

PDP-9 checkerboard

IDENTIFICATION

Product Code: MAINDEC-9A-D1BB-D (D)
Product Name: PDP-9 Extended Memory
 Checkerboard
Date Created: December 2, 1968
Maintainer: Diagnostic Group
Author: J.W. Richardson

1. ABSTRACT

The PDP-9 Extended Memory Checkerboard diagnostic is designed to verify the operational status of core memory by testing for core failure on half selected lines under worst case noise conditions. The program will test, in 4K segments, any memory configuration of from 8K to 32K words. The program uses eight data patterns for testing.

2. REQUIREMENTS

2.1 Equipment

A standard PDP-9 equipped with from 8K to 32K words of core memory.

2.2 Storage

The program occupies locations 200 to 3323 when residing in the lower 4K field of an 8K bank, and locations 10200 to 13323 when residing in a high order 4K field.

3. LOADING PROCEDURE

3.1 Method

The program may be loaded into any low order 4K field (00000 to 07777) of any 8K memory bank.

- a. The tape supplied is punched in the HRI mode.
- b. Place the tape in the reader.
- c. Set ADDRESS switches to 200
- d. Press I/O RESET, and then READ-IN.

Put computer in 9 mode

4. STARTING PROCEDURE

4.1 Starting Address

The program is self-starting at the completion of loading.

4.2 Restarting Address

Restart from 215 (or 10215) *(to setup the test limits and ACS.)*

Restart from 200 (or 10200) *(to retain previous program conditions.)*

4.3 Operator Action

At the completion of program load the program will print "TEST LIMITS."

The operator must then specify via the teletype keyboard the amount
of core memory to test, followed by a carriage return.

The program assumes the 4K fields numbered from 0 through 7. Memory bank
0 would contain field 0 and 1; memory bank 1, fields 2 and 3, etc. The
even numbered fields represent locations 00000 to 07777 of any 8K bank,
and the odd numbered fields represent location 10000 to 17777 of any 8K
memory bank.

Use the following procedure to specify the test limits:

- a. Type two octal numbers, separating the numbers with a comma, and
then a carriage return.

- b. The first number typed signifies the first 4K field to test, and the second number the last 4K field to test.
 - c. The program will begin testing with the lowest order 4K field to test and will test all consecutive fields up to and including the highest specified.
 - d. The 4K field containing the program may be included. It will be tested after program relocation takes place. Program relocation is described in Section 5.3.1.
3. If an error is made during typing, press the RUBOUT key. "TEST LIMITS" will be printed again. Previous input is ignored.
- f. 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 field to test.
 - g. Any single field or any 2 or more consecutive fields may be specified.

For the following examples assume that the program is in field 0 (00000 to 07777 of bank 0), and the PDP-9 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)

Field 3 will be tested alone. Field 3 is locations 30000 to 37777
of memory bank 2.

Example D:

TEST LIMITS

4,6)

Locations 40000 through 67777 (field 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 whn a single field
is selected which currently contains the program.

"TEST LIMITS" is printed again, and the operator must then correct the
test limits.

Operation of the program is unpredictable if the amount of core memory selected for testing exceeds the actual amount available, i.e., selecting 32K for testing on a PDP-9 equipped with a maximum of 24K.

4.3.1 Setup ACS

After the test limits are specified, the program will print "SETUP ACS."

For normal program operation, the ACS must be set to 000000 octal.

Press any key on the Teletype keyboard after setting the ACS to all 0's.

The program will then 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 memory field.

5. OPERATING PROCEDURE

5.1 Program and Operator Action

- a. Load the program into memory field 0 as described in Section 3.
- b. Specify the test limits via keyboard as described in Section 4.3.
- c. The message "SETUP ACS" will be printed. Set the ACS to 000000 octal, and press any keyboard key.
- d. The program will perform all four tests on all of core memory specified, after which, automatic program relocation takes place.

5.2 Operational Switch Settings

Normal operation of the program requires the ACS set to 000000 octal.

Refer to Section 8.2. Applications, for switch settings provided for

trouble-shooting.

5.3

Subroutine Abstracts

A total of eight data patterns are generated by the program. The eight patterns are divided into four tests. Each test generates one pattern and its complement, resulting in eight patterns. Each test uses common write and read loops, and all patterns are manipulated in the same manner. The following sequence is performed by each test on each 4K field under test.

- a. Write one pattern in all 4K fields selected.
- b. Set up for the lowest order 4K field, and perform a read, complement, write sequence on each location in the 4K field. No error checking is performed.
- c. Repeat step (b) 15 more times on the same 4K field.
- d. Check each location in the 4K field for error by reading one location, complementing the data and rewriting in the same location. This read/write sequence is repeated 4 times on each location. Error checking is performed on each second read.
- e. Repeat step (d) on each location in the 4K field.
- f. After checking the 4K field, repeat steps (d) and (e) again. This time, the program stalls for a random period of time after reading and testing each consecutive 200 (octal) word block. The minimum stall time is approximately 10 US, and the maximum

approximately 24 MS. The tested 4K field is not referenced by the program during the stall period.

- g. Set up for the next 4K field and repeat the read and test sequence in steps (b) through (f).

When all fields have been tested in this manner, the next checkerboard pattern is written and steps (b) through (f) repeated.

When all four tests have been executed, the program then relocates and performs all four tests again.

5.3.1 Program Relocation

Program 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 may 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. This procedure is repeated until stopped by the operator with ACS 0. As an example, if the program is initially in field 0, and 32K of memory is selected, the program would relocate from 0 to field 7, then to field 7, 5, 4, 3, 2, 1, 0 in that order. The program does 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. ~~4~~

Fields 0 through 3 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 8.2.6).
- 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).

This 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 printout may be deleted by placing ACS 11 on a 1 at any time. The printout 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 set up 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 relocations, the address in error, and the "good" and "bad" data words are printed. The error printout 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 Printouts and Description

Immediately after the first error is detected, the header shown below is printed. The header is not printed again until restarting from 200 or 211.

TEST	OCTAL ADR	GOOD	BAD	PAT CONTROL WORD
------	-----------	------	-----	------------------

Where:

TEST = The current test which detected the error.

OCTAL ADR = The octal address which contains the data in error.

GOOD = This will always equal 000000 or 777777 octal

BAD = The data as read from that address. This will always contain one or more bits which are the complement of those shown under *GOOD*.

PAT CONTROL WORD = The control word used by the current test to generate the checkerboard pattern. This will be 037700 or 740074 for test 1, 001774 or 776000 for test 2; 037701 or 740075 for test 3; 001775 or 776001 for test 4.

TEST	OCTAL	ADR	GOOD	BAD	PAT CONTROL WORD
1	014000		000000	000001	074400
1	060200		777777	767777	740074
3	014000		000000	000001	037701
4	037555		000000	400000	776001

In the above example, errors were detected by tests 1, 3 and 4 in memory fields 1, 6 and 3. Test 1 detected a "picked up" bit at location 4000, field 1, and a dropped pit at location 200, field 6. Test 3 detected the same error as test 1 at location 4000, field 1, and test 4 detected a bit 0 error at location 7555 in field 3.

After each error printout, the program continues testing with the next sequential memory location.

Three AC switches may be used to control the error printouts. Placing ACS 0 on a 1 during the printout will cause a program halt after completion of printing. ACS 1 on a 1 will inhibit the printout and cause a program halt. Press CONTINUE to receive the error printout and to continue testing, ACS 2 on a 1 will inhibit printout 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	PAT CONTROL WORD
------	-----------	------	-----	------------------

PROGRAM RELOCATION ERROR

031000	741000	740000
031001	611005	601005
031002	760207	760007

NO MORE ERRORS

The above example shows three 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 SKP 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.

6.1.2 PRINTOUTS INHIBITED

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

This feature is included to prevent lengthy error printouts when the program is being run for an extended period of time unattended. Error printouts may be resumed by restarting the program from location 215.

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 4.3.
- b. 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.

6.1.5 FIRST/LAST ADR IS WITHIN PROGRAM

The operator has specified the first or last address, as indicated by the printout, which is within the program area. Retype a new address. See Section 8.2.8 for setting up scope loops via keyboard.

7. RESTRICTIONS

7.1 Starting Restrictions

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

Start from 215 to retain the present program conditions.

7.2 Operating Restrictions

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

8. MISCELLANEOUS

8.1 Execution Time

The time required to run all four tests on one 4K field is approximately 30 seconds.

8.2 Applications

To give the operator control of the program, the ACS are 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 relocating. The operation may not be inhibited

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

8.2.1 Halt After Test or Error Printout - 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 PC will = 671. The ACS may then be changed if desired. Press CONTINUE to recover. If no ACS changes the program will assume the test which was interrupted. If ACS changes were made, the new settings are stored and executed.

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

8.2.2 Delete Error Printout and Halt on Error - ACS 1

Raising ACS 1 at any time cause all data error printout to be inhibited. A halt will occur with the PC = 671 if an error occurs. Press CONTINUE to receive the error printout 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 raising any combination of ACS 3 through 6. 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 down, 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 EPC and MB Indicators. To retain the program in its current 4K field, place ACS 9 on a 1 at any time.

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 printout place ACS 11 on a 0. This switch does not inhibit the message printout 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 printout 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 in step (d) above, press carriage return to resume automatic relocation.

If a data error has previously detected in the new 4K field, the message "ERROR IN SELECTED 4K" 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. Printouts 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 Scope Loop - ACS 13

The operator may initiate, via keyboard, any single or any group of consecutive locations up to 4K for the program to loop on. Any of the four data patterns may also be requested. No error checking is done in the loop. The loop reads a location, complements the data, and rewrites into the same location. Each location within the limits specified is treated likewise. The limits specified are looped until halted with the STOP key. The program must be restarted from 200 to resume normal operation.

Initiate the loop with the following procedure:

- a. Halt the program with ACS 0.
- b. Place ACS 13 on a 1 and press CONTINUE.
- c. The message "TEST" will be printed. The program now waits for a selection by the operator.
- d. Type the desired test number. Either 1, 2, 3, or 4. An automatic carriage return follows.
- e. The message "FIRST ADR" is printed.
- f. Type the first address of the group to be looped by the program. This number must be a 5 digit octal number. An automatic carriage return follows.
- g. The message "LAST ADR" is printed. Type a 5 digit octal address to indicate the last address of the group. The scope loop is immediately entered after the last address is typed.

The loop may also be initiated by restarting from 200 or 215 and placing ACS 13 on a 1 under "SETUP ACS."

If a typing error is made, press the RUB-OUT key. A ? will be printed, and the input routine will restart with "TEST."

Example A:

TEST 3
FIRST ADR 10100
LAST ADR 10100

Address 100 in field 1 will be looped continuously after checkerboard pattern number 3 is written in the entire 4K field.

Example B:

TEST 2
FIRST ADR 20000
LAST ADR 27777

The entire 4K field number 2 will be looped after pattern number 2 is written.

Example C:

TEST 2
FIRST ADR 27777
LAST ADR 20000

The input routine will reverse the two addresses and perform exactly as example B.

Example D:

TEST 4
FIRST ADR 00100
FIRST ADR IS WITHIN PROGRAM
FIRST ADR 10100
LAST ADR 10200

Example D shows the message printed when an address is selected which is in the field containing the program. In this case, field 0.

Example E:

TEST 5

?

TEST 4

FIRST ADR 70000

LAST ADR 73000

In example E the operator typed an invalid test number. The program printed the question mark, and restarted with TEST.

The instructions in the body of the scope loop appear below:

	.LOC 1400	
SCP1	EEM	/ENABLE EXTEND MODE
	LAC* MEMADR	/READ
	CMA	/COMPLEMENT DATA
	DAC* MEMADR	/WRITE
	LAC MEMADR	/ADDRESS
	SAD LTST	/COMPARE TO LAST
	JMP STSCP	/DONE
	ISZ MEMADR	/INCREMENT ADR
	JMP SCP1	/LOOP
STSCP	LAC ADRA	/FIRST ADR
	DAC MEMADR	/RESTORE COUNTER
	JMP SCP1	/GO TO TOP OF LOOP

Note that one 4K field is the maximum that may be looped by the program.

If 4K field boundaries are overlapped, the checkerboard pattern will be written in the field specified by the FIRST ADR. The scope loop, however, will reference the locations specified.

8.2.9 Bit Suppression - ACS 14

Excessive error printouts due to one or more bits in error may be suppressed by specifying the bit positions via keyboard input. The bit positions will still contain the checkerboard pattern, and will be tested. Error printouts will occur for any bit position not selected.

Use the following procedure:

- a. Halt the program with ACS 0.
- b. Place ACS 14 on a 1, and press CONTINUE.
- c. The message "SUPPRESS" will be printed and the program waits for input.
- d. Place ACS 14 on a 0.
- e. Type in decimal, the desired bit position to be suppressed (0 through 17).
- f. Press the carriage return key.

Error printouts for that position alone will not occur.

To suppress more than one bit position:

- a. Proceed as above, but separate the selected bit positions with a comma in step (e). As an example, to suppress bits 0, 8 and 17, type 0, 8, 17 ~~,~~. It is not necessary to type the positions in numerical order.

Press RUB-OUT and then CARRIAGE RETURN to continue with error print-outs of all bit positions.

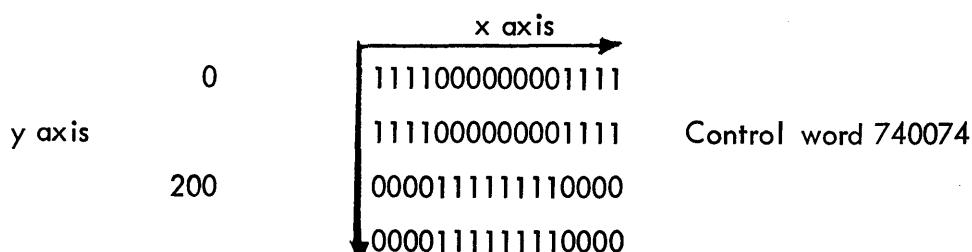
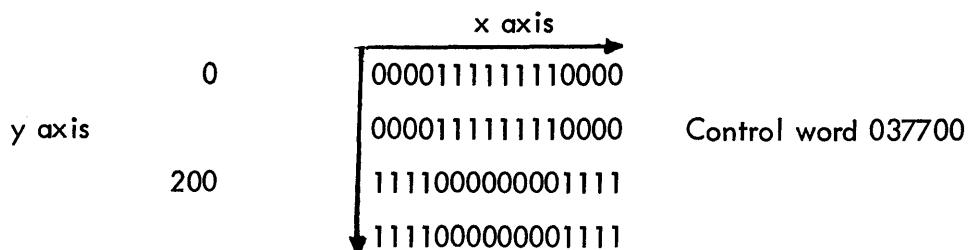
Each time the bit suppressing routine is entered with ACS 14, the previously selected positions must be retyped if suppression is wanted.

The program is not effected in anyway by the typing of letters, or numbers greater than 17. The resulting positions suppressed would be unpredictable.

9. PROGRAM DESCRIPTION

The program is designed to run worst case checkerboard patterns for 2 1/2D memory stacks. A minimum of 8K of core memory is required and a maximum of 32K words may be tested. The program automatically relocates from 4K field to 4K field to test all of memory from each field. The patterns generated by the program are shown below as they would appear in a portion of one bit plane. The patterns are complemented every 200, 400, 600, etc., or every 100, 300, 500, etc., octal locations as shown.

TEST 1



TEST 2

y axis x axis

0	0000000011111111
200	0000000011111111
	1111111100000000
	1111111100000000

Control word 001774

x axis

y axis x axis

0	1111111100000000
200	1111111100000000
	0000000011111111
	0000000011111111

Control word 776000

TEST 3

y axis x axis

0	000011111110000
200	1111000000001111
	1111000000001111
	000011111110000

Control word 037701

x axis

y axis x axis

0	1111000000001111
200	000011111110000
	000011111110000
	1111000000001111

Control word 740075

<u>TEST 4</u>		
	x axis	
0	0000000011111111	
y axis	1111111000000000	Control word 001775
200	1111111000000000	
	0000000011111111	
	x axis	
0	1111111000000000	
y axis	0000000011111111	Control word 776001
200	0000000011111111	
	1111111000000000	

The operator is given a degree of control over the program with the AC switches. The operator may halt the program (0); inhibit error printouts and halt on error (1); substitute the TTY BELL for error indication (2); select any one or a combination of tests (3 through 6); inhibit program relocation (9); relocate the program to any 4K field (12); set up scope loops via keyboard input (13); and inhibit error printouts for one or more bit positions (14).

9CHKR PAGE 1

/COPYRIGHT 1968, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

/POP-9 EXTENDED MEMORY CHECKERBOARD, 8K MINIMUM

/REQUIRED. S.A.=200. RESTART AT 215.

/

.TITLE 9CHKR

.FULL

.LOC 210

/

00200 703302
00201 143165
00202 100663
00203 043166
00204 101674
00205 102002
00206 777777
00207 043235
00210 777700
00211 043304
00212 777770
00213 043163
00214 143164
00215 707704
00216 201636
00217 041621
00220 101615
00221 142715
00222 203215
00223 543214
00224 741000
00225 600236
00226 543166
00227 600202
00230 201103
00231 043305
00232 777770
00233 043211
00234 760000
00235 063305
00236 443305
00237 443211
00240 600235

BEGIN

CAF

DZM FLAGS /CLEAR PROGRAM FLAGS
JMS WHERE /SEE WHERE PROGRAM IS

/SAVE BANK#

DAC INSFLO JMS SLMTS /SETUP TEST LIMITS

JMS SETAC /SETUP ACS

LAW -1 DAC BITSUP /MASK FOR BIT SUPPRESSION

LAW -1'0

DAC MAXERR

LAW -1'

DAC SIXT4

DZM NOPRNT

LEM

LAC GETAD-1

DAC LOCAT+4

JMS LOCAT

DZM PHDR

LAC LAST1

SAD FIRST1

SKP

JMP .+3

SAD INSFLO

JMP REGIN+2

LAC FRTBL

DAC ERWRD

LAW -1'

DAC CT16

LAW

DAC* FRWRD

ISZ ERWRD

ISZ CT16

JMP .-3

/CLEAR PROGRAM FLAGS

/SEE WHERE PROGRAM IS

/SAVE BANK#

/SETUP TEST LIMITS

/SETUP ACS

/MASK FOR BIT SUPPRESSION

/CLEAR PROGRAM FLAGS

/SEE WHERE PROGRAM IS

/SAVE BANK#

/SETUP TEST LIMITS

/SETUP ACS

/MASK FOR BIT SUPPRESSION

/CLEAR PROGRAM FLAGS

/SEE WHERE PROGRAM IS

/SAVE BANK#

/SETUP TEST LIMITS

/SETUP ACS

/MASK FOR BIT SUPPRESSION

/CLEAR PROGRAM FLAGS

/SEE WHERE PROGRAM IS

/SAVE BANK#

/SETUP TEST LIMITS

/SETUP ACS

/MASK FOR BIT SUPPRESSION

/CLEAR PROGRAM FLAGS

/SEE WHERE PROGRAM IS

/SAVE BANK#

/SETUP TEST LIMITS

/SETUP ACS

/MASK FOR BIT SUPPRESSION

/CLEAR PROGRAM FLAGS

/SEE WHERE PROGRAM IS

/SAVE BANK#

/SETUP TEST LIMITS

/SETUP ACS

/MASK FOR BIT SUPPRESSION

/CLEAR PROGRAM FLAGS

/SEE WHERE PROGRAM IS

/SAVE BANK#

/SETUP TEST LIMITS

/SETUP ACS

/MASK FOR BIT SUPPRESSION

/CLEAR PROGRAM FLAGS

/SEE WHERE PROGRAM IS

/SAVE BANK#

/SETUP TEST LIMITS

/SETUP ACS

/MASK FOR BIT SUPPRESSION

/CLEAR PROGRAM FLAGS

/SEE WHERE PROGRAM IS

/SAVE BANK#

/SETUP TEST LIMITS

/SETUP ACS

/MASK FOR BIT SUPPRESSION

/CLEAR PROGRAM FLAGS

/SEE WHERE PROGRAM IS

/SAVE BANK#

/SETUP TEST LIMITS

/SETUP ACS

/MASK FOR BIT SUPPRESSION

EJECT

/LAW = NO ERROR IN TABLE

YCHKR PAGE 2

00241	043167	DAC LAST	/NO ERROR IN LAST
00242	100663	JMS WHERE	
01243	043166	DAC INSFLD	
00244	201103	LAC ERTBL	/ERROR TABLE POINTER
00245	043305	DAC ERWRD	
/RETURN TO STOVER AFTER ANY ACS CHANGES			
/WHILE RUNNING			
/			
00246	750004	STOVER LAS	/READ TEST PARAMETERS
00247	503302	AND K577	
00250	043170	DAC MCWA	
00251	503252	AND K44	
00252	744200	S2A:CLL	
01253	602425	JMP FCDMV	/BIT 12 A1 = FORCE RELOCATE
00254	750004	LAS	
00255	503251	AND K24	
00256	740200	S2A	
00257	601106	JMP KYBRD	/BIT 16 A1 = KEYBOARD INPUT
00260	750004	LAS	
00261	503250	AND K14	
00262	740200	S2A	
00263	101431	JMS SUPBIT	
00264	203170	LAC MCWA	
01265	503300	AND K74K	/PARAMETERS
00266	741200	SNA	/MASK BITS 3 10 6
00267	600323	JMP DUALL	/A L 0 = DO ALL TESTS
/EJECT			

```

        /
        /EXAMINE TEST SWITCHES 3 TO 6
        /
        DV270    203170
        DV271    503276
        DV272    740200
        DV273    600327
        DV274    203275
        DV275    503170
        DV276    740200
        DV277    600340
        DV300    203273
        DV301    503170
        DV302    740200
        DV303    600351
        DV304    203271
        DV305    503170
        DV306    740200
        DV307    600362
        DV310    707704
        DV311    443163
        DV312    600316
        DV313    143164
        DV314    777770
        DV315    043163
        DV316    750004
        DV317    503270
        DV320    740200
        DV321    600246
        DV322    602141

        EXTST    LAC MCWA
                  AND K40K
                  SZA
                  JMP TST1
        EXAM2    LAC K20K
                  AND MCWA
                  SZA
                  JMP TST2
        EXAM3    LAC K10K
                  AND MCWA
                  SZA
                  JMP TST3
        EXAM4    LAC K4K
                  AND MCWA
                  SZA
                  JMP TST4
                  LEM
                  ISZ SIXT4
                  JNP ,+4
                  DZM NOPRNT
                  LAW -1A
                  DAC SIXT4
                  LAS
                  ANI K410
                  SZA
                  JMP STOVER
                  JMP CMUVE /DONE ALL TESTS, SETUP
                           /FOR RELOCATION

        /
        /SETUP TO RUN ALL TESTS
        /
        DOALL    LAC MCWA
                  XOR K74K
                  DAC MCIA
                  JMP TST1
                           /SET ALL TEST BITS
                           /RESTORE
                           /TEST 1

        /
        /TEST 1. WRITE CHECKER PATTERN #1
        /
        DV323    203170
        DV324    243300
        DV325    043170
        DV326    600327

        TST1    LAC PCNA
                  DAC PCW
                  DAC CNTRL
                  LAW 261
                  DAC TNUM
                  JMS NETWK
                  JMS CREAD
                  JMP EXAM2
                  JNP ,-3
                           /TEST 1 PAT. CONTROL WORD
                           /ASCII 1
                           /TEST NUMBER
                           /GO WRITE IN ALL FIELDS
                           /NOW GO READ AND TEST
                           /SEE IF TEST 2 WANTED
                           /DO COMPLEMENT

        /
        /TEST 2,WRITE CHECKER PATTERN #2
        /
        DV340    203174
        DV341    043171

        TST2    LAC PCWB
                  DAC PCW
                           /TEST 2 PAT.CONTROL WORD

```

9CHKR PAGE 4

00342 043172
00343 760262
00344 043203
00345 100373
00346 100404
00347 600300
00350 600345

DAC CNTRL
LAW 262
DAC TNUM
JMS NETWK
JMS CREAD
JMP EXAM3
JMP , -3

/ASCII 2
/TEST NUMBER
/WRITE IN ALL FIELDS
/READ AND TEST EACH FIELD
/SEE IF TEST 3 WANTED
/DO COMPLEMENT

EJECT

9CHKR PAGE 5

/TEST 3. WRITE CHECKER PATTERN # 3.

00351	203175	TST3	LAC PCWC DAC PCW DAC CNTRL LAW 263 DAC TNUM JMS NETWK JMS CRFAD JMP EXAM4 JMP .-3	/TEST 3 PAT, CONTROL WORD /ASCI 3 /TEST NUMBER /WRITE IN ALL FIELDS /READ AND TEST EACH FIELD /SEE IF TEST 4 SELECTED /DO COMPLEMENT
-------	--------	------	---	--

/TEST 4. WRITE CHECKER PATTERN # 4

00362	203176	TST4	LAC PCWD DAC PCW DAC CNTRL LAW 264 DAC TNUM JMS NETWK JMS CREAD JMP EXAM4+4 JMP .-3	/TEST 4 PAT, CONTROL WORD /ASCI 4 /TEST NUMBER /WRITE IN ALL FIELDS /TEST EACH FIELD /PREPARE TO RELOCATE /DO COMPLEMENT
-------	--------	------	---	--

/ROUTINE TO SETUP ADDRESSES FOR WRITE LOOP

00373	000000	NETWK	0 JMS SETU1 JMS CBANK SAP JMP* NETWK JMS GENPAT JMS NXTBNK JMP NETWK+2 JMP* NETWK	/SETUP 1ST FIELD TO TEST /SEE IF IT HAS PROGRAM /NO /EXIT /ACTUALLY WRITE ONE FIELD /SETUP FOR NEXT FIELD /SEE IT IT HAS PROGRAM /WROTE ALL, EXIT
-------	--------	-------	---	--

/ROUTINE TO SETUP ADDRESSES FOR READ LOOP

00404	000000	CREAD	0 JMS READ LAW -4 XOR PCW SAP CNTRL JMP* CREAD DAC CNTRL ISY CRFAD JMP* CREAD	/ACTUALLY READ AND TEST 1 FIELD /AC=COMPLEMENT OF PCW /ALL DONE IF EQUAL /EXIT /CNTRL=COMPLEMENT PATTERN /RETURN+1 /EXIT AND WRITE COMPLEMENT
-------	--------	-------	---	---

,EJECT

/WRITE ROUTINE FOR ALL PATTERNS

```

00415    0000000          /PATTERN CONTROL WORD
00416    203172           /SAVE
00417    043202           /BIT 17 TO LINK
00420    740020           /64 DECIMAL
00421    203254           /TEST BIT 17
00422    740400           /BIT 17=0 SAYS NO OFFSET
00423    750000           /LWR=0 OR 100
00424    043204           /128 DECIMAL
00425    343255           /UPR=300 OR 200
00426    043205           /-16 DECIMAL
00427    777760           /SHIFT COUNTER
00430    043211           /CURRENT CONTROL WORD
00431    203202           /SAVE AGAIN
00432    043177           /START GENERATING A PATTERN
00433    203177           RCL
00434    744010           DAC PATR
00435    043177           SEL:CLA
00436    751400           CMA
00437    740001           FEM
00440    707702           DAC# MEMADR
00441    063206           ISZ MEMADR
00442    443206           ISZ CT4K
00443    443210           SKP
00444    741400           JMP# GENPAT
00445    620415           ISZ CT16
00446    443211           JMP WLOOP
00447    600433           LAC MEMADR
00450    203206           AND K377
00451    503267           SAD LWR
00452    543204           SAD .+4
00453    600457           JMP .+4
00454    543205           SAD UPR
00455    741400           SKP
00456    600427           JMP WLOOP-4
00457    203202           LAC PATN
00460    740001           CMA
00461    043202           DAC PATN
00462    600427           JMP WLOOP-4

```

.EJECT

```

/READ AND TEST ROUTINE. READ AND TEST ONE 4K
/FIELD AT A TIME. TEST EACH FIELD TWICE, RANDOMLY
/STALLING AFTER EACH 128 WORD BLOCK DURING THE
/SECOND PASS.
/
        READ      N
        JMS      SETU1    /SETUP FOR FIRST FIELD
        JMS      CBANK   /SEE IF IT HAS PROGRAM
        SKP
        JMP*    READ     /NO MORE TO TEST
        JMS      RCOMP   /COMPLEMENT FIELD 16 TIMES
        LAK     -2
        DAC      ADRA    /PASS COUNTER
        LAC      CNTRL   /PAT. CONTROL WORD
        DAC      PATR    /SAVE
        LAW     -10    / -8 DECIMAL
        DAC      CT128   /COUNTS 8- 16 WORD BLOCKS
        RSCNT
        LAK     -20
        DAC      CT16    /SHIFT COUNTER
        LAC
        DAC      PATR
        LAC      PATN
        RLOOP
        DAC      PATN
        DAC      SALICLA /LINK A1 = WORD OF 1'S
        CMA
        DAC      GOOD1   /WHAT DATA SHOULD BE
        LAK     -4
        DAC      CT04
        FEM
        RCMA
        LAK*    MEMADR /READ
        CMA
        DAC*    MEMADR /COMPLEMENT
        LAK*    MEMADR /WRITE
        DAC
        DAC*    MEMADR /READ
        DAC      MEMADR /COMPLEMENT
        DAU*    MEMADR /WRITE
        LAC*    MEMADR /READ
        SAD   GOOD1 /COMPARE
        SKP
        JMS      ERROR   /PRINT INFO
        IS*     CT04    /COMPLEMENT 8 TIMES
        IS*     RCMA
        JMP
        ERRTN
        IS*     MEMADR /ADDRESS +1
        IS*     CT4K    /4K WHEN SKIP
        JMP     CK16
        IS*     ADRA    /2 PASSES IF SKIP
        SKP
        JMP     DREAD
        LAC     JMP1    /C (JMP1) = JMP C128
        DAC     RSTOR
        DAC     RSTOR+4
        JMS     STALL   /STALL RANDOMLY
        LAK     -10000  / -4K
        DAC     CT4K    /4K COUNTER
        043210

```

YCHKR PAGE 8

VI 544	343206	TAD	MEMADR	
VI 545	043206	DAC	MEMADR	
VI 546	600473	JMP	RSCNT-2	/RE- TEST WITH STALLS
VI 547	443211	CK16	ISZ	CT16 /16 WORDS WHEN 0
VI 550	600503		JMP	RLOOP /LOOP
VI 551	203206		LAC	MEMADR /CURRENT ADDRESS
VI 552	503267		AND	K377
VI 553	543204		SAD	LWR /LWR = 0 OR 100
VI 554	600560	RSTOR	JMP	.+4 /TIME TO COMPLEMENT
VI 555	543205		SAD	UPR /UPR = 200 OR 300
VI 556	741000		SKP	/COMPLEMENT PATTERN
VI 557	600477		JMP	RLOOP-4
VI 560	203177		LAC	PATR /PAT. CONTROL WORD
VI 561	740001		CMA	/COMPLEMENT
VI 562	043177		DAC	PATR
VI 563	600477		JMP	RLOOP-4

.EJECT

```

      /
      DV564  200573   DREAD    LAC JMP2
      DV565  040557   DAC RSTOR   /RSTOR=JMP RLOOP AGAIN
      DV566  040563   DAC RSTOR+4
      DV567  100653   JMS NXTBNK  /SETUP FOR NEXT FIELD
      DV570  600465   JMP READ+2  /SEE IF IT HAS PROGRAM
      DV571  620463   JMP* READ   /READ ALL FIELDS. EXIT

      /
      DV572  443212   C128     ISZ CT128  /STALL RANDOMLY AFTER 128
      DV573  600477   JMP2     JMP RLOOP+4
      DV574  100676   JMS STALL  /MAX. OF 67 MS
      DV575  600475   JMP RSCNT  /READ NEXT 128 WORD BLOCK

      /
      DV576  6000572  JMP1     JMP C128  /JMP CONSTANT

      /
      /RCOMP ROUTINE. READ AND COMPLEMENT THE
      /4K FIELD 16 TIMES. READ BACKWARDS
      /
      DV577  000000   RCOMP    N
      DV600  777760   LAK -21   /-16 DECIMAL
      DV601  043211   DAC CT16   /PASS COUNTER
      DV602  203206   LAC MEMADR
      DV603  243272   XUR K7777
      DV604  043206   DAC ME4ADR  /FIRST TO READ=X777
      DV605  707702   FEX
      DV606  223206   LAC* MEMADR  /READ
      DV607  740001   CMA     /COMPLEMENT
      DV610  063206   DAC* ME4ADR  /WRITE
      DV611  203206   LAC MEMADR
      DV612  503272   AND K7777
      DV613  741200   SNA     /DONE IF AT ADDRESS N
      DV614  600621   JMP PASSC
      DV615  777777   LAK -1
      DV616  343206   TAD ME4ADR  /ADDRESS -1
      DV617  043206   DAC ME4ADR
      DV620  600606   JMP CL0OP  /READ

      /
      DV621  443211   PASSC   ISZ CT16  /16 PASSES WHEN SKIP
      DV622  600602   JMP RLOOP+3 /DO ANOTHER PASS
      DV623  620577   JMP* RCOMP /EXIT

      /
      .EJECT

```

```

/
/SETUP FOR FIRST 4K FIELD
/
SETU1    R
        LAC FIRST1      /FIRST TO TEST
        DAC MEMADR     /ADDRESS COUNTER
        DAC SVADR
        LAW -11000
        DAC CT4K        /4K COUNTER
        JMP* SETU1      /EXIT

/
/ROUTINE TO SEE IF TESTED FIELD HAS PROGRAM
/
CBANK    R
        JMS WHERF       /CURRENT PROGRAM FIELD
        SAD MEMADR     /NEXT TO TEST
        SKP
        JMP* CBANK      /EXIT
        JMS NXTBNK     /SEE IF CURRENT IS LAST
        JMP .+3          /NO, ADD 4K
        ISZ CBANK       /RETURN +1
        JMP* CRANK
        LAC SVADR
        TAD K1MK        /CURRENT +4K
        DAC SVADR /NEW FIELD
        DAC MEMADR
        LAW -11000      /-4K
        DAC CT4K        /4K COUNTER
        JMP CBANK+1     /EXIT AND TEST NEW FIELD

/
/ROUTINE TO CHECK FOR LAST FIELD
/
NXTBNK   R
        LAS      /CHECK ACSW FOR HALT
        SPA
        JMS HALT /GO HALT
        LAC SVADR
        SAD LAST1 /ALL DONE IF EQUAL
        JMP NOMOR
        JMP NOMOR+2

/
/ROUTINE TO DETERMINE WHERE PROGRAM IS
/
WHERE    R
        /CONTAINS EPC
        LAC .-1
        AND K7MK        /CLEAR ALL BUT BITS 3,4,5
        JMP* WHERE      /EXIT

.EJECT

```

```

/HALT ROUTINE. PRESS CONTINUE TO RESUME
/TESTING, OR IF ACS CHANGES, TO EXECUTE
/NEW PARAMETERS.
/
    HALT      0
    HLT          /PRESS CONTINUE
    LAS
    AND K577
    SAN MCWA
    JMP* HALT
    JMP STOVER
    /RESUME WHERE LEFT OFF
    /EXECUTE NEW PARAMETERS

/
/STALL ROUTINE
/
    STALL      0
    JMS GENRAN   /GET RANDOM NUMBER
    AND K17S     /77777. MAKE 8K MAXIMUM
    CMA
    DAC CRLF    /USE AS COUNTER
    ISZ CRLF    /STALL
    JMP .-1
    EEM
    JMP* STALL   /EXIT
    /
    /
/RANDOM NUMBER GENERATOR
/
    GENRAN      0
    LFM
    LAC RANDEX
    SAN ENtbl    /CHECK FOR END OF TABLE
    SKP
    JMP RANTAD-1  /GENERATE RANDOM
    LAC TBLTOP
    DAC RANDEX
    LAC RANCON
    LAC RANCON
    CLL:RAL
    SEL          /TEST BIT 0
    TAI K1
    DAC RANCON
    LAC* RANDEX
    RANTAD      TAI RANCON
    DAC* RANDEX
    LAC RANSAV
    RAR
    TAI* RANDEX
    DAC RANSAV
    ISZ RANDEX
    JMP* GENRAN  /INCREMENT POINTER
    /
    RANDX      RANTHL+10
    ENtbl       RANTRL+10
    TBLTOP     RANTRL
    .EJECT

```

/ERROR PRINT-OUT ROUTINE, PLACE ACS0 UP FOR
 /HALT AFTER PRINT-OUT, PRESS CONTINUE

00740	000000	ERROR	0	
00741	707704		LEM	
00742	043216		DAC RAD1	/SAVE BAD DATA
00743	203206		LAC MEMADR	
00744	043217		DAC OCADR	/SAVE FAILING ADDRESS
00745	203305		LAC ERWRD	/ERROR TABLE POINTER
00746	541104		SAD ENFRR	/LAST ADDRESS OF TABLE
00747	741000		SKP	
00750	600754		JMP .+4	
00751	201103		LAC ERTBL	/FIRST ADDRESS OF TABLE
00752	043305		DAC ERWRD	/PUT POINTER TO TOP OF TABLE
00753	600763		JMP SW2	/CHECK AC2 FOR BELL
00754	203217		LAC OCADR	/FAILING ADDRESS
00755	503277		AND K70K	/MASK 3, 4 AND 5
00756	543167		SAD LAST	/NEW ERROR FIELD IF SKIP
00757	600763		JMP .+4	/SAME FIELD AS LAST ERROR
00760	043167		DAC LAST	
00761	063305		DAC* ERWRD	/STORE FIELD# IN TABLE
00762	443305		ISZ ERWRD /	/INCREMENT POINTER
 /				
00763	760000	SW2	LAW	/PRINT INHIBIT IF = LAW
00764	543164		SAD NUPRNT	
00765	600530		JMP ERRTN /NO PRINT	
00766	750004		LAS	
00767	742010		RTL	
00770	740100		SMA	/BELL IF SKIP
00771	600775		JMP SW1	/CHECK ACS 1
00772	760207		LAW 207	/ASCII BELL
00773	102043		JMS PCHAR	/PRINT
00774	601003		JMP SW1	/CHECK ACS 0 FOR HALT
00775	750004	SW1	LAS	
00776	740010		RAL	
00777	740100		SMA	/NO SKIP = PRINT INFO
01000	601007		JMP DUFRR	/PRINT
01001	100067		JMS HALT /HALT	
01002	601007		JMP DUERR /PRINT INFO	
01003	750004	SW0	LAS	
01004	741100		SPA	/NO SKIP = HALT
01005	100067		JMS HALT	
01006	600530		JMP ERRTN /RETURN TO READ ROUTINE	
 /				
.EJECT				

```

/SETUP TO PRINT ERROR
/
v1007    203216
v1010    741200
v1011    601024
v1012    740001
v1013    741200
v1014    601024
v1015    740001
v1016    503235
v1017    740200
v1020    740001
v1021    503235
v1022    741200
v1023    600530
v1024    202715
v1025    741200
v1026    102715
v1027    203203
v1030    102043
v1031    777767
v1032    043213
v1033    102115
v1034    203217
v1035    042106
v1036    102123
v1037    777772
v1040    043213
v1041    102115
v1042    203220
v1043    042106
v1044    102123
v1045    777776
v1046    043213
v1047    102115
v1050    203216
v1051    042106
v1052    102123
v1053    777773
v1054    043213
v1055    102115
v1056    203172
v1057    042106
v1060    102123
v1061    102106
v1062    443304
v1063    601003
v1064    777700
v1065    043304
v1066    760000
v1067    043164

DOERR    LAC RAD1 /BAD DATA
          SNA
          JMP STER-6
          CMA
          SNA
          JMP STER-6
          CMA
          AND BITSUP      /MASK SUPPRESSED BITS
          SZA
          CMA
          AND RITSUP
          SNA           /NEW ERROR IF SKIP
          JMP ERRTN /ERROR IS SUPPRESSED
          LAC PHDR
          SVA
          JMS PHIR
          LAC TNUM        /ASCII TEST NUMBER
          JMS PCHAR        /PRINT TST NO.
          LAW -11
          DAC CT#4        /USED FOR SPACING COUNT
          JMS SPING       /SPACE 9
          LAC OCADR        /OCTAL ADDRESS
          DAC CRLF         /SAVE TEMPORARILY
          JMS PROCTL       /PRINT FAILING ADDRESS
          LAW -6
          DAC CT#4
          JMS SPING
          LAC GOOD1        /WHAT DATA SHOULD BE
          DAC CRLF
          JMS PROCTL       /PRINT THE GOOD
          LAW -2
          DAC CT#4
          JMS SPING       /SPACE 5
          LAC BAD1         /DATA READ
          DAC CRLF
          JMS PROCTL       /SAVE
          LAW -5
          DAC CT#4
          JMS SPING       /SPACE 5
          LAC CNTRL        /CURRENT CONTROL WORD
          DAC CRLF
          JMS PROCTL       /PRINT PATTERN CONTROL WORD
          JMS CRLF         /CR, LF
          ISZ MAXERR       /CHECK FOR MAX. PRINT-OUTS
          JMP SW1
          LAW -140
          DAC MAXERR       /CHECK ACS &
          LAW
          DAC NOPRNT        /NO MORE ERROR PRINT-OUTS
          /UNTIL RESTART FROM 100

INDY
          /EJECT

```

YCHKR PAGE 14

01070	201105	LAC PTO	
01071	043221	DAC PRNT	
01072	102050	JMS PNXT	/PRINT-OUTS INHIBITED
01073	102106	JMS CRLF	/CR,LF
01074	777766	LAW -12	/-10 DECIMAL
01075	043213	DAC CT04	
01076	760212	LAW 212	/LF
01077	102043	JMS PCHAR	
01100	443213	ISZ CT04	/10 LINE FEEDS
01101	601076	JMP .-3	
01102	601003	JMP SWA	
/			
01103	003306	ERTBL	ERWRD+1
01104	003316	ENERR	FRWRD+11
01105	003134	PTO	PTOI
/			
.EJECT			

```

/
/KEYBOARD INPUT ROUTINES
/
01106    703302      KYBRD     CAF
01107    707704      LEM
/
/TYPE TEST# AND WAIT FOR INPUT. CARRIAGE
/RETURN ONLY MEANS USE LAST PATTERN WRITTEN
/
01110    202022      TSTNO     LAC TSNX      /POINTER FOR TEST#
01111    043221      DAC PRNT
01112    102106      JMS CRLF
01113    102050      JMS PNXT
01114    101573      JMS KEYIN
01115    543267      SAI K377
01116    601106      JMP KYBRD
01117    543257      SAI K215
01120    601142      JMP ADR1
01121    777517      LAW -261
01122    343213      TAI CT^4
01123    740100      SMA
01124    601127      JMP .+3
01125    102707      JMS WUTIS
01126    601110      JMP TSTNO
01127    203213      LAC CT^4
01130    740001      CMA
01131    343245      TAI K1
01132    343262      TAI K264
01133    740100      SRA
01134    601137      JMP .+3
01135    102707      JMS WUTIS
01136    601110      JMP TSTNO
01137    760000      TSTN     LAW
01140    243213      XOR CT^4
01141    043203      DAC TNUM
/
/EJECT

```

```

/
/WAIT FOR FIRST 5 DIGIT ADDRESS TO LOOP ON
/
v1142 201666      ADR1      LAC ROTB           /POINTER FOR LAW-XX
v1143 041665      DAC ROTA
v1144 221665      LAC* ROTA
v1145 043227      DAC NRota          /LEFT SHIFT COUNTER
v1146 760000      LAK
v1147 043225      DAC ADRA
v1150 143222      DZM AURCW         /SAVES PARTIAL ADDRESS
v1151 102106      JMS CRLF
v1152 202023      LAC ADRX
v1153 043221      DAC PRNT
v1154 102050      JMS PNXT
v1155 202024      LAC AU1R           /PRINT FIRST ADR.
v1156 043224      DAC OVER
v1157 202025      LAC DON1           /C(AD1R) = ADR1
v1160 043223      DAC EXIT
v1161 101573      FADR     JMS KEYIN          /WAIT FOR INPUT
v1162 101602      JMS LEGAL          /SEE IF VALID
v1163 203213      LAC CTM4           /ASCII INPUT
v1164 503247      AND K7
v1165 043213      DAC CTM4
v1166 101637      JMS GETAD          /SHIFT LEFT TO FORM ADDRESS
v1167 601161      JMP FADR          /GET NEXT DIGIT
v1170 203222      LAC AURCW
v1171 043225      DAC ADRA
v1172 201666      DFST    LAC ROTB          /POINTER FOR LAW TABLE
v1173 541665      SAD ROTA          /NOT EQUAL = <5 CHARACTERS
v1174 601177      JMP .+3            /O.K.
v1175 041665      DAC ROTA
v1176 601613      JMP QUERY          /PRINT QUERY AND RESTART
v1177 101414      JMS PROG          /NOW SEE IF 1ST ADR. IS
                                         /IN SAME 4K AS PROGRAM
                                         /OK
v1200 601216      JMP ADR2
v1201 102106      JMS CRLF          /CR, LF
v1202 202033      LAC ADR1P
v1203 043221      DAC PRNT
v1204 102050      JMS PNXT          /PRINT "FIRST"
v1205 202035      LAC OVRLP
v1206 043221      DAC PRNT
v1207 102050      JMS PNXT          /PRINT "ADR. IS WITHIN PROGRAM"
v1208 760000      LAW
v1211 041621      DAC LOCAT+4
v1212 101615      JMS LOCAT          /TELL WHERE PROGRAM IS
v1213 201636      LAC GETAD-1
v1214 041621      DAC LOCAT+4
v1215 601142      JMP ADR1          /START OVER
                                         /EJECT

```

```

    /
    /WAIT FOR LAST 5 DIGIT ADDRESS OF BLOCK.
    /
    v1216    760000          ADR2      LAW
    v1217    043226          DAC ADRB
    v1220    143222          DEM ADRCW
    v1221    221665          LAC# ROTA
    v1222    043227          DAC NROTA
    v1223    1V2106          JMS CRLF
    v1224    202V26          LAC ADXR
    v1225    043221          DAC PRNT
    v1226    102050          JMS PNXT
    v1227    202V27          LAC AD2R
    v1230    043224          DAC OVER
    v1231    202V30          LAC DON2
    v1232    043223          DAC EXIT
    v1233    101573          LADR     JMS KEYIN
    v1234    101602          JMS LEGAL
    v1235    203213          LAC CT4
    v1236    503247          AND K7
    v1237    043213          DAC CT44
    v1240    101637          JMS GETAD
    v1241    601233          JMP LADR
    v1242    2V3222          LAC ADRCW
    v1243    043226          DAC ADRB
    v1244    2V1666          LAC RUTB
    v1245    541665          SAD RUTA
    v1246    601251          JNE .+5
    v1247    041665          DAC RUTA
    v1250    6.1613          JNE QUREY
    v1251    101414          JMS PRIG
    v1252    601270          JMP STLP
    v1253    102106          JMS CRLF
    v1254    2V2034          LAC ADP2P
    v1255    043221          DAC PRNT
    v1256    102050          JMS PNXT
    v1257    2V2135          LAC OVRLP
    v1260    043221          DAC PRNT
    v1261    102050          JMS PNXT
    v1262    760000          LAC
    v1263    041621          DAC LOCAT+4
    v1264    101615          JMS LOCAT
    v1265    201636          LAC GETAD-1
    v1266    041621          DAC LOCAT+4
    v1267    601216          JMP ADR2
    /
    .EJECT

```

/CHK9L - TAPE 2

/

/SETUP ADDRESSES AND PATTERNS BEFORE LOOPING.

/

v1270	760000	STLP	LAW	
v1271	543225		SAD ADR	/A LAW = NO 1ST ADDRESS
v1272	741000		SKP	
v1273	601302		JMP CKLST	
v1274	543226		SAD ADRB	/A LAW = NO LAST ADDRESS
v1275	600215		JMP RTN1	/RESTART PROGRAM
v1276	203226		LAC ADXB	
v1277	043225		DAC ADXA	/ONLY ONE SELECTED
v1300	043230		DAC LTST	/LAST OF BLOCK
v1301	601344		JMP SIMU	
v1302	543226	CKLST	SAD ADRB	/A LAW = NO LAST ADDRESS
v1303	741000		SKP	
v1304	601311		JMP C8OTH	
v1305	203225		LAC ADXA	/ONLY 1 ADDRESS WANTED
v1306	043226		DAC ADXB	
v1307	043230		DAC LTST	/LAST OF BLOCK
v1310	601344		JMP SIMU	
v1311	203225	C8OTH	LAC ADXA	/FIRST ADR.
v1312	503277		AND K7:K	/MASK BITS 3,4 AND 5
v1313	043213		DAC CT:4	/SAVE
v1314	203226		LAC ADXB	/LAST ADR.
v1315	503277		AND K7:K	/MASK 3,4 AND 5
v1316	543213		SAD CT:4	/BOTH MUST = SAME 4K
v1317	601322		JMP ,+3	/OK
v1320	102707		JMS WOTIS	/PRINT QUERY
v1321	601142		JMP ADY1	/START OVER
v1322	203225		LAC ADXA	/FIRST ADDRESS
v1323	740001		CMA	
v1324	343245		TAD K1	/2'S COMPLEMENT
v1325	343226		TAD ADXB	/SUBTRACT LAST ADDRESS
v1326	740100		SMA	/FIRST IS > LAST IF NEG.
v1327	601340		JMP SIMU-4	/LEAVE AS IS
v1330	203226		LAC ADXB	
v1331	042106		DAC CRLF	
v1332	203225		LAC ADXA	
v1333	043226		DAC ADXB	/FIRST IS NOW LAST
v1334	043230		DAC LTST	
v1335	202106		LAC CRLF	
v1336	043225		DAC ADXA	/LAST IS NOW FIRST
v1337	601344		JMP ,+5	
v1340	203226		LAC ADXB	
v1341	043230		DAC LTST	
v1342	203225		LAC ADXA	
v1343	043206		DAC MEMADR	
.EJECT				

01344	760261	SIMU	LAW 261	
01345	543203		SAD TNUM	/TEST 1 IF EQUAL
01346	001360	JMP SIM1		
01347	760262		LAW 262	
01350	543203		SAD TNUM	/TEST 2 IF EQUAL
01351	601372		JMP SIM2	
01352	760263		LAW 263	
01353	543203		SAD TNUM	/TEST 3 IF EQUAL
01354	601374		JMP SIM3	
01355	760264		LAW 264	
01356	543203		SAD TNUM	/TEST 4 IF EQUAL
01357	601376		JMP SIM4	
<hr/>				
01360	203173	SIM1	LAC PCWA	
01361	043172		DAC CNTRL	
01362	102106		JMS CRLF	
01363	203225		LAC ADRA	
01364	503277		AND K21K	
01365	043206		DAC MEMADR	
01366	770000		LAK -10000	
01367	043210		DAC CT4K	
01370	100415		JMS GENPAT	/WRITE PATTERN #1
01371	601411		JMP STSCP	
<hr/>				
01372	203174	SIM2	LAC PCWB	/WRITE PATTERN #2
01373	601361		JMP SI+1+1	
<hr/>				
01374	203175	SIM3	LAC PCNC	/PATTERN #3
01375	601361		JMP SI+1+1	
<hr/>				
01376	203176	SIM4	LAC PCND	/PATTERN #4
01377	601361		JMP SI+1+1	
<hr/>				
01400	707702	SCP1	FEN	/SYNC
01401	223206		LAI* MEMADR	/READ
01402	740001		CMA	/COMPLEMENT
01403	063206		DAC* MEMADR	/WRITE
01404	203206		LAC MEMADR	
01405	543230		SAD LTST	/CHECK FOR END OF BLOCK
01406	601411		JMP .+3	
01407	443206		TSZ MEMADR	/ADDRESS+1
01410	601400	STSCP	JMP SCP1	
01411	203225		LAC ADRA	/STARTING ADDRESS
01412	043206		DAC MEMADR	
01413	601400		JMP SCP1	
<hr/>				
.EJECT				

```

/
/ CHECK WANTED ADDRESS AND PROGRAM AREA
/
v1414    000000      PROG      0
v1415    100663      JMS WHFRE
v1416    042106      DAC CRLF      /SAVE
v1417    760000      LAW
v1420    543222      SAD ADRCW
v1421    621414      JMP* PROG
v1422    203222      LAC ADRCW
v1423    503277      AND K7WK
v1424    542106      SAD CRLF      /C(CRLF) = CURRENT 4K BANK
v1425    741000      SKP      /EQUAL
v1426    621414      JMP* PROG
v1427    441414      ISZ PRG      /EXIT
v1430    621414      JMP* PROG      /RETURN+1
v1430    621414      /EXIT

/
/ BIT SUPPRESSION INPUT ROUTINE, TYPE A
/ CARRIAGE RETURN TO RESUME TESTING ALL BITS.
/ TO SUPPRESS, TYPE THE DECIMAL BIT POSITION (S)
/ SEPARATING EACH WITH A COMMA, TERMINATE WITH
/ A C.R. PRESS RUBOUT TO RESTART THE LINE IN
/ CASE OF TYPING ERROR, PRESS RUBOUT, THEN C.R.
/ TO RESUME TFSTING ALL BIT POSITIONS.
/
v1431    000000      SUPBIT    0
v1432    143234      DEM SCW      /SUPPRESSION CONTROL WORD
v1433    707704      LEM
v1434    703302      CAF
v1435    202031      LAC SUPX      /POINTER FOR SUPPRESS
v1436    043221      DAC PRNT
v1437    202032      LAC SUPXA      /C(SUPXA) = SUPBIT+1
v1440    043224      DAC OVER
v1441    102106      JMS CRLF      /CR,LF
v1442    102050      JMS PNXT      /PRINT "SUPPRESS"
v1443    101573      JMS KEYIN      /FIRST CHAR.
v1444    543257      SAD K215      /C.R. = NO NEW INPUTS
v1445    621431      JMP* SUPBIT
v1446    601452      JMP .+4      /EXIT

/
v1447    101573      AGAIN     JMS KEYIN      /WAIT FOR INPUT
v1450    543257      SAD K215      /CHECK FOR C.R.
v1451    601516      JMP FUT      /DONE SELECTING
v1452    543267      SAD K377      /CHECK FOR RUB-OUT
v1453    601432      JMP SUPBIT+1
v1454    543260      SAD K254      /CHECK FOR COMMA
v1455    601447      JMP AGAIN      /WAIT FOR NEXT BIT POS.
v1456    101546      JMS NUMB      /DETERMINE INPUT NUMBER
v1457    601613      JMP QUERY      /NOT VALID RESTART

/
.EJECT

```

01460	741200	SNA	/CHECK FOR 0
01461	601542	JMP ZERO	/POSITION 0
01462	043231	DAC TTYW	/SAVE DIGIT
01463	101573	JMS KEYIN	/WAIT FOR SECOND DIGIT
01464	543260	SAD K254	/CHECK FOR COMMA
01465	601524	JMP EOM	/2 DIGIT POSITION
01466	543257	SAD K215	/CHECK FOR C.R.
01467	601530	JMP EOTA	/DONE
01470	543267	SAD K377	/RUB-OUT IF NO SKIP
01471	601432	JMP SUPBIT+1	/START OVER
01472	101546	JMS NUMB	/DETERMINE NUMBER
01473	601613	JMP QUREY	/NOT VALID, RESTART
01474	043232	DAC TTYX	/SAVE NUMBER
01475	203231	LAC TTYW	/PREVIOUS DIGIT
01476	744010	RCL; RCL; RCL	
01477	744010		
01500	744010		
01501	243232	XOR TTYX	/COMBINE DIGITS
01502	740001	CMA	/1'S COMPLEMENT
01503	043233	DAC TTYY	/SAVE
01504	777777	LAC -1	
01505	343233	TAD TTYY	/SUBTRACT 1
01506	043233	DAC TTYY	
01507	203301	LAC K400K	/400000
01510	744020	RCL	
01511	443233	TSZ TTYY	/SHIFT COUNT
01512	601510	JMP .-2	
01513	243234	XOR SCW	/INSERT IN CONTROL WORD
01514	043234	DAC SCW	
01515	601447	JMP AGAIN	/WAIT FOR NEXT BIT POSITION

.EJECT

YCHKR PAGE 22

01516	203234	EOT	LAC SCW	/SELECTION COMPLETED
01517	740001		CMA	
01520	043235		DAC BITSUP	
01521	143234		DEM SCW	
01522	102106		JMS CRLF	/CR,LF
01523	621431		JMP* SUPBIT	/EXIT
/				
01524	203231	EOM	LAC TTYW	/SINGLE DIGIT
01525	740001		CMA	
01526	343245		TAD K1	
01527	601506		JMP ROTOR	
/				
01530	203231	EOTA	LAC TTYW	/INPUT DIGIT
01531	740001		CMA	
01532	343245		TAD K1	/2'S COMPLEMENT
01533	043233		DAC TTYY	
01534	203301		LAC K400K	/400000
01535	744020		RCR	
01536	443233		ISZ TTYY	/SHIFT COUNTER
01537	601535		JMP -2	
01540	243234		XOR SCW	
01541	601517		JMP EUT+1	/EXIT
/				
01542	203234	ZERO	LAC SCW	
01543	243301		XOR K400K	/400000
01544	043234		DAC SCW	
01545	601447		JMP AGAIN	/WAIT FOR NEXT
/				
EJECT				

9CHKR PAGE 23

		NUMB		
01546	000000		I	
01547	203213		LAC CT#4	/ASCII INPUT
01550	503266		AND K370	
01551	543261		SALI K260	
01552	741000		SKP	
01553	601560		JMP .+5	
01554	441546		ISZ NUMB	/CHECK FOR A 270 OR 271
01555	203213		LAC CT#4	/RETURN+1
01556	503247		ANI K7	
01557	621546		JMP* NUMB	/EXIT
01560	543263		SALI K270	/= 8 OR 9 IF EQUAL
01561	741000		SKP	
01562	621546		JMP* NUMB	/INVALID
01563	203213		LAC CT#4	/ASCII INPUT
01564	503245		ANI K1	/= A 8 IF BIT 17 = 0
01565	740200		SZA	
01566	601571		JMP .+3	/A 9
01567	777777		LAW -11	/SHIFT COUNT OF 8
01570	601506		JMP ROTOR	
01571	777767		LAW -11	/SHIFT COUNT OF 9
01572	601506		JMP ROTOR	
			/	
			EJECT	

```
/  
/CHARACTER INPUT ROUTINE  
/  
v1573    0000000      KEYIN      0  
v1574    700312       KRB          /INITIALIZE  
v1575    700301       KSF          /WAIT FOR INPUT  
v1576    601575       JMP .-1  
v1577    700312       KRB          /READ BUFFER  
v1600    043213       DAC CT04     /SAVE  
v1601    621573       JMP# KEYIN  
/  
/  
/CHECK VALIDITY OF INPUT CHARACTER  
/  
v1602    0000000      LEGAL      0  
v1603    203213       LAC CT04     /ASCII INPUT  
v1604    543267       SAD K377     /IS IT A RUBOUT  
v1605    601106       JMP KYRRD    /START OVER  
v1606    543257       SAD K215     /CHECK FOR C.R.  
v1607    623223       JMP# EXIT    /LINE TERMINATED  
v1610    503266       AND K370     /  
v1611    543261       SAD K260     /SHOULD EQUAL 260  
v1612    621602       JMP# LEGAL   /O.K.  
v1613    102707       JMS WOTIS    /PRINT QUESTION MARK  
v1614    623224       JMP# OVER    /START LINE OVER  
/  
.EJECT
```

```

/PRINT AREA CONTAINING PROGRAM
/
01615    0000000      LOCAT     0
01616    750004        LAS
01617    503254        AND K100
01620    740200        SZA
01621    621615        JMP* LOCAT
01622    102106        JMS CRLF
01623    202036        LAC PISIN
01624    043221        DAC PRNT
01625    102050        JMS PNXT
01626    100063        JMS WHERE
01627    744010        RTL;      RTL;
01630    742010
01631    742010
01632    742010
01633    343261        TAB K200
01634    102043        JMS PCHAR
01635    102106        JMS CRLF
01636    621615        JMP* LOCAT
/
/
/GENERATE 5 DIGIT ADDRESSES FROM KEYBOARD INPUT
/
01637    0000000      GETAD     0
01640    707704        LEM
01641    221665        LAC* ROTA
01642    043227        DAC NROTA
01643    203213        LAC CT#4
01644    443227        ISY NROTA
01645    601663        JMP GOLEFT
01646    243222        XOR ADRCW
01647    043222        DAC ADRCW
01650    777777        LAW -1
01651    561665        SAD* ROTA
01652    601655        JMP .+3
01653    441665        ISZ ROTA
01654    621637        JMP* GETAD
01655    201666        LAC RUTB
01656    041665        DAC RUTA
01657    221665        LAC* RUTA
01660    043227        DAC NRUTA
01661    441637        ISZ GETAD
01662    621637        JMP* GETAD
/
.EJECT

```

9CHKR PAGE 26

V1663	744010	GOLEFT	RCL	
V1664	601644		JMP CNROT	
/				
V1665	001667	ROTA	.+2	/=LOC. OF FIRST LAW
V1666	001667	ROTB	.+1	/LOC. OF FIRST LAW
V1667	777763	LAW	-15	/ROTATE 12 LEFT FOR 1ST DIGIT
V1670	777766	LAW	-12	/9 LEFT FOR 2ND
V1671	777771	LAW	-7	/6 LEFT FOR 3RD
V1672	777774	LAW	-4	/3 LEFT FOR 4TH
V1673	777777	LAW	-1	/NONE FOR 5TH
/				
/ROUTINE TO ACCEPT TEST LIMITS FROM KEYBOARD INPUT				
/				
V1674	000000	SLMTS	A	
V1675	707704		LEM	
V1676	201636	LAC	GETAD-1	
V1677	041621	DAC	LUCAT+4	/RESTORE JMP*
V1700	102106	JMS	CRLF	
V1701	202037	LAC	TLMX	/TEST LIMITS POINTER
V1702	043221	DAC	PRNT	
V1703	102050	JMS	PNXT	/PRINT "TEST LIMITS"
V1704	102106	JMS	CRLF	/CR,LF
V1705	202040	LAC	SLMX	/C (SLMX)=SLMTS+1
V1706	043224	DAC	OVER	
V1707	202041	LAC	DON3	/RETURN ADDRESS=CHEVR
V1710	043223	DAC	EXIT	
V1711	101573	JMS	KEYIN	/WAIT FOR INPUT
V1712	543267	SAD	K377	
V1713	601675	JMP	SLMTS+1	
V1714	101602	JMS	LEGAL	/SEE IF VALID
V1715	203213	LAC	CTM4	/ASCII INPUT
V1716	503247	AND	K7	/MASK 15,16 AND 17
V1717	744020	RCK	; RTR	
V1720	742420			
V1721	742020			
V1722	742020			
/				

.EJECT

YCHKR PAGE 27

01723	043214	DAC FIRST1	/FIRST TO TEST
01724	101573	JMS KEYIN	/WAIT FOR COMMA
01725	543260	SAD K254	
01726	741000	SKP	
01727	601613	JMP QUERY	/PRINT QUERY, AND RESTART
01730	101573	JMS KEYIN	/WAIT FOR LAST
01731	543267	SAD K377	
01732	601675	JMP SLMTS+1	
01733	1W1602	JMS LEGAL	/SEE IF VALID
01734	203213	LAC CT-4	/ASCII INPUT
01735	503247	AND K7	
01736	744020	RCR; RTR; RTR; RTR	
01737	742020		
01740	742020		
01741	742020		
01742	043215	DAC LAST1	/LAST TO TEST
01743	777777	LAK -1	
01744	043210	DAC CT4K	
01745	443210	ISZ CT4K	/NO 2ND DIGIT IF NO SKIP
01746	601613	JMP QUERY	/PRINT QUERY AND RESTART
01747	203214	LAC FIRST1	/FIRST FIELD
01750	740001	CMA	
01751	343245	TAD K1	
01752	343215	TAD LAST1	/FIRST IS >LAST IF NEG,
01753	740100	SRA	
01754	601763	JMP OKAS	/FIRST IS LOWEST ORDER

.EJECT

Page Missing From Original Document

Page Missing From Original Document

```

/
/PRINT ROUTINES FOR MESSAGES
/
/PRINT ONE CHARACTER AND EXIT
/
02043 0000000 PCHAR   0
02044 700406    TLS
02045 700401    TSF
02046 602045    JMP .-1
02047 622043    JMP* PCHAR
/
/PRINT A STRING AND EXIT.
/
02050 0000000 PNXT    0
02051 777775    LAW -3
02052 043213    DAC CT4
02053 443221    ISZ PRNT
02054 223221    LAC# PRNT
02055 741200    SNA
02056 622050    JMP* PNXT
02057 042106    MASK    DAC CRLF
02060 503253    AND K77
02061 543253    SAD K77
02062 602073    JMP CK3
02063 040707    DAC GENRAN
02064 777740    LAW -4
02065 340707    TAD GENRAN
02066 740100    SMA
02067 602103    JMP CRLF-3
02070 242064    XOH .-4
02071 243264    XOH K3#0
02072 102043    JMS PCHAR
02073 443213    ISY CT4
02074 741000    SKP
02075 602051    JMP PNXT+1
02076 202106    LAC CRLF
02077 742020    RTR; RTR; RTR
02100 742020
02101 742020
02102 602057    JMP MASK
02103 200707    LAC GENRAN
02104 343255    TAD K2#0
02105 602072    JMP CK3-1
/
/CARRIAGE RETURN, LINE FEED
/
02106 0000000 CRLF    0
02107 760215    LAW 21#
02110 102043    JMS PCHAR
02111 542113    SAD .+2
02112 622106    JMP* CRLF
02113 760212    LAW 21#
02114 602110    JMP CRLF+2
/
.EJECT

```

```
/PRINT SPACES
/
02115 000000 SPING 0
02116 760240 LAW 240 /ASCII SPACE
02117 102043 JMS PCHAR
02120 443213 ISZ CT#4 /COUNTER
02121 602116 JMP SPING+1
02122 622115 JMP* SPING /EXIT

/PRINT SIX DIGIT OCTAL NUMBERS
/
02123 000000 PROCTL N
02124 777772 LAW -6
02125 043213 DAC CT#4 /DIGIT COUNTER
02126 202106 LAC CRLF /OCTAL NUMBER
02127 744010 POSITN RCL; RTL
02130 742010
02131 042106 DAC CRLF
02132 740010 RAL
02133 503247 AND K7
02134 343261 TAD K260
02135 102043 JMS PCHAR /PRINT
02136 443213 ISZ CT#4
02137 602126 JMP POSITN-1 /POSITION NEXT DIGIT
02140 622123 JMP* PROCTL /EXIT

/EJECT
```

```

/
/CHKR9L - TAPE 3
/
/ROUTINE TO DETERMINE FIELD FOR RELOCATION
/
02141    707704      CMOVE    LEM
02142    201103      LAC      ERTBL
02143    043305      DAC      ERWRD
02144    203215      LAC      LAST1
02145    543214      SAD      FIRST1
02146    600215      JMP      RTN1
02147    203165      LAC      FLAGS
02150    741100      SPA
02151    600215      JMP      RTN1
02152    7400020     RAR
02153    741400      SEL
02154    602241      JMP      NXTMV
02155    443165      ISZ      FLAGS
02156    203215      LAC      LAST1
02157    043166      DAC      INSFLD
02160    770000      LAW      ~10000
02161    343166      TAD      INSFLD
02162    043236      DAC      NXLOC
02163    100663      JMS      WHERE
02164    543166      SAD      INSFLD
02165    602224      JMP      SUB1
/
.EJECT

```

```

        /NOW CHECK FOR ERROR RECORDED IN NEW FIELD
        /
02166    760000      CKERR     LAW
02167    563305      SAD*      ERWRD      /NO ERRORS IF = LAW
02170    602201      JMP       STMV       /INITIALIZE MOVE
02171    223305      LAC*      ERWRD
02172    543166      SAD      INSFLD      /ERROR IN FIELD IF EQUAL
02173    602213      JMP       EQUAL
02174    443305      ISZ       ERWRD      /POINTER + 1
02175    203305      LAC       ERWRD
02176    541104      SAD       ENERR      /END OF TABLE IF EQUAL
02177    741000      SKP
02200    602171      JMP       CKERR+3
        /
02201    201103      STMV      LAC       ERTBL
02202    043305      DAC       ERWRD      /RESTORE POINTER
02203    203166      LAC       INSFLD      /NEW FIELD
02204    043240      DAC       DESTN
02205    100663      JMS       WHERE
02206    043237      DAC       SOURCE
02207    543240      SAD       DESTN
02210    600215      JMP       RTN1      /NEW AND CURRENT ARE EQUAL
02211    203240      LAC       DESTN
02212    602460      JMP       WHWAY+4      /MOVE PROGRAM
        /
        /ERROR IN NEW FIELD. TRY NEXT LOWER
        /
02213    543214      EQUAL     SAD       FIRST1      /DON'T TRY NEXT IF EQUAL
02214    602236      JMP       DNMVE
02215    741200      SNA
02216    602222      JMP       .+4
02217    770000      LAC       -10000      /YES
02220    343166      TAI       INSFLD      /-4K
02221    043236      DAC       NXLOC      /SUBTRACT 4K FROM NEW FIELD
02222    201103      LAC       ERTBL      /NEXT NEW FIELD
02223    043305      DAC       ERWRD      /RESTORE POINTER
        /
02224    203236      SUB1     LAC       NXLOC      /NEXT NEW FIELD
02225    543166      SAD       INSFLD      /IS IT = CURRENT NEW FIELD
02226    602213      JMP       EQUAL      /TRY NEXT LOWER
02227    043166      DAC       INSFLD      /NEW NEW FIELD
02230    543214      SAD       FIRST1      /DOES IT = LOWEST FIELD
02231    602166      JMP       CKERR      /CHECK FOR ERROR
02232    770000      LAC       -10000      /SUBTRACT 4K
02233    343166      TAI       INSFLD      /NEW FIELD FOR NEXT PASS
02234    043236      DAC       NXLOC
02235    602166      JMP       CKERR
        /
02236    201103      DNMVE     LAC       ERTBL
02237    043305      DAC       ERWRD      /RESTORE POINTER
02240    600215      JMP       RTN1      /START OVER
        /
        .EJECT

```

```

/ROUTINE TO DETERMINE PROGRAM DEST'N AFTER MAKING ONE MOVE
/
02241 100663 NXTMV JMS WHERE           /WHERE IS PROGRAM NOW
02242 043237 DAC SOURCE
02243 760000 CKNXT LAW
02244 563305 SAD* ERWRD
02245 602260 JMP STNXT
02246 201103 LAC ERTBL
02247 043305 DAC FRWRD
02248 223305 LAC* ERWRD
02249 543236 SAD NXLOC
02250 602300 JMP SUB2
02251 443305 TS? ERWRD
02252 203305 LAC ERWRD
02253 541104 SAD ENERR
02254 741W00 SKP
02255 6W2250 JMP CKNXT+5
/
02260 201103 STNXT LAC ERTBL
02261 043305 DAC ERWRD
02262 2W3236 LAC NXLOC
02263 543166 SAD INSLFD
02264 602267 JMP .+3
02265 543214 SAD FIRST1
02266 6W2315 JMP MVBK
02267 543214 SAD FIRST1
02270 6W2311 JMP NXTHI
02271 043166 DAC INSLFD
02272 77W000 LAC -10000
02273 343166 TAD INSLFD
02274 043236 DAC NXLOC
02275 203166 LAC INSLFD
02276 043240 DAC DESTN
02277 602460 JMP WHWAY+4
/
02300 203236 SUR2 LAC NXLOC
02301 543214 SAD FIRST1
02302 6W2236 JMP DMNVE
02303 77W000 LAC -10000
02304 343236 TAD NXLOC
02305 043236 DAC NXLOC
02306 543166 SAD INSLFD
02307 602301 JMP SUR2+1
02310 6W2246 JMP CKNXT+3
/
.EJECT

```

YCHKR PAGE 35

02311	203215	NXTHI	LAC	LAST1	/LAST TO TEST
02312	503277		AND	K70K	
02313	043236		DAC	NXLOC	/LAST = NEXT FIELD
02314	602246		JMP	CKNXT+3	/CHECK FOR E ROR
		/			
02315	100663	MVRK	JMS	WHERE	
02316	043237		DAC	SOURCE	
02317	203236		LAC	NXLOC	
02320	043166		DAC	INSLFD	
02321	043240		DAC	DESTN	
02322	143165		D2M	FLAGS	
02323	602460		JMP	WHWAY+4	
		/			
/ROUTINE TO RELOCATE 4K FIELDS					
		/			
02324	707702	MOVE	EEM		
02325	142675		D2M	LOCER	
02326	770000		LAW	-10000	
02327	043210		DAC	CT4K	
02330	203240		LAC	DESTN	
02331	043241		DAC	MOVED	/MOVE FROM HERE
02332	203237		LAC	SOURCE	
02333	043242		DAC	MOVES	/TO HERE
02334	223242	MVRTN	LAC*	MOVES	/GET ONE FROM THIS FIELD
02335	043203		DAC	TNUM	/SAVF
02336	503303		AND	K700K	/MASK OP CODE BITS
02337	243303		XOR	K700K	
02340	741200		SNA		/OP CODE IS 7X IF 0
02341	602364		JMP	OPRAT	/DON'T MODIFY IT
02342	203203		LAC	TNUM	
02343	243273		XOR	K10K	/MODIFY BY 1 000
02344	102544		JMS	RT19L	
02345	063241	MVOPR	DAC*	MOVED	/PLACE IN NEW 4K FIELD
02346	223241		LAC*	MOVED	
02347	243273		XOR	K10K	/RESTORE BIT 5
02350	563242		SAD*	MOVES	/COMPARE
02351	602354		JMP	.+3	/OK
02352	243273		XOR	K10K	/PRINT ERROR
02353	102644		JMS	MVERR	
02354	443210		ISZ	CT4K	
02355	203203		LAC	TNUM	
02356	542757		SAD	DLMT	/C (DLMT) = 752525
02357	602374		JMP	MVCST	/MOVF CONSTANT TABLES
02360	443242		ISZ	MOVES	
02361	443241		ISZ	MOVED	/SHOULD NEVER SKIP
02362	602334		JMP	MVRTN	
02363	740040	E2	HLT		/SHOULD NEVER GET HERE
		/			
		.	EJECT		

02364	203203	OPRAT	LAC	TNUM	/OPERATE INST.
02365	102544		JMS	RT19L	/ROTATE 19 LEFT
02366	063241		DAC*	MOVED	/DEPOSIT NEW FIELD
02367	223241		LAC*	MOVED	
02370	563242		SAD*	MOVES	/COMPARE
02371	602354		JMP	E2-7	/OK
02372	102644		JMS	MVERR	/PRINT ERROR
02373	602354		JMP	E2-7	
<hr/>					
02374	443242	MVCST	ISZ	MOVES	
02375	443241		ISZ	MOVED	
02376	223242		LAC*	MOVES	
02377	142544		JMS	RT19L	
02400	063241		DAC*	MOVED	
02401	223241		LAC*	MOVED	
02402	563242		SAD*	MOVES	
02403	741000		SKP		
02404	102644		JMS	MVERR	
02405	443210		ISZ	CT4K	/MOVFD ALL 4K WHEN SKIP
02406	602374		JMP	MVCST	
02407	102620		JMS	ENOT	/MAKE CERTAIN IT MOVED
<hr/>					
02410	203243		LAC	BGNLO	/JMP RTN1 IN LO 4K
02411	503272		AND	K7777	
02412	243240		XOR	DESTN	/PUT FIELD # ON RTN1 ADR.
02413	043243		DAC	BGNLO	
02414	203244		LAC	BGNHI	/JMP RTN1 IN HI 4K
02415	503272		AND	K7777	/7777
02416	243240		XOR	DESTN	/PUT FIELD# ON RTN1 ADR.
02417	043244		DAC	BGNHI	
02420	202420		LAC	.	
02421	503273		AND	K10K	
02422	740200		SZA		/TEST BIT 5
02423	623243		JMP*	RGNL0	/HI 4K TO LO 4K
02424	623244		JMP*	RGNHI	/LO 4K TO HI 4K
<hr/>					
.EJECT					

```

    /ROUTINE TO FORCE MOVE THE PROGRAM. DESTINATION
    /FIELD# MUST BE TYPED IN BY THE OPERATOR (0-7 OCTAL).
    /
    02425 203301 FCDMV LAC K400K
    02426 740001 CMA AND FLAGS
    02427 503165 XOR K400K /SET BIT 0 FOR FCDMV FLAG
    02430 243301 DAC FLAGS
    02431 043165 LAC ERTBL
    02432 201103 DAC ERWRD /RESTORE TABLE POINTER
    02433 043305 LEM
    02434 707704 JMS GOTO /PRINT GO TO FIELD
    02435 102557

    /
    /CHECK FOR ERROR IN NEW FIELD
    /
    02436 760000 CKFCD LAW
    02437 563305 SAD* ERWRD /NO ERRORS IF 1ST = LAW
    02440 602454 JMP WHWAY /SEE WHERE TO GO
    02441 223305 LAC* ERWRD
    02442 543240 SAD DESTN /DOES ERROR = NEW FIELD
    02443 602451 JMP XPRS /YES, PRINT MESSAGE
    02444 443305 ISZ ERWRD /POINTER+1
    02445 203305 LAC ERWRD
    02446 541104 SAD ENERR /SEE IF END OF TABLE
    02447 602452 JMP .+3 /DONE AND NO ERRORS
    02450 602441 JMP CKFCD+3 /PRINT ERROR IN SELECTED 4K
    02451 102535 XPRT JMS PRSEL

    /
    02452 201103 LAC ERTBL
    02453 043305 DAC ERWRD
    02454 203240 WHWAY LAC DESTN /NEW FIELD
    02455 543237 SAD SOURCE /DOES IT EQUAL PRESENT
    02456 600215 JMP RTN1
    02457 043166 DAC INSFLD /NEW CURRENT FIELD
    02460 503273 AND K10K /MASK BIT 5
    02461 740200 SZA /TEST BIT 5
    02462 602470 JMP .+6 /DEST'N IS TO HI 4K
    02463 100663 JMS WHFRE /WHERE IS PROGRAM
    02464 503273 AND K10K /MASK BIT 5
    02465 740200 SZA /TEST BIT 5
    02466 602324 JMP MOVE /MOVE FROM HI 4K TO LO 4K
    02467 602474 JMP MSAME /MOVE LO 4K TO LO 4K
    02470 100663 JMS WHERE /WHERE IS PROGRAM
    02471 503273 AND K10K /MASK BIT 5
    02472 741200 SNA /TEST BIT 5
    02473 602324 JMP MOVE /MOVE LO 4K TO HI 4K
    /
    .EJECT

```

.EJECT

```

/
/PRINT ERROR IN SELECTED 4K
02535    000000   ?RSEL    0
02536    102106   JMS      CRLF    /CR,LF
02537    202756   LAC      ERSEL    /TEXT POINTER
02540    043221   DAC      PRNT    /PRINT
02541    102050   JMS      PNXT    /PRINT
02542    102106   JMS      CRLF    /WAIT FOR ANOTHER CHOICE
02543    602425   JMP      FCDMV

/
/ROTATE INSTRUCTION 19 LEFT BEFORE MOVING
/
RT19L    0
02544    000000   CLL      TNUM    /LINK = 0
02545    744000   DAC      TNUM    /SAVE
02546    043203   LAW     -11     /-9 DECIMAL
02547    777767   DAC      CT04    /SHIFT COUNT
02550    043213   LAC      TNUM    /INSTRUCTION
02551    203203   RAL
02552    740010   RTL
02553    742010   ISX     CT04
02554    443213   JMP     .-?
02555    602553   JMP*    RT19L
02556    622544

/
.EJECT

```


YCHKR PAGE 41

02620	000000	ENOT	0	
02621	202675		LAC	LOCER
02622	741200		SNA	
02623	622620		JMP*	ENOT
		/		/NO ERRORS IF 0 /ENTER NEW FIELD
02624	707704		LEM	
02625	142675		DZM	LOCER
02626	102106		JMS	CRLF
02627	202705		LAC	NERN
02630	043221		DAC	PRNT
02631	102050		JMS	PNXT
02632	102106		JMS	CRLF
02633	203165		LAC	FLAGS
02634	741100		SPA	
02635	602425		JMP	FCDMV
02636	602260		JMP	STNXT
		/		/ACS 0 A 1 = FORCED MOVE /WAIT FOR ANOTHER CHOICE /TRY NEXT FIELD LOWER
02637	203301	CFLG	LAC	K400K
02640	740001		CMA	
02641	503165		AND	FLAGS
02642	043165		DAC	FLAGS
02643	600215		JMP	RTN1
		/		/CLEAR THE FORCED MOVE FLAG /START OVER

.EJECT

02644	000000					
02645	707704	MVERR	0			
02646	043216		LEM			
02647	203241		DAC	RAD1	/SAVE INCORRECT INSTRUCTION	
02650	043217		LAC	MOVED	/FIELD AND A DRESS	
02651	223242		DAC	OCADR	/SAVE	
02652	043220		LAC*	MOVES	/CORRECT INSTRUCTION	
02653	202715		DAC	GOOD1	/SAVE	
02654	741200		LAC	PHDR		
02655	102715		SNA			
02656	202675		JMS	PHDR		
02657	741200		LAC	LOCER		
02660	102675		SNA		/DON'T PRINT IF 1	
02661	202706		JMS	LOCER	/PRINT PROGRAM RELOCATION ERROR	
02662	041055		LAC	JMP3	/JMP LOCER-3	
02663	102106		DAC	INDY		
02664	777766		JMS	CRLF		
02665	601032		LAW	-12	/-10 DECIMAL	
			JMP	STER	/PRINT INFO	
02666	201033	/	LAC	STER+1	/EQUALS JMS SPING	
02667	041055		DAC	INDY		
02670	750004		LAS			
02671	741100		SPA			
02672	100667		JMS	HALT		
02673	707702		EEM			
02674	622644		JMP*	MVERR	/EXIT	
02675	000000	/	0			
02676	102106	LOCER	JMS	CRLF		
02677	202704		LAC	RELUC	/CR,LF	
02700	043221		DAC	PRNT	/TEXT PUNTER	
02701	102750		JMS	PNXT		
02702	102106		JMS	CRLF	/PRINT PROGRAM RELOCATION ERROR	
02703	622675		JMP*	LOCER	/CR,LF X 2	
					/EXIT AND PRINT THE ERROR	
02704	003057	/	RELOC	PRGR		
02705	003071	NOTIS	NERN	NUMO		
02706	602666		JMP3	JMP	LOCER-7	/RETURN FROM PRINT ERROR
02707	000000	/	0			
02710	102106	NOTIS	JMS	CRLF		
02711	760277		LAW	277	/CR,LF	
02712	102043		JMS	PCHAR	/QUERY MARK	
02713	102106		JMS	CRLF	/PRINT	
02714	622707		JMP*	NOTIS	/CR,LF	
					/EXIT	
					.EJECT	

```

/HEADER ROUTINE
/
02715 000000 PHDR 0
02716 102106 JMS CRLF /CR,LF
02717 202751 LAC TSTX /POINTER FOR "TEST"
02720 043221 DAC PRNT
02721 102050 JMS PNXT /PRINT TEST
02722 102744 JMS CLMN /SPACE 5
02723 202752 LAC ADRXA //OCTAL ADR."
02724 043221 DAC PRNT
02725 102050 JMS PNXT
02726 102744 JMS CLMN /SPACE 5
02727 202753 LAC GUATX //GOOD"
02730 043221 DAC PRNT
02731 102050 JMS PNXT
02732 102744 JMS CLMN /SPACE 5
02733 202754 LAC BOATX //BAD"
02734 043221 DAC PRNT
02735 102050 JMS PNXT
02736 102744 JMS CLMN /SPACE 5
02737 202755 LAC PCWX //PAT.CONTROL WORD"
02740 043221 DAC PRNT
02741 102050 JMS PNXT
02742 102106 JMS CRLF /CR, LF
02743 622715 JMP* PHDR /DONE

/
CLMN
02744 000000
02745 777773 LAW -5
02746 043213 DAC CTM4
02747 102115 JMS SPING /SPACE
02750 622744 JMP* CLMN

/
TSTX
02751 002760 TST /POINTER FOR "TEST"
02752 002764 ADRX A //OCTAL ADR."
02753 002771 GDATA GDATA //GOOD"
02754 002775 RDATA RDATA //BAD"
02755 003000 PCWX PCWR //PAT.CONTROL WORD
02756 003010 ERSEL SLTER

/
DLMT
02757 752525 752525

.EJECT

```

```

/
/CONSTANTS FOR PRINT ROUTINE TEXTS. PACKED
/3 CHARACTERS PER WORD.
/
/*TEST*/
TST      .
        230524; 777724; 0

/
ADR      .
        240317; 401401; 220401; 0

/
GDAT     .
        171707; 777704; 0

/
BDAT     .
        140102; 0

/
PCWR     .
        240120; 170340; 222416; 401417

/
SLTER    .
        222205; 402217; 401611; 140523

/
TSTNR    .
        230524; 774024; 0

/
FADR1   .
        221106; 402423; 220401; 777740

```

YCHKR PAGE 45

03033	0000000	/	0
03034	003034	LADR1	
03035	230114		230114; 014024; 402204
03036	014024		
03037	402204		
03040	0000000	/	0
03041	003041	SUPR	
03042	202523		202523; 052220; 402323; 0
03043	052220		
03044	402323		
03045	0000000	/	
03046	003046	OVRLAP	
03047	220401		220401; 231140; 112740; 111024
03050	231140		
03051	112740		
03052	111024		
03053	204016		204016; 071722; 150122; 0
03054	071722		
03055	150122		
03056	0000000	/	
03057	003057	PROR	
03060	172220		172220; 012207; 224015; 171405
03061	012207		
03062	224015		
03063	171405		
03064	240103		240103; 161711; 220540; 221722
03065	161711		
03066	220540		
03067	221722		
03070	0000000	/	0
03071	003071	NOMO	
03072	401716		401716; 221715; 054005; 172222
03073	221715		
03074	054005		
03075	172222		
03076	772322		772322; 0
03077	0000000	/	

EJECT

03100	003100	/ PROIS	
03101	172220		172220; 012207; 114015; 114023
03102	012207		
03103	114015		
03104	114023		
03105	064016		064016; 140511; 774004; 0
03106	140511		
03107	774004		
03110	000000		
03111	003111	/ TSLM	
03112	230524		230524; 144024; 111511; 772324
03113	144024		
03114	111511		
03115	772324		
03116	000000		
03117	003117	/ STACS	
03120	240523		240523; 402025; 230301; 0
03121	402025		
03122	230301		
03123	000000		
03124	003124	/ FRST	
03125	221106		221106; 402423; 0
03126	402423		
03127	000000		
03130	003130	/ LSTA	
03131	230114		230114; 774024; 0
03132	774024		
03133	000000		
03134	003134	/ PTOI	
03135	112220		112220; 402416; 242517; 114023
03136	402416		
03137	242517		
03140	114023		
03141	111016		111016; 241102; 770405; 0
03142	241102		
03143	770405		
03144	000000		
03145	003145	/ GOFLD	
03146	401707		401707; 401724; 051106; 400414
03147	401724		
03150	051106		
03151	400414		
03152	000000		
03153	003153	/ PUT12	
03154	242520		242520; 030140; 614023; 174062
03155	030140		
03156	614023		

YCHKR PAGE 47

03157	174062	
03160	014016	014016; 776040; 0
03161	776040	
03162	000000	

EJECT

/STORAGE AND CONSTANT REGISTERS

03163	777770	SIXT4	LAW -1	/COUNTS 64 PASSES BETW EN /ERROR PRINT SUPPRESSION.
03164	000000	NOPRNT	0	/INDICATES END OF ERROR PRINT-OUTS
03165	000000	FLAGS	0	/SAVES SUBROUTINE FLAGS
03166	000000	INSFLD	0	/CURRENT FIELD WITH PROGRAM
03167	000000	LAST	0	/LAST FIELD WITH DATA ERROR
03170	000000	MCWA	0	/SAVES ACS SETTINGS
03171	000000	PCW	0	/CURRENT PAT.CONTROL WORDS
03172	000000	CNTRL	0	/SAME AS PCW
03173	037700	PCWA	037700	/CONTROL WORD FOR TEST 1
03174	001774	PCWB	001774	/FOR TEST 2
03175	037701	PCWC	037701	/FOR TEST 3
03176	001775	PCWD	001775	/FOR TEST 4
03177	000000	PATR	0	/ROTATES CONTROL WORD
03200	000000	PATG	0	/SAVES GOOD DATA DURING READ
03201	000000	PATWD	0	/SAME AS PATG BUT HAS SUPPRESSED BITS
03202	000000	PATN	0	/HAS CONTROL WORD TO ROTATE
03203	000000	TNUM	0	/ASCII TEST NUMBER
03204	000000	LWR	0	/LWR WILL = 0 OR 100
03205	000000	UPR	0	/UPR WILL = 100 OR 300, LWR /AND UPR USED TO OFFSET PATTERNS
03206	000000	MEMADR	0	/ADDRESS COUNTER
03207	000000	SVADR	0	/FIELD COUNTER
03210	000000	CT4K	0	/4K COUNTER
03211	000000	CT16	0	/COUNTS 16 ROTATES
03212	000000	CT128	0	/COUNTS 128 LOCATIONS
03213	000000	CT04	0	/UTILITY COUNTER
03214	000000	FIRST1	0	/FIRST FIELD TO TEST
03215	000000	LAST1	0	/LAST FIELD TO TEST
03216	000000	BAD1	0	/SAVES BAD DATA
03217	000000	OCADR	0	/SAVES FAILING OCTAL ADDRESS
03220	000000	GOOD1	0	/GOOD DATA
03221	000000	PRNT	0	/POINTER FOR PRINT ROUTINES
03222	000000	ADRCN	0	/PARTIAL ADDRESS WORD
03223	000000	EXIT	0	/TO DISMISS
03224	000000	OVER	0	/POINTER TO START OF SUBROUTINES
03225	000000	ADRA	0	/1ST ADR. FROM KEYBOARD INPUT
03226	000000	ADRB	0	/LAST ADR. FROM KEYBOARD INPUT
03227	000000	NROTA	0	/ROTATE COUNTER
03230	000000	LTST	0	/LAST ADR. FOR SCOPE L OPS
03231	000000	TTYW	0	/TTYW THRU YY USED FOR KEYBOARD
03232	000000	TTYX	0	/INPUT BIT SUPPRESSION
03233	000000	TTYY	0	
03234	000000	SCW	0	/TEMP STORAGE OF SUPPRESSED BITS
03235	777777	BITSUP	LAW -1	/EACH SUPPRESSED BIT = 0
03236	000000	NXLOC	0	/NEXT FIELD TO MOVE INTO
03237	000000	SOURCE	0	/FIELD TO MOVE FROM
03240	000000	DESTN	0	/FIELD TO MOVE TO
03241	000000	MMOVED	0	/ADDRESS COUNTER FOR MOVING
03242	000000	MOVES	0	/SAVE AS MOVED
03243	000215	RGNLO	RTM1	/EXIT ADR. TO A LU 4K FIELD
03244	010215	RGNHI	RTM1+10000	/EXIT ADR. TO A HI 4K FIELD
			EJECT	

03245	000001	K1	1
03246	000002	K2	2
03247	000007	K7	7
03250	000010	K10	10
03251	000020	K20	20
03252	000040	K40	40
03253	000077	K77	77
03254	000100	K100	100
03255	000200	K200	200
03256	000212	K212	212
03257	000215	K215	215
03260	000254	K254	254
03261	000260	K260	260
03262	000264	K264	264
03263	000270	K270	270
03264	000300	K300	300
03265	000331	K331	331
03266	000370	K370	370
03267	000377	K377	377
03270	000400	K400	400
03271	0004000	K4K	4000
03272	007777	K7777	7777
03273	010000	K10K	10000
03274	017777	K17S	17777
03275	020000	K20K	20000
03276	040000	K40K	40000
03277	070000	K70K	70000
03300	074000	K74K	74000
03301	400000	K400K	400000
03302	577777	K577	577777
03303	700000	K700K	700000
03304	777700	MAXERR	LAW -1 10
03305	003306	ERWRD	+1
03306	760000		LAW
03307	760000		LAW
03310	760200		LAW
03311	760400		LAW
03312	760600		LAW
03313	760800		LAW
03314	760000		LAW
03315	760000		LAW
03316	123456	RANCON	123456
03317	654321	RANTAL	654321
03320	361416		361416
03321	055363		055363
03322	546060		546060
03323	243035		243035
03324	762572		762572
03325	453237		453237
03326	150214		150214
03327	000000	RANSAV	0
03330	000000	/	
	000200		.END BEGIN NO ERROR LINES

YCHKR PAGE 50

ADR	02764
ADRA	03225
ADRB	03226
ADRCW	03222
ADRX	02023
ADRXA	02752
ADRX1	01142
ADRXP	02033
AU-2	01216
AUR2P	02034
AUXR	02026
AD1R	02024
AD2R	02027
AGAIN	01447
ALUK	01776
BAL1	03216
BOAT	02775
BOATX	02754
BEGIN	00200
BGAH1	03244
BGAL0	03243
BITSUP	03235
CBANK	00633
CBOOTH	01311
CFLG	02637
CKERR	02166
CKFCN	02436
CKLST	01302
CKNXT	02243
CK16	00547
CKS	02073
CLMN	02744
CLCF	700004
CLCN	700044
CLCOP	00606
CLSF	700001
CMGVE	02141
CNRDT	01644
CTRL	03172
CREAD	00404
CREVR	01745
CRLF	02106
CT24	03213
CT128	03212
CT16	03211
CT4K	03210
C128	00572
DESTN	03240
DFST	01172
DLMT	02757
DLST	01244
DNYVE	02236
DUALL	00323
DOERR	01007
DOA1	02025

YCHKR PAGE 51

UON2	02030
UON3	02041
URFAD	00564
ENERR	01104
ENOT	02620
ENTBL	00736
EOM	01524
EOT	01516
EOTA	01530
EQUAL	02213
ERHOR	00740
ERRTN	00530
ERSEL	02756
ERTBL	01143
ERWRD	03305
EXAM2	00274
EXAM3	00300
EXAM4	00304
EXIT	03223
EXTST	00270
E2	02363
FAUR	01161
FAUR1	03026
FCDMV	02425
FIKST1	03214
FLAGS	03165
FRST	03124
GDAT	02771
GUATX	02753
GENPAT	00415
GENRAN	00707
GETAD	01637
GOFL	02617
GOFLO	03145
GOLEFT	01663
GOOD1	03220
GOTO	02557
HALT	00667
INUY	01055
INSFLD	03166
JMP	01346
JMP1	00576
JMP2	00573
JMP3	02706
KEYIN	01573
KRH	700312
KSF	700301
KYHRD	01106
K1	03245
K1W	03250
K1WK	03273
K1W0	03254
K1WS	03274
K2	03246
K2W	03251

YCHKR PAGE 52

K2VK	03275
K2V0	03255
K212	03256
K215	03257
K254	03260
K260	03261
K264	03262
K270	03263
K3V0	03264
K351	03265
K370	03266
K377	03267
K4K	03271
K4L	03252
K4VK	03276
K4V0	03270
K4V0K	03301
K577	03302
K7	03247
K7VK	03277
K7V0K	03303
K74K	03300
K77	03253
K7777	03272
LAUR	01233
LAI R1	03034
LAST	03167
LAST1	03215
LEGAL	01602
LOCAT	01615
LOUER	02675
LSTA	03130
LTST	03230
LWR	03204
MASK	02057
MAXERR	03304
MCWA	03170
MEMADR	03206
MOSOM	02504
MOVE	02324
MOVED	03241
MOVES	03242
MSAME	02474
MVHK	02315
MVST	02374
MVERP	02644
MVOPR	02345
MVHTN	02334
NERN	02705
NETWK	00373
NONO	03071
NONOR	00642
NOPRNT	03164
NUSW	02575
NRUTA	03227

YCHKR PAGE 53

NUMB	01546
NXLOC	03236
NXTBNK	00653
NXTHI	02311
NXTMV	02241
UCADR	03217
UKAS	01763
UPRAT	02364
JVER	03224
UVRLAP	03046
UVRLP	02035
PASSC	00621
PATG	03200
PATN	03202
PATR	03177
PATWD	03201
PCF	700202
PCCHAR	02043
PCW	03171
PCWA	03173
PCWB	03174
PCWC	03175
PCWD	03176
PCWR	03000
PCWX	02755
PHUR	02715
PISIN	02036
PNXT	02050
POSITN	02127
PRNT	03221
PRUCTL	02123
PROG	01414
PROIS	03100
PROK	03057
PRSEL	02535
PSA	700204
PSB	700244
PSF	700201
PTC	01105
PTOI	03134
PTWLV	02616
PUT12	03153
QUERY	01013
RANCON	03316
RANDEX	00735
RANSAV	03330
RANTAD	00725
RANTBL	03317
RCF	700102
RCMA	00514
RCOMP	00577
READ	00463
RELOC	02704
RLOOP	00503
ROTA	01665

YCHKR PAGE 54

ROTB	01666
ROTOR	01506
RRH	700112
RSA	700104
RSM	700144
RSCNT	00475
RSF	700101
RSTOR	00557
RTA1	00215
RT19L	02544
SCP1	01400
SCY	03234
SETAC	02002
SETU1	00624
SETX	02042
SIMU	01344
SIM1	01360
SIM2	01372
SIM3	01374
SIM4	01376
SIXT4	03163
SLMTS	01674
SLMX	02040
SLTER	03010
SOURCE	03237
SPING	02115
STACS	03117
STALL	00676
STER	01032
STLP	01270
STMV	02201
STAXT	02260
STCVER	00246
STSCP	01411
SUH1	02224
SUH2	02300
SUPBIT	01431
SUPR	03041
SUPX	02031
SUPXA	02032
SVADDR	03207
SW0	01003
SW1	00775
SW2	00763
TBLTOP	00737
TCF	700402
TLMX	02037
TLS	700406
TNUM	03203
TSF	700401
TSLM	03111
TSMX	02022
TST	02760
TSTN	01137
TSTNO	01110

YCHKR PAGE 55

TSTNR	03022
TSTX	02751
TST1	00327
TST2	00340
TST3	00351
TST4	00362
TTYW	03231
TTYX	03232
TTYY	03233
UPR	03205
WHERE	00663
MHWAY	02454
WLOOP	00433
NOTIS	02707
XPRTR	02451
ZERO	01542

9CHKR PAGE 56

BEGIN	00200
RTN1	00215
STOVER	00246
EXTST	00270
EXAM2	00274
EXAM3	00300
EXAM4	00304
JOALL	00323
TST1	00327
TST2	00340
TST3	00351
TST4	00362
NETWK	00373
URLOAD	00404
GENPAT	00415
NLOOP	00433
READ	00463
RSCNT	00475
RLOOP	00503
RCMA	00514
ERHTN	00530
CK16	00547
RSTOR	00557
UREAD	00564
C128	00572
JMP2	00573
JMP1	00576
RCCMP	00577
CLUOP	00606
PASSC	00621
SETU1	00624
CBANK	00633
NOMOR	00642
VXTBNK	00653
WHERE	00663
HALT	00667
STALL	00676
GENRAN	00707
RANTAU	00725
RANDFX	00735
ENTBL	00736
TBLTOP	00737
ERROR	00740
SW2	00763
SW1	00775
SW2	01003
JOERR	01007
STER	01032
INDY	01055
ERTBL	01103
ENERR	01104
PTG	01105
KYRD	01106
TSTNO	01110
TSTN	01137

YCHKR PAGE 57

ADH1	01142
FAUR	01161
JFST	01172
ADH2	01216
LAUR	01233
ULST	01244
STLP	01270
UKLST	01302
CBOTH	01311
SIMU	01344
JMP	01346
SIM1	01360
SIM2	01372
SIM3	01374
SIM4	01376
SCP1	01400
STSCP	01411
PRUG	01414
SUPBIT	01431
AGAIN	01447
RUTOR	01506
EOT	01516
EOM	01524
EOTA	01530
ZERO	01542
NUMB	01546
KEYIN	01573
LEGAL	01602
JURY	01613
LOCAT	01615
GETAD	01637
UNROT	01644
GOLEFI	01663
ROTA	01665
ROTB	01666
SLMTS	01674
JREVR	01745
UKAS	01763
ALOK	01776
SETAC	02002
TSNX	02022
ADRX	02023
ADIR	02024
JON1	02025
AUXR	02026
AUZP	02027
JON2	02030
SUPX	02031
SUPXA	02032
ADH1P	02033
ADH2P	02034
JVRLP	02035
PISIN	02036
TLMX	02037
SLMX	02040

YCHKR PAGE 58

JON3	02041
SETX	02042
PCHAR	02043
PNXT	02050
MASK	02057
CKS	02073
CRLF	02106
SPING	02115
PROCTL	02123
POSITN	02127
UMOVE	02141
CKERR	02166
STMV	02201
EQUAL	02213
SUM1	02224
UNMVF	02236
VXTMV	02241
CKNXT	02243
ST^XT	02260
SUM2	02300
VXTHT	02311
MVK	02315
MOVE	02324
MV^TN	02334
IVIPR	02345
E2	02363
JPHAT	02364
IVCST	02374
FCUMV	02425
CKFCDN	02436
APHT	02451
WHWAY	02454
MSAMF	02474
MOSOM	02504
PRSEL	02535
RT19L	02544
GOTO	02557
IVSW	02575
HTWLV	02616
SUFL	02617
ENUT	02626
GFLG	02637
IVERR	02644
LOCER	02675
RELOC	02704
VERN	02705
JMP3	02706
NOTIS	02707
PHDR	02715
CLMN	02744
FSTX	02751
ADHXA	02752
UDATX	02753
3DATX	02754
PCWX	02755

YCHKR PAGE 59

ERSEL	02756
DLMT	02757
TST	02760
ADR	02764
GDAT	02771
HDAT	02775
PCWR	03000
SLTER	03010
ISTNR	03022
FAIR1	03026
LADR1	03034
SUPR	03041
OVRLAP	03046
PROR	03057
NOMO	03071
PRUIS	03100
ISLM	03111
STACS	03117
FRST	03124
LSTA	03130
PTOI	03134
GOFLD	03145
PUT12	03153
SIXT4	03163
NOPRNT	03164
FLAGS	03165
INSFLU	03166
LAST	03167
ICWA	03170
PCW	03171
UNTRL	03172
PCWA	03173
PCWB	03174
PCWC	03175
PCWD	03176
PATR	03177
PATG	03200
PATWD	03201
PATN	03202
FNUM	03203
LWR	03204
UPR	03205
MEMADR	03206
SVADR	03207
CT4K	03210
CT16	03211
CT128	03212
CT4	03213
FIRST1	03214
LAST1	03215
BAU1	03216
UCADR	03217
GOOD1	03220
PRNT	03221
ADRCW	03222

YCHKR PAGE 60

EXIT	03223
OVER	03224
ADRA	03225
ADRB	03226
VROTA	03227
LST	03230
ITYW	03231
ITYX	03232
ITYY	03233
SCW	03234
BITSUP	03235
NXLOC	03236
SOURCE	03237
DESTN	03240
MOVED	03241
MOVES	03242
BGNL0	03243
BGNHI	03244
K1	03245
K2	03246
K7	03247
K1V	03250
K2V	03251
K4V	03252
K7I	03253
K140	03254
K2V0	03255
K212	03256
K215	03257
K254	03260
K260	03261
K264	03262
K270	03263
K370	03264
K331	03265
K370	03266
K377	03267
K4V0	03270
K4K	03271
K7777	03272
K1VK	03273
K17S	03274
K2VK	03275
K4VK	03276
K7VK	03277
K74K	03300
K400K	03301
K577	03302
K700K	03303
MAXERH	03304
ERWRD	03305
RANCON	03316
RANTBL	03317
RANSAV	03330
ULSF	700001

YCHKR PAGE 61

CLOF	700004
CLDN	700044
KSF	700101
KCF	700102
RSA	700104
RRH	700112
RSH	700144
PSF	700201
PCF	700202
PSA	700204
PSH	700244
KSF	700301
KRH	700312
TSF	700401
TCF	700402
TLS	700406