN /DESTN NOW = CDF N
3315 1323 TAD SOURCE
3316 7041 CIA DESTN
3317 1327 TAD DESTN
3320 7650 SNA CLA
3321 5530 JMP IXRTN
3322 3074 DCA LOOP /4K COUNTER
3323 0000 SOURCE, 0 /WILL = CDF N
3324 1474 TAD I LOOP /TAKE FROM HERE
3325 3347 DCA SAVGD /SAVE INSTRUCTION
3326 1347 TAD SAVGD /GET IT BACK
3327 0000 DESTN,
3330 3474 DCA I LOOP /PUT IN HERE
3331 1474 TAD I LOOP /GET INFORMATION STORED
3332 7041 CIA
3333 1347 TAD SAVGD /COMPARE TO THIS VALUE
3334 7650 SNA CLA /WERE THEY THE SAME
3335 5340 JMP +3 /YES CONTINUE
3336 7402 HLT /NO, RELOCATION ERROR
3337 5323 JMP SOURCE /TRY SAME AGAIN
3342 2074 ISZ LOOP /DONE 4K WHEN SKIP
3341 5323 JMP SOURCE /KEEP MOVING
3342 1126 TAD KCIF /6202

/

PAL10	V141	2=NOV=71	0119	PAGE 1=32
3343	1016	TAD INSFLD	/NEW FIELD	
3344	3345	DCA :+1		
3345	6202	CIF 00	/CHANGE TO NEW FIELD	
3346	5530	JMP I XRTN	/EXIT TO RTN1 IN	
3347	0000	SAVGD, 0	/NEW FIELD	
		\$		

/ PAL10 V141 2-Nov-71 0119 PAGE 1-34

4000
4100

4200
4300

4400
4500

4600
4700

5000
5100

5200
5300

5400
5500

5600
5700

6000
6100

6200
6300

6400
6500

6600
6700

7000
7100

7200
7300

7400
7500

7600
7700

PAL10

PAGE 1-35

0119

2=NOV=71

1455

V141	22220	FLCNT	0073	LEGAL	2314
	ALOK	FLDN	2444	LOCAT	2400
	BAD	FLOAD	0017	LOOP	3074
	BADD	GENRAN	2716	M10	0141
	BEGIN	GODD	2555	M14	0140
	BSPCE	GOOD	2164	M20	0056
	CBANK	HALT	2641	M4	0060
	CDF	INSFLD	0016	M40	0057
	CDON1	JMS1	0110	M5	0061
	CFLD	JMS2	0111	MCHA	0033
	CHAR	K10	0034	MEMADR	0122
	CHNXT	K100	0040	MOVE	3307
	CIF	K177	2956	MVBK	3302
	CKBNK	K11	1036	NXLLOC	0191
	CKERR	K1000	0042	NXTBNK	1624
	CKNT	K1000	0040	NXTTHI	3275
	CKNXT	K177	2956	NXTMV	3200
	CLOP1	K12	0042	OADDR	2165
	CLOP2	K20	0042	OKAS	2256
	CLOP3	K200	0037	OTLDR	2541
	CLOP4	K207	2167	OVER	1706
	CMOVE	K212	2171	PREAD	2605
	CNXT	K215	0143	PFILD	2500
	COUNT	K240	2170	PGOOD	2524
	CRLF	K254	2353	PHDR	2474
	DATFLD	K260	0142	PLIMT	2452
	DESTN	K261	0052	PLOCT	2412
	ENTBL	K262	0053	POCDR	2512
	EPRNT	K263	0054	POSITN	2131
	EQUAL	K264	0055	PRERR	2146
	EREXT	K277	0146	PROCTL	2126
	ERROR	K370	0145	PSTSR	2652
	ERTBL	K377	0144	PTSTN	2617
	ERWRD	K40	0041	QUERY	2344
	EXAM1	K400	0036	RANCON	2741
	EXAM2	K7	0133	RANDEX	2742
	EXAM3	K740	0035	RANTAB	2754
	EXAM4	KCDF	0125	RANTAD	2734
	EXHDR	KCIF	0126	RANTBL	2743
	EXIT	K153	KEYIN	TSTNL	2457
	EXT1	0322	KRXT	TSTN1	2626
	EXT1C	0343	KXT1	TSTN2	0452
	EXT2	0423	KXT1C	TST3C	0477
	EXT2C	0445	KXT2	TST4	0400
	EXT3	0475	KXT2C	TST4C	0425
	EXT3C	0517	KXT3	TSTL	0452
	EXT4	0623	KXT3C	W0011	0652
	EXT4C	0647	KXT4	W0110	0704
	FEILD	1646	KXT4C	W1001	0721
	FILD	2532	KXT4C	W1100	0667
	FIRST1	0123	KXT4C	W1200	0667
	FLAGS	0015	KXT4C	W1300	0665

2•NOV•71 V141 0119 PAGE 1•36

XTST10	1244	XTST3	0246
XCF1	0076	XTST4	2047
XCHK1	0100		
XCHK1C	0077		
XCHK2	0101		
XCHK2C	0102		
XCHK3	0102		
XCHK3C	0103		
XCHK4	0104		
XCHK4C	0105		
XCHKA	1236		
XCRLF	0147		
XFILE	0131		
XHDR	0135		
XHLT	0127		
XIT1	0320		
XIT1C	0341		
XIT2	0421		
XIT2C	0443		
XIT3	0473		
XIT3C	0515		
XIT4	0621		
XIT4C	0643		
XKBNK	0070		
XLCAT	2352		
XLMTS	0043		
XLOPA	1237		
XLOPB	1240		
XLOPC	1241		
XLOPD	1242		
XLOPE	1243		
XMOVE	0050		
XMVE	3124		
XPERR	0136		
XPHDR	2644		
XPING	0137		
XPRER	0132		
XRALL	0075		
XRD1	0115		
XRD2	0116		
XRD3	0117		
XRD4	0120		
XRROR	0121		
XRTN	0130		
XSALL	0152		
XSETU	0051		
XSRCE	3123		
XSTS1	0277		
XTBNK	0071		
XTLIM	2351		
XTMV	3125		
XTST1	0044		
XTST2	2045		

PAL10 V141 2-Nov-71 0119 PAGE 1-37

ERRORS DETECTED: 0

LINKS GENERATED: 0

RUN-TIME: 13 SECONDS

2K CORE USED.