

IDENTIFICATION

Product Code: MAINDEC-15-DIFA-D(D)
Product Name: PDP-15 Extended Memory Address Test
Date Created: September 22, 1969
Maintainer: Diagnostics Group
Author: John W. Richardson

1. Abstract

The PDP-15 Extended Memory Address Test tests all of core memory not occupied by the program to ensure that each location can be uniquely addressed. This is done by a series of four tests. The first two tests write the address and complement address of each memory location into itself, and then check each location to make sure each is correct. The third test slides a one through a word of zeroes in each memory field. Test four writes and reads a pattern designed to detect adjacent bit positions shorted within a memory stack.

The program relocates automatically from field to field, and tests all of core memory from each field.

2. Requirements

2.1 Equipment

A PDP-15 equipped with a minimum of 8K of core memory.

2.2 Storage

The program utilizes approximately 3K (decimal) words of its resident 4K memory field.

3. Loading Procedure

3.1 Method

The tape supplied is for HRM, and may be loaded into any low order 4K field (X0000-X7777).

Place the tape in the reader; place the BANK MODE switch on a 1; set the ADDRESS switches to 017700; press I/O RESET and then READ-IN.

4. Starting Procedure

Start from 200 to initialize the program.

Restart from 221 to retain current operating parameters (amount of core to test, etc.)

4.1 Program/Operator Action

After program load or restarting from 200, the operator must indicate to the program the amount of core memory to be tested, and any special function to execute. The amount of core memory is indicated via keyboard.

The program will first print the message "TEST LIMITS". The operator must then specify, following the procedure listed below, the amount of core memory to be tested. The program expects the 4K memory fields to be numbered octally beginning with field 0 (locations 00000 to 07777) through 7 (locations 70000 to 77777). For a maximum of 32K of memory the memory addresses for the 4K fields would be:

<u>Field #</u>	<u>Memory Addresses (octal)</u>
0	00000 to 07777
1	10000 to 17777
2	20000 to 27777
3	30000 to 37777
4	40000 to 47777
5	50000 to 57777
6	60000 to 67777
7	70000 to 77777

Use the following procedure to specify TEST LIMITS.

- a. Type two numbers, separating the numbers with a comma.
- b. Press the carriage return key.
- c. The first number typed signifies the first 4K field to test, and the second number the last 4K field to test.
- d. The program will begin testing with the lowest order 4K field specified, and will test all consecutive fields up to and including the highest specified.
- e. The 4K field containing the program may be included. It will be tested after program relocation takes place. Program relocation is discussed in section 5.3.1.
- f. If a typing error is made, press the RUBOUT key. "TEST LIMITS" will be printed again. Previous input is ignored.

- g. The highest 4K field to test may be typed first. The program will reverse the two numbers so as to make the first number the last to test.
- h. Any single field or any two or more consecutive fields may be specified.

For the following examples assume that the program is in field 0 , and the PDP-15 being used is equipped with 32K of core memory.

Example A:

TEST LIMITS
0, 7 ↵ (↵ denotes carriage return)

The program will test all 32K of memory.

Example B:

TEST LIMITS
7, 0 ↵

The program will perform exactly as Example A.

Example C:

TEST LIMITS
3, 3 ↵

Only field 3 will be tested.

Example D:

TEST LIMITS
4, 6 ↵

Fields 4, 5, and 6 will be tested.

Example E:

TEST LIMITS
0, 0 PROGRAM IS IN FIELD 0
TEST LIMITS
0, 1 ↵

Example E shows the message printed by the program when a

single field is specified which currently contains the program. "TEST LIMITS" is printed again, and the operator must then correct the test limits.

4.1.1 SETUP ACS

After specifying the test limits, the program will print "SETUP ACS". For normal program operation the ACS must be set to 000000 octal. Press any key (except RUBOUT) on the keyboard after setting the ACS to all 0's. "RUBOUT" will restart the program with "TEST LIMITS". After terminating with a key, the program will run until stopped by the operator. Normal program operation is defined as performing all eight checkerboard patterns on all of available memory from every 4K field.

5. Operating Procedure

- a. Load the program into memory field 0 as described in section 3.
- b. Specify the test limits as described in section 4.1.
- c. The message "SETUP ACS" will be printed. Set the ACS to 000000, and press any key except RUBOUT.
- d. The program will perform all four tests on all of memory specified, then automatically relocate to the highest field number under test.

5.1 ACS Settings

Normal operation of the program requires the ACS set to 000000. Refer to section 8.2, Applications for switch settings provided for trouble-shooting.

5.2 Subroutine Abstracts

The program executes a series of four tests on core memory. Each test writes a unique pattern, and checks each location for error.

Test 1 writes the value of each memory location into itself, from the lowest order to the highest order field under test. The address pattern is then read, and checked for error, in the same direction i.e., from the lowest to highest field. The pattern is then read and checked for error in the reverse direction, i.e., from the highest field to the lowest field. This sequence is repeated twice before test 2 is initiated. Test 1 may be run by itself by placing ACS 3 on a 1.

Tests 2, 3 and 4 write and read their patterns into one field at a time. The rest of memory will contain an all 1's pattern. After each pattern is written and read, the rest of memory is checked to make sure that its pattern has not changed. After one field has been tested the next higher field in sequence will contain the pattern, and the rest of memory will equal all 1's. This is continued for one test until all fields have contained the test pattern. The next test in sequence will then be initialized. All of memory is set to 1's before the next field in sequence is tested.

Test 2 first writes 1's into all of memory, and then writes the complement value of each address into itself. The pattern is written once in the forward direction in one field. The rest of memory is then read and checked for error (in the forward direction). The field with the address pattern is then read once in the forward and reverse directions, after which the rest of memory is again checked.

The same field with the address pattern is then reset to all 1's, and the same address pattern is then written in the reverse direction. The rest of memory is then checked; the field with the pattern is read in forward and reverse directions, and finally the rest of memory is again checked. The next higher field in sequence will be tested in the same manner. After all fields have been tested in this manner, test 3 is initialized. Test 2 may be run alone by placing ACS 4 on a 1.

Test 3 first writes 1's into all of memory, and then writes a sliding 1 pattern into one field. Each location of the field will contain a word of all 0's except for one bit position. The bit set is rotated one place to the left for each memory location, starting with bit 17. The following test sequence is repeated 18 times, resulting in every bit in each memory location being set.

- a. Write all 1's into all of memory.
- b. Write a sliding one pattern into one field.
- c. Read and test rest of memory.
- d. Read and test the field with the pattern in the forward direction only.
- e. Read and test rest of memory.
- f. Repeat steps b through e 17 more times before testing the next sequential field in the same manner.

After all fields have been tested with a sliding 1, test 4 is initialized. Test 3 may be run alone by placing ACS 5 on a 1.

Test 4 writes ones into all of memory. A pattern consisting of 1 word of 0's followed by 1 word of 1's is then written into one field. The following sequence is then executed.

- a. Write the pattern in one field in the forward direction.
- b. Read the field with the pattern in the forward direction.
- c. Read again in the reverse direction.
- d. Read rest of memory.
- e. Write the complement pattern in the forward direction, and repeat steps b, c, and d, then do step f.
- f. Write the pattern in the reverse direction, and do steps b, c, and d, then do step g.
- g. Write the complement pattern in the reverse direction, and do steps b, c and d, then setup to repeat a through g on the next higher field in sequence.

After all fields have been tested, the program then relocates automatically and starts over with test 1. Test 4 may be run alone by placing ACS 6 on a 1.

5.2.1 Program Relocation

The program relocates itself in order to test addressing from all fields to every other field. Relocation depends upon the amount of core memory being tested. Relocation is always within the group of 4K fields selected for testing, and under certain conditions the program will not relocate at all, but will remain in the current field to perform the tests (see below). The program normally first relocates to the highest order 4K field under test. From there it relocates to the next lower 4K field, after performing all tests. The program keeps relocating to the next lower 4K field until it reaches the lowest order 4K field under test. The testing and relocation cycle is then repeated. As an example, suppose the program initially is in field 0, and 32K of memory is selected for test. The tests are run from field 0, and then the program relocates to field 7, then to fields 6, 5, 4, 3, 2, 1, 0 in that order. The program will not relocate to any field which is not included in the test limits. If fields 4, 5 and 6 were selected, relocation

would be from 0 to 6, then to 5 and 4. Fields 0 through 3 and field 7 would not contain the program again until included in the test limits.

The program will not relocate if any of the conditions described below exist:

- a. A forced relocation has been made (section 8.2.7).
- b. Only one 4K field is selected for testing.
- c. An error was detected in all of the available 4K fields under test.
- d. ACS 9 is on a 1 to inhibit program relocation (section 8.2.5).

The location of the program is indicated by the message "PROGRAM IS IN FIELD X", where X is the field number. This message occurs immediately after each program relocation. The message print-out may be deleted by placing ACS 11 on a 1 at any time. The print-out will resume when ACS 11 is placed on a 0.

The program provides a degree of protection for itself by not relocating to any field which has an error. The number of the field in error is saved, and is compared to the destination field number before relocation takes place. If equal, the next lower field is setup as the destination providing it has no error. The first field found to be error-free is set up as the destination. Relocation will not take place if all fields have shown errors. The program will resume relocating to a field whenever the error condition does not exist.

During the relocation process the program tests each data word transferred to the new field by performing the transfer, reading the word back and comparing the word with the correct data in the current field. This is done on a one for one basis until the process is completed. The entire 4K field is moved to enable loaders or any other data to be carried with the program. If an error is found during relocation, the address in error, and the "good" and "bad" data words are printed. The error print-out format is described in section 6.

One pass of the program is defined as all four tests performed on all of memory from each 4K field.

6. ERRORS

6.1 Error Print-outs and Description

Immediately after the first error is detected, the header shown below is printed.

TEST	OCTAL ADR.	GOOD	BAD	FIELD WITH PAT.
------	------------	------	-----	-----------------

Where:

TEST = the current test which detected an error.
OCTAL = the memory location which contains the data in error.
GOOD = what the data should have been in that location.
BAD = the data as read from that location.
FIELD WITH PAT. = the current field under test which contains the pattern of the failing test. For test 1, this will equal "ALL", since test 1 writes an address pattern into all memory under test. For test 2, 3 or 4, it will equal 0, 1, 2, 3, 4, 5, 6 or 7, depending upon the amount of core memory available for test.

Example:

TEST	OCTAL ADR.	GOOD	BAD	FIELD WITH PAT.
1	060100	060100	060000	ALL
2	060100	777677	776677	6
3	023000	000002	000003	2
4	047777	777777	000000	4
4	047776	777777	000000	4
4	017777	777777	000000	4

During test 1 address 100 in field 6 was found to be in error. From the example, it can be seen that bit 11 was dropped. Bit 8 was dropped during test 2 in the same address. Bit 17 was picked up at location 3000 in field 2 during the sliding 1 test. Three consecutive addresses in field 4 were in error during test 4. The test was reading in the reverse direction at the time, because the addresses are printed in descending order. Also, when checking rest of memory, location 07777 in field 1 was found to be incorrect.

After each print-out the program continues with the next memory location to test.

Three AC switches may be used to control the error print-outs. Placing ACS 0 on a 1 during the print-out will cause a program halt

after completion of printing. ACS 1 on a 1 will inhibit the print-out and cause a program halt. Press CONTINUE to receive the error print-out and to continue testing, ACS 2 on a 1 will inhibit print-out and ring the TTY BELL for each error. The use of these switches is described in section 8.2 in more detail.

6.1.1 PROGRAM RELOCATION ERROR

This message will be printed upon detection of a relocation error. The error information will immediately follow as in the example below. After all errors have been printed the message "NO MORE ERRORS" is printed, and the program will then set up to relocate to the next lower field if one is available.

Example:

TEST	OCTAL ADR.	GOOD	BAD	FIELD WITH PAT.
PROGRAM RELOCATION ERROR				
	031000	740100	740000	
	031001	611005	601005	
	031002	760207	760007	
NO MORE ERRORS				

The above example shows those consecutive errors during program relocation to field 3. Field 2 would be set up for relocation. Location 1000 in field 3 should have contained a SMA instruction, but bit 11 was dropped during the transfer. Bit 5 was dropped in the JMP instruction in 1001, and bit 10 dropped in the LAW instruction in 1002.

PRINT-OUTS INHIBITED

The above message is printed whenever 64 (decimal) consecutive print-outs have occurred. Error print-outs will be inhibited until after all four tests have been run eight times, after which the error print-outs will resume for 64 more print-outs. This feature is not used with program relocation errors.

This feature is included to prevent lengthy error print-outs when the program is being run for an extended period of time unattended. Error print-outs may be resumed by restarting the program from location 200.

6.1.3 PROGRAM IS IN FIELD X

Where "X" is a field number. This message is printed if one of the following conditions exist:

- a. The operator has specified a single field for testing and that field contains the program. Select another field, refer to section 4.1.
- b. The operator has requested to relocate the program to a 4K field which currently contains the program. See section 8.2.7 for instruction to force the program to another field.
- c. After every program relocation.

6.1.4 ERROR IN SELECTED FIELD

This message is printed when a forced program relocation is attempted and the program has previously detected a data error in that field. Type a new field number, or press carriage return to resume automatic program relocation. See section 8.2.7 for instructions to force the program to another field.

7. Restrictions

7.1 Starting Restrictions

Start from 200 to set up the test limits and ACS and to reinitialize the program.

Start from 221 to retain the present program conditions.

7.2 Operating Restrictions

Don't use the STOP key to halt the program. Place ACS 0 on a 1.

8.2 Applications

To give the operator control of the program, the ACS were assigned unique functions. The ACS assignments and their effect on the program are described below. Please note that it is important that the program be halted with ACS 0 rather than the STOP key. Using the STOP key may result in a halt while the program is in the process of relocating, which is disastrous. Any ACS listed may be raised or lowered while the program is running. The operation may not be

initiated immediately since most of the ACS are sensed only after all tests have been performed.

8.2.1 Halt after Test or Error Print-out - ACS 0

Placing ACS 0 on a 1 at any time while the program is running will cause a halt after the current test is completed on one 4K field. The MO will = 1366. The ACS may then be changed if desired. Press CONTINUE to recover. If no ACS changes the program will resume the test which was interrupted. If ACS changes were made the new setting are stored and executed.

Raising ACS 0 during an error print-out will cause a halt at the same location mentioned above, after the print-out.

8.2.2 Delete Error Print-out and Halt on Error - ACS 1

ACS 1 on a 1 at any time causes all data error print-outs to be inhibited. A halt will occur with the MO = 1366 if an error occurs. Press CONTINUE to receive the error print-out and to resume testing. ACS changes may be made.

8.2.3 Bell on Error - ACS 2

ACS 2 on a 1 causes the program to ring the TTY BELL whenever an error occurs. This is convenient when testing with power supply margins. ACS 1 has no effect if ACS 2 and 1 should both happen to be on a 1. If ACS 0 and 2 are 1, a halt occurs after the bell. Proceed as described in 8.2.1.

8.2.4 Test Selection - ACS 3 through 6

Any one, or any combination of tests may be executed by placing combination of ACS 3 through 6 on a 1. ACS 3 specifies test 1; ACS 4, test 2; ACS 5, test 3; ACS 6, test 4. The test specified by the most significant ACS will be performed first.

If all four ACS are 0, all four tests are performed in sequence.

The ACS may be changed while the program is running. The new tests will be recognized after the last of the current selection is performed.

8.2.5 Inhibit Program Relocation - ACS 9

The program normally relocates automatically as indicated by the print-outs. To retain the program in its current 4K

field place ACS 9 on a 1 at any time. Placing it on a 0 enables relocation to resume.

8.2.6 Inhibit "PROGRAM IS IN FIELD" - ACS 11

The program normally prints the field number containing the program immediately after each relocation. The message may be suppressed by placing ACS 11 on a 1 at any time. To resume the print-out place ACS 11 on a 0. This switch does not inhibit the message print-out when an operator error is made.

8.2.7 Program Relocation - ACS 12

The operator may relocate the program to any 4K field by specifying a forced relocation with ACS 12 on a 1. Use the following procedure:

- a. Halt the program with ACS 0.
- b. Place ACS 12 on a 1 and ACS 0 on a 0. Press CONTINUE.
- c. A print-out will occur which instructs the operator to place ACS 12 on a 0. The program will loop until this is done.
- d. With ACS 12 on a 0 the message GO TO FIELD is printed followed by the program waiting for a field number.
- e. Type the desired field number (0 through 7).
- f. Relocation is done immediately, and the program is executed in the new field.

The program will not relocate again until restarted from 200 or 221, or in step d above, press carriage return to resume automatic relocation.

If a data error was previously detected in the new 4K field, the message "ERROR IN SELECTED FIELD" is printed, followed by step d repeated. Type another field number, or carriage return to resume normal operation.

Each word transferred to the new field is tested in the same manner as described in section 5.3.1, Program Relocation. Print-outs occur for each relocation error. Step d will be repeated after all error reporting is done. Type another field number, or carriage return to resume normal operation.

At times the program will automatically restart at 200 and print TEST LIMITS. This will occur whenever a single field has been selected for testing, and the operator relocates the program to that field. New test limits must be specified since the program cannot run the tests on its own 4K field. Proceed as described in section 4.3.

8.2.8 Request Keyboard Input - ACS 13

ACS 13, when up, indicates to the program that the operator wishes to select one test, along with one or more addresses to be suppressed, and that one or more addresses are to be tested, disregarding all other addresses. ACS 13 is recognized immediately after restarting from 200, 221 or after pressing CONTINUE after a halt with ACS 0.

The program will print the following information waiting for input from the keyboard after each line:

TEST # -
SUPPRESS -
BLOCK # 1 -
BLOCK # 2 -

The program expects input information for each line. Typing only a carriage return indicates that the function represented by that line is not wanted. The next line will be printed. An explanation of each line follows.

TEST # - Type the test number desired (1 to 4). Any number less than 1 or greater than 4 is an error, and a ? will be printed, followed by TEST # - being printed again. If no particular test is wanted, type a carriage return only. The pattern used by the last test in progress will be used. In the case of no tests being previously run, test 1 will be used. TEST # is ignored.

SUPPRESS - Error print-outs for one or more individual addresses, or a block of consecutive addresses may be suppressed. The addresses typed must be 5 digit octal numbers. Up to 256 (decimal) addresses, individual or a block, may be suppressed. Any amount over 256 (decimal) is not an error, but will be ignored by the program.

Suppression of individual addresses is indicated by separating each 5 digit address with a colon, terminating the line with a carriage return after the last address or colon. A block is indicated by typing the first address of the block, and the last address of the block, separating the two by a comma. An automatic carriage return is provided after the second address is typed. The individual or block of addresses do not have to be typed in numerical order.

Any input which differs from the above will result in a ? being printed, followed by SUPPRESS - being printed again. Typing a carriage return only indicates no addresses are to be suppressed.

BLOCK #1 and BLOCK # 2 - Any length block of consecutive addresses to be tested may be indicated in either BLOCK # 1 or BLOCK #2, or both. The block limits must not overlap the program or exceed the amount of memory available. If the limits of either block overlap the program a message will be printed giving the location of the program. The current line will be re-printed, and the operator must then specify new limits. Indicate the block to be tested by typing the first address and last address of the block, separating the two 5 digit addresses with a comma.

The program handles the blocks to be tested as if each were a separate memory field. That is, if test 1 were selected, each address within either block would contain its own value. If any other test were selected, BLOCK 1 would contain the pattern, and BLOCK 2 would contain an all 1's pattern. The rest of memory outside of the block limits is ignored. Refer to section 5.2, Subroutine Abstracts, for a description of methods of testing, and patterns generated by each test.

When two blocks are selected, their limits should not overlap when tests 2, 3 or 4 are selected. BLOCK 2 will always contain an all 1's pattern with these tests, and error print-outs will occur if both block limits overlap.

If only one block is selected, the pattern will be written in that block, and the rest of memory is ignored. A single block may be indicated after BLOCK 1 or BLOCK 2; one of the lines being left blank.

If no blocks are to be tested type a carriage return after BLOCK # 1 and BLOCK # 2. The program will start over with test 1, or the test selected in ACS 3-6.

If a block is selected, program relocation will not take place.

Press the RUBOUT key if a typing error is made. The routine will start over with "TEST" # -.

If the pattern generated by test 4 is to be used, the operator may select one address, or a block which has an even total of addresses, i.e., 2, 4, 6, 10, etc. This is necessary due to the method of generating and reading the pattern. A guide would be to always

make the first address even (XXX0), and the last address of the block odd (XXX7). If an odd total is requested, invalid error print-outs will occur indicating an error at each location. This restriction applies to BLOCK # 1 when two blocks are selected, or to a single block.

Examples:

Assume the program to be located in field 0.

A. TEST # -4
SUPPRESS -
BLOCK # 1 - 10100, 10101
BLOCK # 2 -

The pattern generated by test 4 will be written and checked for error from 100 to 101 of field 1. The rest of memory (including field 1) will contain an all 1's pattern, and will be ignored by the program since only one block is selected.

B. TEST# -4
SUPPRESS -
BLOCK #1 - 10101, 10100
BLOCK #2 -

The program will perform exactly as described in example A.

C. TEST # -3
SUPPRESS -
BLOCK #1 - 37777, 36000
BLOCK #2 - 10000, 16000

The pattern generated by test 3 will be written and checked for error in the first block from 36000 to 37777. An all 1's pattern will be written and checked for error in the second block from 10000 to 16000.

D. TEST# -
SUPPRESS - 37777:36100:35000
BLOCK #1 - 10000, 16000
BLOCK #2 - 36000, 37777

The pattern used by the last test in progress will be written in BLOCK 1, and an all 1's pattern in BLOCK 2 (note that the blocks are reversed from Example C). Any error at the two addresses 36100 or 37777 will not be printed. Address 35000 is meaningless since it is located outside the test limits.


```

F.   TEST # - 8
      ?
      TEST # -1
      SUPPRESS - 036000
      ?
      SUPPRESS - 36000, 37777
      BLOCK #1 - 40100, 40100
      BLOCK #2 - 00000, 16000
      BLOCK #2 - LIMITS OVERLAP PROGRAM
      (PROGRAM IS LOCATED IN FIELD 0)
      BLOCK #2 - 30000, 30100

```

Example F indicates program response to format errors. The first, TEST #, is self-explanatory, since there are only 4 tests to choose from. The second, SUPPRESS, was in error because the number exceeds 5 digits. The program is assumed to be in field 0 for these examples, and the first address selected for BLOCK 2 is within this area. The example shows the error message printed for BLOCK 2, followed by the operator's correction.

```

G.   TEST # - 2
      SUPPRESS - 10000, 10377
      BLOCK #1 -
      BLOCK #2 -

```

The program will not loop on test 2, but will restart with test 1, or the test specified in ACS 3-6. TEST # is used only when one or more blocks are specified. All of memory specified will be tested.

Any errors detected within the block indicated after SUPPRESS will not be printed, regardless of which test is running.

To return to normal program operation type a carriage return after each of the four lines, or restart the program from 200.

9. Program Description

The Extended Memory Address test is intended for use with PDP-15s equipped with 8K or more of core memory. A total of four tests are executed by the program. Each test writes a unique pattern into core memory and then checks for error. The first test writes the value of each memory address into that address, with all available banks containing the pattern. The remaining four tests write their patterns into one 4K field at a time, with rest of memory containing an "all ones" pattern. The patterns were chosen so as to detect word and bit errors, as well as shorted wires within any bank (see sect. 5.2).

Control of the program is given to the operator by means of the ACS. The operator may halt the program inhibit error print-outs, substitute the TTY BELL for error indication, halt after print-out, select any one or group of tests, inhibit program relocation, specify any single address or group of addresses to be suppressed, any single address or up to two blocks of addresses for testing, relocate the program to any 4K area, and vary the number of 4K fields to test. See section 8.2 for the ACS designations for the above functions.

The program automatically relocates after performing all specified tests on the amount of core memory selected.

10. Listing

```
/
/COPYRIGHT 1969, DIGITAL EQUIPMENT CORP.,
/MAYNARD, MASS.
/
/
/PDP-15 EXTENDED MEMORY ADDRESS TEST.
/SA = 200. RESTART AT 221.
/8K MINIMUM CORE REQUIRED
/
/J. RICHARDSON
/
/IOT DEFS.
/

```

```
700406 TLS=700406
700401 TSF=700401
700312 KRB=700312
700301 KSF=700301
707762 EPA=707762
/
```

		.TITLE XAD15	
		.AHS	
00001		.LOC 1	
00001	600001	JMP 1	
00002	000002	2	
00003	000003	3	
00004	777777	LAW -1	
00005	777777	LAW -1	
	/		
00200		.LOC 200	
	/		
00200	707762	BEGIN EPA	/ENTER PDP-15 MODE
00201	144270	DZM FLAGS	/CLEAR PROGRAM FLAGS
00202	101360	JMS WHERE	/SEE WHERE PROGRAM IS
00203	044271	DAC INFLD	/SAVE FIELD NUMBER
00204	103555	JMS SLMTS	/SETUP ADDRESS LIMITS
00205	103662	JMS SETAC	/SETUP ACS
00206	204012	LAC STBL	
00207	044013	DAC SUPTBL	/POINTER FOR SUPPRESS TABLE
00210	760000	LAW	
00211	044320	DAC BLOC1	
00212	044322	DAC BLOC3	
00213	064013	DAC* SUPTBL	/LAW = NONE SUPPRESSED
00214	777700	LAW -100	/-64 DECIMAL
00215	044324	DAC MAXERR	/COUNTS 64 ERRORS
00216	777770	LAW -10	
00217	044266	DAC SIXT4	
00220	144267	DZM NOPRNT	
00221	202335	LAC STLOOP-1	
00222	042320	DAC LOCAT+4	
00223	707762	EPA	
00224	102314	JMS LOCAT	
00225	143746	DZM PHDR	
00226	204301	LAC LAST1	/LAST FIELD TO TEST
00227	544300	SAD FIRST1	/FIRST FIELD TO TEST
00230	741000	SKP	/THE TWO ARE EQUAL
00231	600234	JMP .+3	
		.EJECT	

00232	044271	SAC	INSFLD	/DO THEY EQUAL CURRENT FIELD
00233	00201	JMP	REGIN+1	/SETUP NEW ADDRESS LIMITS
00234	004043	LAC	ERTBL	
00235	004325	DAC	ERWRD	/ERROR TABLE POINTER
00236	077770	LAX	-10	
00237	044341	DAC	WRCNT	
00240	060000	LAW		
00241	064325	DAC*	ERWRD	/LAW = NO ERROR IN TABLE
00242	444341	ISZ	WRCNT	/FILL TABLE WITH LAW'S
00243	600240	JMP	.-3	
00244	044336	DAC	LAST	/EQUALS LAST FIELD IN ERROR
00245	101360	JMS	WHERE	/SEE WHERE PROGRAM IS
00246	044271	DAC	INSFLD	/SAVE FIELD NUMBER
00247	004043	LAC	ERTBL	
00250	044325	DAC	ERWRD	/RESTORE TABLE POINTER
/				
00251	750004	STOVER	LAS	/READ AC SWITCHES
00252	504406	AND	K177	
00253	044272	DAC	MCWA	/SAVE
00254	504353	AND	K40	
00255	744200	SZA:CLL		/BIT 12 A 1 = FORCE RELOCATE
00256	603273	JMP	FCDMV	/RELOCATE
00257	750004	LAS		
00260	504352	AND	K20	
00261	740200	SZA		/BIT 13 A 1 = KEYBOARD INPUT
00262	601623	JMP	KYBRD	
00263	004272	LAC	MCWA	
00264	504404	AND	K74K	
00265	741200	SNA		/DO ALL TESTS IF 0
00266	600321	JMP	DOALL	
/				
.EJECT				

```

/
/EXAMINE TEST SWITHES 3 TO 7
/
00267 204272          LAC      MCWA
00270 504402          AND      K40K
00271 740200          SZA
00272 600324          JMP      TST1          /BIT 3 A 1 = TEST 1
00273 204401          EXAM2   LAC      K20K
00274 504272          AND      MCWA
00275 740200          SZA
00276 600453          JMP      TST2          /BIT 4 A 1 = TEST 2
00277 204400          EXAM3   LAC      K10K
00300 504272          AND      MCWA
00301 740200          SZA
00302 600700          JMP      TST3          /BIT 5 A 1 = TEST 3
00303 204376          EXAM4   LAC      K4K
00304 504272          AND      MCWA
00305 740200          SZA
00306 601035          JMP      TST4          /BIT 6 A 1 = TEST 4
00307 444266          ISZ     SIXT4         /64 PASSES WHEN SKIP
00310 600314          JMP      .+4
00311 144267          DZM     NOPRNT        /CLEAR INHIBIT PRINT FLAG
00312 777770          LAW     -10
00313 044266          DAC     SIXT4         /RESTORE COUNTER
00314 750004          LAS
00315 504375          AND      K400
00316 740200          SZA
00317 600221          JMP      RTN1
00320 603111          JMP      CMOVE        /BIT 9 A 1 = DON'T MOVE
/STAY IN CURRENT FIELD
/DONE ALL TESTS.  SETUP FOR RELOCATION

/
/SETUP TO RUN ALL TESTS
/
00321 204272          DOALL   LAC      MCWA
00322 244404          XOR     K74K          /SET ALL TEST BITS
00323 044272          DAC     MCWA          /SAVE
/
.EJECT

```

```

      /TEST 1. EACH LOCATION WILL CONTAIN ITS
      /OWN VALUE. ALL OF MEMORY SPECIFIED WILL CONTAIN
      /THE PATTERN.
      /
00324 123170   TST1     JMS      WRT1S      /WRITE 1'S INTO ALL OF MEMORY
00325 760261   LAX      261
00326 444345   DAC      INUM      /TEST NUMBER
00327 102733   JMS      SET1     /SETUP FOR FIRST FIELD
00328 121276   JMS      CBANK    /SEE IF IT HAS PROGRAM
00330 741000   SKP
00332 600360   JMP      READ1    /READ ANT TEST ALL
00333 777776   WBLK1   LAW      -2
00334 444342   DAC      RPETE    /DELAY COUNTER
00335 777775   LAW      -3
00336 444341   DAC      WRCNT    /COUNTS 3 TIMES FOR EACH ADDRESS
00337 204275   WLOP1   LAC      MEMADR
00340 164275   DAC*    MEMADR    /WRITE C(MEMADR) INTO SAME
00341 444342   ISZ     RPETE
00342 600341   JMP     .-1     /DELAY 9 US
00343 777776   LAW     -2
00344 444342   DAC     RPETE
00345 444341   ISZ     WRCNT
00346 600337   JMP     WLOP1   /TOTAL 22.5 US BETWEEN WRITES
00347 444275   ISZ     MEMADR  /ADDRESS + 1
00350 444277   ISZ     CT4K    /4K WHEN SKIP
00351 600335   JMP     WLOP1-2 /WRITE IN NEXT
00352 760000   LAW
00353 544320   SAD     RLOC1   /NO BLOCK IF = LAW
00354 741000   SKP
00355 602440   JMP     BLKA1   /SETUP FOR BLOCK 2
00356 101344   JMS     NXTBNK  /SETUP FOR NEXT FIELD
00357 600330   JMP     WBLK1-3 /SEE IF IT HAS PROGRAM
      .EJECT

```

```

/
/READ AND CHECK FOR ERROR. READ FROM LO FIELD
/TO HI FIELD AND THEN DECREMENT FROM HI TO LO.
/REPEAT THE SEQUENCE TWICE BEFORE FINISHING.
/
00360 777776 READ1 LAX -2
00361 044343 DAC LOOPT
00362 102733 JMS SET1 /SETUP FOR 1ST FIELD
00363 744002 STL
00364 101276 JMS CBANK /SEE IF IT HAS PROGRAM
00365 741000 SKP /NO
00366 600420 JMP RBAK1 /READ BACKWARDS
00367 777766 RLOP1 LAX -12
00370 044342 DAC RPETE /LOOP 10 TIMES ON EACH READ
00371 204275 LAC MEMADR
00372 044274 DAC PATR
00373 224275 LAC* MEMADR /READ
00374 544275 SAG MEMADR /COMPARE
00375 600400 JMP .+3 /OK
00376 101377 JMS ERROR /PRINT ERROR
00377 600402 JMP .+3
00400 444342 ISZ RPETE
00401 600374 JMP RLOP1+5 /LOOP 10 TIMES
00402 444275 ISZ MEMADR /ADDRESS + 1
00403 444277 ISZ CT4K /4K WHEN SKIP
00404 600367 JMP RLOP1 /READ NEXT
00405 760000 LAX
00406 544320 SAG RLOC1 /0 BLOCKS IF = LAW
00407 741000 SKP
00410 602454 JMP RLKB1 /SETUP FOR BLOCK 2
00411 744002 STL
00412 101344 JMS NXTBNK /SETUP FOR NEXT FIELD
00413 600364 JMP RLOP1-3 /SEE IF IT HAS PROGRAM
00414 600420 JMP RBAK1 /READ FROM HI TO LO FIELD
/
.EJECT

```



```

/
00415 000343 LOOP1 ISZ LOOP1 /DONE IF 0
00416 000362 JMP READ1+2 /READ FORWARD ONCE MORE
00417 000373 JMP EXAM2 /CHECK FOR TEST 2.
/
/READ TEST 1 FROM HI FIELD TO LO FIELD
/
00420 102722 PBAK1 JMS SETBAK /SETUP FOR LAST FIELD
00421 102742 JMS CKBAK /SEE IF IT HAS PROGRAM
00422 741000 SKP /NO
00423 600415 JMP LOOP1 /ALL DONE
00424 777766 BAK1 LAW -12
00425 144342 DAC RPETE
00426 024275 LAC MEMADR
00427 144274 DAC PATR
00430 024275 LAC* MEMADR /READ ONE
00431 544275 SAD MEMADR /COMPARE
00432 600435 JMP .+3
00433 101377 JMS ERROR /PRINT ERROR
00434 600437 JMP .+3
00435 444342 ISZ RPETE
00436 600430 JMP BAK1+4
00437 777777 LAW -1
00440 344275 TAD MEMADR /ADDRESS - 1
00441 144275 DAC MEMADR
00442 444277 ISZ CT4K /4K WHEN SKIP
00443 600424 JMP BAK1 /READ NEXT
00444 760000 LAW
00445 544320 SAD BLOC1 /NO BLOCKS IF = LAW
00446 741000 SKP
00447 602471 JMP BLKC1 /SETUP FOR BLOCK 2
00450 102765 JMS NXBAK /SETUP FOR NEXT FIELD
00451 600421 JMP RBAK1+1 /SEE IF IT HAS PROGRAM
00452 600415 JMP LOOP1 /READ FORWARD AGAIN
.EJECT

```

```

/
/TEST2. WRITE COMPLEMENT ADDRESSES INTO ONE
/FIELD AND 77777 IN ALL OTHER FIELDS. WRITE IN
/ BOTH DIRECTIONS (LO TO HI AND HI TO LO) AND READ
/ IN BOTH DIRECTIONS. REST OF MEMORY IS CHECKED
/ AFTER EACH WRITE AND READ.
/
00453 103070 TST2 JMS WRT1S /WRITE 1'S INTO ALL OF MEMORY
00454 102733 JMS SET1 /SETUP FOR FIRST FIELD
00455 744000 CLL
00456 101276 LOP2 JMS CBANK /SEE IF IT HAS PROGRAM
00457 741000 SKP /NO
00460 600501 JMP CKB2 /ALL DONE
00461 777776 LAW -2
00462 044342 DAC RPETE /DELAY COUNTER
00463 777775 LAW -3
00464 044341 DAC WRCNT
00465 204275 WLOP2 LAC MEMADR /ADDRESS
00466 740001 CMA /COMPLEMENT
00467 064275 DAC* MEMADR /ITE INTOSAME
00470 444342 ISZ RPETE /DELAY 4.5 US
00471 600470 JMP .-1
00472 777776 LAW -2
00473 044342 DAC RPETE
00474 444341 ISZ WRCNT /39 US TOTAL BETWEEN WRITES
00475 600465 JMP WLOP2
00476 444275 ISZ MEMADR /ADDRESS + 1
00477 444277 ISZ CT4K /4K WHEN 0
00500 600463 JMP WLOP2-2
00501 760000 CKB2 LAW
00502 544320 SAD BLOC1 /NO BLOCK IF = LAW
00503 741000 SKP
00504 602511 JMP BLKA2 /SETUP FOR BLOCK 2
00505 100511 JMS RST2 /READ REST OF MEMORY
00506 100562 JMS RFWD2 /READ TESTED FIELD LO TO HI
00507 100511 JMS RST2 /READ REST OF MEMORY AGAIN
00510 600637 JMP TST2A /WRITE HI TO LO IN TESTED FIELD
.EJECT

```

```

/
/SETUP TO READ REST OF MEMORY, THEN READ AND TEST
/THE FIELD WITH THE ADDRESS PATTERN.
/
RST2      0
          LAW      262
          DAC      TNUM          /TEST NUMBER
          LAC      LAST1
          SAD      FIRST1       /ONLY 1 SELECTED IF EQUAL
          JMP*     RST2         /NONE TO READ
RTN2      LAC      FIRST1       /FIRST TO TEST
RST2A     DAC      MEMADR
          LAW      -10000       /-4K
          DAC      CT4K
          LAC      KRTN2       /KRTN2 = LOCATION RTN2
          DAC      EXIT
          LAC      MEMADR
          SAD      PATBNK       /DOES FIELD HAVE PATTERN
          JMP      .+3          /YES
          JMS      REST2       /NO. CHECK IT FOR 777777
          JMP      .-4          /SETUP FOR NEXT
          SAD      LAST1       /IS IT THE LAST
          JMP*     RST2         /YES, NO MORE TO TEST
          LAC      PATBNK
          TAD      K10K        /ADD 4K TO ADDRESS
          JMP      RST2A
/
.EJECT

```

```

00511 004300
00512 767262
00513 144345
00514 204301
00515 544300
00516 620511
00517 204300
00520 044275
00521 770000
00522 144277
00523 204003
00524 044307
00525 204275
00526 544340
00527 600532
00530 100537
00531 600525
00532 544301
00533 620511
00534 204340
00535 344400
00536 600520

```

```

/
/READ ALL OF MEMORY EXCEPT FIELD WITH ADDRESS PATTERN.
/
00537 000000 REST2 0
00540 141323 JMS CBNK /SEE IF FIELD HAS PROGRAM
00541 777777 LAW -1
00542 244274 DAC PATR /COMPARE CONSTANT
00543 224275 ALL1 LAC* MEMADR /READ
00544 544274 SAD PATR /DOES IT = 777777
00545 741000 SKP /OK
00546 101377 JMS ERROR /PRINT ERROR
00547 444275 ISZ MEMADR /ADDRESS + 1
00550 444277 ISZ CT4K /4K IF 0
00551 600543 JMP ALL1
00552 760000 LAW
00553 544320 SAD RLOC1 /NO BLOCKS IF = LAW
00554 741000 SKP
00555 624307 JMP* EXIT
00556 744002 STL
00557 101276 JMS CBANK /SETUP FOR NEXT FIELD
00560 620537 JMP* REST2
00561 624307 JMP* EXIT

/
/ROUTINE TO READ THE FIELD WITH THE ADDRESS
/PATTERN FROM LO TO HI.
/
00562 000000 RFWD2 0
00563 204340 LAC PATBNK /FIELD WITH ADDRESS PATTERN
00564 044275 DAC MEMADR
00565 770000 LAW -10000
00566 044277 DAC CT4K
00567 204275 FWD2 LAC MEMADR /ADDRESS
00570 740001 CMA /COMPLEMENT
00571 044274 DAC PATR
00572 224275 LAC* MEMADR /READ
00573 544274 SAD PATR /COMPARE
00574 741000 SKP /OK
00575 101377 JMS ERROR /PRINT ERROR
00576 444275 ISZ MEMADR /ADDRESS + 1
00577 444277 ISZ CT4K
00600 600567 JMP FWD2 /READ ANOTHER
00601 760000 LAW
00602 544320 SAD RLOC1 /NO BLOCK IF = LAW
00603 600605 JMP RBAK2 /READ BACKWARDS
00604 602523 JMP RLKC2 /SETUP FOR BLOCK

/
.EJECT

```

```

/
/ NOW READ SAME FIELD FROM HI TO LO, THEN RECHECK
/ REST OF MEMORY.
/
00605 244340
00606 244377
00607 244275
00610 777700
00611 244277
00612 777774
00613 244342
00614 244275
00615 740101
00616 244274
00617 224275
00620 544274
00621 600624
00622 101377
00623 600626
00624 444342
00625 600617
00626 777777
00627 344275
00630 244275
00631 444277
00632 600612
00633 760300
00634 544320
00635 620562
00636 602525

/RAK2 LAC PATBNK /FIELD WITH PATTERN
XOR K7777
DAC MEMADR
LAW -10000
DAC CT4K
BAK2 LAW -4
DAC RPETE
LAC MEMADR
CMA
DAC PATR
LAC* MEMADR /READ
SAD PATR /COMPARE
JMP .+3
JMS ERROR /PRINT ERROR
JMP .+3
ISZ RPETE /READ EACH 4 TIMES
JMP RAK2+5
LAW -1
TAD MEMADR /ADDRESS MINUS 1
DAC MEMADR
ISZ CT4K /4K WHEN 0
JMP RAK2
LAW
SAD BLOC1 /NO BLOCK IF = LAW
JMP* RFWD2 /EXIT AND RECHECK REST OF MEMORY
JMP BLKD2 /SETUP NEXT BLOCK
.EJECT

```

```

/
/TEST 2A. WRITE SAME PATTERN IN SAME FIELD
/FROM HI TO LO.
/
00637 103070 TST2A JMS WRT1S /WRITE 1'S INTO ALL OF MEMORY
00640 204340 LAC PATBNK /FIELD WITH PATTERN
00641 244377 XOR K7777
00642 044275 DAC MEMADR
00643 770000 LAX -10000
00644 044277 DAC CT4K

/
00645 204275 BAK2A LAC MEMADR /ADDRESS
00646 740001 CMA /COMPLEMENT
00647 064275 DAC* MEMADR /WRITE INTO SAME
00650 777777 LAX -1
00651 344275 TAD MEMADR /ADDRESS MINUS 1
00652 044275 DAC MEMADR
00653 444277 ISZ CT4K /4K WHEN 0
00654 600645 JMP BAK2A
00655 760000 LAX
00656 544320 SAD RLOC1 /NO BLOCK IF = LAW
00657 741000 SKP
00660 602511 JMP BLKA2
00661 100511 JMS RST2 /SETUP TO CHECK REST OF MEMORY
00662 100562 JMS RFWD2 /READ LO TO HI; HI TO LO
00663 100511 JMS RST2 /RECHECK REST OF MEMORY AGAIN

/
/SETUP TO WRITE ADDRESS PATTERN IN NEXT
/SEQUENTIAL FIELD
/
00664 204340 NXPT2 LAC PATBNK /CURRENT TEST FIELD
00665 544301 SAD LAST1 /WAS IT THE LAST
00666 600277 JMP EXAM3 /YES. CHECK FOR TEST 3
00667 103070 JMS WRT1S /WRITE 1'S INTO ALL OF MEMORY
00670 204340 LAC PATBNK
00671 344400 TAD K10K /ADD 4K TO CURRENT FIELD
00672 044340 DAC PATBNK /NEW FIELD
00673 044275 DAC MEMADR
00674 744000 CLL
00675 101276 JMS CBANK
00676 600461 JMP LOP2+3
00677 600277 JMP EXAM3

/
.EJECT

```

```

/
/TEST 3. SLICE A 1 THRU 1 FIELD. REPEAT 18 TIMES
/PER FIELD TO CHECK EACH BIT POSITION. REST OF MEMORY
/WILL CONTAIN ALL 1'S. CHECK REST OF MEMORY AFTER
/EACH WRITE AND READ IN THE FIELD BEING TESTED.
/
00700 760263 TST3 LAW 263
00701 044345 DAC TNUM /TEST NUMBER
00702 103070 JMS WRT1S /WRITE 1'S INTO ALL OF MEMORY
00703 102733 JMS SET1 /SETUP FOR 1ST FIELD
00704 744000 CLL

/
00705 101276 LOP3 JMS CBANK /SEE IF IT HAS PROGRAM
00706 741000 SKP /NO
00707 600724 JMP CKB3 /ALL DONE
00710 204346 LAC K1 /K1 = 1
00711 044344 DAC BITN /C(BITN) = STARTING BIT POSITION
00712 204344 LAC BITN
00713 044274 DAC PATR
00714 744000 CLL
00715 204274 WLOP3 LAC PATR
00716 064275 DAC* MEMADR /WRITE THE BIT
00717 740010 RAL /NEXT POSITION
00720 044274 DAC PATR
00721 444275 ISZ MEMADR /ADDRESS + 1
00722 444277 ISZ CT4K /4K WHEN SKIP
00723 600716 JMP WLOP3+1 /WRITE IN NEXT
00724 760000 CKR3 LAW
00725 544320 SAD BLOC1
00726 741000 SKP
00727 602547 JMP BLKA3
00730 100734 JMS RST3 /SET UP TO READ REST OF MEMORY
00731 100757 JMS RFWD3 /READ FIELD LO TO HI
00732 100734 JMS RST3 /RECHECK REST OF MEMORY
00733 601007 JMP CK18B /SEE IF 18 POSITIONS YT
.EJECT

```

```

/
/SETUP TO READ REST OF MEMORY.
/
00734      000000
00735      204301
00736      544300
00737      600734
00740      204300
00741      044275
00742      770000
00743      044277
00744      204004
00745      044307
00746      204275
00747      544340
00750      600753
00751      100537
00752      600746
00753      544301
00754      620734
00755      344400
00756      600741

RST3      0
          LAC      LAST1
          SAD      FIRST1
RTN3      JMP*     RST3
          LAC      FIRST1
RST3A     DAC      MEMADR
          LAW      -10000
          DAC      CT4K
          LAC      KRTN3
          DAC      EXIT
          LAC      MEMADR
          SAD      PATBNK
          JMP      .+3
          JMS      REST2
          JMP      .-4
          SAD      LAST1
          JMP*     RST3
          TAD      K10K
          JMP      RST3A

          /LAST FIELD TO TEST
          /FIRST TO TEST
          /NONE TO TEST

          /-4K
          /4K COUNTER
          /KRTN3 = LOCATION RTN3

          /FIELD WITH PATTERN
          /READ REST OF MEMORY

          /NO MORE TO READ
          /ADD 4K TO CURRENT
          /TRY NEXT

/
.EJECT

```


/XAN05-TAPE 2
 /READ TEST 3 PATTERN FROM LO TO HI.

00757	000000	RFWD3	LAC	PATBNK	/FIELD WITH BIT PATTERN
00760	000340		LAC	MEMADR	
00761	044275		LAC	MEMADR	
00762	770000		LAC	-10000	
00763	044277		DAC	CT4K	/4K COUNTER
00764	004344		LAC	RITN	/CURRENT STARTING POSITION
00765	044274		DAC	PATR	
00766	744000		CLL		
00767	024275	FWD3	LAC*	MEMADR	/READ
00770	044274		SAD	PATR	/COMPARE
00771	741000		SKP		/OK
00772	101377		JMS	ERROR	/PRINT ERROR
00773	004274		LAC	PATR	
00774	741200		SNA		
00775	744002		STL		
00776	740010		RAL		/NEXT BIT POSITION
00777	044274		DAC	PATR	
01000	044275		ISZ	MEMADR	/ADDRESS + 1
01001	044277		ISZ	CT4K	/4K WHEN SKIP
01002	000767		JMP	FWD3	
01003	760000		LAC		
01004	044320		SAD	RLOC1	/NO CHECK IF = LAW
01005	000757		JMP*	RFWD3	/EXIT
01006	002561		JMP	RLKC3	
/					
01007	004344	CK18B	LAC	RITN	
01010	044405		SAD	K400K	/DONE IF = 400000
01011	001021		JMP	NXPT3	/SETUP FOR NEXT FIELD
01012	744010		RCL		
01013	044344		DAC	RITN	
01014	004340		LAC	PATBNK	/CURRENT FIELD
01015	044275		DAC	MEMADR	
01016	770000		LAC	-10000	
01017	044277		DAC	CT4K	
01020	000712		JMP	WLOP3-3	
/					
/SETUP NEXT FIELD WITH BIT PATTERN					
/					
01021	004340	NXPT3	LAC	PATBNK	
01022	044301		SAD	LAST1	
01023	000303		JMP	EXAM4	/ALL DONE. CHECK FOR TEST 4
01024	103070		JMS	WRT1S	/WRITE 1'S INTO ALL OF MEMORY
01025	004340		LAC	PATBNK	
01026	044400		TAD	K10K	/ADD 4K
01027	044340		DAC	PATBNK	/NEW FIELD
01030	044275		DAC	MEMADR	
01031	744000		CLL		
01032	101276		JMS	CBANK	
01033	000710		JMP	LOP3+3	
01034	000303		JMP	EXAM4	

.EJECT

```

/
/TEST 4. WRITE A PATTERN CONSISTING OF
/ALTERNATE WORDS OF 777777 AND 000000,
/FROM LO TO HI, AND HI TO LO. THE PATTERN IS
/READ THE SAME WAY. CHECK REST OF MEMORY
/AFTER EACH WRITE AND READ SEQUENCE, THEN
/COMPLEMENT THE PATTERN AND REPEAT.
/
TST4      JMS      WRT1S      /WRITE 1'S INTO ALL OF MEMORY
          LAW      264
          DAC      TNUM      /TEST NUMBER
          DZM      CNTRL
          JMS      SET1      /SETUP FOR FIRST FIELD
          CLL
LOP4      JMS      CBANK      /SEE IF IT HAS PROGRAM
          SKP
          JMP      CKB6      /ALL DONE
          LAC      CNTRL
WLOP4     DAC      PATR
          LAW      -4
          DAC      RPETE      /4 WRITES PER LOCATION
          LAC      PATR
          CMA      /777777 OR 000000
          DAC*     MEMADR     /WRITE
          ISZ      RPETE      /4 WRITES WHEN SKIP
          JMP      -2
          ISZ      MEMADR
          ISZ      CT4K      /4K WHEN SKIP
          JMP      WLOP4
CKB6      LAW      BLOC1     /NO BLOCK IF = LAW
          SAD
          SKP
          JMP      BLKA4
          JMS      RFWO4     /READ LO TO HI IN TESTED FIELD
          JMS      REST4     /READ REST OF MEMORY
/
/CHECK FOR COMPLEMENT PATTERN
/
          LAC      CNTRL
          SZA
          JMP      TST4A     /WRITE BACKWARDS
          CMA
          DAC      CNTRL
          LAC      PATBNK    /FIELD WITH PATTERN
          DAC      MEMADR
          LAW      -10000    /MINUS 4K
          DAC      CT4K
          JMP      LOP4+3    /WRITE COMPLEMENT
/
.EJECT

```

```

01035 103070
01036 760264
01037 044345
01040 444273
01041 102733
01042 744000
01043 101276
01044 741000
01045 601062
01046 204273
01047 444274
01050 777774
01051 044342
01052 204274
01053 740001
01054 764275
01055 444342
01056 601054
01057 444275
01060 444277
01061 601047
01062 760000
01063 544320
01064 741000
01065 602610
01066 101126
01067 101102

```

```

01070 204273
01071 740200
01072 601215
01073 740001
01074 044273
01075 204340
01076 044275
01077 770000
01100 044277
01101 601046

```

```

/SETUP TO READ REST OF MEMORY
/
01102 000000
01103 000001
01104 044300
01105 621102
01106 204300
01107 044275
01110 770000
01111 044277
01112 204305
01113 044307
01114 204275
01115 544340
01116 601121
01117 100537
01120 601114
01121 544301
01122 621102
01123 204340
01124 344400
01125 601107

REST4 0
LAC LAST1 /LAST TO TEST
SAD FIRST1 /ONLY 1 SELECTED IF EQUAL
RTN4 JMP* REST4 /NONE TO READ
LAC FIRST1 /FIRST TO TEST
REST4A DAC MEMADR
LAW -10000
DAC CT4K
LAC KRTN4 /KRTN4 = LOCATION RTN4
DAC EXIT
LAC MEMADR
SAD PATBNK /DOES FIELD HAVE PATTERN
JMP .+3 /YES
JMS REST2 /READ REST OF MEMORY
JMP .-4
SAD LAST1 /DOES IT = LAST TO TEST
JMP* REST4 /NO MORE TO READ
LAC PATBNK /CURRENT FIELD
TAD K10K /+ 10000
JMP REST4A

/
/READ THE FIELD WITH THE WORD PATTERN FROM LO TO
/HI, THEN HI TO LO.
/
01126 000000
01127 204340
01130 044275
01131 770000
01132 044277
01133 204273
01134 044274
01135 777774
01136 044342
01137 204274
01140 740001
01141 044274
01142 224275
01143 544274
01144 601147
01145 101377
01146 601151
01147 444342
01150 601142
01151 444275
01152 444277
01153 601135
01154 760000
01155 544320
01156 601160
01157 624307

RFWD4 0
LAC PATBNK /FIELD WITH PATTERN
DAC MEMADR
LAW -10000 /MINUS 4K
DAC CT4K
RBLK4 LAC CNTRL /CNTRL = 777777 OR 000000
DAC PATR
LAW -4
DAC RPETE /READ EACH LOCATION 4 TIMES
LAC PATR
CMA
DAC PATR
FWD4 LAC* MEMADR /READ
SAD PATR /COMPARE
JMP .+3 /OK
JMS ERROR /PRINT ERROR
JMP .+3
ISZ RPETE
JMP FWD4
ISZ MEMADR /ADDRESS + 1
ISZ CT4K /4K WHEN SKIP
JMP RBLK4+2 /READ ANOTHER
LAW
SAD RLOC1 /NO BLOCK IF = LAW
JMP RBAK4 /READ BACKWARD
JMP* EXIT /SETUP FOR NEXT BLOCK

/

```

.EJECT

```

/
/READ SAME FIELD WITH WORD PATTERN FROM
/HI TO LO, THEN RECHECK REST OF MEMORY.
/
01160 204340 PBAK4 LAC PATBNK /CURRENT TEST FIELD
01161 244377 XOR K7777
01162 044275 DAC MEMADR
01163 770000 LA* -10000 /MINUS 4K
01164 144277 DAC CT4K
01165 204273 LAC CNTRL /CNTRL = 777777 OR 000000
01166 740001 CMA
01167 044274 DAC PATR
01170 777770 BAK4 LA* -10 /READ EACH LOCATION 8 TIMES
01171 044342 DAC RPETE
01172 204274 LAC PATR
01173 740001 CMA
01174 044274 DAC PATR
01175 224275 LAC* MEMADR /READ
01176 544274 SAD PATR /COMPARE
01177 601202 JMP .+3 /OK
01200 101377 JMS ERROR /PRINT ERROR
01201 601204 JMP .+3
01202 444342 ISZ RPETE
01203 501175 JMP BAK4+5
01204 777777 LA* -1 /MINUS 1
01205 344275 TAD MEMADR /ADDRESS MINUS 1
01206 144275 DAC MEMADR
01207 444277 ISZ CT4K /4K WHEN SKIP
01210 601170 JMP BAK4 /READ ANOTHER
01211 760000 LA*
01212 544320 SAD BLOC1 /NO BLOCK IF = LA*
01213 521126 JMP* RWD4 /READ REST OF MEMORY
01214 624307 JMP* EXIT

```

```

/
.EJECT

```

```

/
/TEST 4A. WRITE IN SAME FIELD FROM HI TO LO.
/
/
01215 777777 TST4A LAW -1
01216 044273 DAC CNTRL
01217 044274 DAC PATR
01220 004340 LAC PATBNK /CURRENT FIELD WITH PATTERN
01221 044377 XOR K7777
01222 044275 DAC MEMADR
01223 770000 LAW -10000
01224 044277 DAC CT4K
01225 777760 LOP4A LAW -20
01226 044342 DAC RPFTE /WRITE EACH LOCATION 16 TIMES
01227 004274 LAC PATR
01230 740001 CMA
01231 044274 DAC PATR
01232 044275 DAC* MEMADR /WRITE
01233 444342 ISZ RPETE /16 WRITES WHEN SK P
01234 601232 JMP .-2
01235 777777 LAW -1
01236 044275 TAD MEMADR /ADDRESS - 1
01237 044275 DAC MEMADR
01240 444277 ISZ CT4K /4K WHEN SKIP
01241 601225 JMP LOP4A /WRITE ANOTHER
01242 760000 LAW
01243 544320 SAD BLOC1 /NO BLOCK IF = LAW
01244 741000 SKP
01245 602642 JMP RLKF4 /SETUP FOR NEXT BLOCK
01246 004273 LAC CNTRL /CNTRL = 777777 OR 000000
01247 740001 CMA
01250 044273 DAC CNTRL /COMPLEMENT FOR READ FORWARD
01251 101126 JMS RFWD4 /READ FORWARD
01252 101102 JMS REST4 /READ REST OF MEMORY
01253 004340 LAC PATBNK
01254 044275 DAC MEMADR
01255 004273 LAC CNTRL
01256 740200 SZA /CHECK FOR COMPLEMENT
01257 601262 JMP NXTP4
01260 044274 DAC PATR
01261 601220 JMP TST4A+3
/

```

```

.EJECT

```

```

/
/SETUP TO WRITE IN NEXT FIELD
/
01250 004340 NXP4 LAC PATBNK /CURRENT FIELD
01255 044301 SAE LAST1
01264 000307 JMP EXAM4+4 /SETUP TO RELOCATE
01265 003770 JMS WRT1S /WRITE 1'S INTO ALL OF MEMORY
01266 004340 LAC PATBNK
01267 344400 TAD K10K /PLUS 4K
01270 044340 DAC PATBNK
01271 044275 DAC MEMADR
01272 744000 CLL
01273 001276 JMS CBANK
01274 001046 JMP LOP4+3
01275 000307 JMP EXAM4+4

/
/ROUTINE TO SEE IF TESTED FIELD HAS PROGRAM
/
01276 000000 CBANK 0
01277 770000 LAW -10000
01300 044277 DAC CT4K
01301 001360 JMS WHERE /CURRENT PROGRAM FIELD
01302 044275 SAE MEMADR /NEXT TO TEST
01303 001317 JMP CBNK-4
01304 021276 JMP* CBANK /EXIT
NOMOR ISZ CBANK /RETURN +1
01305 041276 JMP* CBANK
01306 021276 LAC MEMADR
01307 004275 TAD K10K /CURRENT +4K
01310 344400 DAC MEMADR
01311 044275 SNL
01312 740400 DAC PATBNK
01313 044340 LAW -10000 /-4K
01314 770000 DAC CT4K /4K COUNTER
01315 044277 JMP CBANK+1 /EXIT AND TEST NEW FIELD
01316 001277

/
01317 004275 LAC MEMADR
01320 344400 TAD K10K
01321 044275 DAC MEMADR
01322 001344 JMS NXTBNK
.EJECT

```

```

01323 000000 CBNK 0
01324 000000 LAW -10000
01325 044277 DAC CT4K
01326 044275 TAD MEMADR
01327 044301 SA LAST1
01330 024307 JMP* EXIT
01331 001360 JMS WHERE /SEE WHERE PROGRAM IS
01332 044275 SAD MEMADR /NEXT TO TEST
01333 041000 SKP /ADD 10K
01334 021323 JMP* CBNK /READ ALL 1'S
01335 044301 SAD LAST1 /SEE IF LAST
01336 024307 JMP* EXIT /NO MORE TO READ
01337 044400 TAD K10K
01340 044275 DAC MEMADR /NEW FIELD
01341 044340 SAD PATBNK
01342 001335 JMP .-5
01343 021323 JMP* CBNK /EXIT
/
/Routine TO CHECK FOR LAST FIELD
/
01344 000000 NXTBNK 0
01345 0750004 LAS /CHECK ACS0 FOR HALT
01346 041100 SPA
01347 001364 JMS HALT /GO HALT
01350 0770000 LAW -10000 /-4K
01351 044275 TAD MEMADR /SUBTRACT 4K
01352 044301 SAD LAST1 /ALL DONE IF EQUAL
01353 001305 JMP NOMOR
01354 001344 LAC NXTBNK
01355 040010 RAL /RESTORE LINK
01356 004275 LAC MEMADR /NEXT FIELD
01357 001312 JMP NOMOR+5
/
/
/Routine TO DETERMINE WHERE PROGRAM IS
/
01360 000000 WHERE 0 /CONTAINS EPC
01361 001360 LAC .-1
01362 004403 AND K70K /CLEAR ALL BUT BITS 3,4,5
01363 021360 JMP* WHERE /EXIT
/
/
/HALE ROUTINE. PRESS CONTINUE TO RESUME
/TESTING, OR IF ACS CHANGES, TO EXECUTE
/NEW PARAMETERS
/
01364 000000 HALT 0
01365 040040 HLT
01366 0750004 LAS
01367 040010 RAL
01370 041100 SPA
01371 021364 JMP* HALT
01372 040020 RAH
01373 004406 AND K177
01374 044272 SAD MCWA

```

PAGE 22

XAF15

01375 21364
01376 21251

JMP# HALT
JMP STOVER
.EJECT


```

/
/ERROR PRINT-OUT ROUTINE. PLACE ACS0 UP FOR
/HALE AFTER PRINT-OUT. PRESS CONTINUE
/
ERROR 0
01377 000000
01400 044302 DAC RAD1 /SAVE BAD DATA
01401 204275 LAC MEMADR
01402 044303 DAC OCADR /SAVE FAILING ADDRESS
01403 204274 LAC PATR
01404 044304 DAC GOOD1
01405 204325 LAC FRWRD /ERROR TABLE POINTER
01406 544044 SAD FNERR /LAST ADDRESS OF TABLE
01407 741000 SKP
01410 601414 JMP .+4
01411 204043 LAC ERTBL /FIRST ADDRESS OF TABLE
01412 044325 DAC ERWRD /PUT POINTER TO TOP OF TABLE
01413 601423 JMP SW2 /CHECK AC2 FOR BELL
01414 204303 LAC OCADR /FAILING ADDRESS
01415 504403 AND K70K /MASK 3,4 AND 5
01416 544336 SAD LAST /NEW ERROR FIELD IF SKIP
01417 601423 JMP .+4 /SAME FIELD AS LAST ERROR
01420 044336 DAC LAST
01421 064325 DAC* ERWRD /STORE FIELD# IN TABLE
01422 444325 ISZ ERWRD /INCREMENT POINTER

/
SW2 LAW /PRINT INHIBIT IF = LAW
01423 760000
01424 544267 SAD NOPRNT
01425 601443 JMP SW0 /NO PRINT
01426 750004 LAS
01427 742010 RTL
01430 740100 SMA /BELL IF SKIP
01431 601435 JMP SW1 /CHECK ACS 1
01432 760207 LAW 207 /ASCII BELL
01433 101571 JMS PCHAR /PRINT
01434 601443 JMP SW0
01435 750004 SW1 LAS
01436 740010 RAL
01437 740100 SMA /NO SKIP = PRINT INFO
01440 601450 JMP DOERR /PRINT
01441 101364 JMS HALT /HALT
01442 601450 JMP DOERR /PRINT INFO
01443 750004 SW0 LAS
01444 741100 SPA /NO SKIP = HALT
01445 101364 JMS HALT
01446 707702 FEM
01447 621377 JMP* ERROR /RETURN TO READ ROUTINE
.EJECT

```

```

/
/SETUP TO PRINT ERROR
/
01451 762000 DOFRR LAW SUPTRL /IF 1ST LOC. OF SUPTRL = LAW,
01451 564013 SADR# SUPTRL /NO ADR. SUPPRESSION WANTED

01452 601455 JMP .+3
01453 101551 JMS CSUP /SEE IF THIS ERROR IS SUPPRESSED
01454 601443 JMP SW0 /YES.
01455 103737 JMS CRLF /CR,LF
01456 203746 LAC PHDR
01457 741200 SNA
01460 103746 JMS PHDR
01461 204345 LAC TNUM /TEST NUMBER
01462 101571 JMS PCHAR
01463 777767 LAW -11 /-9
01464 043737 STER DAC CRLF /SPACING COUNT
01465 101615 JMS SPING /SPACE 9
01466 204303 LAC OCADR
01467 043737 DAC CRLF
01470 101576 JMS PROCTL /PRINT FAILING ADDRESS
01471 777772 LAW -6
01472 043737 DAC CRLF
01473 101615 JMS SPING /SPACE 6
01474 204304 LAC GOOD1 /WHAT DATA SHOULD BE
01475 043737 DAC CRLF
01476 101576 JMS PROCTL /PRINT THE GOOD
01477 777776 LAW -2
01500 043737 DAC CRLF
01501 101615 JMS SPING /SPACE 2
01502 204302 LAC RAD1 /DATA READ
01503 043737 DAC CRLF
01504 101576 JMS PROCTL /PRINT THE BAD
01505 777773 LAW -5
01506 043737 DAC CRLF
01507 101615 INDY JMS SPING /SPACE 5
01510 760261 LAW 261
01511 544345 SADR TNUM
01512 741000 SKP /PRINT "ALL"
01513 601520 JMP .+5
01514 204006 LAC LAL
01515 044305 DAC PRNT
01516 103701 JMS PNXT
01517 601527 JMP CMAX
01520 204340 LAC PATBNK
01521 744010 RCL; RTL; /FIELD WITH PAT.
01522 742010 RTL;
01523 742010
01524 742010

```

.EJECT

01525	344362	TAF	K260	
01526	121571	JMS	PCHAR	
01527	444324	CMAX	ISZ	MAXERR
01530	601443	JMP	SW0	
01531	777700	LAW	-100	
01532	044324	DAC	MAXERR	
01533	760000	LAW		
01534	044267	DAC	NOPRNT	
01535	103737	JMS	CRLF	
01536	204007	LAC	PTO	
01537	044305	DAC	PRNT	
01540	103701	JMS	PNXT	/PRINT-OUTS INHIBITED
01541	103737	JMS	CRLF	
01542	777766	LAW	-12	
01543	043737	DAC	CRLF	
01544	760212	LAW	212	
01545	101571	JMS	PCHAR	/10 LINE FEEDS
01546	443737	ISZ	CRLF	
01547	601544	JMP	.-3	
01550	601443	JMP	SW0	
/				
/CHECK FOR SUPPRESSED ADDRESS				
/				
01551	000000	CSUP	0	
01552	224013	LAC*	SUPTBL	
01553	544303	SAD	OCADR	/COMPARE WITH CURRENT ADDRESS
01554	601564	JMP	TOP	/SUPPRESSED
01555	544337	SAD	LSTSUP	/SEE IF DONE WITH LIST
01556	601567	JMP	SPEXT	/YES
01557	204013	LAC	SUPTBL	
01560	544014	SAD	ESTBL	/SEE IF DONE WITH TABLE
01561	601567	JMP	SPEXT	/YES
01562	444013	ISZ	SUPTBL	/POINTER +1
01563	601552	JMP	CSUP+1	
01564	204012	TOP	LAC	STBL
01565	044013	DAC	SUPTBL	/RESTORE POINTER
01566	621551	JMP*	CSUP	/EXIT
01567	441551	SPEXT	ISZ	CSUP.
01570	601564	JMP	TOP	/EXIT ADDRESS +1
/				
01571	000000	PCHAR	0	
01572	700406	TLS		
01573	700401	TSF		
01574	601573	JMP	.-1	
01575	621571	JMP*	PCHAR	
/				
.EJECT				

```

/
/PRINT 6 DIGIT OCTAL NUMBERS
/
01576 000000 PROCTL 0
01577 777772          LAW      -6
01600 44342          DAC      RPETE
01601 203737          LAC      CRLF
01602 744010 POSN    RCL;    RTL
01603 742010          DAC      CRLF
01604 043737          RAL
01605 740010          AND      K7
01606 504350          XOR      K260
01607 244362          JMS     PCHAR
01610 101571          ISZ     RPETE
01611 444342          SKP
01612 741000          JMP*   PROCTL
01613 621576          JMP     POSN-1
01614 601601

/
/SPACE ROUTINE
/
01615 000000 SPING 0
01616 760240          LAW      240
01617 101571          JMS     PCHAR
01620 443737          ISZ     CRLF
01621 601617          JMP     .-2
01622 621615          JMP*   SPING

/
.EJECT

```

```

/MASK AC 15-17
/MAKE ASCII
/PRINT 1
/6 DIGITS WHEN SKIP
/EXIT
/POSITION NEXT NUMBER

```

```

/PRINT A SPACE
/DONE WHEN SKIP
/EXIT

```

```

/XAD15-TAPE 3
/
/ROUTINES TO ACCEPT KEYBOARD INPUT FOR TEST SELECTION;
/ADDRESS SUPPRESSION AND BLOCK TEST LIMITS. PLACE ACS 13 DOWN
/BEFORE RE-INITIATING MAIN PROGRAM.
/
01623 707762 KYBRD EPA
01624 144344 DZM BITN /TEMP. STORAGE FOR INPUT CHARS.
/
/TYPE "TEST#" AND WAIT FOR INPUT
/
01625 204015 TSTNO LAC TSNX
01626 244305 DAC PRNT
01627 103737 JMS CRLF /CR,LF
01630 103701 JMS PNXT /PRINT "TEST#"
01631 102272 JMS KEYIN /GO WAIT FOR INPUT
01632 144344 DAC BITN /SAVE TTY CHAR.
01633 544374 SAD K377 /IS INPUT A RUBOUT
01634 601623 JMP KYBRD /YES. START OVER
01635 544360 SAD K215 /NO TEST WANTED IF A C.R.
01636 601660 JMP SUPIN /LAST TEST PATTERN WILL BE USED
01637 777517 LAK -261
01640 344344 TAL BITN
01641 740100 SMA /IF AC IS NEG., TEST # IS <1
01642 601645 JMP .+3 /IT IS >1
01643 103547 JMS WOTIS /PRINT QUESTION MARK
01644 601623 JMP KYBRD /START OVER
01645 204344 LAC BITN
01646 740001 CMA
01647 344346 TAD K1 /2'S COMPLEMENT TEST #
01650 344366 TAD K264
01651 740100 SMA /IF AC IS NEG., TEST # IS >4
01652 601655 JMP .+3
01653 103547 JMS WOTIS /THERE ARN'T MORE THAN 4 TESTS
01654 601623 JMP KYBRD /START OVER
01655 204344 TSTN LAC BITN
01656 044345 DAC TNUM
01657 601660 JMP SUPIN /WAIT FOR C.R.
/
.EJECT

```

```

/DONE WITH TEST#.  NOW DO ADR. SUPPRESSION
/
01660  204012  SUPIN  LAC  STRL  /1ST LOCATION IN SUPPRESS TABLE
01661  44013   DAC  SUPTBL /POINTER
01662  204011  LAC  ROTB
01663  044010  DAC  ROTA
01664  224010  LAC* ROTA
01665  044311  DAC  NROTA  /SHIFT COUNTER
01666  760000  LAW
01667  064013  DAC*  SUPTBL  /A LAW IN 1ST LOC. SAYS NO
                                /ADDRESS TO BE SUPPRESSED
                                /USED TO STORE 15 BIT ADDRESS
01670  144306  DZM  ADRCW
01671  103737  JMS  CRLF
01672  204016  LAC  SUPSX
01673  044305  DAC  PRNT
01674  103701  JMS  PNXT  /PRINT "SUPPRESS"
/
/ACCEPT 1ST ADDRESS AND THEN WAIT FOR A COLON
/OR A COMMA
/
01675  204017  NXSUP LAC  INSUP  /C(INSUP) = SUPIN
01676  044310  DAC  OVER
01677  204020  LAC  SUPDN  /C(SUPDN) = DNSUP
01700  044307  DAC  EXIT
01701  102272  JMS  KEYIN  /WAIT FOR INPUT
01702  102301  JMS  LEGAL  /CHECK VALIDITY
01703  224010  LAC*  ROTA
01704  044311  DAC  NROTA  /C(NROTA) = COUNT FOR LEFT SHIFTS
01705  204344  LAC  BITN
01706  504350  AND  K7  /MASK 15-17
01707  044344  DAC  BITN  /SAVE
01710  103043  JMS  GENADR /START ASSEMBLING 1ST ADDRESS
01711  601675  JMP  NXSUP  /GET NEXT INPUT
/
/RETURN HERE FROM GENADR AFTER 5 CHARS. REC'D.
/
01712  204306  LAC  ADRCW  /FIRST ADDRESS
01713  064013  DAC* SUPTBL /STORE IN 1ST LOC. OF TABLE
01714  444013  ISZ  SUPTBL
01715  144306  DZM  ADRCW  /CLEAR
01716  102272  JMS  KEYIN  /WAIT FOR A: , OR C.R.
01717  544374  SAD  K377  /CHECK FOR RUBOUT
01720  601623  JMP  KYBRD /START OVER WITH TEST #
01721  544360  SAD  K215  /CHECK FOR C.R.
01722  602027  JMP  DNSUP
01723  544361  SAD  K254  /CHECK FOR COMMA
01724  601761  JMP  SUPBLK /A COMMA = SUPPRESS A BLOCK
01725  544370  CKCLN SAD  K272  /CHECK FOR COLON
01726  741000  SKP
01727  602312  JMP  QUERY  /NONE OF THE ABOVE.
/
.EJECT

```

```

/ACCEPT INDIVIDUAL ADDRESSFS
/
SUP1      JMS      KEYIN      /WAIT FOR INPUT
          JMS      LEGAL      /CHECK VALIDITY
          LAC*     ROTA       /COUNTS LEFT SHIFTS MADE
          LAC      NROTA
          LAC      RITN
          AND      K7         /MASK ACS 15-17
          DAC      RITN
          JMS      GENADR     /ASSEMBLE ADDRESS
          JMP      SUP1       /WAIT FOR NEXT CHAR.
          LAC      ADRCW     /COMPLETE ADDRESS
          DAC*     SPTBL      /STORE IN SUPPRESSION TABLE
          LAC      LSTSUP     /LSTSUP = LAST TO SUPPRESS
          LAC      SPTBL
          SAD      ESTBL      /CHECK FOR 257 ADDRESSES
          JMP      DNSUP     /WAIT FOR C.R.
          ISZ     SPTBL      /INCREMENT POINTER
          DZM     ADRCW
          JMS      KEYIN     /WAIT FOR COLON INPUT
          SAD     K215       /DONE IF C.R.
          JMP     DNSUP
          JMP     CKCLN      /IS IT REALLY A COLON

          /
          JMS     KEYIN     /WAIT FOR C.R.
          SAD     K215
          JMP     DNSUP
          JMP     QUERY     /NOT A C.R.

          .EJECT

```

```

/
/ACCEPT TWO INPUTS FOR A SUPPRESSED BLOCK
/
01761 102272 SUPBLK JMS KEYIN /WAIT FOR INPUT
01762 102301 JMS LEGAL /CHECK VALIDITY
01763 224010 LAC* ROTA
01764 044311 DAC NROTA /COUNT LEFT SHIFTS
01765 204344 LAC RITN
01766 504350 AND K7 /MASK ACS 15-17
01767 044344 DAC RITN
01770 103043 JMS GENADR /ASSEMBLE ADDRESS
01771 601761 JMP SUPBLK /WAIT FOR NEXT

/
01772 204306 LAC ADRCW /COMPLETE ADDRESS
01773 044337 DAC LSTSUP /SAVE
01774 204012 LAC STBL
01775 044013 DAC SUPTBL /SETUP TABLE POINTER
01776 224013 LAC* SUPTBL
01777 740001 CMA
02000 344346 TAD K1 /2'S COMP. 1ST ADDRESS
02001 344337 TAD LSTSUP /SUBTRACT 2ND ADDRESS
02002 740100 SMA /1ST IS > LAST IF SKIP
02003 602012 JMP SETSUP-3
02004 224013 LAC* SUPTBL /REVERSE THE TWO ADDRESSES
02005 043737 DAC CRLF /SAVE FIRST
02006 204337 LAC LSTSUP
02007 064013 DAC* SUPTBL /LAST IS NOW FIRST
02010 203737 LAC CRLF
02011 044337 DAC LSTSUP /FIRST IS NOW LAST
02012 224013 LAC* SUPTBL
02013 544337 SAD LSTSUP /ARE THEY EQUAL
02014 602027 JMP DNSUP /YES
02015 204013 SETSUP LAC SUPTBL
02016 544014 SAD ESTBL
02017 602027 JMP DNSUP /256 CHARS STORED. IGNORE ANY MORE
02020 224013 LAC* SUPTBL
02021 344346 TAD K1
02022 444013 ISZ SUPTBL /INCREMENT POINTER
02023 064013 DAC* SUPTBL
02024 544337 SAD LSTSUP /WAS LAST THE LAST TO SUPPRESS
02025 602027 JMP DNSUP /YES
02026 602015 JMP SETSUP
.EJECT

```



```

/
/RESTORE POINTERS BEFORE ENTERING NEXT LINE
/
02027 234 12      DNSUP  LAC      STRL
02030 044013     DAC      SUPTRL
02031 204011     LAC      ROTB      /ROTB AND ROTA MUST BE EQUAL
02032 544010     SAD      ROTA
02033 642044     JMP      RLK1
02034 044010     DAC      ROTA      /RESTORE SHIFT COUNT POINTER
02035 103547     JMS      WOTIS      /LAST ADDRESS WAS <5 CHARS
02036 601660     JMP      SUPIN      /START OVER WITH SUPPRESS
/
02037 777763     ROTC   LAW -15     /ROTATE 12 LEFT FOR 1ST DIGIT
02040 777766     LAW -12     /9 LEFT FOR 2ND
02041 777771     LAW -7      /6 LEFT FOR 3RD
02042 777774     LAW -4      /3 LEFT FOR 4TH
02043 777777     LAW -1      /NONE FOR 5TH
/
.EJECT

```

```

/
/INPUT ROUTINE FOR ADDRESS LIMITS OF BLOCK #1.
/PRESS CR IF NO BLOCKS WANTED.
/
02044 144306 BLK1 DZM ADRCW
02045 760000 LAW
02046 144320 DAC BLOC1
02047 144321 DAC BLOC2
02050 202335 LAC STLOOP-1
02051 142320 DAC LOCAT+4
02052 224010 LAC* ROTA
02053 144311 DAC NROTA /SHIFT COUNTER
02054 103737 JMS CRLF

/
/TYPE "BLOCK#1" AND WAIT FOR INPUT
/
02055 204025 BLKN1 LAC BLKSX
02056 144305 DAC PRNT
02057 103701 JMS PNXT /PRINT BLOCK #1
02060 204023 LAC DONE2 /C(DONE2) = DBLK1
02061 144307 DAC EXIT
02062 204021 LAC NBLK /C(NBLK) = BLK1
02063 144310 DAC OVER
02064 102272 BLK1A JMS KEYIN /WAIT FOR INPUT
02065 102301 JMS LEGAL /CHECK FOR LEGAL CHAR.
02066 204344 LAC RITN /INPUT CHAR.
02067 504350 AND K7 /MASK AC 15-17
02070 144344 DAC BITN
02071 103043 JMS GENADR /ASSEMBLE ADDRESS
02072 602064 JMP BLK1A

/
/ENTER HERE AFTER FIRST ADDRESS RECEIVED
/
02073 204306 LAC ADRCW /ASSEMBLED ADDRESS
02074 144320 DAC BLOC1 /FIRST ADR. OF FIRST BLOCK
02075 144306 DZM ADRCW /CLEAR
02076 102272 JMS KEYIN /WAIT FOR COMMA
02077 544361 SAD K254 /CHECK FOR COMMA
02100 741000 SKP /O.K.
02101 602312 JMP QUERY /PRINT QUERY MARK
02102 102272 BLK1B JMS KEYIN /WAIT FOR INPUT
02103 102301 JMS LEGAL /SEE IF IT'S LEGAL
02104 204344 LAC RITN /INPUT CHAR.
02105 504350 AND K7 /MASK AC 15-17
02106 144344 DAC BITN
02107 103043 JMS GENADR /ASSEMBLE SECOND ADR.
02110 602102 JMP BLK1B

.EJECT

```

```

/
/ENTER HERE AFTER SECOND ADDRESS
/
02111 204306 LAC ADCRW /ASSEMBLED ADDRESS
02112 044321 DAC RLOC2 /LAST ADR. OF 1ST BLOCK
02113 144306 DEM ADCRW

/
DBLK1 02114 204011 LAC ROTB
02115 544010 SAD ROTA /MUST BE EQUAL
02116 602121 JMP .+3 /O.K.
02117 044010 DAC ROTA
02120 602312 JMP QUERY /ADDRESS IS NOT 5 CHARS.
02121 760000 IAW /LAW = NO INPUT
02122 544320 SAD RLOC1
02123 602163 JMP RLK2
02124 544321 SAD RLOC2 /CHECK FOR 2 INPUTS
02125 602312 JMP QUERY /PRINT QUERY AND START OVER
02126 204320 LAC RLOC1
02127 504403 AND K70K
02130 044344 DAC BITN /TEMP. SAVE ADR. BITS 3,4 AND 5
02131 204321 LAC RLOC2
02132 504403 AND K70K
02133 043737 DAC CRLF /DO SAME WITH UPPER LIMIT
02134 102151 JMS PROG /SEE IF LIMITS OVERLAP PROGRAM
02135 602163 JMP RLK2 /O.K.
02136 103737 JMS CRLF /LIMITS IN SAME 4K AS PROGRAM
02137 204025 LAC RLKSX
02140 044305 DAC PRNT
02141 103701 JMS PNXT /PRINT BLOCK #1
02142 204026 LAC OVRLP
02143 044305 DAC PRNT
02144 103701 JMS PNXT /PRINT BLOCK #1
02145 204407 LAC K700K
02146 042320 DAC LOCAT+4 /CHANGE JMP* TO NOP
02147 102314 JMS LOCAT /PRINT AREA OF PROGRAM
02150 602044 JMP RLK1 /START OVER

/
PROG 02151 000000 0
02152 101360 JMS WHERE
02153 544344 SAD BITN /CHECK IF SAME AS LOW LIMIT
02154 741000 SKP /YES. ERROR
02155 602160 JMP .+3 /NO
02156 442151 ISZ PROG
02157 622151 JMP* PROG
02160 543737 SAD CRLF /SEE IF SAME AS HIGH LIMIT
02161 442151 ISZ PROG /YES
02162 622151 JMP* PROG /EXIT

/
.EJECT

```

```

/
/INPUT ROUTINE FOR BLOCK #2
/
02163 144306 PLK2 DZM ADRCW
02164 760000 LAR
02165 044322 DAC BLOC3
02166 044323 DAC BLOC4
02167 103737 JMS CRLF
/
/TYPE BLOCK #2 AND WAIT FOR INPUT
/
02170 204027 BLKN2 LAC BLKTX
02171 044305 DAC PRNT
02172 103701 JMS PNXT /PRINT BLOCK #2
02173 204024 LAC DONE3 /C(DONE3)=DBLK2
02174 044307 DAC EXIT
02175 204022 LAC TBLK /C(TBLK) = BLK2
02176 044310 DAC OVER
02177 102272 BLK2A JMS KEYIN /WAIT FOR INPUT
02200 102301 JMS LEGAL /CHECK FOR LEGAL CHAR.
02201 204344 LAC BITN
02202 504350 AND K7 /MASK AC 15-17
02203 044344 DAC BITN
02204 103043 JMS GENADR /ASSEMBLE ADDRESS
02205 602177 JMP BLK2A
/
/ENTER HERE AFTER FIRST ADDRESS
/
02206 204306 LAC ADRCW /ASSEMBLED ADDRESS
02207 044322 DAC BLOC3 /1ST ADR. OF 2ND BLOCK
/
02210 144306 DZM ADRCW
02211 102272 JMS KEYIN /WAIT FOR COMMA
02212 544361 SAD K254
02213 741000 SKP
02214 602312 JMP QUERY /PRINT QUERY
02215 102272 BLK2B JMS KEYIN /WAIT FOR INPUT
02216 102301 JMS LEGAL /SEE IF IT'S LEGAL
02217 204344 LAC BITN
02220 504350 AND K7
02221 044344 DAC BITN
02222 103043 JMS GENADR
02223 602215 JMP BLK2B
/
/RETURN HERE AFTER 2ND ADDRESS
/
02224 204306 LAC ADRCW /ASSEMBLED ADDRESS
02225 044323 DAC BLOC4 /LAST ADR. OF 2ND BLOCK
02226 144306 DZM ADRCW
02227 102272 JMS KEYIN /WAIT FOR C.R.
02230 544360 SAD K215
02231 741000 SKP /DONE
02232 602312 JMP QUERY /PRINT QUERY
/
.EJECT

```

```

/
OBLK2
02235 044011 LAC ROTB
02236 044010 SAD ROTA /MUST BE EQUAL
02236 02240 JMP .+3 /O.K.
02236 044010 DAC ROTA
02237 042312 JMP QUERY /ADR. IS NOT 5 CHARS.
02240 760000 LAW
02241 044322 SAD BLOC3 /LAW = NO INPUT
02242 042336 JMP STLOOP
02243 044323 SAD BLOC4 /CHECK FOR 2 INPUTS
02244 042232 JMP DBLK2-1 /START OVER
02245 044322 LAC BLOC3
02246 044403 AND K70K
02247 044344 DAC RITN /TEMP. SAVE MA 3, 4 AND 5
02250 044323 LAC BLOC4
02251 044403 AND K70K
02252 043737 DAC CRLF /TEMP. SAVE HIGH LIMIT
02253 102151 JMS PROG /SEE IF LIMITS OVERLAP PROGRAM
02254 042336 JMP STLOOP /SETUP TO START LOOPS
02255 103737 JMS CRLF /LIMITS IN SAME 4K AS PROGRAM
02256 044027 LAC BLKTX
02257 044305 DAC PRNT
02260 103701 JMS PNXT /PRINT BLOCK #2
02261 044026 LAC OVRLP
02262 044305 DAC PRNT
02263 103701 JMS PNXT /PRINT LIMITS OVERLAP PROGRAM
02264 044407 LAC K700K
02265 042320 DAC LOCAT+4 /CHANGE JMP* TO NOP
02266 102314 JMS LOCAT /PRINT AREA OF PROGRAM
02267 042335 LAC STLOOP-1
02270 042320 DAC LOCAT+4 /RESTORE THE JMP*
02271 042163 JMP BLK2 /START OVER
.EJECT

```

```

/
/CHARACTER INPUT ROUTINE
/
02270 000000 KEYIN 0
02273 700312 KRR
02274 700301 KSF /INITIALIZE
02275 602274 JMP -1 /WAIT
02276 700312 KRR
02277 044344 DAC BITN /TEMP. SAVE
02300 622272 JMP* KEYIN /EXIT

/
/SEE IF CHARACTER IS LEGAL
/
02301 000000 LEGAL 0
02302 204344 LAC BITN /TTY CHAR.
02303 544374 SAD K377 /CHECK FOR RUBOUT
02304 601623 JMP KYBRD /START OVER WITH TEST #
02305 544360 SAD K215 /CHECK FOR C.R.
02306 624307 JMP* EXIT /TERMINATE
02307 504373 AND K370
02310 544362 SAD K260
02311 622301 JMP* LEGAL /EXIT
02312 103547 QUERY JMS WOTIS /PRINT QUESTION MARK
02313 624310 JMP* OVER /START PRESENT LINE OVER

/
.EJECT

```

```

/PRINT AREA CONTAINING PROGRAM
/
02314 100000 LOCAT 0
02315 000004 LAC
02316 004355 AND K100
02317 040200 SZA
02320 022314 JMP* LOCAT
02321 003737 JMS CRLF /CR, LF
02322 004051 LAC PISIN
02323 044305 DAC PRNT
02324 003701 JMS PNXT /PRINT "PROGRAM IS NI FIELD"
02325 001360 JMS WHERE /WHERE IS IT
02326 044010 RCL; RTL; RTL; RTL
02327 042010
02330 042010
02331 042010
02332 044362 TAD K260 /ASCII FIELD NUMBER
02333 001571 JMS PCHAR
02334 003737 JMS CRLF /CR, LF
02335 022314 JMP* LOCAT /EXIT
/
/SETUP ROUTINES FOR TESTING BLOCKS OF CORE
/
02336 003737 STLOOP JMS CRLF
02337 060000 LAW /SEE IF ANY BLOCKS SELECTED
02340 044320 SAD BLOC1
02341 041000 SKP /LAW=NONE SELECTED
02342 002346 JMP SETU1
02343 044322 SAD BLOC3
02344 000251 JMP STOVER /NO BLOCKS SELECTED
02345 002377 JMP SETU2
/
02346 004345 SETU1 LAC TNUM /SEE IF A TEST WANTED
02347 041200 SNA
02350 060261 LAW 261 /NO, USE TEST 1
02351 044345 DAC TNUM
/
02352 004320 LAC BLOC1
02353 040001 CMA
02354 044346 TAD K1 /2'S COMPLEMENT UPPER LIMIT
02355 044321 TAD BLOC2
02356 041100 SPA /REVERSE IF NEG.
02357 002426 JMP REVR1
02360 004320 SINGL LAC BLOC1
02361 004403 AND K70K
02362 044340 DAC PATBNK
02363 004345 LAC TNUM /DETERMINE PATTERN
02364 004374 AND K377
02365 044363 SAD K261
02366 002435 JMP SIMU1 /TEST 1 PATTERN
02367 044364 SAD K262
02370 002506 JMP SIMU2 /TEST 2 PATTERN
02371 044365 SAD K263
02372 002544 JMP SIMU3 /TEST 3 PATTERN
02373 044366 SAD K264

```

PAGE 38

XAD15

02374 602604
02375 204345
02376 740040

JMP
LAC
HLT
.EJECT

SIMU4
TNUM

/TEST 4 PATTERN
/SHOULD NEVER GET HERE


```

/
02377 204345 /SETU2 LAC TNUM
02400 741200 SNA
02401 760261 LAW 261
02402 044345 DAC TNUM
02403 204322 LAC BLOC3
02404 740001 CMA
02405 344346 TAD K1
02406 344323 TAD BLOC4
02407 741100 SPA /LAST ADR. IS >FIRST IF 0
02410 602421 JMP /REVERSE CONTENTS
02411 204322 LAC BLOC3 /1ST TO TEST
02412 044320 DAC BLOC1 /LAST TO TEST
02413 204323 LAC BLOC4 /A LAW = NO BLOCK SELECTED
02414 044321 DAC BLOC2
02415 760000 LAW
02416 044322 DAC BLOC3
02417 044323 DAC BLOC4
02420 602360 JMP SINGL

/
02421 204323 /REVR2 LAC BLOC4
02422 044320 DAC BLOC1
02423 204322 LAC BLOC3
02424 044321 DAC BLOC2
02425 602415 JMP REVR2-4

/
02426 204321 /REVR1 LAC BLOC2
02427 044344 DAC BITN
02430 204320 LAC BLOC1
02431 044320 DAC BLOC1
02432 204344 LAC BITN
02433 044320 DAC BLOC1
02434 602360 JMP SINGL

/
.EJECT

```

```

/
/SETUP FOR TEST 1
/
02435 103170 SIMU1 JMS WRT1S /PUT 1'S INTO ALL OF CORE
02436 102775 JMS SETB1 /SETUP 1ST BLOCK
02437 600333 JMP WBLK1 /WRITE PATTERN IN BLOCK1
/
/RETURN HERE AFTER WRITING BLOCK 1
/
02440 777777 BLKA1 LAW -1
02441 344275 TAD MEMADR /LAST ADDRESS
02442 544323 SAD BLOC4 /SEE IF 2 BLOCKS WRITTEN
02443 602452 JMP T1RDF /SETUP TO READ FORWARD
02444 760000 LAW
02445 544322 SAD BLOC3
02446 602452 JMP T1RDF
02447 102704 JMS CKADR
02450 103006 JMS SETB2 /SETUP 2ND BLOCK
02451 600333 JMP WBLK1 /WRITE PATTERN IN BLOCK 2
/
/SETUP TO READ FORWARD
/
02452 102775 T1RDF JMS SETB1 /SETUP 1ST BLOCK
02453 600367 JMP RLOP1 /READ BLOCK 1 LO TO HI
/
/RETURN HERE AFTER READING A BLOCK LO TO HI
/
02454 777777 RLKB1 LAW -1
02455 344275 TAD MEMADR /LAST ADDRESS
02456 544323 SAD BLOC4 /DONE BOTH BLOCKS IF EQUAL
02457 602465 JMP T1RDA /READ 2 BLOCKS HI TO LO
02460 760000 LAW
02461 544322 SAD BLOC3 /SEE IF 2 SELECTED
02462 602467 JMP T1RDB /NO. READ 1 HI TO LO
02463 103006 JMS SETB2 /SETUP 2ND BLOCK
02464 600367 JMP RLOP1 /READ BLOCK 2 LO TO HI
/
02465 103031 T1RDA JMS STB2 /SETUP BLOCK 2 FOR HI TO LO
02466 600424 JMP BAK1 /READ BLOCK 2 HI TO LO
/
02467 103017 T1RDB JMS STB1 /SETUP BLOCK 1 FOR HI TO LO
02470 600424 JMP BAK1 /READ BLOCK 1 HI TO LO
/
/RETURN HERE AFTER READING A BLOCK HI TO LO
/
02471 444275 BLKC1 ISZ MEMADR
02472 740000 NOP
02473 204275 LAC MEMADR /LAST ADDRESS
02474 544320 SAD BLOC1 /ALL DONE IF EQUAL
02475 741000 SKP
02476 602467 JMP T1RDB /SETUP BLOCK 1
02477 750004 LAS
02500 741100 SPA /CHECK ACS 0 FOR ALT
02501 101364 JMS HALT /GO TO ALT
02502 504352 ANI K20

```

02013 740200
02014 601623
02015 602435

SZA
JMP
JMP
.EJECT

KYBRD
SIMU1

/START OVER
/KEEP LOOPING

```

/
/SETUP FOR TEST 2
/
02510 103170 SIMU2 JMS WRT1S /WRITE 1'S INTO ALL OF CORE
02517 102775 JMS SETB1 /SETUP FOR BLOCK 1
02516 600461 JMP LOP2+3 /WRITE BLOCK 1
/
02511 760000 BLKA2 LAW /SEE IF 2 BLOCKS SELECTED
02512 544322 SAD BLK3 /ONLY ONE SELECTED
02513 602521 JMP BLKB2
02514 102704 JMS CKADR
02515 103006 JMS SETB2 /SETUP 2ND BLOCK
02516 204030 LAC DONE5 /DONE5 = LOCATION BLKB2
02517 044307 DAC EXIT
02520 600541 JMP REST2+2 /READ REST OF MEMORY
/
02521 102775 BLKB2 JMS SETB1 /SETUP BLOCK 1
02522 600567 JMP FWD2 /READ BLOCK 1 LO TO HI
/
02523 103017 BLKC2 JMS STR1 /SETUP BLOCK 1
02524 600612 JMP BAK2 /READ BLOCK 1 HI TO LO
/
02525 760000 BLKD2 LAW /SEE IF 2 BLOCKS SELECTED
02526 544322 SAD BLK3 /ONLY ONE SELECTED
02527 602534 JMP BLKE2
02530 103006 JMS SETB2
02531 204031 LAC DONE6 /DONE6 = LOCATION BLKE2
02532 044307 DAC EXIT
02533 600541 JMP REST2+2 /READ REST OF MEMORY
/
02534 103017 BLKE2 JMS STR1 /RESTORE LIMITS
02535 750004 LAS
02536 741100 SPA /CHECK ACS0 FOR HALT
02537 101364 JMS HALT /GO HALT
02540 504352 AND K20
02541 740200 SZA
02542 601623 JMP KYBRD
02543 600645 JMP BAK2A /WRITE BLOCK 1 HI TO LO
.EJECT

```

```

/
/SETUP FOR TEST 3
/
02544 103070 SIMU3 JMS WRT1S /WRITE 1S INTO ALL OF CORE
02545 102775 JMS SETB1 /SETUP FOR BLOCK1
02546 600710 JMP LOP3+3 /WRITE BLOCK1
/
02547 204032 RLKA3 LAC DONE7 /DONE7 = LOCATION BLKB3
02550 204307 DAC EXIT
02551 760000 LAW
02552 544322 SAD RLOC3 /SEE IF 2 BLOCKS SELECTED
02553 602557 JMP RLKB3 /NO
02554 102704 JMS CKADR
02555 103006 JMS SETB2 /SETUP FOR BLOCK 2
02556 600541 JMP REST2+2 /READ BLOCK 2
/
02557 102775 RLKB3 JMS SETB1 /SETUP FOR BLOCK 1
02560 600764 JMP FWD3-3 /READ BLOCK 1 AGAIN
/
02561 204033 RLKC3 LAC DONE8 /DONE8 = LOCATION BLKD3
02562 204307 DAC EXIT
02563 760000 LAW
02564 544322 SAD BLOC3 /SEE IF 2 BLOCKS SELECTED
02565 602567 JMP RLKD3 /NO
02566 602555 JMP RLKB3-2
/
02567 750004 RLKD3 LAS
02570 741100 SPA /CHECK ACS 0 FOR HALT
02571 101364 JMS HALT /GO HALT
02572 504352 AND K20
02573 740200 SZA
02574 601623 JMP KYBRD
02575 204344 LAC BITN
02576 544405 SAD K400K /DONE 18 BITS IF EQUAL
02577 602544 JMP SIMU3 /START OVER
02600 744010 RCL
02601 204344 DAC /NEXT STARTING POSITION
02602 102775 JMS SETB1 /SETUP FOR BLOCK 1
02603 600712 JMP LOP3+5 /WRITE BLOCK 1
.EJECT

```

```

/
/SETUP FOR TEST 1
/
02609 103170 SIMU4 JMS WRT1S /WRITE IS INTO ALL OF CORE
02610 102775 JMS SETB1
02611 144273 DCH CNTRL
02612 601046 JAP LOP4+3 /WRITE BLOCK 1 LO TO HI
/
02610 204034 BLKA4 LAC DONE9 /DONE9 = LOCATION BLKB4
02611 044307 DAC EXIT
02612 102775 JMS SETB1
02613 601133 JMP RBLK4 /READ BLOCK 1 LO TO HI
/
02614 204035 BLKB4 LAC DON10 /DON10 = LOCATION BLKC4
02615 044307 DAC EXIT
02616 103017 JMS STR1 /SETUP FOR BLOCK 1
02617 601165 JMP BAK4-3 /READ BLOCK 1 HI TO LO
/
02620 204036 BLKC4 LAC DON11 /DON11=LOCATION RLKD4
02621 044307 DAC EXIT
02622 760000 LAW
02623 544322 SADR RLOC3 /SEE IF 2 BLOCKS SELECTED
02624 602630 JMP RLKD4 /NO
02625 102704 JMS CKADR
02626 103006 JMS SETB2 /SETUP FOR BLOCK 2
02627 600541 JMP REST2+2 /READ BLOCK 2
/
.EJECT

```

		/			
02630	741273	BLKD4	LAC	CNTRL	
02631	741270		SZA		
02632	602637		JMP	BLKE4-1	/WRITE HI TO LO
02633	741201		CMA		
02634	744273		DAC	CNTRL	
02635	102775		JMS	STB1	/SETUP FOR BLOCK 1
02636	601146		JMP	LOP4+3	/WRITE COMPLEMENT LO TO HI
		/			
02637	744274		DAC	PATR	
02640	103017	BLKE4	JMS	STB1	/SETUP BLOCK 1
02641	601224		JMP	LOP4A-1	/WRITE BLOCK 1 HI TO LO
		/			
02642	204037	BLKF4	LAC	DON12	/DON12 = LOCATION BLKG4
02643	744307		DAC	EXIT	
02644	760000		LAW		
02645	544322		SAD	BL0C3	/SEE IF 2 BLOCKS SELECTED
02646	602650		JMP	BLKG4	/NO
02647	604033		JMP	DON11-3	/READ BLOCK 2
		/			
02650	204040	BLKG4	LAC	DON13	/DON13 = LOCATION BLKH4
02651	044307		DAC	EXIT	
02652	102775		JMS	STB1	/SETUP BLOCK 1
02653	204273		LAC	CNTRL	
02654	741201		CMA		
02655	601134		JMP	RBLK4+1	/READ BLOCK 1 LO TO HI
		/			
02656	204041	BLKH4	LAC	DON14	/DON14 = LOCATION BLKJ4
02657	044307		DAC	EXIT	
02660	204273		LAC	CNTRL	
02661	744274		DAC	PATR	
02662	103017		JMS	STB1	
02663	601170		JMP	BAK4	
		/			
02664	204042	BLKJ4	LAC	DON15	/DON15 = =LOCATION BLKL4
02665	044307		DAC	EXIT	
02666	760000		LAW		
02667	544322		SAD	BL0C3	/SEE IF 2 BLOCKS SELECTED
02670	602672		JMP	BLKL4	/NO
02671	604033		JMP	DON11-3	/READ BLOCK 2
		/			
02672	204273	BLKL4	LAC	CNTRL	
02673	741201		SNA!CMA		
02674	602637		JMP	BLKE4-1	/WRITE COMPLEMENT HI TO LO
02675	750004		LAS		
02676	741100		SPA		/CHECK ACS 0 FOR HALT
02677	101364		JMS	HALT	/GO HALT
02700	504352		AND	K20	
02701	740200		SZA		
02702	601623		JMP	KYRRD	
02703	602604		JMP	SIMU4	
		/			
					.EJECT

02744	200100	CKADR	0		
02745	204322		LAC	RLOC3	
02746	740001		CMA		
02747	344346		TAD	K1	
02710	344323		TAD	BLOC4	
02711	740100		SMA		/1ST IS >2ND IF SKIP
02712	622704		JMP*	CKADR	/EXIT
02713	204322		LAC	RLOC3	
02714	043737		DAC	CRLF	
02715	204323		LAC	BLOC4	
02716	044322		DAC	BLOC3	/REVERSE BLOC3 AND BLOC4
02717	203737		LAC	CRLF	
02720	044323		DAC	RLOC4	
02721	622704		JMP*	CKADR	/EXIT
/					
02722	000000	SETBAK	0		
02723	204301		LAC	LAST1	/LAST TO TEST
02724	044340		DAC	PATBNK	
02725	044276		DAC	SVADR	
02726	244377		XOR	K7777	
02727	044275		DAC	MEMADR	
02730	770000		LAW	-10000	
02731	044277		DAC	CT4K	
02732	622722		JMP*	SETBAK	
/					
02733	000000	SET1	0		
02734	204300		LAC	FIRST1	
02735	044275		DAC	MEMADR	
02736	044340		DAC	PATBNK	
02737	770000		LAW	-10000	
02740	044277		DAC	CT4K	
02741	622733		JMP*	SET1	
/					
02742	000000	CKBAK	0		
02743	204275		LAC	MEMADR	/LAST ADDRESS
02744	504403		AND	K70K	/MASK BITS 3,4,5
02745	043737		DAC	CRLF	/SAVE
02746	101360		JMS	WHERE	/SEE WHERE PROGRAM IS
02747	543737		SAD	CRLF	/NEXT HAS PROGRAM IF EQUAL
02750	741000		SKP		
02751	622742		JMP*	CKBAK	/EXIT
02752	102765		JMS	NXBAK	/SEE IF CURRENT IS LAST
02753	602756		JMP	+.3	/SUBTRACT 4K
02754	442742	NONE	ISZ	CKBAK	/RETURN +1
02755	622742		JMP*	CKBAK	
02756	770000		LAW	-10000	
02757	044277		DAC	CT4K	
02760	344276		TAD	SVADR	
02761	044276		DAC	SVADR	/NEXT FIELD
02762	244377		XOR	K7777	
02763	044275		DAC	MEMADR	
02764	602743		JMP	CKBAK+1	/CHECK NEW FIELD

.EJECT


```

/
/CHECK FOR LAST FIELD
/
02765 000000 NXRAK 0
02766 750004 LAS
02767 741100 SPA /CHECK ACS 0 FOR HALT
02770 101364 JMS HALT
02771 274276 LAC SVADR
02772 544300 SAD FIRST1
02773 612754 JMP NONE /NO MORE TO TEST
02774 602756 JMP NONE+2 /SETUP FOR NEXT

/
/SETUP ADDRESSES FOR 1ST BLOCK LO TO HI
/
02775 000000 SETB1 0
02776 204320 LAC BLOC1 /1ST ADDRESS
02777 044275 DAC MEMADR
03000 740001 CMA
03001 344346 TAD K1 /2'S COMPLEMENT
03002 344321 TAD BLOC2 /SUBTRACT BLOC2
03003 740001 CMA /COMPLEMENT RESULT
03004 044277 DAC CT4K
03005 622775 JMP* SETB1 /EXIT

/
/SETUP ADDRESSES FOR 2ND BLOCK LO TO HI
/
03006 000000 SETB2 0
03007 204322 LAC BLOC3 /1ST ADDRESS
03010 044275 DAC MEMADR
03011 740001 CMA
03012 344346 TAD K1 /2'S COMPLEMENT
03013 344323 TAD BLOC4 /SUBTRACT
03014 740001 CMA
03015 044277 DAC CT4K
03016 623006 JMP* SETB2 /EXIT

.EJECT

```

```

/
/SETUP BLOCK 1 ADDRESSES FOR HI TO LO
/
03017 000000
03020 204321
03021 044275
03022 204320
03023 740001
03024 344346
03025 344321
03026 740001
03027 044277
03030 623017

/SETUP BLOCK 2 ADDRESSES FOR HI TO LO
/
03031 000000
03032 204323
03033 044275
03034 204322
03035 740001
03036 344346
03037 344323
03040 740001
03041 044277
03042 623031

STR1 0
LAC RLOC2 /LAST ADDRESS
DAC MEMADR
LAC RLOC1
CMA
TAD K1 /2'S COMPLEMENT
TAD RLOC2 /SUBTRACT
CMA
DAC CT4K
JMP* STR1

STR2 0
LAC RLOC4 /LAST ADDRESS
DAC MEMADR
LAC RLOC3
CMA
TAD K1 /2'S COMPLEMENT
TAD RLOC4 /SUBTRACT
CMA
DAC CT4K
JMP* STR2

/
.EJECT

```

```

/
/GENERATE BINARY ADDRESS FROM KEYBOARD INPUT
/
03043  000000
03044  024010
03045  044311
03046  024344
03047  444311
03050  603066
03051  244306
03052  044306
03053  777777
03054  564010
03055  603060
03056  444010
03057  623043
03060  204011
03061  044010
03062  424010
03063  044311
03064  443043
03065  623043

03066  744010
03067  603047

/
/ROUTINE TO WRITE 1'S INTO ALL OF MEMORY
/
03070  000000
03071  244300
03072  044275
03073  044276
03074  770000
03075  044277
03076  744002
03077  101276
03100  741000
03101  623070
03102  777777
03103  064275
03104  444275
03105  444277
03106  603103
03107  744002
03110  101344

GENADR  0
        XCT*  ROTA          /XCT A LAW MINUS X.
        DAC   NROTA
        LAC   RITN         /INPUT NUMBER
CNROT   ISZ   NROTA         /SHIFT COUNT+1
        JMP   GOLEFT       /ROTATE 1 LEFT
        XOR   ADRCW        /XOR WITH PARTIAL ADDRESS
        DAC   ADRCW
        LAW   -1
        SADR* ROTA         /DONE 5 DIGITS IF EQUAL
        JMP   .+3
        ISZ   ROTA         /LAW POINTER+1
        JMP*  GENADR       /GO WAIT FOR NEXT
        LAC   ROTB
        DAC   ROTA         /RESTORE LAW POINTER
        XCT*  ROTA         /GET THE LAWS
        DAC   NROTA
        ISZ   GENADR       /RETURN+1
        JMP*  GENADR       /EXIT

GOLEFT  RCL
        JMP   CNROT

/
WRT1S   0
        LAC   FIRST1
        DAC   MEMADR
        DAC   SVADR
        LAW   -10000
        DAC   CT4K
        JMS   CBANK        /SEE IF IT HAS PROGRAM
        SKP
        JMP*  WRT1S        /NO
        /EXIT
RITE    LAW   -1          /AC=777777
        DAC*  MEMADR       /WRITE
        ISZ   MEMADR       /ADDRESS+1
        ISZ   CT4K         /DONE 4K WHEN ZERO
        JMP   .-3
        STL
        JMS   NXTBNK       /SETUP FOR NEXT
        .EJECT

```

XAD15 - TAPE 4
ROUTINE TO DETERMINE FIELD FOR RELOCATION

03111	044343	MOVE	LAC	ERTBL	
03112	044325		DAC	ERWRD	
03113	044301		LAC	LAST1	/LAST TO TEST
03114	544300		SAD	FIRST1	/DON'T MOVE IF EQUAL
03115	600221		JMP	RTN1	/RETURN
03116	044270		LAC	FLAGS	/PROGRAM FLAGS
03117	741100		SPA		/FORCED MOVE MADE IF A 1.
03120	600221		JMP	RTN1	/DON'T MOVE
03121	740020		RAR		/LINK = BIT 17
03122	741400		SZL		/FIRST MOVE IF SKIP
03123	603210		JMP	NXTMV	/SETUP FOR NEXT MOVE
03124	444270		ISZ	FLAGS	/SET FLAG FOR 1ST MOVE
03125	044301		LAC	LAST1	/LAS=% NEWCURRENT FIELD
03126	044271		DAC	INSFLD	
03127	770000		LAW	-10000	/-4K
03130	344271		TAD	INSFLD	/SUBTRACT 4K FROM CURRENT
03131	044312		DAC	NXLOC	/NXLOC = DEST'N FOR NEXT TIME.
03132	101360		JMS	WHERE	/WHERE ARE WE NOW
03133	544271		SAD	INSFLD	/ALREADY IN LAST 1 IF EQUAL
03134	603173		JMP	SUB1	/TRY NEXT LOWER

.EJECT

```

/
/NOV CHECK FOR ERROR RECORDED IN NEW FIELD
/
03135  204420  CKERR  LAW
03136  544325  SAC*   ERWRD  /NO ERRORS IF = LAW
03137  603150  JMP    STMV  /INITIALIZE MOVE
03140  224325  LAC*   ERWRD
03141  544271  SAD    INSFLD /ERROR IN FIELD IF EQUAL
03142  603162  JMP    EQUAL
03143  444325  ISZ    ERWRD  /POINTER + 1
03144  204325  LAC    ERWRD
03145  544044  SAD    ENERR  /END OF TABLE IF EQUAL
03146  741000  SKP
03147  603140  JMP    CKERR+3
/
03150  204043  STMV   LAC    ERTBL
03151  044325  DAC    ERWRD  /RESTORE POINTER
03152  204271  LAC    INSFLD /NEW FIELD
03153  044314  DAC    DESTN
03154  111360  JMS    WHERE
03155  044313  DAC    SOURCE
03156  544314  SAD    DESTN
03157  600221  JMP    RTN1  /NEW AND CURRENT ARE EQUAL
03160  204314  LAC    DESTN
03161  603325  JMP    MOVE  /MOVE PROGRAM
/
/ERROR IN NEW FIELD. TRY NEXT LOWER
/
03162  544300  EQUAL  SAD    FIRST1 /DON'T TRY NEXT IF EQUAL
03163  603205  JMP    DNMVE
03164  741200  SNA
03165  603171  JMP    .+4  /IS IT FIELD 0
03166  770000  LAW    -10000 /YES
03167  344271  TAD    INSFLD /-4K
03170  044312  DAC    NXLOC  /SUBTRACT 4K FROM NEW FIELD
03171  204043  LAC    ERTBL  /NEXT NEW FIELD
03172  044325  DAC    ERWRD  /RESTORE POINTER
/
03173  204312  SUB1   LAC    NXLOC  /NEXT NEW FIELD
03174  544271  SAD    INSFLD /IS IT = CURRENT NEW FIELD
03175  603162  JMP    EQUAL /TRY NEXT LOWER
03176  044271  DAC    INSFLD /NEW NEW FIELD
03177  544300  SAD    FIRST1 /DOES IT = LOWEST FIELD
03200  603135  JMP    CKERR  /CHECK FOR ERROR
03201  770000  LAW    -10000
03202  344271  TAD    INSFLD /SUBTRACT 4K
03203  044312  DAC    NXLOC  /NEW FIELD FOR NEXT PASS
03204  603135  JMP    CKERR
/
03205  204043  DNMVE  LAC    ERTBL
03206  044325  DAC    ERWRD  /RESTORE POINTER
03207  600221  JMP    RTN1  /START OVER
/
.EJECT

```

```

/ROUTINE TO DETERMINE PROGRAM DEST'N AFTER MAKING ONE MOVE
/
03210 004360 NXTMV JMS WHERE /WHERE IS PROGRAM NOW
03211 044313 DAC SOURCE
03212 700000 CKNXT LAX
03213 564325 SAG* ERWRD /NO ERRORS IF 1ST = LAW
03214 603227 JMP STNXT
03215 204043 LAC ERTBL
03216 044325 DAC ERWRD
03217 224325 LAC* ERWRD /GET AN ERROR ADDRESS
03220 544312 SAG NXLOC
03221 603247 JMP SUR2 /ERROR IN NEXT FIELD TRY NEXT
03222 444325 ISZ ERWRD
03223 204325 LAC ERWRD
03224 544044 SAG ENERR
03225 741000 SKP
03226 603217 JMP CKNXT+5

/
03227 204043 STNXT LAC ERTBL
03230 044325 DAC ERWRD /RESTORE POINTER
03231 204312 LAC NXLOC /NEW FIELD
03232 544271 SAG INSFLD /DOES IT = CURRENT FIELD
03233 603236 JMP .+3
03234 544300 SAG FIRST1 /DOES IT = LOWEST FIELD
03235 603264 JMP MVBK /YES, CLEAR FLAGS AND MOVE
03236 544300 SAG FIRST1 /DOES THE CURRENT ALSO=
/ THE LOWEST FIELD.
/ YES, SETUP FOR HIGHEST FIELD
03237 603260 JMP NXTHI /NEW CURRENT FIELD
03240 044271 DAC INSFLD /-4K
03241 770000 LAW -10000
03242 344271 TAD INSFLD
03243 044312 DAC NXLOC /NEW NEXT FIELD
03244 204271 LAC INSFLD
03245 044314 DAC DESTN
03246 603325 JMP MOVE /MOVE FROM HERE TO C (DESTN)

/
03247 204312 SUB2 LAC NXLOC
03250 544300 SAG FIRST1 /IS NEXT = FIELD 0 OR 1ST TO TEST
03251 603205 JMP ONMVE /YES, DON'T MOVE
03252 770000 LAW -10000 /-4K
03253 344312 TAD NXLOC /NEW NEXT FIELD
03254 044312 DAC NXLOC
03255 544271 SAG INSFLD /DOES IT = CURRENT FIELD
03256 603250 JMP SUB2+1 /YES
03257 603215 JMP CKNXT+3 /SEE IF ERROR IN NEW FIELD

```

.EJECT

03260	044301	NXTHI	LAC	LAST1	/LAST TO TEST
03261	044303		AND	K70K	
03262	044312		DAC	NXLOC	/LAST = NEXT FIELD
03263	044315		JMP	CKNXT+3	/CHECK FOR ERROR
		/			
03264	101360	MVBK	JMS	WHERE	
03265	044313		DAC	SOURCE	
03266	044312		LAC	NXLOC	
03267	044271		DAC	INSFLD	
03270	044314		DAC	DESTN	
03271	144270		DZM	FLAGS	
03272	044325		JMP	MOVE	
			.EJECT		

```

/
/ROUTINE TO FORCE MOVE THE PROGRAM. DESTINATION
/FIELD# MUST BE TYPED IN BY THE OPERATOR (0-7 OCTAL).
/
03273 044405 FCDMV LAC K400K
03274 740001 CMA
03275 544270 AND FLAGS
03276 244405 XOR K400K /SET BIT 0 FOR FCDMV FLAG
03277 244270 DAC FLAGS
03300 204043 LAC ERTBL
03301 044325 DAC ERWRD /RESTORE TABLE POINTER
03302 103426 JMS GOTO /PRINT GO TO FIELD
/
/CHECK FOR ERROR IN NEW FIELD
/
03303 760000 CKFCD LAW
03304 564325 SAD* ERWRD /NO ERRORS IF 1ST = LAW
03305 603321 JMP WHWAY /SEE WHERE TO GO
03306 224325 LAC* ERWRD
03307 544314 SAD DESTN /DOES ERROR = NEW FIELD
03310 603316 JMP XPRT /YES, PRINT MESSAGE
03311 444325 ISZ ERWRD /POINTER+1
03312 204325 LAC ERWRD
03313 544044 SAD ENERR /SEE IF END OF TABLE
03314 603317 JMP .+3 /DONE AND NO ERRORS
03315 603306 JMP CKFCD+3
03316 103404 XPRT JMS PRSEL /PRINT ERROR IN SELECTED 4K
/
03317 204043 LAC ERTBL
03320 044325 DAC ERWRD
03321 204314 WHWAY LAC DESTN /NEW FIELD
03322 544313 SAD SOURCE /DOES IT EQUAL PRESENT
03323 600221 JMP RTN1
03324 044271 DAC INSFLD /NEW CURRENT FIELD
/
.EJECT

```



```

/
ROUTINE TO RELOCATE THE PROGRAM
/
03320 044341 MOVE    DEM    LOCER
03321 044340     LAX    -10000    /-4K
03322 044277     DAC    CT4K      /4K COUNTER
03331 044313     LAC    SOURCE    /CURRENT FIELD
03332 044316     DAC    MOVES
03333 044314     LAC    DESTN     /NEW FIELD
03334 044315     DAC    MOVED
03335 044316     LAC*   MOVES     /MOVE FROM CURRENT
03336 044345     DAC    TNUM      /SAVE
03338 044313     JMS    RT19L
03337 044315     DAC*   MOVED     /PUT IN NEW FIELD
03340 024315     LAC*   MOVED     /READ BACK
03341 044316     SAD*   MOVES     /COMPARE
03342 044300     SKP
03343 044310     JMS    MVERR     /PRINT ERROR INFO
03344 044316     ISZ    MOVES     /INCREMENT ADDRESSES
03345 044315     ISZ    MOVED
03346 044377     ISZ    CT4K
03347 044300     SKP
03350 044376     JMP
03351 044345     LAC    TNUM
03352 044302     SAD    DLMT      /DELIMITING CHARACTER
03353 044300     SKP      /ADJUST INDIRECTS
03354 044334     JMP    MOSOM
03355 024316     LAC*   MOVES
03356 044064     SAD    DLMTA     /DONE INDIRECTS IF EQUAL
03357 044335     JMP    MOSOM+1
03360 044377     AND    K7777    /MASK ADDRESS BITS
03361 044314     XOR    DESTN    /PUT FIELD NUMBER ON IT
03362 044313     JMS    RT19L
03363 044315     DAC*   MOVED     /PUT IN NEW FIELD
03364 024315     LAC*   MOVED     /READ BACK
03365 044377     AND    K7777
03366 044313     XOR    SOURCE
03367 044316     SAD*   MOVES     /COMPARE
03370 044300     SKP      /OK
03371 044310     JMS    MVERR     /PRINT ERROR INFO
03372 044316     ISZ    MOVES     /INCREMENT ADDRESSES
03373 044315     ISZ    MOVED
03374 044277     ISZ    CT4K
03375 044335     JMP    AJIN
          .EJECT

```

33674	444365
33677	444317
33401	444377
33401	444314
33402	444317
33403	444317

/
DIND

JMS	FNOT
LAF	RGNLO
AND	47777
XOR	DESTN
DAC	RGNLO
JMP*	RGNLO

.EJECT

/WAS TRANSFER MADE OK
 /C(RGNLO) = LOC RTN1
 /MASK ADDRESS

/EXIT FROM HERE TO LOC
 /RTN1 IN NEW FIELD

```

/
/PRINT ERROR IN SELECTED 4K
/
ERSEL      0
           000000      CRLF      /CR,LF
           1443737     LAC       /TEXT POINTER
           2004252     DAC       PRNT
           344305      JMS       /PRINT
           403701      JMS       CRLF
           503737      JMS       CRLF
           603273      JMP       FCDMV      /WAIT FOR ANOTHER CHOICE
/
/ROTATE INSTRUCTION 19 LEFT BEFORE MOVING
/
RT19L      0
           000000      CLL
           744000      DAC       /LINK = 0
           844345      DAC       /SAVE
           944345      LAK      -11  /-9 DECIMAL
           044344      DAC       RITN  /SHIFT COUNT
           144345      LAC       TNUM  /INSTRUCTION
           240010      RAL
           342010      RTL
           444344      ISZ       RITN
           503422      JMP       .-2
           623413      JMP*     RT19L
/
           .EJECT

```

KEYBOARD ROUTINE FOR FORCED RELOCATION

```

/
03426 504040 GOTO 0
03427 750004 LAS /READ ACS
03430 504353 AND K40
03431 741200 SNA /CHECK BIT 12
03432 603444 JMP NOSW /EQUALS 0
03433 103737 JMS CRLF /CR,LF
03434 204045 LAC PTWLV /TEXT POINTER
03435 044305 DAC PRNT
03436 103701 JMS PNXT /PRINT PUT ACS 12 ON A 0
03437 750004 LAS
03440 504353 AND K40
03441 740200 SZA /WAIT FOR THE 0
03442 603437 JMP .-3
03443 103737 JMS CRLF /CR,LF X 2
03444 103737 NOSW JMS CRLF
03445 204046 LAC GOFL /TEXT POINTER
03446 044305 DAC PRNT
03447 103701 JMS PNXT /PRINT GO TO FIELD -
03450 102272 JMS KEYIN /WAIT FOR INPUT
03451 544360 SAD K215 /A CR = NO FORCED MOVE
/AND RESUME AUTO RELOCATE
03452 603503 JMP CFLG /CLEAR THE FORCED MOVE FLAG
/
03453 740020 RAR /NOT = RO OR CR, SO IT
/MUST BE A NUMBER
03454 742020 RTR; RTR; RTR
03455 742020
03456 742020
03457 504403 AND K70K /MASK 3,4 AND 5
03460 044314 DAC DESTN /NEW FIELD
03461 101360 JMS WHERE /WHERE ARE WE NOW
03462 044313 DAC SOURCE /CURRENT FIELD
03463 103737 JMS CRLF /CR, LF
03464 623426 JMP* GOTO /CHECK FOR ERROR
/
.EJECT

```

03460	111000	FNOT			
03461	203540		LAC	LOCER	
03462	741200		SJA		/NO ERRORS IF 0
03470	603465		JMP*	FNOT	/ENTER NEW FIELD
/					
03471	143540		DEX	LOCER	
03472	103737		JMS	CRLF	/CR,LF
03473	204050		LAC	NERN	/TEXT POINTER
03474	044305		DAC	PRNT	
03475	103701		JMS	PNXT	/PRINT NO MORE ERRORS
03476	103737		JMS	CRLF	/CR,LF
03477	204270		LAC	FLAGS	
03500	741100		SPA		/ACS 0 A 1 = FORCED MOVE
03501	603273		JMP	FCDMV	/WAIT FOR ANOTHER CHOICE
03502	603227		JMP	STNXT	/TRY NEXT FIELD LOWER
/					
03503	204405	CFLG	LAC	K400K	
03504	740001		CMA		
03505	504270		AND	FLAGS	/CLEAR THE FORCED MOVE FLAG
03506	044270		DAC	FLAGS	
03507	600221		JMP	RTN1	/START OVER
/					
.EJECT					

```

/
03510 000000 MVERR 0
03511 044302 DAC RAD1 /SAVE INCORRECT INSTRUCTION
03512 204315 DAC MOVED /FIELD AND ADDRESS
03513 044303 DAC QCADR /SAVE
03514 224316 LAC* MOVES /CORRECT INSTRUCTION
03515 044304 DAC GOOD1 /SAVE
03516 203746 LAC PHDR
03517 741200 SNA
03520 103746 JMS PHDR
03521 203540 LAC LOCER
03522 741200 SNA /DON'T PRINT IF 1
03523 103540 JMS LOCER /PRINT PROGRAM RELOCATION ERROR
03524 203531 LAC JMP1 /JMP RETURN
03525 041507 DAC INDY
03526 103737 JMS CRLF
03527 77766 LAW -12 /-10 DECIMAL
03530 601464 JMP STER /PRINT INFO
03531 603532 JMP1 JMP .+1

03532 201465 LAC STER+1 /EQUALS JMS SPING
03533 041507 DAC INDY
03534 750004 LAS
03535 741100 SPA
03536 101364 JMS HALT
03537 623510 JMP* MVERR /EXIT

/
LOCER 0
03540 000000
03541 103737 JMS CRLF /CR,LF
03542 204047 LAC RELOC /TEXT POINTER
03543 044305 DAC PRNT
03544 103701 JMS PNXT /PRINT PROGRAM RELOCATION ERROR
03545 103737 JMS CRLF /CR,LF X 2
03546 623540 JMP* LOCER /EXIT AND PRINT THE ERROR

/
WOTIS 0
03547 000000
03550 103737 JMS CRLF /CR,LF
03551 760277 LAW 277 /QUERY MARK
03552 101571 JMS PCHAR /PRINT
03553 103737 JMS CRLF /CR,LF
03554 623547 JMP* WOTIS /EXIT

.EJECT

```

```

/
/ROUTINE TO ACCEPT TEST LIMITS FROM KEYBOARD INPUT
/
SLMTS
03554 042020
03556 204335
03557 042320
03560 103737
03561 204053
03562 044305
03563 103701
03564 103737
03565 204054
03566 044310
03567 204055
03570 044307
03571 102272
03572 544374
03573 603556
03574 102301
03575 204344
03576 504350
03577 744020
03600 742020
03601 742020
03602 742020
03603 044300
03604 102272
03605 544361
03606 741000
03607 602312
03610 102272
03611 544374
03612 603556
03613 102301
03614 204344
03615 504350
03616 744020
03617 742020
03620 742020
03621 742020
03622 044301
03623 777777
03624 044277
03625 444277
03626 602312
03627 204300
03630 740001
03631 344346
03632 344301
03633 740100
03634 603643

DAC STLOOP-1
DAC LOCAT+4
JMS CRLF /CR,LF
LAC TLMX /TEST LIMITS POINTER
DAC PRNT
JMS PNXT /PRINT "TEST LIMITS"
JMS CRLF /CR,LF
LAC SLMX /C (SLMX)=SLMTS+1
DAC DON3 /RETURN ADDRESS=CREVR
DAC EXIT
JMS KEYIN /WAIT FOR INPUT
SAD K377
JMP SLMTS+1
JMS LEGAL /SEE IF VALID
LAC BITN /ASCII INPUT
AND K7 /MASK 15,16 AND 17
RCR; RTR; RTR; RTR

DAC FIRST1 /FIRST TO TEST
JMS KEYIN /WAIT FOR COMMA
SAD K254
SKP
JMP QUERY /PRINT QUERY, AND RESTART
JMS KEYIN /WAIT FOR LAST
SAD K377
JMP SLMTS+1
JMS LEGAL /SEE IF VALID
LAC BITN /ASCII INPUT
AND K7
RCR; RTR; RTR; RTR

DAC LAST1 /LAST DO TEST
LAW -1
DAC CT4K
ISZ CT4K /NO 2ND DIGIT IF NO SKIP
JMP QUERY /PRINT QUERY AND RESTART
LAC FIRST1 /FIRST FIELD
CMA
TAD K1 /2'S COMPLEMENT
TAD LAST1 /FIRST IS >LAST IF NEG.
SMA
JMP OKAS /FIRST IS LOWEST ORDER
/

```

.EJECT

```

03635 004300 LAC FIRST1
03636 044344 DAC RITN /SAVE
03637 044301 LAC LAST1
0364 044300 DAC FIRST1 /LAST IS NOW FIRST
03641 044344 LAC RITN
03642 044301 DAC LAST1 /FIRST IS NOW LAST
03643 004301 OKAS LAC LAST1
03644 044300 SAD FIRST1 /SEE IF ONLY 1 SELECTED
03645 741000 SKP /YES, SEE IF IT HAS PROGRAM
03646 603656 JMP ALOK
03647 544271 SAD INFLD /REJECT IF EQUAL.
03650 741000 SKP /TELL WHERE IT IS
03651 603656 JMP ALOK
03652 004407 LAC K700K
03653 042320 DAC LOCAT+4
03654 102314 JMS LOCAT
03655 603556 JMP SLMTS+1 /RESTART
03656 102272 ALOK JMS KEYIN /WAIT FOR A C.R.
03657 544360 SAD K215
03660 623555 JMP* SLMTS /EXIT
03661 602312 JMP QUERY /PRINT QUERY AND RESTART
/
/SETUP ACS. PRESS CARRIAGE RETURN TO EXIT
/
03662 000000 SETAC 0
03663 103737 JMS CRLF /CR,LF
03664 204056 LAC SETX /POINTER
03665 044305 DAC PRNT
03666 103701 JMS PNXT /PRINT "SETUP ACS"
03667 700312 KRB
03670 700301 KSF
03671 603670 JMP .-1
03672 700312 KRB
03673 544374 SAD K377 /CHECK FOR A RO
03674 603556 JMP SLMTS+1 /START OVER
03675 750004 LAS
03676 044272 DAC MCWA
03677 103737 JMS CRLF /CR,LF
03700 623662 JMP* SETAC /EXHT
/
.EJECT

```



```

/
/PRINT A STRING AND EXIT.
/
PNXT      0
          LAW      -3
          DAC      WRCNT      /CHARACTER COUNTER
          ISZ      PRNT      /WORD POINTER+1
          LAC*     PRNT
          SNA
          JMP*     PNXT      /ALL DONE OF 0
          DAC      CRLF      /EXIT
MSK       043737      /SAVE WORD
          AND      K77      /MASK 6 BIT CHARACTER
          SAD      K77      /CHECK IF RUBOUT
          JMP      CK3
          DAC      RPETE      /SAVE CHAR
          LAW      -40
          TAD      RPETE
          SMA
          JMP      CRLF-3      /NEG. = ALPHA
          XOR      .-4      /NUMERIC
          TAD      K300      /MAKE ALPHA
          JMS      PCHAR      /PRINT ACS 10-17
          ISZ      WRCNT      /CHECK FOR 3 CHARACTERS
          SKP
          JMP      PNXT+1      /GET NEXT 3 CHARACTERS
          LAC      CRLF      /POSITION NEXT
          RTR;     RTR;     RTR
          JMP      MSK      /PRINT IT
          LAC      RPETE
          TAD      K200      /MAKE NUMERIC
          JMP      CK3-1
/
/CARRIAGE RETURN, LINE FEED
/
CRLF      0
          LAW      215      /ASCII CR
          JMS      PCHAR
          SAD      .+2
          JMP*     CRLF      /EXIT
          LAW      212      /LF
          JMP      CRLF+2
/
.EJECT

```

```

/XAD15 - TAPE 5
/
/HEADER ROUTINE
/
PHDR      0
03746     000000
03747     103737      JMS      CRLF      /CR, LF
03750     204057      LAC      TSTX      /POINTER FOR "TEST"
03751     044305      DAC      PRNT
03752     103701      JMS      PNXT      /PRINT TEST
03753     103775      JMS      CLMN      /SPACE 5
03754     204060      LAC      ADRXA     /"OCTAL ADR."
03755     044305      DAC      PRNT
03756     103701      JMS      PNXT
03757     103775      JMS      CLMN      /SPACE 5
03760     204061      LAC      GDATX     /"GOOD"
03761     044305      DAC      PRNT
03762     103701      JMS      PNXT
03763     103775      JMS      CLMN      /SPACE 5
03764     204062      LAC      BDATX     /"BAD"
03765     044305      DAC      PRNT
03766     103701      JMS      PNXT
03767     103775      JMS      CLMN      /SPACE 5
03770     204063      LAC      BWPA      /"FIELD WITH PAT."
03771     044305      DAC      PRNT
03772     103701      JMS      PNXT
03773     103737      JMS      CRLF      /CR, LF
03774     623746      JMP*     PHDR      /DONE

/
/CLMN      0
03775     000000
03776     777773      LAW      -5
03777     043737      DAC      CRLF
04000     101615      JMS      SPING     /SPACE
04001     623775      JMP*     CLMN
.EJECT

```

```

/
/RETURN ADDRESSES (INDIRECTS)
/
04000 752523 DLMT 752523
04003 000516 KRTN2 RTN2
04004 000737 KRTN3 RTN3
04005 001105 KRTN4 RTN4
04006 004263 LAL ALL
04007 004252 PTO PTO1
04008 002037 ROTA ROTC
04011 002037 ROTB ROTC
04012 004412 STBL KEND+2
04013 000000 SUPTBL 0
04014 005011 ESTBL KEND+401
04015 004114 TSNX TSN
04016 004121 SUPSX SUPS
04017 001660 INSUP SUPIM
04020 002027 SUPDN DNSUP
04021 002044 NBLK BLK1
04022 002163 TBLK BLK2
04023 002114 DONE2 DBLK1
04024 002233 DONE3 DBLK2
04025 004127 BLKSX BLKS
04026 004154 OVRLP OVLAP
04027 004135 BLKTX BLKT
04030 002521 DONE5 BLKB2
04031 002534 DONE6 BLKE2
04032 002557 DONE7 BLKB3
04033 002567 DONE8 BLKD3
04034 002614 DONE9 BLKB4
04035 002620 DON10 BLKC4
04036 002630 DON11 BLKD4
04037 002650 DON12 BLKG4
04040 002656 DON13 BLKH4
04041 002664 DON14 BLKJ4
04042 002672 DON15 BLKL4
04043 004326 ERTBL ERWRD+1
04044 004336 ENERR ERWRD+11
04045 004176 PTWLV PUT12
04046 004243 GOFD GOFLD
04047 004206 RELOC PROR
04050 004220 NERN NOMO
04051 004143 PISIN PROIS
04052 004164 ERSEL SLTER
04053 004227 TLMX TSLM
04054 003556 SLMX SLMTS+1
04055 003625 DON3 CREVR
04056 004235 SETX STACS
04057 004065 TSTX TST
04060 004071 ADRXA ADR
.EJECT

```

24061	24476	GDATX	GDAT
24062	24182	RDATX	RDAT
24063	24185	RWPA	RWPAT
24064	752524	DLMTA	752524 .EJECT

04065 004065
 04066 230524
 04067 777724
 04070 000000

04071 004071
 04072 240317
 04073 401401
 04074 220401
 04075 000000

04076 004076
 04077 171707
 04100 777704
 04101 000000

04102 004102
 04103 040102
 04104 000000

04105 004105
 04106 051106
 04107 400414
 04110 241127
 04111 204010
 04112 562401
 04113 000000

04114 004114
 04115 230524
 04116 434024
 04117 405540
 04120 000000

04121 004121
 04122 202523
 04123 052220
 04124 402323
 04125 774055
 04126 000000

04127 004127
 04130 171402
 04131 401303

/
 /CONSTANTS FOR PRINT ROUTINE TEXTS, PACKED
 /3 CHARACTERS PER WORD.

/ "TEST"
 TST .
 230524; 777724; 0

/
 ADR .
 240317; 401401; 220401; 0

/
 GDAT .
 171707; 777704; 0

/
 RDAT .
 040102; 0

/
 RWPAT .
 051106; 400414; 241127; 204010

562401; 0

/
 /TEST # -
 /
 TSN .
 230524; 434024; 405540; 0

/
 /SUPPRESS -
 /
 SUPS .
 202523; 052220; 402323; 774055

0

/
 /BLOCK #1 -
 /
 RLKS .
 171402; 401303; 406143; 774055

04132 406143
04133 774055
04134 000000

/
/BLOCK #2 -
/

04135 004135
04136 171402
04137 401303
04140 406243
04141 774055
04142 000000

RLKT .
171402; 401303; 406243; 774055

/
/ .EJECT

```

/PROGRAM IS IN FIELD
/
PROIS .
      172220; 012207; 114015; 114023
      064016; 140511; 774004; 0
/
/IS WITH IN PROGRAM
/
OVLAP .
      402311; 241127; 161110; 222040
      220717; 771501; 0
/
/ERROR IN SELECTED FIELD
/
SLTER .
      222205; 402217; 401611; 140523
      240305; 400405; 051106; 770414
      0
/
/PUT ACS 12 ON A 0
/
PUT12 .
      242520; 030140; 614023; 174062
      014016; 776040; 0
/
/PROGRAM RELOCATION ERROR
/
PROR .
      172220; 012207; 224015; 171405
      240103; 161711; 220540; 221722

```

```

04143 004143
04144 172220
04145 012207
04146 114015
04147 114023
04150 064016
04151 140511
04152 774004
04153 000000

```

```

04154 004154
04155 402311
04156 241127
04157 161110
04160 222040
04161 220717
04162 771501
04163 000000

```

```

04164 004164
04165 222205
04166 402217
04167 401611
04170 140523
04171 240305
04172 400405
04173 051106
04174 770414
04175 000000

```

```

04176 004176
04177 242520
04200 030140
04201 614023
04202 174062
04203 014016
04204 776040
04205 000000

```

```

04206 004206
04207 172220
04210 012207
04211 224015
04212 171405
04213 240103

```

04214	161711
04215	220540
04216	221722
04217	000000

/
/NO MORE ERRORS

04220	004220
04221	401716
04222	221715
04223	054005
04224	172222
04225	772322
04226	000000

/
NOMO .
401716; 221715; 054005; 172222

772322; 0

04227	004227
04230	230524
04231	144024
04232	111511
04233	772324
04234	000000

/
/TEST LIMITS
/
TSLM .
230524; 144024; 111511; 772324

0
.EJECT


```

/
/SETUP APTS
/
STACS .
      240523; 402025; 030140; 777723
      0
/
/GO TO FIELD
/
GOFLD .
      401707; 401724; 051106; 400414
      774055
      0
/
/PRINT OUTS INHIBITED
/
PTOI .
      112220; 402416; 242517; 114023
      111016; 241102; 770405; 0
/
/ALL
/
ALL .
      141401; 0
/
      .EJECT

```

```

04235 004235
04236 240523
04237 402025
04240 030140
04241 777723
04242 000000

```

```

04243 004243
04244 401707
04245 401724
04246 051106
04247 400414
04250 774055
04251 000000

```

```

04252 004252
04253 112220
04254 402416
04255 242517
04256 114023
04257 111016
04260 241102
04261 770405
04262 000000

```

```

04263 004263
04264 141401
04265 000000

```

```

/STORAGE AND CONSTANT REGISTERS
/
04266 00000000 SIXT4 LAW -14 /COUNTS 64 PASSES BETWEEN
04267 00000000 NOPRNT 0 /ERROR PRINT SUPPRESSION
04270 00000000 FLAGS 0 /INDICATES END OF ERROR PRINT-OUTS
04271 00000000 INSFLD 0 /SAVES SUBROUTINE FLAGS
04272 00000000 MCWA 0 /CURRENT FIELD WITH PROGRAM
04273 00000000 CNTRL 0 /SAVES ACS SETTINGS.
04274 00000000 PATR 0 /PATTERN GENERATOR
/SAVES GOOD DATA
/AND UPR USED TO OFFSET PATTERNS
04275 00000000 MEMADR 0 /ADDRESS COUNTER
04276 00000000 SVADR 0 /FIELD COUNTER
04277 00000000 CT4K 0 /4K COUNTER
04300 00000000 FIRST1 0 /FIRST FIELD TO TEST
04301 00000000 LAST1 0 /LAST FIELD TO TEST
04302 00000000 BAD1 0 /SAVES BAD DATA
04303 00000000 OCADR 0 /SAVES FAILING OCTAL ADDRESS
04304 00000000 GOOD1 0 /GOOD DATA
04305 00000000 PRNT 0 /POINTER FOR PRINT ROUTINES
04306 00000000 ADRCW 0 /PARTIAL ADDRESS WORD
04307 00000000 EXIT 0 /TO DISMISS
04310 00000000 OVER 0 /POINTER TO START OF SUBROUTINES
04311 00000000 NROTA 0 /ROTATE COUNTER
04312 00000000 NXLOC 0 /NEXT FIELD TO MOVE INTO
04313 00000000 SOURCE 0 /FIELD TO MOVE FROM
04314 00000000 DESTN 0 /FIELD TO MOVE TO
04315 00000000 MOVED 0 /ADDRESS COUNTER FOR MOVING
04316 00000000 MOVES 0 /SAVE AS MOVED
04317 000221 BGNLO RTN1 /EXIT ADR. TO A LO 4K FIELD
04320 760000 BLOC1 LAW /SAVES 1ST ADR. FOR BLOCK 1
04321 760000 BLOC2 LAW /LAST ADR. FOR BLOCK1
04322 760000 BLOC3 LAW /SAVES 1ST ADR. FOR BLOCK2
04323 760000 BLOC4 LAW /LAST ADR. FOR BLOCK2
04324 000000 MAXERR 0 /COUNTS 64 ERROR PRINT-OUT
04325 004326 ERWRD .+1 /SAVES UP TO 8 FIELDS IN ERROR
04326 760000 LAW
04327 760000 LAW
04330 760000 LAW
04331 760000 LAW
04332 760000 LAW
04333 760000 LAW
04334 760000 LAW
04335 760000 LAW
04336 760000 LAST LAW /SAVES LAST FIELD IN ERROR
04337 000000 LSTSUP 0 /LAST ADR. TO BE SUPPRESSED
04340 000000 PATBNK 0 /CURRENT FIELD WITH PATTERN
04341 000000 WRCNT 0 /UTILITY COUNTER
04342 000000 RPETE 0 /UTILITY COUNTER
04343 000000 LOOPT 0 /UTILITY COUNTER
04344 000000 BITN 0 /UTILITY STORAGE
04345 000000 TNUM 0 /TEST NUMBER
/

```

.EJECT

```

/
04346 000001 K1 1
04347 000002 K2 2
04350 000007 K7 7
04351 000010 K10 10
04352 000020 K20 20
04353 000040 K40 40
04354 000077 K77 77
04355 000100 K100 100
04356 000200 K200 200
04357 000212 K212 212
04360 000215 K215 215
04361 000254 K254 254
04362 000260 K260 260
04363 000261 K261 261
04364 000262 K262 262
04365 000263 K263 263
04366 000264 K264 264
04367 000270 K270 270
04370 000272 K272 272
04371 000300 K300 300
04372 000331 K331 331
04373 000370 K370 370
04374 000377 K377 377
04375 000400 K400 400
04376 004000 K4K 4000
04377 007777 K7777 7777
04400 010000 K10K 10000
04401 020000 K20K 20000
04402 040000 K40K 40000
04403 070000 K70K 70000
04404 074000 K74K 74000
04405 400000 K400K 400000
04406 177777 K177 177777
04407 700000 K700K 700000
04410 004411 KEND .+1
000000 .END
NO ERROR LINES
```

ADR	04071
ADRCW	04306
ADRXA	04060
AJIN	03355
ALL	04263
ALL1	00543
ALOK	03656
BAD1	04302
BAK1	00424
BAK2	00612
BAK2A	00645
BAK4	01170
BDAT	04102
BDATX	04062
BEGIN	00200
BGNLO	04317
BITN	04344
BLKA1	02440
BLKA2	02511
BLKA3	02547
BLKA4	02610
BLKB1	02454
BLKB2	02521
BLKB3	02557
BLKB4	02614
BLKC1	02471
BLKC2	02523
BLKC3	02561
BLKC4	02620
BLKD2	02525
BLKD3	02567
BLKD4	02630
BLKE2	02534
BLKE4	02640
BLKF4	02642
BLKG4	02650
BLKH4	02656
BLKJ4	02664
BLKL4	02672
BLKN1	02055
BLKN2	02170
BLKS	04127
BLKSX	04025
BLKT	04135
BLKTX	04027
BLK1	02044
BLK1A	02064
BLK1B	02102
BLK2	02163
BLK2A	02177
BLK2B	02215
BLOC1	04320
BLOC2	04321
BLOC3	04322
BLOC4	04323

BWPA	04063
RWPAT	04115
CBANK	01276
CBNK	01323
CFLG	03503
CKADR	02704
CKBAK	02742
CKB2	00501
CKB3	00724
CKB6	01062
CKCLN	01725
CKERR	03135
CKFCD	03303
CKNXT	03212
CK18B	01007
CK3	03724
CLMN	03775
CLOF	700004
CLOH	700044
CLSF	700001
CMAK	01527
CMOVE	03111
CNROT	03047
CNTRL	04273
CREVR	03625
CRLF	03737
CSUP	01551
CT4K	04277
DBLK1	02114
DBLK2	02233
DESTN	04314
DIND	03376
DLMT	04002
DLMTA	04064
DNMVE	03205
DNSUP	02027
DOALL	00321
DOERR	01450
DONE2	04023
DONE3	04024
DONE5	04030
DONE6	04031
DONE7	04032
DONE8	04033
DONE9	04034
DON10	04035
DON11	04036
DON12	04037
DON13	04040
DON14	04041
DON15	04042
DON3	04055
ENERR	04044
FNOT	03465
FPA	707762

FQUAL	03162
FRROR	01377
FRSEL	04052
FRTRL	04043
FRWRD	04325
ESTBL	04014
EXAM2	00273
EXAM3	00277
EXAM4	00303
EXIT	04307
FCDMV	03273
FIRST1	04300
FLAGS	04270
FWD2	00567
FWD3	00767
FWD4	01142
GDAT	04076
GDATX	04061
GENADR	03043
GOFL	04046
GOFLD	04243
GOLEFT	03066
GOOD1	04304
GOTO	03426
HALT	01364
INDY	01507
INSFLD	04271
INSUP	04017
JMP1	03531
KEND	04410
KEYIN	02272
KRB	700312
KRTN2	04003
KRTN3	04004
KRTN4	04005
KSF	700301
KYBRD	01623
K1	04346
K10	04351
K10K	04400
K100	04355
K177	04406
K2	04347
K20	04352
K20K	04401
K200	04356
K212	04357
K215	04360
K254	04361
K260	04362
K261	04363
K262	04364
K263	04365
K264	04366
0	04367

K272	04370
K300	04371
K331	04372
K370	04373
K377	04374
K4K	04376
K40	04353
K40K	04402
K400	04375
K400K	04405
K7	04350
K70K	04403
K700K	04407
K74K	04404
K77	04354
K7777	04377
LAL	04006
LAST	04336
LAST1	04301
LEGAL	02301
LOCAT	02314
LOCER	03540
LOOP1	04343
LOOP1	00415
LOP2	00456
LOP3	00705
LOP4	01043
LOP4A	01225
LSTSUP	04337
MAXERR	04324
MCWA	04272
MEMADR	04275
MOSOM	03334
MOVE	03325
MOVED	04315
MOVES	04316
MSK	03710
MVBK	03264
MVERR	03510
NBLK	04021
NERN	04050
NOMO	04220
NOMOR	01305
NONE	02754
NOPRNT	04267
NOSW	03444
NROTA	04311
NXBAK	02765
NXLOC	04312
NXPT2	00664
NXPT3	01021
NXSUP	01675
NXTBnk	01344
NXTHI	03260
NXTMV	03210

NXTP4	01262
OCADR	04303
OKAS	03643
OVER	04310
OVLAP	04154
OVRLP	04026
PATBNK	04340
PATR	04274
PCF	700202
PCHAR	01571
PHDR	03746
PISIN	04051
PNXT	03701
POSN	01602
PRNT	04305
PROCTL	01576
PROG	02151
PROIS	04143
PROR	04206
PRSEL	03404
PSA	700204
PSB	700244
PSF	700201
PTO	04007
PTOI	04252
PTWLV	04045
PUT12	04176
QUERY	02312
RBAK1	00420
RBAK2	00605
RBAK4	01160
RBLK4	01133
RCF	700102
READ1	00360
RELOC	04047
REST2	00537
REST4	01102
REST4A	01107
REVR1	02426
REVR2	02421
RFWD2	00562
RFWD3	00757
RFWD4	01126
RITE	03102
RLOP1	00367
ROTA	04010
ROTB	04011
ROTC	02037
RPETE	04342
RRB	700112
RSA	700104
RSB	700144
RSF	700101
2	00511
2A	00520

RST3	00734
RST3A	00741
RTN1	00221
RTN2	00516
RTN3	00737
RTN4	01105
RT19L	03413
SETAC	03662
SETBAK	02722
SETB1	02775
SETB2	03006
SETSUP	02015
SETU1	02346
SETU2	02377
SETX	04056
SET1	02733
SIMU1	02435
SIMU2	02506
SIMU3	02544
SIMU4	02604
SINGL	02360
SIXT4	04266
SLMTS	03555
SLMX	04054
SLTER	04164
SOURCE	04313
SPEXT	01567
SPING	01615
STACS	04235
STBL	04012
STB1	03017
STB2	03031
STER	01464
STLOOP	02336
STMV	03150
STNXT	03227
STOVER	00251
SUB1	03173
SUB2	03247
SUPBLK	01761
SUPDN	04020
SUPIN	01660
SUPS	04121
SUPSX	04016
SUPTBL	04013
SUP1	01730
SVADR	04276
SW0	01443
SW1	01435
SW2	01423
TBLK	04022
TCF	700402
TLMX	04053
TLS	700406
TNLM	04345

TOP	01564
TSF	700401
TSLM	04227
TSN	04114
TSNX	04015
TST	04065
TSTN	01655
TSTNO	01625
TSTX	04057
TST1	00324
TST2	00453
TST2A	00637
TST3	00700
TST4	01035
TST4A	01215
T1RDA	02465
T1RDB	02467
T1RDF	02452
WBLK1	00333
WHERE	01360
WHWAY	03321
WLOP1	00337
WLOP2	00465
WLOP3	00715
WLOP4	01047
WOTIS	03547
WRCNT	04341
WRT1S	03070
XPRT	03316

BEGIN	00200
RTN1	00221
STOVER	00251
EXAM2	00273
EXAM3	00277
EXAM4	00303
DOALL	00321
TST1	00324
WBLK1	00333
WLOP1	00337
READ1	00360
RLOP1	00367
LOOP1	00415
RBAK1	00420
BAK1	00424
TST2	00453
LOP2	00456
WLOP2	00465
CKB2	00501
RST2	00511
RTN2	00516
RST2A	00520
REST2	00537
ALL1	00543
RFWD2	00562
FWD2	00567
RBAK2	00605
BAK2	00612
TST2A	00637
BAK2A	00645
NXPT2	00664
TST3	00700
LOP3	00705
WLOP3	00715
CKB3	00724
RST3	00734
RTN3	00737
RST3A	00741
RFWD3	00757
FWD3	00767
CK18B	01007
NXPT3	01021
TST4	01035
LOP4	01043
WLOP4	01047
CKB6	01062
REST4	01102
RTN4	01105
REST4A	01107
RFWD4	01126
RBLK4	01133
FWD4	01142
RBAK4	01160
BAK4	01170
TST4A	01215

LOP4A	01225
NXTP4	01262
CRANK	01276
NOMOR	01305
CBANK	01323
NXTBNK	01344
WHERE	01360
HALT	01364
ERROR	01377
SW2	01423
SW1	01435
SW0	01443
DOERR	01450
STER	01464
INDY	01507
CMAX	01527
CSUP	01551
TOP	01564
SPEXT	01567
PCHAR	01571
PROCTL	01576
POSN	01602
SPING	01615
KYBRD	01623
TSTNO	01625
TSTN	01655
SUPIN	01660
NXSUP	01675
CKCLN	01725
SUP1	01730
SUPBLK	01761
SETSUP	02015
DNSUP	02027
ROTC	02037
BLK1	02044
BLKN1	02055
BLK1A	02064
BLK1B	02102
DBLK1	02114
PROG	02151
BLK2	02163
BLKN2	02170
BLK2A	02177
BLK2B	02215
DBLK2	02233
KEYIN	02272
LEGAL	02301
QUERY	02312
LOCAT	02314
STLOOP	02336
SETU1	02346
SINGL	02360
SETU2	02377
SEVR2	02421
SEVR1	02426

SIMU1 02435
 RLKA1 02440
 T1RDF 02452
 RLKR1 02454
 T1RDA 02465
 T1RDR 02467
 BLKC1 02471
 SIMU2 02506
 RLKA2 02511
 BLKB2 02520
 RLKC2 02527
 RLKD2 02535
 BLKE2 02534
 SIMU3 02544
 RLKA3 02547
 BLKR3 02557
 RLKC3 02561
 RLKD3 02567
 SIMU4 02604
 RLKA4 02614
 BLKR4 02614
 RLKC4 02620
 RLKD4 02630
 BLKE4 02640
 BLKF4 02642
 BLKG4 02650
 BLKH4 02656
 BLKJ4 02664
 BLKL4 02672
 CKADR 02704
 SETBAK 02722
 SET1 02733
 CKBAK 02742
 NONE 02754
 NXBAK 02765
 SETB1 02775
 SETB2 03006
 STB1 03017
 STB2 03031
 GENADR 03043
 CNROT 03047
 GOLEFT 03066
 WRT1S 03070
 RITE 03102
 CMOVE 03111
 CKERR 03135
 STMV 03150
 EQUAL 03162
 SUB1 03173
 DNMVE 03205
 NXTMV 03210
 CKNXT 03212
 STNXT 03227
 SUB2 03247
 NXTHI 03250

MVEK	03264
FCDMV	03273
CKFCD	03303
XPRT	03316
WHWAY	03321
MOVE	03325
MOSOM	03334
AJIN	03355
DIND	03376
PRSEL	03404
RT19L	03413
GOTO	03426
NOSW	03444
ENOT	03465
CFLG	03503
MVERR	03510
JMP1	03531
LOCER	03540
WOTIS	03547
SLMTS	03555
CREVR	03625
OKAS	03643
ALOK	03656
SETAC	03662
PNXT	03701
MSK	03710
CK3	03724
CRLF	03737
PHDR	03746
CLMN	03775
DLMT	04002
KRTN2	04003
KRTN3	04004
KRTN4	04005
LAL	04006
PTO	04007
ROTA	04010
ROTB	04011
STBL	04012
SUPTBL	04013
ESTBL	04014
TSNX	04015
SUPSX	04016
INSUP	04017
SUPDN	04020
NBLK	04021
TBLK	04022
DONE2	04023
DONE3	04024
BLKSX	04025
OVRLP	04026
BLKTX	04027
DONE5	04030
DONE6	04031
DONE7	04032

DONE8	04033
DONE9	04034
DON10	04035
DON11	04036
DON12	04037
DON13	04040
DON14	04041
DON15	04042
ERTBL	04043
ENERR	04044
PTWLV	04045
GOFL	04046
RELOC	04047
NERN	04050
PISIN	04051
ERSEL	04052
TLMX	04053
SLMX	04054
DON3	04055
SETX	04056
TSTX	04057
ADRXA	04060
GDATX	04061
BDATX	04062
BWPA	04063
DLMTA	04064
TST	04065
ADR	04071
GDAT	04076
BDAT	04102
BWPAT	04105
TSN	04114
SUPS	04121
BLKS	04127
BLKT	04135
PROIS	04143
OVLAP	04154
SLTER	04164
PUT12	04176
PROR	04206
NOMO	04220
TSLM	04227
STACS	04235
GOFLD	04243
PTOI	04252
ALL	04263
SIXT4	04266
NOPRNT	04267
FLAGS	04270
INSFLD	04271
MCHA	04272
CNTRL	04273
PATR	04274
MEMADR	04275
SVADR	04276

CT4K	04377
FIRST1	04301
LAST1	04301
BAD1	04302
OCADR	04303
GOOD1	04304
PRNT	04305
ADRCW	04306
EXIT	04307
OVER	04310
NROTA	04311
NXLOC	04312
SOURCE	04313
DESTN	04314
MOVED	04315
MOVES	04316
BGNLO	04317
BLOC1	04320
BLOC2	04321
BLOC3	04322
BLOC4	04323
MAXERR	04324
ERWRD	04325
LAST	04336
LSTSUP	04337
PATBNK	04340
WRCNT	04341
RPETE	04342
LOOP1	04343
BITN	04344
TNUM	04345
K1	04346
K2	04347
K7	04350
K10	04351
K20	04352
K40	04353
K77	04354
K100	04355
K200	04356
K212	04357
K215	04360
K254	04361
K260	04362
K261	04363
K262	04364
K263	04365
K264	04366
K270	04367
K272	04370
K300	04371
K331	04372
K370	04373
K377	04374
K380	04375

K4K	04375
K7777	04377
K10K	04400
K20K	04401
K40K	04402
K70K	04403
K74K	04404
K400K	04405
K177	04406
K700K	04407
KEND	04410
CLSF	700001
CLOF	700004
CLON	700044
RSF	700101
RCF	700102
RSA	700104
RRB	700112
RSH	700144
PSF	700201
PCF	700202
PSA	700204
PSB	700244
KSF	700301
KRB	700312
TSF	700401
TCF	700402
TLS	700406
EPA	707762