

IDENTIFICATION

PRODUCT CODE: MAINDEC-08-DHLQA-B-D
PRODUCT NAME: LQP-8 PRINTER DIAGNOSTIC
PRODUCT DATE: JULY 1977
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: DAVID ORIN

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1977, BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	HARDWARE
2.2	STORAGE
2.3	PREPEQUISITE SOFTWARE
3.0	RESTRICTIONS
4.0	STANDARD TEST PROCEDURE
4.1	LOADING THE PROGRAM
4.2	PROGRAM INITIALIZATION
4.2.1	INITIALIZATION FOR A NON-ACTIVE CONSOLE PACKAGE
4.2.2	INITIALIZATION FOR AN ACTIVE CONSOLE PACKAGE
4.3	CHANGING DEVICE CODES
4.3.1	CHANGING DEVICE CODES ON A NON-ACTIVE CONSOLE PACKAGE
4.3.2	CHANGING DEVICE CODES ON AN ACTIVE CONSOLE PACKAGE
4.4	INTERNAL LOGIC/DATA TEST DESCRIPTION
4.4.1	RUN INTERNAL LOGIC/DATA TEST ON A NON-ACTIVE CONSOLE PACKAGE
4.4.2	RUN INTERNAL LOGIC/DATA TEST ON AN ACTIVE CONSOLE PACKAGE
4.5	DYNAMIC PRINTER TEST DESCRIPTION
4.5.1	RUN DYNAMIC PRINTER TEST ON A NON-ACTIVE CONSOLE PACKAGE
4.5.2	RUN DYNAMIC PRINTER TEST ON AN ACTIVE CONSOLE PACKAGE
4.6	RUN SPECIAL DYNAMIC PRINTER TESTS
4.6.1	REQUIREMENTS
4.6.2	PROCEDURES
4.7	OPERATION OF CONSOLE PACKAGE
4.7.1	CONTROL CHARACTERS
4.7.2	SELECTING TEST NUMBER
4.7.3	CHANGING PSUEDO SWITCH REGISTER
5.0	ERRORS
5.1	LOGIC ERRORS
5.1.1	LOGIC ERRORS ON A NON-ACTIVE CONSOLE
5.1.2	LOGIC ERRORS ON AN ACTIVE CONSOLE
5.2	STATUS ERRORS
5.2.1	STATUS ERRORS ON A NON-ACTIVE CONSOLE
5.2.2	STATUS ERRORS ON AN ACTIVE CONSOLE
5.3	DATA ERRORS
5.3.1	DATA ERRORS ON A NON-ACTIVE CONSOLE
5.3.2	DATA ERRORS ON AN ACTIVE CONSOLE
6.0	SWITCH REGISTER SETTINGS
6.1	NORMAL OPERATION SWITCHES
6.2	ERROR RELATED SWITCHES
7.0	REVISIONS
8.0	ADDITIONAL INFORMATION
8.1	STARTING ADDRESSES
8.2	SPECIAL ADDRESSES
9.0	LISTING

1.0 ABSTRACT

THE LQP-8 PRINTER DIAGNOSTIC WILL TEST ALL THE LOGIC ON THE LQP-8 PARALLEL INTERFACE MODULE (M8366) THAT CAN BE TESTED VIA PROGRAM INSTRUCTION. IN ORDER TO TEST BOTH THE INTERFACE LOGIC AND PRINTER LOGIC COMPLETELY, BOTH THE LOGIC AND DYNAMIC TESTS MUST BE RUN.

THIS DIAGNOSTIC CAN ALSO BE USED ON THE VT78 SYSTEM TO TEST THE PARALLEL INTERFACE AND PRINTER LOGIC.

A CONSOLE PACKAGE WITH SOFTWARE CONTROLLED SWITCH REGISTER HAS BEEN INCORPORATED INTO THIS DIAGNOSTIC TO ALLOW THE PROGRAM TO RUN WITH OR WITHOUT A HARDWARE SWITCH REGISTER, TO INHIBIT ALL HALTS WITHIN THE PROGRAM, AND TO HAVE COMMUNICATIONS WITH THE DIAGNOSTIC VIA A TERMINAL. IF THE CONSOLE SWITCH REGISTER PACKAGE IS NOT ACTIVE, THEN THE PROGRAM WILL RUN WITH OR WITHOUT THE HARDWARE SWITCH REGISTER. ALL ERRORS AND END OF PASS INDICATIONS WILL RESULT IN A HALT.

THE INFORMATION NECESSARY TO INTERPRET ERROR HALTS OR MESSAGES IS CONTAINED IN SECTION 5 UNDER ERRORS.

2.0 REQUIREMENTS

2.1 HARDWARE

THE FOLLOWING HARDWARE IS REQUIRED FOR EXECUTION OF THIS PROGRAM:

PROCESSOR:	PDP-8A, 8E, 8F, 8M; VT78
MEMORY:	MINIMUM OF 4K R/W MEMORY IN FIELD 0
OPTIONS:	LQP-8 PARALLEL INTERFACE (M8366) CONSOLE SERIAL INTERFACE WITH TERMINAL (OPTIONAL)
SPECIAL:	DIABLO HYTYPE II PRINTER

2.2 STORAGE

THE PROGRAM MUST BE LOADED INTO FIELD 0 CONTAINING AT LEAST 4K OF READ/WRITE MEMORY. THE PROGRAM OCCUPIES LOCATIONS 0000 TO 7577.

2.3 PREREQUISITE SOFTWARE

ALL OF THE BASIC PDP-8A, E, F, M OR VT78 CPU AND MEMORY TESTS MUST HAVE BEEN RUN SUCCESSFULLY.

3.0 RESTRICTIONS

- A. RUNNING OF THE INTERNAL LOGIC/DATA TEST REQUIRES THAT THE LQ PRINTER BE PLACED OFF LINE BY

PRESSING THE SWITCH LABELED "PAUSE", OR BY REMOVING THE MAINTENANCE COVER LOCATED AT THE FRONT OF THE PRINTER. ON SOME OLDER PRINTERS, PRESSING "PAUSE" MAY NOT DROP "PRINTER READY", BIT 0, WHICH WILL CAUSE A STATUS ERROR. IN THIS CASE, REMOVE THE MAINTENANCE COVER.

- B. PRESSING A KEY ON THE CONSOLE TERMINAL WITH CONSOLE PACKAGE NON-ACTIVE WILL HAVE NO EFFECT ON THE PROGRAM. PRESSING A NON-CONTROL KEY, WITH ACTIVE CONSOLE, DURING EXECUTION OF THE PROGRAM WILL RESULT IN A PROGRAM STALL, THE CHARACTER BEING ECHOED, AND THE PROGRAM BEING RESUMED.
- C. RUNNING OF THE DYNAMIC PRINTER TESTS REQUIRES THAT THE LQ PRINTER BE PLACED ON LINE BY REPLACING THE MAINTENANCE COVER AND/OR PRESSING THE SWITCH LABELED "PAUSE".

4.0 STANDARD TEST PROCEDURE

TO FULLY TEST THE LQ PRINTER, CONNECTING CABLE, AND INTERFACE LOGIC, THE FOLLOWING SECTIONS MUST BE RUN COMPLETELY:

- PARAGRAPH 4.4 INTERNAL LOGIC/DATA TEST
- PARAGRAPH 4.5 DYNAMIC PRINTER TEST
- PARAGRAPH 4.6 SPECIAL DYNAMIC PRINTER TESTS

4.1 LOADING THE PROGRAM

PROCESSOR = PDP-8A, E, F, M;
LOAD THE PROGRAM INTO FIELD 0 USING THE STANDARD BINARY LOADER TECHNIQUE.

PROCESSOR = VT78:

FOR VT78 SYSTEMS THIS DIAGNOSTIC IS PROVIDED ALONG WITH A SYSTEM MONITOR ON FLOPPY MEDIA. TO LOAD THE SYSTEM MONITOR SIMPLY INSERT THE FLOPPY DISKETTE CONTAINING THE DIAGNOSTIC INTO EITHER DRIVE RXA0 OR RXA1 AND PRESS THE VT78 START BUTTON. THE SYSTEM WILL RESPOND ON THE VIDEO DISPLAY WITH A START MESSAGE, FOLLOWED BY A PROMPT CHARACTER. TO CALL AND START THE DIAGNOSTIC TYPE 'R DHLQAB' FOLLOWED BY THE RETURN KEY. THE SYSTEM WILL RESPOND WITH A DISPLAY OF THE PROGRAM NAME AND CURRENT PSEUDO SWITCH REGISTER SETTING AND WAIT FOR USER SELECTION (SEE SECTION 6.0). THIS DIAGNOSTIC ON THE FLOPPY DISKETTE HAS BEEN PRE-INITIALIZED FOR USE ON A VT78 SYSTEM (LOC 22 BIT 2 = 1). ALSO STARTING THE DIAGNOSTIC BY ISSUING 'R DHLQAB' WILL EXECUTE THE DYNAMIC PRINTER TEST AND NOT THE INTERNAL LOGIC/DATA TEST. THE VT78 CPU DIAGNOSTIC CHECKS OUT THE INTERNAL INTERFACE LOGIC.

4.2 PROGRAM INITIALIZATION

THE PROGRAM, WHEN LOADED, IS INITIALIZED TO RUN AS FOLLOWS:

- A. NO HARDWARE SWITCH REGISTER (USES LOCATION 0020 AS PSEUDO SWITCH REGISTER)

- B. 4K OF READ/WRITE MEMORY - THE MINIMUM REQUIREMENT
- C. CONSOLE PACKAGE INACTIVE (PAPER TAPE MEDIA)
- D. CONSOLE PACKAGE ACTIVE (DISK MEDIA)

IF IT IS DESIRED TO CHANGE THE PROGRAM INITIALIZATION FROM THE ABOVE, GO TO THE APPROPRIATE PARAGRAPH LISTED BELOW. OTHERWISE, GO TO PARAGRAPH 4.3 - CHANGING DEVICE CODES.

4.2.1 INITIALIZATION FOR A NON-ACTIVE CONSOLE PACKAGE

- A. TO USE HARDWARE SWITCH REGISTER, LOAD ADDRESS TO 0021 IN FIELD 0, AND DEPOSIT 4000. THIS IS HARDWARE CONFIGURATION WORD ONE, AND SETTING BIT 0 TO A ONE (1) TELLS THE PROGRAM TO USE THE HARDWARE SWITCH REGISTER.
- B. TO USE PSUEDO SWITCH REGISTER, LOAD ADDRESS TO 0021 IN FIELD 0, AND DEPOSIT 0000. LOAD ADDRESS TO 0020 AND DEPOSIT THE DESIRED SWITCH REGISTER CONTENTS.
 - LOC 0020 FIELD 0 PSUEDO SWITCH REGISTER
 - LOC 0021 FIELD 0 HARDWARE CONFIG. WORD ONE
 - LOC 0022 FIELD 0 HARDWARE CONFIG. WORD TWO
- C. TO DEACTIVATE THE CONSOLE PACKAGE, LOAD ADDRESS TO 0022 AND DEPOSIT 0000.

4.2.2 INITIALIZATION FOR AN ACTIVE CONSOLE PACKAGE

- A. TO USE HARDWARE SWITCH REGISTER, FOLLOW INSTRUCTIONS IN SECTION 4.2.1 A.
- B. TO USE PSUEDO SWITCH REGISTER, FOLLOW INSTRUCTIONS IN SECTION 4.2.1 B.
- C. TO ACTIVATE THE CONSOLE PACKAGE, LOAD ADDRESS TO 0022 AND DEPOSIT 0400.

4.3 CHANGING DEVICE CODES

THE PROGRAM, WHEN LOADED, IS INITIALIZED FOR DEVICE CODE 650X FOR THE LO PRINTER. IF IT IS DESIRED TO CHANGE THIS DEVICE CODE, FOLLOW THE INSTRUCTIONS IN THE APPROPRIATE PARAGRAPH BELOW FOR A NON-ACTIVE OR AN ACTIVE CONSOLE PACKAGE. IF NO CHANGE IN DEVICE CODE IS REQUIRED, GO TO PARAGRAPH 4.4 - RUN INTERNAL LOGIC/DATA TEST.

4.3.1 CHANGING DEVICE CODE ON A NON-ACTIVE CONSOLE PACKAGE

USING PSUEDO SWITCH REGISTER:

- A. LOAD THE DESIRED DEVICE CODE INTO BITS 3-8 OF LOCATION 0020 (THE PSUEDO SWITCH REGISTER).
- B. LOAD ADDRESS TO 202, PRESS "CLEAR" THEN "CONT" ON A PDP-8E, F, OR M. PRESS "INIT" THEN "RUN" ON A PDP-8A.

- C. PROGRAM WILL HALT WITH OLD DEVICE CODE IN AC BITS 3-8.
- D. PRESS "CONT". PROGRAM WILL HALT, AFTER CHANGING IOT DEVICE CODES, WITH THE NEW DEVICE CODE IN AC BITS 3-8.
- E. PRESSING "CONT" WILL CAUSE THE PROGRAM TO START THE LOGIC TEST OR THE PROGRAM MAY NOW BE STARTED AT ONE OF THE OTHER STARTING ADDRESSES.

USING HARDWARE SWITCH REGISTER:

- A. LOCATION 0021 OF FIELD 0 MUST CONTAIN 4XXX.
- B. LOAD ADDRESS TO 202.
- C. SET SWITCH REGISTER BITS 3-8 TO DESIRED DEVICE CODE.
- D. PRESS "CLEAR" THEN "CONT" ON A PDP-8E, F, M, OR PRESS "INIT" THEN "RUN" ON A PDP-8A.
- E. PROGRAM WILL HALT WITH OLD DEVICE CODE IN AC BITS 3-8.
- F. PRESS "CONT". PROGRAM WILL HALT, AFTER CHANGING IOT DEVICE CODES, WITH THE NEW DEVICE CODE IN AC BITS 3-8.
- G. PRESSING "CONT" WILL CAUSE THE PROGRAM TO START THE LOGIC TEST OR THE PROGRAM MAY NOW BE STARTED AT ONE OF THE OTHER STARTING ADDRESSES.

4.3.2 CHANGING DEVICE CODES ON AN ACTIVE CONSOLE PACKAGE

USING PSUEDO SWITCH REGISTER:

- A. LOAD THE DESIRED DEVICE CODE INTO BITS 3-8 OF LOCATION 0020 (THE PSUEDO SWITCH REGISTER).
- B. START THE PROGRAM AT ADDRESS 202.
- C. THE PROGRAM WILL CHANGE THE IOT DEVICE CODES, PRINT THE MESSAGE "DSC = 00X0" (WHERE "XX" EQUALS THE NEW DEVICE CODE) AND ASK THE SWITCH REGISTER EQUALS QUESTION (SR=).

USING HARDWARE SWITCH REGISTER:

- A. LOCATION 0021 OF FIELD 0 MUST CONTAIN 4XXX.
- B. LOAD THE DESIRED DEVICE CODE INTO BITS 3-8 OF THE SWITCH REGISTER.
- C. START THE PROGRAM AT ADDRESS 202.
- D. THE PROGRAM WILL CHANGE THE IOT DEVICE CODES, PRINT THE MESSAGE "DSC=0XX0" (WHERE "XX" EQUALS THE NEW DEVICE CODE) AND ASK THE SWITCH REGISTER EQUALS QUESTION (SR=).

4.4 INTERNAL LOGIC/DATA TEST DESCRIPTION

THE LOGIC TEST CONSISTS OF THE FOLLOWING TESTS:

DATA BUFFER TESTS:

```

IOT 2          MOVE PAPER
--- -          ----

TEST 1         LOAD ALL ONES AND ALL ZEROS
TEST 2         LOAD ALTERNATING ONES AND ZEROS

```


TEST 3 FLOATING ONE BIT CHECK
TEST 4 FLOATING ZERO BIT CHECK
TEST 5 BINARY COUNT, EVERY NUMBER

IOT 3 MOVE CARRIAGE

TEST 6 LOAD ALL ONES AND ALL ZEROS
TEST 7 LOAD ALTERNATING ONES AND ZEROS
TEST 10 FLOATING ONE BIT CHECK
TEST 11 FLOATING ZERO BIT CHECK
TEST 12 BINARY COUNT, EVERY NUMBER

IOT 4 PRINT CHARACTER

TEST 13 LOAD ALL ONES AND ALL ZEROS
TEST 14 LOAD ALTERNATING ONES AND ZEROS
TEST 15 FLOATING ONE BIT CHECK
TEST 16 FLOATING ZERO BIT CHECK
TEST 17 BINARY COUNT, EVERY NUMBER

ACCUMULATOR TEST

TEST 20 CHECK THAT CAF INSTRUCTION CLEARS DATA BUFFER
TEST 21 CLEAR AC USING IOT2
TEST 22 CLEAR AC USING IOT3
TEST 23 CLEAR AC USING IOT4
TEST 24 CHECK THAT AC IS NOT CHANGED BY IOT0
TEST 25 CHECK THAT AC IS NOT CHANGED BY IOT7
TEST 26 CLEAR AC USING IOT6

DONE FLAG TESTS

TEST 27 SET DONE USING IOT6 CLEAR DONE IOT5
TEST 30 SET DONE USING IOT6 CLEAR DONE IOT2
TEST 31 SET DONE USING IOT6 CLEAR DONE IOT3
TEST 32 SET DONE USING IOT6 CLEAR DONE IOT4
TEST 33 SET DONE USING IOT6 CLEAR DONE IOT7
TEST 34 SET DONE USING IOT6 CLEAR DONE USING CAF

STATUS REGISTER TESTS

TEST 35 LOAD ALL ONES AND ALL ZEROS BITS 5, 8, 9, 10, 11
TEST 36 BIT 5 CHECK
TEST 37 BIT 8 CHECK

TEST 40 BIT 9 CHECK
TEST 41 BIT 10 CHECK
TEST 42 BIT 11 CHECK
TEST 43 CLEAR STATUS REGISTER USING CAF INSTRUCTION

SKIP AND INTERRUPT TESTS

TEST 44 CHECK SKIP AND INTERRUPT ENABLE

4.4.1 RUN INTERNAL LOGIC/DATA TEST ON A NON-ACTIVE CONSOLE PACKAGE

- A. PLACE PRINTER OFF LINE BY REMOVING MAINTENANCE COVER
- B. SEE SECTION 6.0 FOR SWITCH REGISTER SETTINGS.
- C. START THE PROGRAM AT ADDRESS 200.
- D. THE LOGIC TEST WILL LOOP CONTINUOUSLY.
- E. PRESSING "CONT" WILL RESTART THE LOGIC TEST.

4.4.2 RUN INTERNAL LOGIC/DATA TEST ON AN ACTIVE CONSOLE PACKAGE

- A. PLACE PRINTER OFF LINE BY REMOVING MAINTENANCE COVER.
- B. SEE SECTION 6.0 FOR SWITCH REGISTER SETTINGS.
- C. START THE PROGRAM AT ADDRESS 200.
- D. THE LOGIC TEST WILL LOOP 4096 TIMES AND PRINT "END OF LOGIC TEST" AND "PASS COUNT XXXX", WHERE "XXXX" IS A 4-DIGIT OCTAL NUMBER.

NOTE: IF AN ATTEMPT IS MADE TO RUN THE LOGIC TEST WITH THE PRINTER "ON LINE" (READY), A STATUS ERROR WILL RESULT.

4.5 DYNAMIC PRINTER TEST DESCRIPTION

THE STANDARD DYNAMIC PRINTER TESTS CONSIST OF:

TEST 45	HORIZONTAL SPACING TEST /BIT CHECK
TEST 46	VERTICAL SPACING TEST /BIT CHECK
TEST 47	CARRIAGE POSITIONING TEST
TEST 50	ONE LINE OF EACH CHARACTER, 96 LINES, BI-DIRECTIONAL PRINTING
TEST 51	SWIRL PATTERN, 96 LINES
TEST 52	RIBBON LIFT AND DROP WITH OVERPRINT
TEST 53	RANDOM CHARACTER LINE PAIRS
TEST 54	NON-PRINTING CHARACTER CODES TEST
TEST 55	PRINT WHEEL VIBRATION TEST

THE SPECIAL DYNAMIC PRINTER TESTS CONSIST OF:

TEST 56	OPERATOR GENERATED PRINT LINE
TEST 57	RAISE AND LOWER RIBBON
TEST 60	FRICTION FEED TEST 10 BY 10 RANDOM MATRIX

A. TEST 45 HORIZONTAL SPACING TEST

THIS TEST BEGINS BY RESTORING THE PRINTER AND OVERPRINTING A DOT. THE PRINTER LINE FEEDS AND CARRIAGE RETURNS, PRINTS A DOT AND THEN MOVES THE CARRIAGE TO THE RIGHT BY THE MAGNITUDE OF BIT 11 = 1/120 INCH, AND PRINTS ANOTHER DOT. THIS PROCESS CONTINUES UNTIL ALL BITS ARE TESTED. BITS 1-11 ARE TESTED FOR PROPER MAGNITUDE AND BIT 0 IS TESTED DURING A CARRIAGE RETURN.

THE PRINT PATTERN SHOULD APPEAR AS FOLLOWS:

DATA BUFFER BITS	CARRIAGE MOVEMENT MAGNITUDE
BITS 1-11 = 0	OVERPRINT (NO MOVEMENT)

D. TEST 50 ONE LINE OF EACH CHARACTER, 96 LINES, BI-DIRECTIONAL PRINTING.

THIS TEST BEGINS BY RESTORING THE PRINTER AND SENDING ASCII CODE 040 TO THE PRINTER (SEE MAINTENANCE MANUAL FOR CHARACTER PRINTED). AN ENTIRE LINE OF EACH OF THE 96 POSSIBLE CHARACTERS IS PRINTED FOR ASCII CODES 040 THROUGH 177. THE FIRST LINE IS PRINTED FROM LEFT TO RIGHT, THE SECOND LINE IS PRINTED FROM RIGHT TO LEFT, AND SO ON. ALL ODD NUMBERED LINES WILL BE PRINTED FROM LEFT TO RIGHT, ALL EVEN NUMBERED LINES WILL BE PRINTED FROM RIGHT TO LEFT.

NOTE: SINCE BITS 4-10 OF THE DATA BUFFER ARE USED FOR CHARACTER PRINTING THE ASCII CODE IS SHIFTED LEFT ONE (1) POSITION BY THE PROGRAM BEFORE BEING SENT TO THE PRINTER, THUS, ASCII CODE 040 BECOMES 100. IN ADDITION, ALL UNUSED BITS OF THE DATA BUFFER ARE SET TO THE ONE (1) CONDITION FOR MAXIMUM CROSS TALK WORSE CASE TESTING, THEREFORE THE ACTUAL CODE SENT TO THE PRINTER FOR ASCII CODE 040 WOULD BE 7501, (UNUSED BITS 0-3 AND BIT 11 ARE ALL SET TO THE ONE (1) STATE).

E. TEST 51 SWIRL PATTERN, 96 LINES

THIS TEST BEGINS BY RESTORING THE PRINTER AND PRINTING A LINE STARTING WITH ASCII CODE 040 FOR THE FIRST CHARACTER, CODE 041 FOR THE SECOND CHARACTER, ETC. THE NEXT LINE BEGINS WITH ASCII CODE 041, ETC. REFER TO MAINTENANCE MANUAL FOR CHARACTER PRINTED FOR EACH ASCII CODE.

NOTE: ALL 96 ASCII CODES 040-177 ARE PRINTING CHARACTERS, ASCII CODE 040 IS NOT A BLANK, REFER TO MAINTENANCE MANUAL FOR CORRECT CHARACTERS.

F. TEST 52 RIBBON LIFT AND DROP WITH OVERPRINT

THIS TEST BEGINS BY RESTORING THE PRINTER AND PRINTING A CAPITAL "M" WITH THE RIBBON LIFTED. THE PRINTER CARRIAGE MOVES RIGHT ONE PRINT POSITION AND A CAPITAL "M" IS AGAIN STRUCK WITH THE RIBBON DROPPED. THE NEXT "M" IS PRINTED WITH THE RIBBON UP, ETC. EVERY EVEN NUMBERED "M" IS STRUCK WITH THE RIBBON DOWN. EVERY ODD NUMBERED "M" IS STRUCK WITH THE RIBBON UP. THE SAME LINE IS THEN OVERPRINTED FROM RIGHT TO LEFT. THE PATTERN CONTINUES FOR FIVE(S) LINES.

THE PATTERN SHOULD APPEAR AS FOLLOWS:

M M M M M M M M ... ETC.

NOTE: FOR BEST RESULTS, USE A MYLAR (SINGLE SIDED) RIBBON AND CLEAN PRINT WHEEL. OTHERWISE THE PRINT WHEEL WILL RETAIN INK (CLOTH RIBBON) AND THE "M"'S STRUCK WITH RIBBON DROPPED WILL APPEAR ONLY SLIGHTLY LIGHTER THAN THOSE STRUCK WITH RIBBON UP.

G. TEST 53 RANDOM CHARACTER LINE PAIRS

THIS TEST PRINTS PAIRS OF RANDOMLY GENERATED LINES. EACH LINE OF A PAIR SHOULD BE IDENTICAL TO THE OTHER LINE OF THE PAIR. 24 PAIRS OF LINES WILL BE PRINTED, EACH PAIR SEPARATED BY A BLANK LINE.

H. TEST 54 NON-PRINTING CHARACTER CODES TEST

THIS TEST SENDS ASCII CODES 00 THROUGH 37 TO THE PRINTER. THE CHARACTER PRINTED SHOULD BE "w" FOR ALL PRINT LINES, SINCE THIS IS THE "HOME" POSITION OF THE PRINT WHEEL, 10 LINES OF THE CHARACTER "w" SHOULD BE PRINTED.

I. TEST 55 PRINT WHEEL VIBRATION TEST

THIS TEST PRINTS 10 LINES CONSISTING OF "ACA:ACA:". THESE CHARACTERS ARE ADJACENT POSITIONS ON THE PRINT WHEEL WITH LETTER "A" BETWEEN "C" AND ";". THE PRINTING OF THIS PATTERN WILL RESULT IN A VERY RAPID VIBRATION OF THE PRINT WHEEL.

J. TEST 56 OPERATOR GENERATED PRINT LINE

THIS TEST ALLOWS THE OPERATOR TO TYPE IN A CHARACTER STRING OF UP TO 132 CHARACTERS AND CAUSE THIS TEXT STRING TO BE PRINTED ON THE LQ PRINTER BY TERMINATING THE CHARACTER STRING WITH A CARRIAGE RETURN. THE LQ PRINTER WILL REPEAT THIS CHARACTER STRING UNTIL ANOTHER CHARACTER IS TYPED FOLLOWING THE CARRIAGE RETURN.

K. TEST 57 RAISE AND LOWER RIBBON

THIS TEST IS DESIGNED FOR TROUBLESHOOTING THE RIBBON LIFT MECHANISM. THE PRINTER IS RESTORED AND THEN THE RIBBON LIFTS AND DROPS AT A RATE DETERMINED BY A DELAY SUBROUTINE.

L. TEST 60 FRICTION FEED TEST 10 BY 10 RANDOM MATRIX

THIS TEST VERIFIES THE FRICTION FEED, BI-DIRECTIONAL PAPER MOVEMENT CAPABILITIES OF THE PRINTER. A CHARACTER MATRIX, 10 BY 10 CHARACTERS (9X9 INCHES) IS PRINTED WITH ONE (1) INCH SPACING BETWEEN CHARACTERS. THE PRINT POSITIONS ARE SELECTED RANDOMLY. THE MATRIX MAY ALSO BE PRINTED AS DOTS BY SETTING LOCATION "CSWTCH" TO ALL ZEROS.

4.5.1 RUN DYNAMIC PRINTER TEST ON A NON-ACTIVE CONSOLE PACKAGE

-
- A. LOAD ADDRESS TO 201.
 - B. SELECT ANY SWITCH REGISTER OPTIONS (SEE SECTION 6.0).
 - C. PRESS "CLEAR" THEN "CONT" ON A PDP-8E, F, M OR "INIT" THEN "RUN" ON A PDP-8A.
 - D. THE PROGRAM WILL CYCLE THROUGH TESTS 45 THROUGH 55 CONTINUOUSLY UNLESS HALT AT END OF PASS IS SELECTED (SEE SECTION 6.0).

4.5.2 RUN DYNAMIC PRINTER TEST ON AN ACTIVE CONSOLE PACKAGE

- A. SELECT ANY SWITCH REGISTER OPTIONS (SEE SECTION 6.0).
- B. START THE PROGRAM AT ADDRESS 201.
- C. THE PROGRAM WILL CYCLE THROUGH TESTS 45 THROUGH 55 CONTINUOUSLY UNLESS HALT AT END OF PASS IS SELECTED (SEE SECTION 6.0).
- D. REFER TO SECTION 4.7 FOR CONSOLE PACKAGE OPERATION.

4.6 RUN SPECIAL DYNAMIC PRINTER TESTS

4.6.1 REQUIREMENTS

- A. TEST 56 REQUIRES A CONSOLE AND THAT CONSOLE PACKAGE BE ACTIVE (SEE SECTION 4.7).
- B. TEST 60 REQUIRES THAT THE FORMS TRACTOR BE REMOVED AND THE PAPER FED FROM THE REAR OF THE PLATEN (REFER TO MAINTENANCE MANUAL).

4.6.2 PROCEDURES

A. TEST 56 OPERATOR GENERATED PRINT LINE

WITH CONSOLE PACKAGE ACTIVE AND TERMINAL ON-LINE, THE OPERATOR MAY INPUT UP TO 132 CHARACTERS, WHICH WILL BE ECHOED ON THE TERMINAL WHEN A CARRIAGE RETURN IS TYPED, THE ENTIRE CHARACTER STRING WILL BE OUTPUT TO THE LQ PRINTER CONTINUOUSLY UNTIL ANOTHER CHARACTER IS TYPED ON THE CONSOLE TERMINAL. THE PROGRAM WILL THEN ACCEPT A NEW CHARACTER STRING TERMINATED BY A CARRIAGE RETURN.

TO RUN:

1. START THE PROGRAM AT ADDRESS 203.
2. TYPE CONTROL-G.
3. PROGRAM PRINTS "SR= " AND WAITS FOR KEYBOARD INPUT OF NEW PSEUDO SWITCH REGISTER CONTENTS.
4. TYPE "56" FOLLOWED BY CARRIAGE RETURN.
5. THIS WILL CAUSE THE PROGRAM TO BEGIN TEST 56.
6. THE PROGRAM IS NOW WAITING FOR INPUT OF THE CHARACTER STRING TO BE PRINTED ON THE LQP-8 PRINTER.
7. TYPING CONTROL-T FOLLOWED BY CONTROL-G WILL AGAIN ASK THE "SR= " QUESTION AND A NEW TEST NUMBER MAY NOW BE INPUT IN PSEUDO SWITCH REGISTER BITS 6-11.

NOTE: THE TESTS IN THIS PROGRAM ARE NUMBERED IN OCTAL FORMAT. TYPING OF AN INVALID TEST NUMBER WILL RESULT IN PROGRAM TRANSFER TO TEST 1 OF LOGIC TESTS. VALID TEST NUMBERS ARE 1 THROUGH 60 (OCTAL).

B. TEST 57 RAISE AND LOWER RIBBON

THIS TEST WILL RAISE AND LOWER THE RIBBON ON THE LQP-8 PRINTER AT A RATE DETERMINED BY THE OCTAL NUMBER FOLLOWING

A CALL TO A DELAY SUBROUTINE (SEE PROGRAM LISTING TST57)
THIS TEST IS DESIGNED FOR TROUBLESHOOTING OF THE RIBBON
LIFT MECHANISM.

TO RUN:

CONSOLE NON-ACTIVE:

1. SET HARDWARE OR PSUEDO SWITCH REGISTER EQUAL TO 0057.
2. START THE PROGRAM AT ADDRESS 203.
3. TEST WILL LOOP CONTINUOUSLY.

CONSOLE ACTIVE:

1. START THE PROGRAM AT ADDRESS 203.
2. TYPE CONTROL-G.
3. PROGRAM PRINTS "SR= " AND WAITS FOR KEYBOARD INPUT OF NEW PSUEDO SWITCH REGISTER CONTENTS.
4. TYPE "57" FOLLOWED BY CARRIAGE RETURN.
5. THIS WILL CAUSE THE PROGRAM TO BEGIN TEST 57.
6. TYPING CONTROL-T FOLLOWED BY CONTROL-G WILL AGAIN ASK THE "SR= " QUESTION, AND A NEW TEST NUMBER MAY NOW BE INPUT IN PSUEDO SWITCH REGISTER BITS 6-11.

NOTE: THE TESTS IN THIS PROGRAM ARE NUMBERED IN OCTAL
FORMAT. TYPING OF AN INVALID TEST NUMBER WILL
RESULT IN PROGRAM TRANSFER TO TEST 1 OF LOGIC
TESTS. VALID TEST NUMBERS ARE 1 THROUGH 60
(OCTAL).

C. TEST 60 FRICTION FEED TEST

THIS TEST PRINTS 10 BY 10 CHARACTER OR DOT MATRIX. THE
SPACING BETWEEN CHARACTERS IS SET TO ONE(1) INCH. THIS
SPACING MAY BE CHANGED BY CHANGING THE CONTENTS OF
LOCATION "LININ" FOR LINE SPACING AND LOCATION "COLIN"
FOR COLUMN SPACING (REFER TO PROGRAM LISTING OF TST60).

NOTE: THE FORMS TRACTOR MUST BE REMOVED AND THE PAPER
FED FROM REAR OF PLATEN (REFER TO MAINTENANCE
MANUAL).

THE PRINT PATTERN SHOULD APPEAR AS FOLLOWS:

```
A B C D E F G H I J  
B C D E F G H I J K  
C D E F G H I J K L  
D E F G H I J K L M
```

```
E F G H I J K L M N  
F G H I J K L M N O  
G H I J K L M N O P  
H I J K L M N O P Q  
I J K L M N O P Q R  
J K L M N O P Q R S
```

FOR BEST RESULTS THE ENTIRE PATTERN SHOULD BE PRINTED
ON ONE (1) PAGE, ADJUST THE PAPER FOR TOP OF FORM ABOUT
ONE INCH ABOVE PRINT WHEEL.

TO CHANGE THE PATTERN FROM CHARACTERS TO DOTS, SET LOCATION CSWICH (REFER TO PROGRAM LISTING OF TST60) TO ALL ZEROS.

REFER TO SECTION 4.5 L. FOR FURTHER TEST DESCRIPTION.

TO RUN:

CONSOLE NON-ACTIVE:

1. SET HARDWARE OR PSUEDO SWITCH REGISTER EQUAL TO 0060.
2. START THE PROGRAM AT ADDRESS 203.
3. TEST WILL LOOP CONTINUOUSLY.

CONSOLE ACTIVE:

1. START THE PROGRAM AT ADDRESS 203.
2. TYPE CONTROL-G.
3. PROGRAM PRINTS "SR= " ON CONSOLE AND WAITS FOR KEYBOARD INPUT OF NEW PSUEDO SWITCH REGISTER CONTENTS.
4. TYPE "60" FOLLOWED BY CARRIAGE RETURN.
5. THIS WILL CAUSE THE PROGRAM TO BEGIN TEST 60.
6. TYPING CONTROL-T FOLLOWED BY CONTROL-G WILL AGAIN ASK THE "SR= " QUESTION, AND A NEW TEST NUMBER MAY NOW BE INPUT IN PSUEDO SWITCH REGISTER BITS 6-11.

NOTE: THE TESTS IN THIS PROGRAM ARE NUMBERED IN OCTAL FORMAT. TYPING OF AN INVALID TEST NUMBER WILL RESULT IN PROGRAM TRANSFER TO TEST 1 OF LOGIC TESTS. VALID TEST NUMBERS ARE 1 THROUGH 60 (OCTAL).

4.7 OPERATION OF CONSOLE PACKAGE -----

THE CONSOLE PACKAGE USES THREE LOCATIONS ON PAGE 0 FIELD 0 FOR SPECIAL FUNCTIONS. THESE LOCATIONS ARE:

LOC 20	PSUEDO SWITCH REGISTER
LOC 21	HARDWARE CONFIGURATION WORD ONE
LOC 22	HARDWARE CONFIGURATION WORD TWO

SEE SECTION 6 FOR SWITCH REGISTER SETTINGS.

LOCATION 21

BIT 0 = 0	USE PSUEDO SWITCH REGISTER
BIT 0 = 1	USE HARDWARE SWITCH REGISTER

LOCATION 22

BIT 3 = 0	CONSOLE PACKAGE INACTIVE
BIT 3 = 1	CONSOLE PACKAGE ACTIVE

4.7.1 CONTROL CHARACTERS -----

CTRL/C	RETURN TO OPERATING SYSTEM MONITOR.
--------	-------------------------------------

CTRL/G	ALLOWS CHANGING OF SOFTWARE PSUEDO SWITCH REGISTER, RESTARTS PROGRAM IF FOLLOWED BY A LINE FEED. CONTINUES PROGRAM IF FOLLOWED BY CARRIAGE RETURN.
CTRL/S	STOP PROGRAM AND WAIT FOR A CONTROL CHARACTER.
CTRL/G	CONTINUE PROGRAM FROM CTRL/S.
CTRL/K	RESTORE LQ PRINTER AND RESTART CURRENT TEST.
CTRL/T	PROGRAM WAITS FOR A CTRL/G, ASKS "SR= " QUESTION, DESIRED TEST NUMBER IS NOW ENTERED INTO PSUEDO SWITCH REGISTER BITS 6-11.

4.7.2 SELECTING TEST USING PSUEDO SWITCH REGISTER

STARTING THE PROGRAM AT LOCATION 203 IS EQUIVALENT TO TYPING CTRL/T WHILE THE PROGRAM IS RUNNING. THE PROGRAM WAITS FOR THE USER TO TYPE CTRL/G AND ENTER THE DESIRED TEST NUMBER IN PSUEDO SWITCH REGISTER BITS 6-11.

EXAMPLE:

- A. TYPE CTRL/T WHILE PROGRAM IS RUNNING OR START PROGRAM AT ADDRESS 203.
- B. TYPE CTRL/G PROGRAM ASKS "SR= " QUESTION.
- C. USER TYPES 0045 FOLLOWED BY CARRIAGE RETURN.
- D. PROGRAM WILL NOW BEGIN TEST 45.

NOTE: TESTS ARE NUMBERED IN OCTAL TO PERMIT USE OF PSUEDO SWITCH REGISTER FOR TEST SELECTION.

4.7.3 CHANGING PSUEDO SWITCH REGISTER

- A. WHILE PROGRAM IS RUNNING, USER TYPES CTRL/G ON CONSOLE TERMINAL.
- B. PROGRAM RESPONDS BY PRINTING "G SR=XXXX" WHERE XXXX IS THE CURRENT PSUEDO SWITCH REGISTER CONTENTS.
- C. USER NOW TYPES UP TO FOUR(4) OCTAL DIGITS WHICH WILL BE ENTERED FROM RIGHT TO LEFT INTO THE PSUEDO SWITCH REGISTER. IF A CHARACTER OTHER THAN 0 THRU 7 IS TYPED, THE PROGRAM WILL REASK THE "SR= " QUESTION.
- D. AFTER TYPING NEW PSUEDO SWITCH REGISTER CONTENTS, USER TYPES CARRIAGE RETURN TO CONTINUE THE PROGRAM OR LINE FEED TO RESTART THE CURRENT SECTION OF TESTS. SEE SECTION 6 FOR SWITCH REGISTER OPTIONS.

5.0 ERRORS

IF THE PROGRAM DETECTS AN ERROR EITHER IN DATA OR STATUS OF THE LQ PRINTER A NUMBER OF OPTIONS ARE AVAILABLE TO THE USER:

- A. INHIBIT ERROR HALT
- B. LOOP ON ERROR
- C. LOOP ON TEST
- D. INHIBIT ERROR TIMEOUT

REFER TO SECTION 6 FOR SWITCH SETTINGS TO SELECT THE DESIRED OPTIONS.

5.1 LOGIC ERRORS

DURING THE RUNNING OF THE LOGIC TESTS, IF THE PROGRAM DETECTS A FAULT IN THE INTERFACE LOGIC THE PROGRAM WILL EITHER HALT IF CONSOLE IS NOT ACTIVE OR PRINT AN ERROR MESSAGE IF CONSOLE IS ACTIVE.

5.1.1 LOGIC ERRORS ON A NON-ACTIVE CONSOLE

- A. PROGRAM WILL HALT WITH THE CONTENTS OF THE PROGRAM COUNTER AT THE TIME OF ERROR IN THE ACCUMULATOR.
- B. PRESSING "CONTINUE" WILL RESTORE THE AC TO ITS VALUE AT THE TIME OF THE ERROR AND PROGRAM WILL HALT AGAIN.
- C. PRESSING "CONTINUE" AGAIN WILL CAUSE THE PROGRAM TO RESUME AND EITHER RESTART FAILING TEST OR CONTINUE FROM THE ERROR, DEPENDING ON SWITCH SETTINGS. (SEE SECTION 6).

5.1.2 LOGIC ERRORS ON AN ACTIVE CONSOLE

- A. PROGRAM WILL PRINT THE FOLLOWING ERROR MESSAGE:

DHLQA FAILED

PC=XXXX AC=XXXX MQ=XXXX FL=XXXX
WAITING
- B. THE PROGRAM IS NOW WAITING FOR CTRL/G TO BE TYPED BY USER TO ALLOW SETTING OF PSEUDO SWITCH REGISTER OPTIONS (SEE SECTION 6).

5.2 STATUS ERRORS

THE FOLLOWING STATUS REGISTER BITS ARE OF INTEREST TO THE PROGRAM:

- BIT 0 PRINTER READY (POWER ON)
- BIT 1 CHARACTER READY
- BIT 2 CARRIAGE READY
- BIT 3 PAPER READY
- BIT 6 CHECK ERROR

DURING PRINTER OPERATION (DYNAMIC TESTS) THE PROGRAM WAITS FOR THE APPROPRIATE READY BIT TO EQUAL A ONE (1) PRIOR TO ISSUING THE CORRESPONDING IOT. IF THE READY BIT DOES NOT RETURN WITHIN FIVE (5) SECONDS, THE PROGRAM CONSIDERS THIS TO

BE A STATUS ERROR. IF A CHECK ERROR OCCURS THE PROGRAM WILL DETECT THIS AND GO TO THE STATUS ERROR HANDLING ROUTINE.

5.2.1 STATUS ERRORS ON A NON-ACTIVE CONSOLE

- A. PROGRAM WILL HALT WITH PC AT TIME OF ERROR IN THE AC.
- B. PRESSING "CONTINUE" WILL CAUSE THE PROGRAM TO PLACE EXPECTED STATUS IN AC AND HALT.
- C. PRESSING "CONTINUE" WILL CAUSE PROGRAM TO PLACE ACTUAL STATUS IN AC AND HALT.
- D. PRESSING "CONTINUE" AGAIN WILL CAUSE THE PROGRAM TO RESUME AND EITHER RESTART THE FAILING TEST OR CONTINUE FROM ERROR DEPENDING ON SWITCH REGISTER SETTINGS (SEE SECTION 6).

5.2.2 STATUS ERRORS ON AN ACTIVE CONSOLE

- A. PROGRAM WILL PRINT THE FOLLOWING ERROR MESSAGE:

LQP-8 STATUS ERROR

PC = XXXX
EXPECTED = XXXX ACTUAL = XXXX
WAITING
- B. THE PROGRAM IS NOW WAITING FOR CTRL/G TO BE TYPED BY USER TO ALLOW SETTING OF PSEUDO SWITCH REGISTER OPTIONS (SEE SECTION 6).

5.3 DATA ERRORS

5.3.1 DATA ERRORS ON A NON-ACTIVE CONSOLE

- A. PROGRAM WILL HALT WITH PC AT TIME OF ERROR IN THE AC.
- B. PRESSING "CONTINUE" WILL CAUSE THE PROGRAM TO PLACE THE DATA SENT IN THE AC AND HALT.
- C. PRESSING "CONTINUE" WILL CAUSE THE PROGRAM TO PLACE THE DATA READ BACK IN THE AC AND HALT.
- D. PRESSING "CONTINUE" AGAIN WILL CAUSE THE PROGRAM TO RESUME AND EITHER RESTART THE FAILING TEST OR CONTINUE FROM THE ERROR DEPENDING ON SWITCH REGISTER SETTINGS (SEE SECTION 6).

5.3.2 DATA ERRORS ON AN ACTIVE CONSOLE

- A. THE PROGRAM WILL PRINT THE FOLLOWING ERROR MESSAGE:

LQP-8 DATA ERROR

PC = XXXX SENT: XXXX RECEIVED: XXXX
WAITING
- B. THE PROGRAM IS NOW WAITING FOR CTRL/G TO BE TYPED BY THE USER TO ALLOW SETTING OF PSEUDO SWITCH REGISTER OPTIONS (SEE SECTION 6).

6.0 SWITCH REGISTER SETTINGS

6.1 NORMAL OPERATION

ALL SWITCHES = 0

6.2 ERROR RELATED SWITCHES

SR0 = 1 INHIBIT ERROR HALT
SR1 = 1 LOOP ON ERROR
SR2 = 1 LOOP ON TEST
SR3 = 1 HALT ON COMPLETION OF A PROGRAM PASS
SR4 = 1 INHIBIT ERROR TIMEOUT

7.0 REVISIONS

REVISION B - MODIFICATIONS TO RUN ON VT78 SYSTEMS.

8.0 ADDITIONAL INFORMATION

8.1 STARTING ADDRESSES

LOC 200 LOGIC TESTS
LOC 201 PRINTER TESTS
LOC 202 CHANGE DEVICE CODE
LOC 203 SELECT TEST USING SWITCH REGISTER

8.2 SPECIAL ADDRESSES

THE PROGRAM USES THREE LOCATIONS TO DETERMINE CHARACTER SPACING, LINE SPACING, AND NUMBER OF CHARACTERS PER LINE. THE PROGRAM IS INITIALLY SET TO 132 CHARACTERS PER LINE, 10 CHARACTERS PER INCH, AND 6 LINES PER INCH. THESE LOCATIONS ARE:

SYMBOLIC NAME	ADDRESS	VALUE	
-----	-----	-----	
SPACE	125	0014	12/120 CHARACTER SPACING
LF	126	0020	8/48 LINE SPACING
LTNE	127	0204	132 CHARACTERS PER LINE

NOTE: SET LOCATION "LINE" TO 0110 IF 8 INCH WIDE PAPER IS BEING USED. THIS IS EQUIVALENT TO 72 CHARACTERS PER LINE.

THESE VALUES CORRESPOND WITH A 10 PITCH CHARACTER WHEEL. TO SET THESE VALUES FOR A 12 PITCH CHARACTER WHEEL CHANGE THESE VALUES TO THE FOLLOWING:

SPACE	125	0012	10/120 CHARACTER SPACING
LF	126	0016	6/48 LINE SPACING
LTNE	127	0236	158 CHARACTERS PER LINE

NOTE: SET LOCATION "LINE" TO 0126 IF 8 INCH WIDE PAPER IS BEING USED. THIS IS EQUIVALENT TO 86 CHARACTERS PER LINE.

A. TST47 RMOST LOC 1534 VALUE 3044

SET THIS LOCATION TO 1700 IF USING 8 INCH PAPER.

B. TST60 CSWICH LOC 3111 VALUE 7777

SET THIS LOCATION TO 0000 FOR DOTS INSTEAD OF AN ALPHABETICAL PATTERN.

C. TWO SPECIAL ROUTINES EXIST IN THIS PROGRAM, WHICH ARE NOT EXECUTED DURING THE NORMAL TEST SEQUENCE. THEY ARE FOR USE ONLY BY PERSONS QUALIFIED TO MAKE MECHANICAL ADJUSTMENTS ON THE LQPS PRINTER.

1. RESTORE LOOP

THIS IS A ROUTINE TO ISSUE CONTINUOUS RESTORE INSTRUCTIONS TO THE LQPS PRINTER. START THE PROGRAM AT THE LOCATION LABELED "RESTLP", WHICH IS LOCATED IMMEDIATELY FOLLOWING TEST 60. THIS LOOP WILL RUN CONTINUOUSLY UNTIL A CONTROL CHARACTER IS TYPED BY THE OPERATOR.

2. RAISE OR LOWER RIBBON

THIS IS A ROUTINE TO RAISE OR LOWER THE RIBBON OF THE LQPS PRINTER, TO ALLOW ADJUSTMENT OF RIBBON HEIGHT. START THE PROGRAM AT THE LOCATION LABELED "RIBBON", WHICH IS LOCATED IMMEDIATELY FOLLOWING TEST 60, (SEE PROGRAM LISTING). EACH TIME A KEY IS TYPED ON THE OPERATOR'S TERMINAL, THE RIBBON WILL EITHER RAISE OR LOWER (DEPENDING ON ITS PREVIOUS CONDITION). TYPING A VALID CONTROL CHARACTER (SEE SECTION 4.7)

WILL TERMINATE THIS ROUTINE.

9.0 LISTING

```

1          /LQP-8 PRINTER DIAGNOSTIC
2          /
3          0000      *0
4          0000      LOCO, 0
5          0001      5402      ADDR81, JMP 1      ADDR82
6          0002      4000      ADDR82, INTSVC      /GO TO INTERRUPT SERVICE ROUTINE
7          0003      4015      ADDR83, RESTRT
8          /
9          /
10         0010      *10
11         0010      0000      INDX10, 0
12         0011      0000      INDX11, 0
13         0012      0000      INDX12, 0
14         0013      0000      INDX13, 0
15         0014      0000      INDX14, 0
16         0020      *20
17         /
18         /
19         0020      0000      PSWR, 0      /PSUEDO SWITCH REGISTER
20         0021      0000      OPISL, 0000      /HARDWARE CONFIGURATION WORD ONE
21         0022      0400      DP2SEL, 0400      /HARDWARE CONFIGURATION WORD TWO
22
23         4423      LOSK= JMS I .      /SKIP ON DONE FLAG IOTO
24         3200      LOSKRT
25         4424      LQRB= JMS I .      /READ DATA BUFFER INTO AC IOT1
26         3205      LQRBRT
27         4425      LQMP= JMS I .      /LOAD DATA BUFFER, MOVE PAPER, CLR AC, IOT2
28         3214      LQMPRT
29         4426      LQMC= JMS I .      /LOAD DATA BUFFER, MOVE CARRIAGE IOT3
30         3223      LQMCRT
31         4427      LQPC= JMS I .      /LOAD DATA BUFFER, PRINT CHAR IN AC4-10 IOT4
32         3232      LQPCRT
33         4430      LQRS= JMS I .      /READ STATUS REG, CLR DONE IOT5
34         3237      LQRSRT
35         4431      LQLS= JMS I .      /LOAD STATUS REG, SET DONE IOT6
36         3246      LQLSRT
37         4432      LQRE= JMS I .      /RESTORE CARRIAGE, CHECK RECOVERY IOT7
38         3260      LQRERT
39         4433      BGN1ST= JMS I .      /INITIALIZE AND SET UP FOR TEST
40         3400      TSTBGN
41         4434      ERROR= JMS I .      /COMMON ERROR HANDLER
42         3600      ERRORS
43         4435      LOADDB= JMS I .      /ROUTINE TO LOAD DATA BUFFER
44         3265      DBLDB
45         4436      IN1ST= JMS I .      /ROUTINE TO INITIALIZE FOR TEST WITH PRINTER
46         3324      TSTIN      /OFF LINE
47         4437      DATERR= JMS I .      /ROUTINE TO HANDLE DATA ERRORS
48
49         0037      3430      ERFDAT
50         4440      ENDTST= JMS I .      /ROUTINE TO HANDLE SCOPE LOOPING AT TEST END
51         3342      TSTEND      /IF ERROR ENCOUNTERED
52         4441      WAIT= JMS I .      /WAIT FOR INDIVIDUAL READY BITS ACTIVE
53         4033      AWAIT
54         4442      SWAIT= JMS I .      /WAIT FOR ALL READY BITS ACTIVE
55         4063      WAITS
56         4443      DELAY= JMS I .      /VARIABLE DELAY ROUTINE

```

```

56         0043      5244      ADELAY
57         4444      ECHO= JMS I .      /ROUTINE TO ECHO CHARACTER TYPED ON CONSOLE
58         4673      XECHO
59         4445      CTLCHK= JMS I .      /ROUTINE TO CHECK FOR A VALID CONTROL CHARACTER
60         4630      CHKCTL
61         4446      RDYCHK= JMS I .      /CHECK READY BITS
62         3312      CHKRDY
63         4447      SAVEAC= JMS I .      /ROUTINE TO SAVE CONTENTS OF AC AND LINK
64         4200      ACSAVE
65         4450      RSTRAC= JMS I .      /ROUTINE TO RESTORE CONTENTS OF AC AND LINK
66         4205      ACRSTR
67         4451      STATER= JMS I .      /STATUS ERROR HANDLER
68         4213      ERSTAT
69         4452      KBCHK= JMS I .      /ROUTINE TO CHECK FOR KEYBOARD INTERVENTION
70         4610      CHKKB
71         4453      ROTCHR= JMS I .      /ROUTINE TO SHIFT 7 BIT ASCII ONE PLACE LEFT
72         4526      CHRRUT
73         4454      LINEND= JMS I .      /CHECK FOR END OF LINE
74         4534      ENDLIN
75         4455      MOVCAR= JMS I .      /MOVE CARRIAGE SUBROUTINE
76         4400      CARMOV
77         4456      MOVCR= JMS I .      /MOVE CARRIAGE TO ABSOLUTE COORDINATE
78         4441      CRMV
79         4457      MOVPAR= JMS I .      /MOVE PAPER SUBROUTINE
80         4316      PARMUV
81         4460      LSTLIN= JMS I .      /CHECK FOR LAST LINE
82         4546      LINLST
83         4461      XCRLF= JMS I .      /EXECUTE CARRIAGE RETURN, LINE FEED
84         4471      CRLFRT
85         4462      RSPACE= JMS I .      /SPACE ONE PRINT POSITION TO THE RIGHT
86         4557      SPACER
87         4463      LSPACE= JMS I .      /SPACE ONE PRINT POSITION TO THE LEFT
88         4600      SPACEL
89         4464      PRINT= JMS I .      /ROUTINE TO PRINT ONE CHARACTER ON LP
90         4507      XPRINT
91         4465      LAS= JMS I .      /ROUTINE TO HANDLE LAS INSTRUCTION
92         5274      LASRT
93         4466      CBACTV= JMS I .      /ROUTINE TO CHECK FOR ACTIVE CONSOLE
94         4306      ACTVCB
95         4467      MESSAGE= JMS I .      /MESSAGE ROUTINE
96         4702      MESPRT
97         4470      REGPRT= JMS I .      /ROUTINE TO PRINT THE CONTENTS OF A REGISTER
98         5016      PRTRREG
99         4471      MANCNG= JMS I .      /ROUTINE TO CHANGE DEVICE SELECT CODES
100        0071      5477      CNGMAN
101        4472      RANDOM= JMS I .
102        4112      XRAND
103        4473      SAVGEN= JMS I .
104        4126      XSAVGN
105        4474      RESGEN= JMS I .
106        4135      XRESGN
107        4475      VT8CK= JMS I .
108        0247      CKVT8
109        0117      STMASK, 117      /STATUS REGISTER USED BIT MASK
110        0000      SVAC, 0

```

```

111 0100 0000 SAVAC, 0
112 0101 0000 SAVLNK, 0
113 0102 0000 SAVMQ, 0
114 0103 0000 SAVFLG, 0
115 0104 0003 K003, 0003
116 0105 0007 K007, 0007
117 0106 0021 K021, 0021
118 0107 0023 K023, 0023
119 0110 6000 K6000, 6000
120 0111 5252 C5252, 5252
121 0112 2525 C2525, 2525
122 0113 0000 XDATA, 0 /SAVE AREA FOR EXPECTED DATA
123 0114 0000 SVSTAT, 0 /SAVE STATUS READ
124 0115 0000 LINCNT, 0 /SAVE LINE COUNT
125 0116 0000 INTFLG, 0 /SAVE INTERRUPT CONDITION
126 0117 0000 XSTAT, 0 /SAVE AREA FOR EXPECTED STATUS
127 0120 0000 SAVMV, 0 /SAVE AREA FOR MOVE CARRIAGE MAGNITUDE
128 0121 0000 SAVMOV, 0 /SAVE CARRIAGE MOVE MAGNITUDE
129 0122 0000 SAVDAT, 0 /SAVE AREA FOR EXPECTED STATUS
130 0123 0000 SAVPR, 0 /SAVE AREA FOR PAPER MOVEMENT MAGNITUDE
131 0124 0000 CHAR, 0 /SAVE AREA FOR CHARACTER TO PRINT
132 0125 0000 CHARRT, 0 /SAVE AREA FOR CHARACTER SHIFTED LEFT ONE PLACE
133 0126 0000 CARDIR, 0 /SAVE AREA FOR CARRIAGE DIRECTION
134 0127 0014 SPACE, 14 /SAVE AREA FOR CHARACTER SPACING
135 0130 0020 LF, 20 /SAVE AREA FOR LINE FEED MAGNITUDE
136 0131 0204 LINE, 204 /132 CHARACTERS PER LINE
137 0132 0000 LINES, 0 /NUMBER OF LINES TO PRINT
138 0133 0000 WSTAT, 0 /WANTED STATUS SAVE AREA
139 0134 0000 FIRST, 0 /FIRST CHARACTER OF A TEST
140 0135 0000 FSTCHR, 0 /FIRST CHARACTER OF A PATTERN
141 0136 0000 CHRCNT, 0 /SAVE CHARACTER COUNT
142 0137 0000 KESAV, 0 /SAVE KEYBOARD CHARACTER
143 0140 0000 SAVPOS, 0 /SAVE CARRIAGE POSITION
144 0141 0000 MOVSMAG, 0 /SAVE CARRIAGE MOVEMENT MAGNITUDE
145 0142 0500 DSCNOW, 0500 /CURRENT DEVICE SELECT CODE
146 6007 CAF=6007
147 6102 SPL=6102
148 7501 MGA=7501
149 7421 MQL=7421
150 6030 KCF=6030
151 6004 GTF=6004
152 6005 RTF=6005
153
154 7002 BSW=7002
155 6035 KIE=6035
156 /
157 /

158 / /STATUS REGISTER BITS
159 /
160 4000 LQRDY=4000 /BIT 0 PRINTER READY
161 2000 LQCHR=2000 /BIT 1 CHARACTER READY
162 1000 LQCAR=1000 /BIT 2 CARRIAGE READY
163 0400 LQPAR=0400 /BIT 3 PAPER READY
164 3400 LQDON=3400 /BITS 1,2,3 ALL OPERATIONS DONE
165 0100 LPLFT=0100 /BIT 5 SELECT LEFT PLATEN

```

```

166 0040 LQCHK=0040 /BIT 6 CHECK ERROR CONDITION
167 0020 LQOUT=0020 /BIT 7 OUT OF PAPER
168 0004 LQRED=0004 /BIT 9 PRINT RED
169 0002 LQBLK=0002 /BIT 10 PRINT BLACK
170 0001 LQIEN=0001 /BIT 11 INTERRUPT ENABLE
171 7400 ALRDY=7400 /BITS 0,1,2,3, ALL READY BITS SET
172 /
173 /LQP-8 IOT'S NOT USED IN SUBROUTINE
174 6500 XLQSK=6500 /SKIP ON DONE FLAG
175 6501 XLQRB=6501 /READ DATA BUFFER
176 6502 XLQMP=6502 /MOVE PAPER
177 6503 XLQMC=6503 /MOVE CARRIAGE
178 6504 XLQPC=6504 /PRINT CHARACTER
179 6505 XLQRS=6505 /READ STATUS
180 6506 XLQLS=6506 /LOAD STATUS
181 6507 XLQRE=6507 /RESTORE
182 /
183 /DATA BUFFER CONSTANTS
184 /
185 4000 PRRET=4000 /PAPER RETRACT BIT 0 = ONE
186 0000 PRADV=0 /PAPER ADVANCE BIT 0 = ZERO
187 4000 CARLFT=4000 /MOVE CARRIAGE LEFT BIT 0 = ONE
188 0000 CARRGT=0 /MOVE CARRIAGE RIGHT BIT 0 = ZERO
189 /
190 0200 *200
191 /
192 /
193 0200 4204 JMS START /GO TO BEGINNING OF LOGIC TESTS
194 0701 4204 JMS START /GO TO BEGINNING OF PRINTER TESTS
195 0202 4204 JMS START /GO TO DSC MODIFICATION ROUTINE
196 0703 4204 JMS START /GO TO SELECTED TEST
197 /
198 /
199 0704 0000 START, 0
200 0705 7340 CLA CLL CMA
201 0706 1204 TAD START
202 0707 3230 DCA SAVADD
203 0710 6030 KCF /CLEAR KYBD FLAG
204 0711 4466 C8ACTV /CHECK FOR ACTIVE CONSOLE
205 0712 7410 SKP /SKIP IF CONSOLE IS ACTIVE
206 0713 5216 JMP ,+3 /CONSOLE NOT ACTIVE, SKIP MESSAGE
207 0714 4467 MESSAGE
208 0715 6160 MANDEC
209 0716 3777 DCA PASS /CLEAR PASS COUNT
210 0717 1376 TAD (-200
211 0720 1230 TAD SAVADD
212 0721 1375 TAD (STARTB
213 0722 3231 DCA GOADD
214 0723 1631 TAD I GOADD
215 0724 3232 DCA RSTART
216 0725 4475 VT78CK /VT78/ SYSTEM UNDER TEST = VT78 ?
217 0726 4737 JMS INSW /VT78/ YES = INSERT OVERWRITES FOR VT78 TESTING
218 0727 5632 JMP I RSTART
219 /
220 0730 0000 SAVADD, 0

```



```

221 0231 0000 GOADD, 0
222 0232 0000 RSTART, 0 /RESTART ADDRESS
223 0233 0400 STARTB, TST1
224 0234 1400 TST45
225 0235 5456 IOTCNG
226 0236 5623 TSELCT
227 /
228 /
229 /VT78/ ROUTINE TO INSERT OVERWRITES FOR TESTING ON VT78 SYSTEM.
230
231 0237 0000 INSOW, 0 /VT78/
232 0240 1374 TAD C003 /VT78/
233 0241 3076 DCA STMASK /VT78/ OVERWRITE STATUS REG MASK
234 0242 1373 IAD (NOP /VT78/ OVERWRITES TO ACCOUNT FOR DATA BUFFER INVERSION IN VT78
235 0243 3772* DCA V780W1 /VT78/
236 0244 1371 TAD (IAC /VT78/
237 0245 3770* DCA V780W2 /VT78/
238 0246 5637 JMP I INSOW /VT78/
239
240 /
241 /VT78/ ROUTINE TO CHECK IF SYSTEM UNDER TEST IS A VT78
242
243 0247 0000 CKVT78, 0 /VT78/
244 0250 1022 TAD OP2SEL /VT78/ SYS UNDER TEST = VT78 ? (LOC 22 BIT 2 = 1)
245 0251 7006 RTL /VT78/
246 0252 7710 SPA CLA /VT78/
247 0253 5647 JMP I CKVT78 /VT78/ YES - RETURN TO CALL+1
248 0254 2247 LSZ CKVT78 /VT78/
249 0255 5647 JMP I CKVT78 /VT78/ NO - RETURN TO CALL+2
250
251 0270 3301
252 0271 7001
253 0272 3270
254 0273 7000
255 0274 0003
256 0275 0233
257 0276 7600
258 0277 1365
259 0400 *400
260 /ROUTINE TO LOAD ALL ONES AND ZEROS USING IOT2
261 /
262 0400 4436 TST1, INTST /INIT AND SET UP FOR SCOPE LOOP
263 0401 7340 CLA CLL CMA /SET UP TO LOAD ALL ONES
264 0402 4435 LOADDB /LOAD DATA BUFFER AND CHECK FOR DATA ERRORS
265 0403 4425 LGMP
266 0404 4437 DATERR /DATA ERROR
267 0405 7300 CLA CLL /SET UP TO LOAD ALL ZEROS
268 0406 4435 LOADDB
269 0407 4425 LGMP
270 0410 4437 DATERR /DATA ERROR
271 0411 4440 ENDTST /CHECK FOR TEST LOOPING ON ERROR
272 /
273 /ROUTINE TO LOAD ALTERNATING ONES AND ZEROS USING IOT2
274 /

```

```

275 0412 4436 TST2, INTST
276 0413 1111 TAD C5252 /LOAD 5252
277 0414 4435 LOADDB
278 0415 4425 LGMP
279 0416 4437 DATERR /DATA ERROR
280 0417 1112 TAD C2525 /LOAD 2525
281 0420 4435 LOADDB
282 0421 4425 LGMP
283 0422 4437 DATERR /DATA ERROR
284 0423 4440 ENDTST
285 /
286 /ROUTINE TO FLOAT A ONE USING IOT2
287 /
288 0424 4436 TST3, INTST
289 0425 7301 CLA CLL IAC /AC=0001
290 0426 4435 LOADDB
291 0427 4425 LGMP
292 0430 4437 DATERR /DATA ERROR
293 0431 1077 TAD SVAC
294 0432 7100 CLL
295 0433 7004 RAL
296 0434 7420 SNL
297 0435 5226 JMP .-7 /LOOP UNTIL ALL BITS CHECKED
298 0436 4440 ENDTST
299 /
300 /ROUTINE TO FLOAT A ZERO USING IOT2
301 /
302 0437 4436 TST4, INTST
303 0440 7344 CLA CLL CMA RAL /AC=7776 L=1
304 0441 4435 LOADDB
305 0442 4425 LGMP
306 0443 4437 DATERR /DATA ERROR
307 0444 1077 TAD SVAC
308 0445 7120 STL
309 0446 7004 RAL
310 0447 7430 SZL
311 0450 5241 JMP .-7 /LOOP UNTIL ALL BITS CHECKED
312 0451 4440 ENDTST
313 /
314 /ROUTINE TO LOAD EVERY NUMBER USING IOT2
315 /
316 0452 4436 TST5, INTST
317 0453 7301 CLA CLL IAC
318 0454 4435 LOADDB
319 0455 4425 LGMP
320 0456 4437 DATERR
321 0457 1077 TAD SVAC
322 0460 7001 IAC
323 0461 7420 SNL
324 0462 5254 JMP .-6
325 0463 4440 ENDTST
326 /
327 /ROUTINE TO LOAD ALL ONES AND ZEROS USING IOT3
328 /
329 0464 4436 TST6, INTST

```

```

330 0465 7340      CLA CLL CMA      /AC=7777
331 0466 4435      LOADDB
332 0467 4426      LGMC
333 0470 4437      DATERR          /DATA ERROR
334 0471 7300      CLA CLL          /SET UP TO LOAD ALL ZEROS
335 0472 4435      LOADDB
336 0473 4426      LGMC
337 0474 4437      DATERR          /DATA ERROR
338 0475 4440      ENDTST         /CHECK FOR TEST LOOPING ON ERROR
339
340
341
342 0476 4436      /ROUTINE TO LOAD ALTERNATING ONES AND ZEROS USING IOT3
343
344 0477 1111      TST7, INTST
345 0500 4435      TAD C5252       /LOAD 5252
346 0501 4426      LOADDB
347 0502 4437      LGMC
348 0503 1112      DATERR          /DATA ERROR
349 0504 4435      TAD C2525
350 0505 4426      LOADDB
351 0506 4437      LGMC
352 0507 4440      DATERR          /DATA ERROR
353
354
355
356 0510 4436      /ROUTINE TO FLOAT A ONE USING IOT3
357
358 0511 7301      TST10, INTST
359 0512 4435      CLA CLL IAC     /AC=0001
360 0513 4426      LOADDB
361 0514 4437      LGMC
362 0515 1077      DATERR          /DATA ERROR
363 0516 7100      TAD SVAC
364 0517 7004      CLL
365 0520 7420      RAL
366 0521 5312      SNL
367 0522 4440      JMP .-7         /LOOP UNTIL ALL BITS CHECKED
368
369
370
371 0523 4436      /ROUTINE TO FLOAT A ZERO USING IOT3
372
373 0524 7344      TST11, INTST
374 0525 4435      CLA CLL CMA RAL /AC=7776 L=1
375 0526 4426      LOADDB
376 0527 4437      LGMC
377 0530 1077      DATERR          /DATA ERROR
378 0531 7120      TAD SVAC
379 0532 7004      STL
380
381
382
383 0533 7430      SZL
384 0534 5325      JMP .-7         /LOOP UNTIL ALL BITS CHECKED
385 0535 4440      ENDTST
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438

```

```

385 0540 4435      LOADDB
386 0541 4426      LGMC
387 0542 4437      DATERR          /DATA ERROR
388 0543 1077      TAD SVAC
389 0544 7001      IAC
390 0545 7420      SNL
391 0546 5340      JMP .-6
392 0547 4440      ENDTST
393
394
395
396 0550 4436      /ROUTINE TO LOAD ALL ONES AND ZEROS USING IOT4
397
398 0551 7340      TST13, INTST
399 0552 4435      CLA CLL CMA     /INIT AND SET UP FOR SCOPE LOOP
400 0553 4427      LOADDB          /SET UP TO LOAD ALL ONES
401 0554 4437      LGMC
402 0555 7300      DATERR          /DATA ERROR
403 0556 4435      CLA CLL          /SET UP TO LOAD ALL ZEROS
404 0557 4427      LOADDB
405 0560 4437      LGMC
406 0561 4440      DATERR          /DATA ERROR
407 0562 5777      ENDTST         /CHECK FOR TEST LOOPING ON ERROR
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438

```

```

439
440 0625 4436 /
441 0626 7344 TST16, INTST
442 0627 4435 CLA CLL CMA RAL /AC=7776 L=1
443 0630 4427 LOADDB
444 0631 4437 LQPC
445 0632 1077 DATERR
446 0633 7120 TAD SVAC
447 0634 7004 STL
448 0635 7430 RAL
449 0636 5227 SZL
450 0637 4440 JMP *-7 /LOOP UNTIL ALL BITS CHECKED
451 ENDTST
452 /
453 /ROUTINE TO LOAD EVERY NUMBER USING IOT4
454 0640 4436 TST17, INTST
455 0641 7301 CLA CLL IAC
456 0642 4435 LOADDB
457 0643 4427 LQPC
458 0644 4437 DATERR
459 0645 1077 TAD SVAC
460 0646 7001 IAC
461 0647 7420 SNL
462 0650 5242 JMP *-6 /LOOP UNTIL EVERY NUMBER TRIED
463 0651 4440 ENDTST
464 0652 4475 VT7ACK /VT78/ SYSTEM UNDER TEST = VT78 ?
465 0653 5266 JMP TST21 /VT78/ YES = SKIP TEST 20. IN VT78 CAF
466 /VT78/ DOES NOT CLEAR PRINTER DATA BUFFER.
467 /
468 /ROUTINE TO TEST THAT CAF INSTRUCTION CLEARS DATA BUFFER
469 /
470 0654 4436 TST20, INTST
471 0655 7340 CLA CLL CMA /AC=7777
472 0656 4435 LOADDB /SET UP ROUTINE FOR DATA CHECK
473 0657 4425 LQMP /LOAD DATA BUFFER WITH IOT2
474 0660 4437 DATERR /DATA ERROR
475 0661 6007 CAF /INIT BUS, SHOULD CLEAR DATA BUFFER
476 0662 4424 LQRB
477 0663 7440 SZA
478 0664 4434 ERROR /DATA BUFFER NOT CLEARED BY CAF INSTRUCTION
479 0665 4440 ENDTST
480 /
481 /ROUTINE TO CHECK THAT AC IS CLEARED BY IOT2
482 /
483 0666 4436 TST21, INTST
484 0667 7340 CLA CLL CMA /AC=7777
485 0670 4425 LQMP /LOAD DATA BUFFER WITH IOT2
486 0671 7440 SZA
487 0672 4434 ERROR /AC NOT CLEARED BY IOT2
488 0673 4440 ENDTST
489 /
490 /ROUTINE TO CHECK THAT AC IS CLEARED BY IOT3
491 /
492 0674 4436 TST22, INTST
493 0675 7340 CLA CLL CMA /AC=7777

```

```

494 0676 4426 LQMC /LOAD DATA BUFFER WITH IOT3
495 0677 7440 SZA
496 0700 4434 ERROR /AC NOT CLEARED BY IOT3
497 0701 4440 ENDTST
498 /
499 /ROUTINE TO CHECK THAT AC IS CLEARED BY IOT4
500 /
501 0702 4436 TST23, INTST
502 0703 7340 CLA CLL CMA /AC=7777
503 0704 4427 LQPC /LOAD DATA BUFFER WITH IOT4
504 0705 7440 SZA
505 0706 4434 ERROR /AC NOT CLEARED BY IOT4
506 0707 4440 ENDTST
507 /
508 /ROUTINE TO INSURE THAT AC IS NOT CHANGED BY IOT 0
509 /
510 0710 4436 TST24, INTST
511 0711 7340 CLA CLL CMA /AC = 7777
512 0712 4423 LQSK /ISSUE IOT 0
513 0713 7000 NOP
514 0714 7001 IAC /ADD ONE
515 0715 7440 SZA /AC SHOULD = 0
516 0716 4434 ERROR /AC CHANGED BY IOT 0
517 0717 7300 CLA CLL /AC = 0
518 0720 4423 LQSK /ISSUE IOT 0
519 0721 7000 NOP
520 0722 7440 SZA /AC STILL 0
521 0723 4434 ERROR /AC CHANGED BY IOT 0
522 0724 4440 ENDTST
523 /
524 /ROUTINE TO INSURE THAT AC IS NOT CHANGED BY IOT 7
525 /
526 0725 4436 TST25, INTST
527 0726 7340 CLA CLL CMA /AC = 7777
528 0727 4432 LGRE /ISSUE IOT 7
529 0730 7001 IAC /ADD ONE
530 0731 7440 SZA /AC SHOULD = 0
531 0732 4434 ERROR /AC CHANGED BY IOT7
532 0733 7300 CLA CLL /AC = 0
533 0734 4432 LGRE /IOT 7
534 0735 7440 SZA /AC STILL = 0
535 0736 4434 ERROR /NO, AC CHANGED BY IOT7
536 0737 4440 ENDTST
537 /
538 /ROUTINE TO CHECK THAT AC IS CLEARED BY IOT 6
539 /
540 0740 4436 TST26, INTST
541 0741 7340 CLA CLL CMA /AC = 7777
542 0742 4431 LGLS /LOAD STATUS REGISTER IOT 6
543 0743 7440 SZA /AC SHOULD = 0
544 0744 4434 ERROR /AC NOT CLEARED BY IOT 6
545 0745 4440 ENDTST
546 /
547 /ROUTINE TO CHECK THAT DONE FLAG IS SET BY IOT6
548 /AND CLEARED BY IOTS

```

```

549 /
550 0746 4436 TST27, INTST
551 0747 7300 CLA CLL
552 0750 4431 LQLS /SET DONE IOT6
553 0751 4423 LQSK /SKIP ON DONE FLAG
554 0752 4434 ERROR /IOT 6 DID NOT SET DONE FLAG OR SKIP HIGH
555 0753 4430 LQRS /READ STATUS CLEAR DONE IOT 5
556 0754 4423 LQSK /SKIP IF DONE FLAG STILL SET
557 0755 5357 JMP ,+2 /FLAG CLEARED
558 0756 4434 ERROR /FLAG NOT CLEARED BY IOT 6 OR SKIP STUCK LOW
559 0757 4440 ENDTST
560 /
561 /ROUTINE TO CHECK THAT DONE FLAG IS SET BY IOT 6
562 /AND CLEARED BY IOT 2
563 /
564 0760 4436 TST30, INTST
565 0761 7300 CLA CLL
566 0762 4431 LQLS /SET DONE IOT 6
567 0763 4423 LQSK /SKIP ON DONE FLAG
568 0764 4434 ERROR /IOT 6 DID NOT SET DONE FLAG OR SKIP STUCK HIGH
569 0765 4425 LQMP /IOT 2, SHOULD CLEAR DONE
570 0766 4423 LQSK /SKIP IF DONE FLAG STILL SET
571 0767 5371 JMP ,+2 /FLAG CLEARED
572 0770 4434 ERROR /FLAG NOT CLEARED BY IOT 2 OR SKIP STUCK LOW
573 0771 4440 ENDTST
574 0772 5777* JMP TST31
575 /
576 0777 1000 PAGE
577 1000 /
578 /
579 /ROUTINE TO CHECK THAT DONE FLAG IS SET BY IOT 6
580 /AND CLEARED BY IOT 3
581 /
582 1000 4436 TST31, INTST
583 1001 7300 CLA CLL
584 1002 4431 LQLS /SET DONE FLAG IOT 6
585 1003 4423 LQSK /SKIP ON DONE FLAG SET
586 1004 4434 ERROR /DONE FLAG NOT SET BY IOT 6
587 1005 4426 LQMC /CLEAR DONE FLAG IOT 3
588 1006 4423 LQSK /SKIP IF DONE FLAG STILL SET
589 1007 5211 JMP ,+2 /DONE FLAG CLEARED OK
590 1010 4434 ERROR /DONE FLAG NOT CLEARED BY IOT 3
591 1011 4440 ENDTST /OR SKIP STUCK LOW
592 /
593 /ROUTINE TO CHECK THAT DONE FLAG IS SET BY IOT 6
594 /AND CLEARED BY IOT 4
595 /
596 1012 4436 TST32, INTST
597 1013 7300 CLA CLL
598 1014 4431 LQLS /SET DONE FLAG IOT 6
599 1015 4423 LQSK /SKIP ON DONE FLAG SET
600 1016 4434 ERROR /DONE FLAG NOT SET BY IOT 6
601 1017 4427 LQPC /CLEAR DONE FLAG IOT 4
602 1020 4423 LQSK /SKIP IF DONE FLAG STILL SET

```

```

603 1021 5223 JMP ,+2 /DONE FLAG CLEARED OK
604 1022 4434 ERROR /DONE FLAG NOT CLEARED BY IOT 4 OR
605 1023 4440 ENDTST /SKIP STUCK LOW
606 /
607 /ROUTINE TO CHECK THAT DONE FLAG IS SET BY IOT 6
608 /AND CLEARED BY IOT 7
609 /
610 1024 4436 TST33, INTST
611 1025 7300 CLA CLL
612 1026 4431 LQLS /SET DONE FLAG IOT 6
613 1027 4423 LQSK /SKIP ON DONE FLAG SET
614 1030 4434 ERROR /DONE FLAG NOT SET BY IOT 6
615 1031 4432 LQRE /CLEAR DONE FLAG IOT 7
616 1032 4423 LQSK /CHECK DONE FLAG FOR CLEARED CONDITION
617 1033 5235 JMP ,+2 /DONE FLAG CLEARED OK
618 1034 4434 ERROR /DONE FLAG NOT CLEARED BY IOT 7 OR SKIP
619 1035 4440 ENDTST /STUCK LOW
620 /
621 /ROUTINE TO CHECK THAT DONE FLAG IS SET BY IOT 6
622 /AND CLEARED BY CAF INSTRUCTION
623 /
624 1036 4436 TST34, INTST
625 1037 7300 CLA CLL
626 1040 4431 LQLS /SET DONE FLAG IOT 6
627 1041 4423 LQSK /SKIP ON DONE FLAG SET
628 1042 4434 ERROR /DONE FLAG NOT SET BY IOT 6
629 1043 6007 CAF /CLEAR DONE FLAG WITH CAF INSTRUCTION
630 1044 4423 LQSK /CHECK DONE FLAG FOR CLEARED CONDITION
631 1045 5247 JMP ,+2 /DONE FLAG CLEARED OK
632 1046 4434 ERROR /DONE FLAG NOT CLEARED BY CAF OR SKIP STUCK LOW
633 1047 4440 ENDTST
634 /
635 /
636 /
637 /ROUTINE TO CHECK READ-WRITE BITS OF STATUS REGISTER
638 /ALL ONES AND ALL ZEROS
639 /
640 1050 4436 TST35, INTST
641 1051 7300 CLA CLL
642 1052 4431 LQLS /LOAD ZEROS INTO STATUS REGISTER
643 1053 4430 LQRS /READ STATUS
644 1054 0076 AND STMASK /MASK BITS 5,8,9,10,11 (ONLY BITS 10 & 11 ON VT78)
645 1055 7440 SZA /SKIP IF BITS 5,8,9,10,11 ALL ZERO (ONLY BITS 10 & 11 ON VT78)
646 1056 4451 STATER /ONE OR MORE READ-WRITE BITS NOT CLEARED
647 1057 7340 CLA CLL CMA /AC = 7777
648 1060 4431 LQLS /LOAD ONES INTO STATUS REGISTER
649 1061 4430 LQRS /READ STATUS
650 1062 0076 AND STMASK /MASK BITS 5,8,9,10,11 (ONLY BITS 10 & 11 ON VT78)
651 1063 1377 TAD (7661) /ADD TWO'S COMPLEMENT
652 1064 7440 SZA /SKIP IF ALL BITS SET
653 1065 4451 STATER /ONE OR MORE OF BITS 5,8,9,10,11 NOT SET (10 & 11 ON VT78)
654 1066 4440 ENDTST /IN STATUS REGISTER
655 1067 4475 VT78CK /VT78/ SYSTEM UNDER TEST = VT78 ?
656 1070 5776* JMP TST41 /VT78/ YES - SKIP TESTS 36,37 AND 40, IN VT78
657 /VT78/ STATUS BITS 5,8, AND 9 DO NOT EXIST.

```

```

658 /
659 /
660 /ROUTINE TO CHECK BIT 5 OF STATUS REGISTER
661 /
662 1071 4436 TST36, INTST
663 1072 7300 CLA CLL
664 1073 1375 TAD (7677) /ALL ONES EXCEPT BIT 5
665 1074 4431 LQLS /LOAD STATUS
666 1075 4430 LQRS /READ STATUS
667 1076 0374 AND (100) /TEST BIT 5
668 1077 1373 TAD (7700)
669 1100 7440 SZA /SKIP IF BIT 5 IS SET
670 1101 5303 JMP ,+2 /BIT 5 CLEARED OK
671 1102 4451 STATER /BIT 5 NOT CLEARED AFTER LOAD STATUS
672 1103 7300 CLA CLL
673 1104 1374 TAD (100) /ALL ZEROS EXCEPT BIT 5
674 1105 4431 LQLS /LOAD STATUS
675 1106 4430 LQRS /READ STATUS
676 1107 0374 AND (100) /TEST BIT 5
677 1110 1373 TAD (7700)
678 1111 7440 SZA /SKIP IF BIT 5 IS SET
679 1112 4451 STATER /BIT 5 NOT SET AFTER LOAD STATUS
680 1113 4440 ENDTST
681 /
682 /ROUTINE TO CHECK BIT 8 OF STATUS REGISTER
683 /
684 1114 4436 TST37, INTST
685 1115 7300 CLA CLL
686 1116 1372 TAD (7767) /ALL ONES EXCEPT BIT 8
687 1117 4431 LQLS /LOAD STATUS
688 1120 4430 LQRS /READ STATUS
689 1121 0371 AND (10) /TEST BIT 8
690 1122 1370 TAD (7770)
691 1123 7440 SZA /SKIP IF BIT 8 IS SET
692 1124 5326 JMP ,+2 /BIT 8 CLEARED OK
693 1125 4451 STATER /BIT 8 NOT CLEARED AFTER LOAD STATUS
694 1126 7300 CLA CLL
695 1127 1371 TAD (10) /ALL ZEROS EXCEPT BIT 8
696 1130 4431 LQLS /LOAD STATUS
697 1131 4430 LQRS /READ STATUS
698 1132 0371 AND (10) /TEST BIT 8
699 1133 1370 TAD (7770)
700 1134 7440 SZA /SKIP IF BIT 8 IS SET
701 1135 4451 STATER /BIT 8 NOT SET AFTER LOAD STATUS
702 1136 4440 ENDTST
703 /
704 /ROUTINE TO CHECK BIT 9 OF STATUS REGISTER
705 /
706 1137 4436 TST40, INTST
707 1140 7300 CLA CLL
708 1141 1367 TAD (7773) /ALL ONES EXCEPT BIT 9
709 1142 4431 LQLS /LOAD STATUS
710 1143 4430 LQRS /READ STATUS
711 1144 0366 AND (4) /TEST BIT 9
712 1145 1365 TAD (7774)

```

```

713 1146 7640 SZA CLA /SKIP IF BIT 9 IS SET
714 1147 5351 JMP ,+2 /BIT 9 CLEARED OK
715 1150 4451 STATER /BIT 9 NOT CLEARED AFTER LOAD STATUS
716 1151 7300 CLA CLL
717 1152 1366 TAD (4) /ALL ZEROS EXCEPT BIT 9
718 1153 4431 LQLS /LOAD STATUS
719 1154 4430 LQRS /READ STATUS
720 1155 0366 AND (4) /TEST BIT 9
721 1156 1365 TAD (7774)
722 1157 7640 SZA CLA /SKIP IF BIT 9 IS SET
723 1160 4451 STATER /BIT 9 NOT SET AFTER LOAD STATUS
724 1161 4440 ENDTST
725 1162 5776 JMP TST41
726 /
727 1165 7774 /
728 1166 0004 /
729 1167 7773 /
730 1170 7770 /
731 1171 0010 /
732 1172 7767 /
733 1173 7700 /
734 1174 0100 /
735 1175 7677 /
736 1176 1200 /
737 1177 7661 /
738 PAGE
739 /
740 /ROUTINE TO CHECK BIT 10 OF STATUS REGISTER
741 /
742 1200 4436 TST41, INTST
743 1201 7300 CLA CLL
744 1202 1377 TAD (7775) /ALL ONES EXCEPT BIT 10
745 1203 4431 LQLS /LOAD STATUS
746 1204 4430 LQRS /READ STATUS
747 1205 0376 AND (2) /TEST BIT 10
748 1206 1375 TAD (7776)
749 1207 7640 SZA CLA /SKIP IF BIT 10 IS SET
750 1210 5212 JMP ,+2 /BIT 10 NOT CLEARED AFTER LOAD STATUS
751 1211 4451 STATER
752 1212 7300 CLA CLL
753 1213 1376 TAD (2) /ALL ZEROS EXCEPT BIT 10
754 1214 4431 LQLS /LOAD STATUS
755 1215 4430 LQRS /READ STATUS
756 1216 0376 AND (2) /TEST BIT 10
757 1217 1375 TAD (7776)
758 1220 7640 SZA CLA /SKIP IF BIT 10 IS SET
759 1221 4451 STATER /BIT 10 NOT SET AFTER LOAD STATUS
760 1222 4440 ENDTST
761 /
762 /ROUTINE TO CHECK BIT 11 OF STATUS REGISTER
763 /
764 1223 4436 TST42, INTST
765 1224 7300 CLA CLL
766 1225 1375 TAD (7776) /ALL ONES EXCEPT BIT 11

```

```

767 1226 4431 LQLS /LOAD STATUS
768 1727 4430 LQRS /READ STATUS
769 1730 0374 AND (1) /TEST BIT 11
770 1731 1373 TAD (7777)
771 1232 7640 SZA CLA /SKIP IF BIT 11 NOT CLEARED
772 1733 5235 JMP +2 /BIT 11 CLEARED OK
773 1734 4451 STATER /BIT 11 NOT CLEARED AFTER LOAD STATUS
774 1735 7300 CLA CLL
775 1736 1374 TAD (1) /SET BIT 11
776 1737 4430 LQLS /LOAD STATUS
777 1740 4430 LQRS /READ STATUS
778 1241 0374 AND (1) /MASK BIT 11
779 1242 1373 TAD (7777)
780 1243 7640 SZA CLA /SKIP IF SET
781 1244 4451 STATER /BIT 11 NOT SET AFTER LOAD STATUS
782 1245 4440 ENDST
783
784 /ROUTINE TO CHECK THAT STATUS REGISTER IS CLEARED BY CAF
785
786 1246 4436 TST43, INIST
787 1247 7340 CLA CLL CMA /AC = 7777
788 1750 4431 LQLS /LOAD STATUS
789 1751 4430 LQRS /READ STATUS
790 1252 0076 AND STMASK /MASK BITS 5,8,9,10,11 (ONLY 10 & 11 FOR VT78)
791 1253 1372 TAD (7661) /ADD TWO'S COMPLEMENT
792 1254 7640 SZA CLA /SKIP IF BITS 5,8,9,10,11 ALL SET (ONLY 10 & 11 ON VT78)
793 1255 4451 STATER /ONE OR MORE OF BITS 5,8,9,10,11 NOT SET (10 & 11 ON VT78)
794 1256 6007 CAF /CLEAR STATUS REGISTER
795 1257 4430 LQRS /READ STATUS
796 1260 0076 AND STMASK /TEST BITS 5,8,9,10,11 (ONLY 10 & 11 ON VT78)
797 1761 7640 SZA CLA /SKIP IF ALL CLEARED
798 1762 4451 STATER /ONE OR MORE OF BITS 5,8,9,10,11 NOT CLEARED BY CAF
799
800 1763 4440 ENDST
801
802 /ROUTINE TO CHECK THAT INTERRUPT ENABLE (BIT 11)
803 /OF STATUS REGISTER WILL GENERATE INTERRUPT REQUEST
804
805 1264 4436 TST44, INIST
806 1265 7300 CLA CLL
807 1766 4431 LQLS /SET DONE FLAG
808 1767 4423 LQSK /SKIP ON DONE FLAG
809 1270 4434 ERROR /DONE FLAG NOT SET BY LOAD STATUS IOT 6
810 1771 6001 ION /INTERRUPT ON
811 1772 7000 NOP /WAIT FOR INTERRUPT
812 1273 6002 IOF /INTERRUPT OFF
813 1774 7300 CLA CLL
814
815 1275 1116 TAD INTFLG /GET PROGRAM INTERRUPT FLAG
816 1276 7640 SZA CLA /SKIP IF NO INTERRUPT
817 1777 4434 ERROR /DONE FLAG CAUSED INTERRUPT WITH
/INTERRUPT ENABLE CLEARED
818 1300 7300 CLA CLL
819 1301 3116 DCA INTFLG /CLEAR INTERRUPT FLAG
820 1302 4423 LQSK /DONE FLAG STILL SET?
821 1303 4434 ERROR /DONE FLAG NOT SET

```

```

822 1304 7300 CLA CLL
823 1305 6001 ION
824 1306 7201 CLA IAC /BIT 11 - INTERRUPT ENABLE
825 1307 4431 LQLS /SET INTERRUPT ENABLE
826 1310 7000 NOP /WAIT FOR INTERRUPT
827 1311 6002 IOF
828 1312 7300 CLA CLL
829 1313 1116 TAD INTFLG /GET PROGRAM INTERRUPT FLAG
830 1314 7650 SNA CLA /SKIP IF INTERRUPT FLAG SET
831 1315 4434 ERROR /INTERRUPT ENABLE FAILED TO CAUSE INTERRUPT
832 1316 7300 CLA CLL
833 1317 3116 DCA INTFLG /CLEAR INTERRUPT FLAG
834 1320 4430 LQRS /CLEAR DONE FLAG
835 1321 4423 LQSK /SKIP IF DONE FLAG STILL SET
836 1322 7410 SKP /DONE FLAG CLEARED
837 1323 4434 ERROR /DONE FLAG NOT CLEARED BY LQRS IOT 5
838 1324 6001 ION /INTERRUPT ON
839 1325 7000 NOP /WAIT FOR INTERRUPT
840 1326 6002 IOF /INTERRUPT OFF
841 1327 7300 CLA CLL
842 1330 1116 TAD INTFLG /GET PROGRAM INTERRUPT FLAG
843 1331 7440 SZA /SKIP IF CLEAR
844 1332 4434 ERROR /PROGRAM INTERRUPTED WITH DONE FLAG
845
846 1333 2366 ISZ LOOPC /Cleared AND INTERRUPT ENABLE SET
847 1334 5771 JMP TST1 /INCREMENT LOOP COUNTER
848 1335 4466 C$ACTV /CONTINUE LOGIC TESTS
849 1336 7410 SKP /CHECK FOR ACTIVE CONSOLE
850 1337 5347 JMP +10 /SKIP IF CONSOLE ACTIVE
851 1340 4447 MESSAGE /CONSOLE NOT ACTIVE, DO NOT PRINT
852 1341 6125 ENDDIG /PRINT END OF PASS MESSAGE
853 1342 4467 MESSAGE /MESSAGE ADDRESS
854 1343 6137 PASCNT /PRINT PASS COUNT
855 1344 2365 ISZ PASS /INCREMENT PASS COUNT
856 1345 1365 TAD PASS /GET PASS COUNT
857 1346 4470 RECPRT /PRINT PASS COUNT
858 1347 4465 L$S /GET SWITCHES
859 1350 0371 AND (400) /TEST SR3, HALT AT COMPLETION OF PASS
860 1351 7650 SNA CLA /SKIP IF SR3=1
861 1352 5771 JMP TST1 /SR3=0 CONTINUE TEST
862 1353 4466 C$ACTV /CHECK FOR ACTIVE CONSOLE
863 1354 7410 SKP /CONSOLE ACTIVE, DO NOT HALT
864 1355 7402 HLT /CONSOLE NOT ACTIVE
865 1356 4466 C$ACTV /CHECK FOR ACTIVE CONSOLE
866 1357 7410 SKP /SKIP IF CONSOLE IS ACTIVE
867 1360 5771 JMP TST1 /CONTINUE LOGIC TEST
868 1361 4467 MESSAGE /PRINT "WAITING" MESSAGE
869 1362 6000 WAITNG /MESSAGE POINTER
870 1363 4445 CTLCHK /WAIT FOR CONTROL CHARACTER
871 1364 5363 JMP .-1
872
873 1365 0000 PASS, 0 /PASS COUNT
874 1366 0000 LOOPC, 0 /LOOP COUNT
875
876

```

```

877 1371 0400
878 1372 7661
879 1373 7777
880 1374 0001
881 1375 7776
882 1376 0002
883 1377 7775
      1400
884 PAGE
885 /
886 /*****
887 /
888 / BEGINNING OF DYNAMIC TESTS
889 /
890 /*****
891 /
892 /ROUTINE TO CHECK INDIVIDUAL BITS OF DATA BUFFER
893 /FOR CORRECT CARRIAGE POSITIONING
894 /
895 1400 3777* TST45, DCA Pcount /CLEAR PASS COUNT
896 1401 4433 BGNTST
897 1402 1376 TAD (56 /ASCII CODE FOR PERIOD
898 1403 3124 DCA CHAR /SAVE IT
899 1404 3231 DCA POS45 /CLEAR PRINT POSITION COORDINATE
900 1405 7305 CLA CLL IAC RAL /SET BIT 10
901 1406 4431 LQLS /LIFT RIBBON
902 1407 4464 PRINT /PRINT A PERIOD
903 1410 4464 PRINT /OVERPRINT A PERIOD
904 1411 2231 ISZ POS45 /SET BIT 11 OF NEW PRINT POSITION
905 1412 4461 XCRLF /CARRIAGE RETURN, LINE FEED
906 1413 4452 PLP45, KBCHK /CHECK FOR KEYBOARD INTERVENTION
907 1414 4464 PRINT /PRINT AT LEFTMOST POSITION
908 1415 1231 TAD POS45 /GET NEW PRINT POSITION
909 1416 4456 MOVCR /MOVE CARRIAGE TO NEW PRINT POSITION
910 1417 4464 PRINT /PRINT AT NEW PRINT POSITION
911 1420 4461 XCRLF /GO BACK TO LEFTMOST POSITION AND LINE FEED
912 1421 1231 TAD POS45 /GET PRINT POSITION
913 1422 7104 CLL RAL /ROTATE LEFT TO TEST NEXT BIT OF DATA BUFFER
914 1423 3231 DCA POS45 /SAVE UPDATED PRINT POSITION
915 1424 1231 TAD POS45 /GET NEW PRINT POSITION
916 1425 7700 SMA CLA /SKIP IF ALL BITS TESTED
917 1426 5213 JMP PLP45 /CONTINUE PRINTING
918 1427 4440 ENDTST /CHECK FOR TEST LOOP
919 1430 5232 JMP TST46 /GO TO NEXT TEST
920 1431 0000 POS45, 0 /STORAGE AREA FOR PRINT POSITION
921 /
922 /
923 /ROUTINE TO CHECK INDIVIDUAL BITS OF DATA BUFFER FOR CORRECT
924 /PAPER POSITIONING
925 /
926 1432 4433 TST46, BGNTST
927 1433 1375 TAD (137 /ASCII CODE FOR UNDERLINE
928 1434 3124 DCA CHAR /SAVE IT
929 1435 7001 IAC /SET BIT 11
930 1436 3256 DCA MAG46 /SETUP PAPER MOVEMENT MAGNITUDE

```

```

931 1437 7305 CLA CLL IAC RAL /SET BIT 10
932 1440 4431 LQLS /LIFT RIBBON
933 1441 1374 TAD (24 /GET LINE LENGTH = 20 CHARACTERS
934 1442 3275 DCA LIN46 /SAVE IT
935 1443 3136 PLP46A, DCA CHRCNT /CLEAR CHARACTER COUNT
936 1444 4452 PLP46B, KBCHK /CHECK FOR KEYBOARD INTERVENTION
937 1445 4462 RSPACE /MOVE CARRIAGE ONE SPACE RIGHT
938 1446 4464 PRINT /PRINT A CHARACTER
939 1447 1136 TAD CHRCNT /GET CHARACTER COUNT
940 1450 7041 CIA /NEGATE IT
941 1451 1275 TAD LIN46 /ADD NUMBER OF CHARACTERS PER LINE
942 1452 7640 SZA CLA /SKIP IF END OF LINE
943 1453 5244 JMP PLP46B /NOT END OF LINE, CONTINUE PRINTING
944 1454 4457 MOVPAR /MOVE PAPER
945 1455 0000 PRADY /PAPER DIRECTION = ADVANCE
946 1456 0000 MAG46, 0 /PAPER MOVEMENT MAGNITUDE
947 1457 1256 TAD MAG46 /GET PAPER MOVEMENT MAGNITUDE
948 1460 7104 CLL RAL /ROTATE TO TEST NEXT BIT OF DATA BUFFER
949 1461 3256 DCA MAG46 /SAVE NEW PAPER MOVEMENT MAGNITUDE
950 1462 4442 SWAIT /WAIT FOR READY BITS
951 1463 4451 STATER /READY BIT STATUS ERROR
952 1464 4432 LGRE /RESTORE PRINTER
953 1465 4442 SWAIT /WAIT FOR READY BITS
954 1466 4451 STATER /READY BIT STATUS ERROR
955 1467 7300 CLA CLL
956 1470 1256 TAD MAG46 /GET PAPER MOVEMENT MAGNITUDE
957 1471 7640 SZA CLA /SKIP IF ALL BITS TESTED
958 1472 5243 JMP PLP46A /CONTINUE PRINTING
959 1473 4440 ENDTST /CHECK FOR TEST LOOP
960 1474 5276 JMP TST47 /GO TO NEXT TEST
961 1475 0000 LIN46, 0 /STORAGE AREA FOR LINE LENGTH
962 /
963 /
964 /ROUTINE TO PRINT THE LETTER I ON ONE LINE WITH 1/60 INCH SPACING
965 /BETWEEN EACH LETTER, CARRIAGE MOVING ACROSS PAGE FOR EACH NEW
966 /PRINT POSITION
967 /
968 1476 4433 TST47, BGNTST
969 1477 1373 TAD (111 /GET CHARACTER TO PRINT "I"
970 1500 3124 DCA CHAR /SAVE IT
971 1501 3332 DCA LSTART /CLEAR LEFT COORDINATE
972 1502 1334 TAD RMOST /GET RIGHTMOST COORDINATE
973 1503 3333 DCA LEND /SAVE IT
974 1504 7305 CLA CLL IAC RAL /SET BIT 10
975 1505 4431 LQLS /LIFT RIBBON
976 1506 1332 PLP47, TAD LSTART /GET LEFT COORDINATE
977 1507 4456 MOVCR /MOVE CARRIAGE TO NEW POSITION
978 1510 4464 PRINT /PRINT CHARACTER
979 1511 4452 KBCHK /CHECK FOR KEYBOARD INTERVENTION
980 1512 1332 TAD LSTART /GET LEFT COORDINATE
981 1513 7041 CIA /NEGATE IT
982 1514 1333 TAD LEND /ADD RIGHT COORDINATE
983 1515 7650 SMA CLA /SKIP IF NOT EQUAL
984 1516 5330 JMP TST47A /CHECK FOR TEST LOOP
985 1517 1333 TAD LEND /GET RIGHT COORDINATE

```

```

986 1520 4456      MOVCR      /MOVE CARRIAGE
987 1521 4464      PRINT      /PRINT CHARACTER
988 1522 2332      ISZ LSTART
989 1523 2332      ISZ LSTART /ADD 1/60 INCH TO LINE START POSITION
990 1524 7344      CLA CLL CMA RAL /AC = MINUS TWO
991 1525 1333      TAD LEND  /SUBTRACT 1/60 INCH FROM LINE END
992 1526 3333      DCA LEND  /UPDATE LEFT COORDINATE
993 1527 5306      JMP PLP47
994 1530 4440      TST47A, ENDTST /CHECK FOR TEST LOOP
995 1531 5772      JMP TST50
996 1532 0000      LSTART, 0
997 1533 0000      LEND, 0
998 1534 3044      RMOST, 3044 /RIGHTMOST COORDINATE
999 /
1000 /
1001 /
1002 1572 1600
1003 1573 0111
1004 1574 0024
1005 1575 0137
1006 1576 0056
1007 1577 2465
1008 1600
1009 /
1010 /
1011 1600 4433      PAGE
1012 1601 1377      /ROUTINE TO PRINT ONE LINE OF EACH CHARACTER
1013 1602 3124      /
1014 1603 1376      TST50, BGNST
1015 1604 3132      TAD (40 /GET FIRST CHARACTER
1016 1605 3245      DCA CHAR /SAVE IT
1017 1606 7305      TAD (140 /GET NUMBER OF LINES TO PRINT (96)
1018 1607 4431      DCA LINES /SAVE IT
1019 1610 4452      DCA DSWTCH /CLEAR CARRIAGE DIRECTION SWITCH
1020 1611 4464      CLA CLL IAC RAL /SET BIT 10
1021 1612 4454      LQLS /LIFT PRINTER RIBBON
1022 1613 5231      PLP50, KBCHK /CHECK FOR KEYBOARD INTERVENTION
1023 1614 2115      PRINT /PRINT CHARACTER IN LOCATION CHARRT
1024 1615 7000      LINEND /CHECK FOR END OF LINE
1025 1616 1130      JMP NES0A /NOT END OF LINE, CONTINUE PRINTING
1026 1617 3222      ISZ LINCNT /INCREMENT LINE COUNT
1027 1620 4457      NOP /DON'T SKIP IF LINCNT OVERFLOWS
1028 1621 0000      TAD LF /GET LINE FEED MAGNITUDE
1029 1622 0000      DCA ,+3 /SAVE FOR LINE FEED
1030 1623 4460      MOVPAR /CALL MOVE PAPER SUBROUTINE
1031 1624 5240      /DIRECTION EQUAL ADVANCE PAPER
1032 1625 4442      SWAIT
1033 1626 4451      STATER /ONE OR MORE READY BITS NOT READY
1034 1627 4440      ENDTST /CHECK FOR TEST LOOP
1035 1630 5775      JMP TST51 /GO TO NEXT TEST
1036 /
1037 1631 1245      NE50A, TAD DSWTCH /GET CARRIAGE DIRECTION SWITCH
1038 1632 7650      SNA CLA /SKIP IF DIRECTION IS LEFT
1039 1633 4462      RSPACE /DIRECTION IS RIGHT ONE SPACE
1040 1634 1245      TAD DSWTCH /GET CARRIAGE DIRECTION SWITCH
1041 1635 7640      SZA CLA /SKIP IF DIRECTION IS RIGHT
1042 1636 4463      LSPACE /DIRECTION IS LEFT ONE SPACE
1043 1637 5210      JMP PLP50 /CONTINUE PRINTING
1044 1640 2124      NE50B, ISZ CHAR /INCREMENT TO NEXT CHARACTER
1045 1641 1245      TAD DSWTCH /GET CARRIAGE DIRECTION SWITCH
1046 1642 7040      CMA /COMPLEMENT IT
1047 1643 3245      DCA DSWTCH /SAVE NEW SWITCH
1048 1644 5210      JMP PLP50 /CONTINUE PRINTING
1049 /
1050 1645 0000      DSWTCH, 0 /CARRIAGE DIRECTION SWITCH
1051 1775 2000
1052 1776 0140
1053 1777 0040
1054 2000
1055 /
1056 /
1057 /
1058 /
1059 2000 4433      TST51, BGNST /IOF,CHK KYBD,SETUP TEST LOOP,INIT MODULE
1060 2001 1377      TAD (40 /GET FIRST CHARACTER
1061 2002 3135      DCA FSTCHR /SAVE IT
1062 2003 1377      TAD (40 /GET FIRST CHARACTER
1063 2004 3134      DCA FIRST /SAVE FIRST CHARACTER OF TEST

```

```

1029 1622 0000      0000 /MODIFIED LINE FEED MAGNITUDE
1030 1623 4460      LSTLIN /CHECK FOR LAST LINE TO PRINT
1031 1624 5240      JMP NES0B /NOT LAST LINE CONTINUE PRINTING
1032 1625 4442      SWAIT /WAIT FOR LAST CHARACTER TO BE PRINTED
1033 1626 4451      STATER /ONE OR MORE READY BITS NOT READY
1034 1627 4440      ENDTST /CHECK FOR TEST LOOP
1035 1630 5775      JMP TST51 /GO TO NEXT TEST
1036 /
1037 1631 1245      NE50A, TAD DSWTCH /GET CARRIAGE DIRECTION SWITCH
1038 1632 7650      SNA CLA /SKIP IF DIRECTION IS LEFT
1039 1633 4462      RSPACE /DIRECTION IS RIGHT ONE SPACE
1040 1634 1245      TAD DSWTCH /GET CARRIAGE DIRECTION SWITCH
1041 1635 7640      SZA CLA /SKIP IF DIRECTION IS RIGHT
1042 1636 4463      LSPACE /DIRECTION IS LEFT ONE SPACE
1043 1637 5210      JMP PLP50 /CONTINUE PRINTING
1044 1640 2124      NE50B, ISZ CHAR /INCREMENT TO NEXT CHARACTER
1045 1641 1245      TAD DSWTCH /GET CARRIAGE DIRECTION SWITCH
1046 1642 7040      CMA /COMPLEMENT IT
1047 1643 3245      DCA DSWTCH /SAVE NEW SWITCH
1048 1644 5210      JMP PLP50 /CONTINUE PRINTING
1049 /
1050 1645 0000      DSWTCH, 0 /CARRIAGE DIRECTION SWITCH
1051 1775 2000
1052 1776 0140
1053 1777 0040
1054 2000
1055 /
1056 /
1057 /
1058 /
1059 2000 4433      TST51, BGNST /IOF,CHK KYBD,SETUP TEST LOOP,INIT MODULE
1060 2001 1377      TAD (40 /GET FIRST CHARACTER
1061 2002 3135      DCA FSTCHR /SAVE IT
1062 2003 1377      TAD (40 /GET FIRST CHARACTER
1063 2004 3134      DCA FIRST /SAVE FIRST CHARACTER OF TEST

```



```

1064 2005 1376 TAD (140 /GET NUMBER OF LINES TO PRINT (96)
1065 2006 3132 DCA LINES /SAVE IT
1066 2007 7305 CLA CLL IAC RAL /SET BIT 10
1067 2010 4431 LQLS /LIFT PRINTER RIBBON
1068 2011 1135 PLP51A, TAD FSTCHR /GET FIRST CHARACTER
1069 2012 3124 DCA CHAR /SAVE IT TO BEGIN PATTERN
1070 2013 4452 PLP51B, KBCHK /CHECK FOR KEYBOARD INTERVENTION
1071 2014 4464 PRINT /PRINT CHARACTER IN LOCATION CHARRT
1072 2015 4454 LINEND /CHECK FOR END OF LINE
1073 2016 5226 JMP NE51A /NOT END OF LINE, CONTINUE PRINTING
1074 2017 2115 ISZ LINCNT /INCREMENT LINE COUNT
1075 2020 7000 NOP /DUMMY IF LINE COUNT OVERFLOWS
1076 2021 4461 XCRLF /EXECUTE CARRIAGE RETURN AND LINE FEED
1077 2022 4460 LSTLIN /CHECK FOR LAST LINE TO PRINT
1078 2023 5243 JMP NE51B /NOT LAST LINE, CONTINUE PRINTING
1079 2024 4440 ENDTST /CHECK FOR TEST LOOP
1080 2025 5245 JMP TST52 /GO TO NEXT TEST
1081 /
1082 2026 7300 NE51A, CLA CLL /INCREMENT TO NEXT CHARACTER
1083 2027 2124 ISZ CHAR /CHECK FOR LAST CHARACTER
1084 2030 1375 TAD (200 /FORM TWO'S COMPLEMENT
1085 2031 7041 CIA /ADD CURRENT CHARACTER
1086 2032 1124 TAD CHAR /SKIP IF NOT LAST CHARACTER
1087 2033 7650 SNA CLA /LAST CHARACTER
1088 2034 5237 JMP ,+3 /MOVE CARRIAGE TO NEXT PRINT POSITION
1089 2035 4462 RSPACE /CONTINUE PRINTING
1090 2036 5213 JMP PLP51B /MOVE CARRIAGE TO NEXT PRINT POSITION
1091 2037 4462 RSPACE /GET FIRST CHARACTER
1092 2040 1134 TAD FIRST /SET UP TO BEGIN AGAIN
1093 2041 3124 DCA CHAR /CONTINUE PRINTING
1094 2042 5213 JMP PLP51B /INCREMENT STARTING CHARACTER
1095 /
1096 2043 2135 NE51B, ISZ FSTCHR /PRINT NEXT LINE
1097 2044 5211 JMP PLP51A /ROUTINE TO PRINT FIVE LINES OF CAPITAL "M" WITH
1098 / /OVERPRINT AND BI-DIRECTIONAL PRINTING, AND RIBBON LIFT
1099 / /AND DROP EVERY OTHER CHARACTER
1100 / /
1101 / /
1102 / /
1103 / /
1104 / /
1105 / /
1106 / /
1107 2045 4433 TST52, HGNTST /GET ASCII CHARACTER "M"
1108 2046 1374 TAD (115 /SAVE IT
1109 2047 3124 DCA CHAR /5 LINES TO PRINT
1110 2050 1373 TAD (12 /SAVE IT
1111 2051 3132 DCA LINES /CLEAR RIBBON SWITCH
1112 2052 3334 DCA RIBN52 /THIS LOCATION MAY BE MODIFIED TO SLOW PRINT PATTERN
1113 2053 4443 PLP52A, DELAY 7777 /SET BIT 10
1114 2054 7777 LQLS /LIFT RIBBON
1115 2055 7305 CLA CLL IAC RAL /CHECK FOR KEYBOARD INTERVENTION
1116 2056 4431 LQLS /PRINT ONE CHARACTER
1117 2057 4452 PLP52B, KBCHK /CHECK FOR KEYBOARD INTERVENTION
1118 2060 4464 PRINT /PRINT ONE CHARACTER

```

```

1119 2061 4454 LINEND /CHECK FOR END OF LINE
1120 2062 5303 JMP NE52A /NOT END OF LINE, CONTINUE
1121 2063 2115 ISZ LINCNT /INCREMENT LINE COUNT
1122 2064 7000 NOP /DUMMY IN CASE LINE COUNT OVERFLOWS
1123 2065 4443 PLP52C, DELAY 7777 /THIS LOCATION MAY BE MODIFIED TO SLOW PRINT SPEED
1124 2066 7777 LQLS /DROP RIBBON
1125 2067 7300 CLA CLL /CHECK FOR KEYBOARD INTERVENTION
1126 2070 4431 PLP52D, LQLS /PRINT ONE CHARACTER
1127 2071 4452 KBCHK /CHECK FOR END OF LINE
1128 2072 4464 PRINT /NOT END OF LINE, CONTINUE
1129 2073 4454 LINEND /INCREMENT LINE COUNT
1130 2074 5316 JMP NE52B /DUMMY IN CASE LINE COUNT OVERFLOWS
1131 2075 2115 ISZ LINCNT /CHECK FOR LAST LINE OF TEST
1132 2076 7000 NOP /NOT LAST LINE, CONTINUE
1133 2077 4460 LSTLIN /CHECK FOR TEST LOOP
1134 2100 5332 JMP NE52C /GO TO NEXT TEST
1135 2101 4440 ENDTST /SPACE ONE PRINT POSITION RIGHT
1136 2102 5772 JMP TST53 /GET RIBBON SWITCH
1137 / /COMPLEMENT IT
1138 2103 4462 NE52A, RSPACE /SAVE NEW RIBBON SWITCH
1139 2104 1334 TAD RIBN52 /GET NEW RIBBON SWITCH
1140 2105 7040 CMA /COMPLEMENT IT
1141 2106 3334 DCA RIBN52 /SAVE NEW RIBBON SWITCH
1142 2107 1334 TAD RIBN52 /GET NEW RIBBON SWITCH
1143 2110 7650 SNA CLA /SKIP IF RIBBON IS UP
1144 2111 5253 JMP PLP52A /CONTINUE WITH RIBBON UP
1145 2112 4443 DELAY 7777 /THIS LOCATION MAY BE MODIFIED TO SLOW PRINT SPEED
1146 2113 7777 LQLS /LOWER RIBBON
1147 2114 4431 JMP PLP52B /CONTINUE WITH RIBBON DOWN
1148 2115 5257 /
1149 /
1150 2116 4463 NE52B, LSPACE /SPACE ONE PRINT POSITION LEFT
1151 2117 1334 TAD RIBN52 /GET RIBBON SWITCH
1152 2120 7040 CMA /COMPLEMENT IT
1153 2121 3334 DCA RIBN52 /SAVE NEW RIBBON SWITCH
1154 2122 1334 TAD RIBN52 /GET NEW RIBBON SWITCH
1155 2123 7640 SZA CLA /SKIP IF RIBBON DOWN
1156 2124 5265 JMP PLP52C /CONTINUE WITH RIBBON DOWN
1157 2125 4443 DELAY 7777 /THIS LOCATION MAY BE MODIFIED TO SLOW PRINT SPEED
1158 2126 7777 LQLS /RAISE RIBBON
1159 2127 7305 CLA CLL IAC RAL /CONTINUE WITH RIBBON UP
1160 2130 4431 LQLS /EXECUTE CARRIAGE RETURN, LINE FEED
1161 2131 5271 JMP PLP52D /CONTINUE WITH NEXT LINE
1162 / /RIBBON SWITCH
1163 2132 4461 NE52C, XCRLF /EXECUTE CARRIAGE RETURN, LINE FEED
1164 2133 5253 JMP PLP52A /CONTINUE WITH NEXT LINE
1165 2134 0000 RIBN52, 0 /RIBBON SWITCH
1166 / /
1167 / /
1168 2172 2200
1169 2173 0012
1170 2174 0115
1171 2175 0200
1172 2176 0140
1173 2177 0040

```

```

1174          2200 PAGE
1175          /
1176          /ROUTINE TO GENERATE PAIRS OF LINES CONSISTING OF
1177          /RANDOM CHARACTERS, EACH PAIR OF LINES SHOULD BE
1178          /IDENTICAL
1179          /
1180          2200 4433 TST53, BGNSTST /INIT FOR TEST
1181          2201 1377 TAD (30 /24 PAIRS OF RANDOM CHARACTER LINES
1182          2202 3132 DCA LINES /SAVE NUMBER OF LINES TO PRINT
1183          2203 7305 CLA CLL IAC RAL /SET BIT 10
1184          2204 4431 LOLS /LIFT RIBBON
1185          2205 1376 TAD (1234 /GET PRIME
1186          2206 3775 DCA RAN1 /SAVE IT
1187          2207 1374 TAD (5670 /GET SECOND PRIME
1188          2210 3773 DCA RAN2 /SAVE IT
1189          2211 4473 SAVGEN /SAVE PRIMES
1190          2212 4452 PLP53A, KBCHK /CHECK FOR OPERATOR INTERVENTION
1191          2213 4472 RANDOM /GENERATE A RANDOM CHARACTER
1192          2214 3124 DCA CHAR /SAVE IT
1193          2215 1124 TAD CHAR /GET CHARACTER
1194          2216 0372 AND (177 /MASK ASCII BITS
1195          2217 1371 TAD (-40 /IS IT A PRINTING CHARACTER CODE?
1196          2220 7710 SPA CLA /SKIP IF YES
1197          2221 5212 JMP PLP53A /GET A NEW CODE
1198          2222 4464 PRINT /PRINT THE CHARACTER
1199          2223 4454 LINEND /CHECK FOR END OF LINE
1200          2224 5251 JMP NES3A /MOVE THE CARRIAGE
1201          2226 4461 RESGEN /RESTORE THE PRIMES
1202          2227 4452 PLP53B, KBCHK /CHECK FOR OPERATOR INTERVENTION
1203          2230 4472 RANDOM /GET A RANDOM CHARACTER
1204          2231 3124 DCA CHAR /SAVE IT
1205          2232 1124 TAD CHAR /GET IT
1206          2233 0372 AND (177 /MASK ASCII BITS
1207          2234 1371 TAD (-40 /IS IT A PRINTING CHARACTER CODE
1208          2235 7710 SPA CLA /SKIP IF YES
1209          2236 5227 JMP PLP53B /GET A NEW RANDOM NUMBER
1210          2237 4464 PRINT /PRINT THE CHARACTER
1211          2240 4454 LINEND /CHECK FOR END OF LINE
1212          2241 5253 JMP NES3B /MOVE THE CARRIAGE
1213          2242 4461 XCRLF /DO A CR AND LF
1214          2243 4461 XCRLF /DO A CR AND LF
1215          2244 2115 ISZ LINCNT /INCREMENT LINE COUNT
1216          2245 4460 LSTLIN /CHECK FOR LAST LINE TO PRINT
1217          2246 5211 JMP PLP53A-1 /START NEXT PAIR OF RANDOM LINES
1218          2247 4440 ENDTST /CHECK FOR TEST LOOP
1219          2250 5255 JMP TST54 /GO TO NEXT TEST

1220          /
1221          2251 4462 NES3A, RSPACE /SPACE CARRIAGE
1222          2252 5212 JMP PLP53A
1223          /
1224          2253 4462 NES3B, RSPACE /SPACE CARRIAGE
1225          2254 5227 JMP PLP53B
1226          /
1227          /

```

```

1228          /NON-PRINTING CHARACTER CODES TEST
1229          /SHOULD PRINT CHARACTER AT HOME POSITION ON
1230          /PRINT WHEEL IN ALL CASES
1231          /
1232          2255 4433 TST54, BGNSTST /INIT FOR TEST
1233          2256 7300 CLA CLL /INCREMENT CHARACTER CODE
1234          2257 3124 DCA CHAR /CLEAR CHARACTER CODE
1235          2260 7005 IAC RAL /SET BIT 10
1236          2261 4431 LOLS /LIFT RIBBON
1237          2262 1370 TAD (12 /10 LINES TO PRINT
1238          2263 3132 DCA LINES /SAVE IT
1239          2264 4452 PLP54, KBCHK /CHECK FOR OPERATOR INTERVENTION
1240          2265 4464 PRINT /PRINT CHARACTER
1241          2266 4454 LINEND /CHECK FOR END OF LINE
1242          2267 5277 JMP NES4A /GET READY FOR NEXT CHARACTER
1243          2270 2115 ISZ LINCNT /INCREMENT LINE COUNT
1244          2271 7000 NOP /NO SKIP
1245          2272 4461 XCRLF /DO A CR AND LF
1246          2273 4460 LSTLIN /CHECK FOR LAST LINE
1247          2274 5264 JMP PLP54 /START NEXT LINE
1248          2275 4440 ENDTST /CHECK FOR TEST LOOP
1249          2276 5767 JMP TST55 /GO TO NEXT TEST

1250          /
1251          2277 7300 NES4A, CLA CLL /INCREMENT CHARACTER CODE
1252          2300 2124 ISZ CHAR /INCREMENT CHARACTER CODE
1253          2301 1124 TAD CHAR /GET CHARACTER
1254          2302 0366 AND (37 /MASK TO NON-PRINTING CODE
1255          2303 3124 DCA CHAR /SAVE CHARACTER
1256          2304 4462 RSPACE /SPACE CARRIAGE RIGHT
1257          2305 5264 JMP PLP54 /CONTINUE TEST
1258          /
1259          /
1260          /
1261          2366 0037
1262          2367 2400
1263          2370 0012
1264          2371 7740
1265          2372 0177
1266          2373 4145
1267          2374 5670
1268          2375 4144
1269          2376 1234
1270          2377 0030
1271          2400 PAGE
1272          /
1273          /ROUTINE TO PRINT "ACA:" FOR RAPID PRINT WHEEL VIBRATION
1274          /
1275          2400 4433 TST55, BGNSTST /INIT FOR TEST
1276          2401 1377 TAD (12 /10 LINES TO PRINT
1277          2402 3132 DCA LINES /SAVE IT
1278          2403 7005 IAC RAL /SET BIT 10
1279          2404 4431 LOLS /LIFT RIBBON
1280          2405 1376 PLP55, TAD (101 /CODE FOR "A"
1281          2406 3124 DCA CHAR /SAVE CHARACTER

```

```

1282 2407 4452 KBCHK /CHECK FOR OPERATOR INTERVENTION
1283 2410 4464 PRINT /PRINT "A"
1284 2411 4454 LINEND /CHECK FOR END OF LINE
1285 2412 7410 SKP /SKIP IF NOT END OF LINE
1286 2413 5251 JMP ELIN55 /SET UP FOR NEXT LINE
1287 2414 4462 RSPACE /SPACE CARRIAGE
1288 2415 7300 CLA CLL
1289 2416 1375 TAD (103 /GET CODE FOR "C"
1290 2417 3124 DCA CHAR /SAVE IT
1291 2420 4452 KBCHK /CHECK FOR OPERATOR INTERVENTION
1292 2421 4464 PRINT /PRINT "C"
1293 2422 4454 LINEND /CHECK FOR END OF LINE
1294 2423 7410 SKP /SKIP IF NOT END OF LINE
1295 2424 5251 JMP ELIN55 /SET UP FOR NEXT LINE
1296 2425 4462 RSPACE /SPACE CARRIAGE
1297 2426 7300 CLA CLL
1298 2427 1376 TAD (101 /GET CODE FOR "A"
1299 2430 3124 DCA CHAR /SAVE IT
1300 2431 4452 KBCHK /CHECK FOR OPERATOR INTERVENTION
1301 2432 4464 PRINT /PRINT THE "A"
1302 2433 4454 LINEND /CHECK FOR END OF LINE
1303 2434 7410 SKP /SKIP IF NOT END OF LINE
1304 2435 5251 JMP ELIN55 /SET UP FOR NEXT LINE
1305 2436 4462 RSPACE /SPACE CARRIAGE
1306 2437 7300 CLA CLL
1307 2440 1374 TAD (72 /GET CODE FOR ":"
1308 2441 3124 DCA CHAR /SAVE IT
1309 2442 4452 KBCHK /CHECK FOR OPERATOR INTERVENTION
1310 2443 4464 PRINT /PRINT THE ":"
1311 2444 4454 LINEND /CHECK FOR END OF LINE
1312 2445 7410 SKP /SKIP IF NOT END OF LINE
1313 2446 5251 JMP ELIN55 /SET UP FOR NEXT LINE
1314 2447 4462 RSPACE /SPACE CARRIAGE
1315 2450 5205 JMP PLP55 /CONTINUE THIS LINE
1316
1317 2451 2115 ELIN55, ISZ LINCNT /INCREMENT LINE COUNT
1318 2452 7000 NOP /NO SKIP
1319 2453 4461 XCRLF /DO A CR AND LF
1320 2454 4460 LSTLIN /CHECK FOR LAST LINE
1321 2455 5205 JMP PLP55 /CONTINUE TEST
1322 2456 4440 ENDTST /CHECK FOR TEST LOOP
1323 2457 4467 MESSAGE /PRINT PASS COUNT MESSAGE
1324 2460 6137 PASCNT /MESSAGE POINTER
1325 2461 2265 ISZ PCOUNT /INCREMENT PASS COUNT
1326 2462 1265 TAD PCOUNT /GET PASS COUNT
1327 2463 4470 HEGPRT /PRINT PASS COUNT
1328 2464 5773 JMP TST45+1 /LOOP ON PRINTER TESTS
1329 /
1330 /
1331 2465 0000 PCOUNT, 0 /PASS COUNT
1332 /
1333 /ROUTINE TO ACCEPT UP TO ONE LQ PRINT LINE OF CHARACTERS FROM
1334 /OPERATOR'S TERMINAL AND SEND THEM OUT REPEATEDLY TO LQ PRINTER,
1335 /THE PRINTING OF THESE CHARACTERS WILL BEGIN AUTOMATICALLY IF
1336 /THE NUMBER OF CHARACTERS INPUT BY THE OPERATOR EQUALS ONE

```

```

1337 /PRINT LINE IN LENGTH, THE OPERATOR MAY ALSO INITIATE THE
1338 /PRINTING SEQUENCE BY TYPING CARRIAGE RETURN, WHEN ANY
1339 /CHARACTER IS TYPED ON THE OPERATOR'S CONSOLE, THE LQ PRINTING
1340 /WILL CEASE UNTIL A CARRIAGE RETURN IS TYPED OR THE LINE LENGTH
1341 /IS REACHED.
1342 /
1343 2466 4433 TST56, HGTST
1344 2467 1372 TAD (BUFF-1 /GET ADDRESS OF CHARACTER BUFFER
1345 2470 3011 DCA INDX11 /SET UP ADDRESS POINTER
1346 2471 3353 DCA CNT56A /CLEAR LOCATION COUNTER
1347 2472 3411 CLR56A, DCA I INDX11 /CLEAR A CHARACTER BUFFER LOCATION
1348 2473 2353 ISZ CNT56A /INCREMENT LOCATION COUNTER
1349 2474 1131 TAD LINE /GET LINE LENGTH
1350 2475 7041 CIA /NEGATE IT
1351 2476 1353 TAD CNT56A /INCREMENT LOCATION COUNTER
1352 2477 7640 SZA CLA /SKIP IF EQUAL
1353 2500 5272 JMP CLR56A /CONTINUE CLEARING BUFFER
1354 2501 3353 DCA CNT56A /CLEAR LOCATION COUNTER
1355 2502 1372 TAD (BUFF-1 /GET ADDRESS OF CHARACTER BUFFER
1356 2503 3011 DCA INDX11 /SET UP ADDRESS POINTER
1357 2504 6031 LOP56A, KSF /WAIT FOR OPERATOR INTERVENTION
1358 2505 5304 JMP *-1
1359 2506 4452 KBCHK
1360 2507 7300 CLA CLL
1361 2510 1137 TAD KBSAV /GET NEGATED CHARACTER
1362 2511 1371 TAD (15 /WAS IT CARRIAGE RETURN?
1363 2512 7640 SZA CLA /SKIP IF YES
1364 2513 7410 SKP /SKIP IF NO
1365 2514 5326 JMP CLR56B /GO SET UP FOR LQ PRINTING
1366 2515 1137 TAD KBSAV /GET NEGATED CHARACTER
1367 2516 7041 CIA /MAKE IT POSITIVE
1368 2517 3411 DCA I INDX11 /SAVE IT IN BUFFER
1369 2520 2353 ISZ CNT56A /INCREMENT LOCATION COUNTER
1370 2521 1131 TAD LINE /GET LINE LENGTH
1371 2522 7041 CIA /NEGATE IT
1372 2523 1353 TAD CNT56A /ADD LOCATION COUNTER
1373 2524 7640 SZA CLA /SKIP IF EQUAL
1374 2525 5304 JMP LOP56A /CONTINUE STORING CHARACTERS
1375 2526 3354 CLR56B, DCA CNT56B /CLEAR CHARACTER DUMP COUNTER
1376 2527 1372 TAD (BUFF-1 /GET ADDRESS OF CHARACTER BUFFER
1377 2530 3011 DCA INDX11 /SET UP ADDRESS POINTER
1378 2531 7305 CLA CLL IAC RAL /SET BIT 10
1379 2532 4431 LQLS /LIFT RIBBON
1380 2533 4461 XCRLF /EXECUTE CARRIAGE RETURN, LINE FEED ON LQ
1381 2534 1411 LOP56B, TAD I INDX11 /GET A CHARACTER FROM BUFFER
1382 2535 3124 DCA CHAR /SET UP FOR LQ PRINT
1383 2536 6031 KSF /TEST KEYBOARD FLAG
1384 2537 7410 SKP /SKIP IF NOT SET
1385 2540 5266 JMP TST56 /BEGIN AGAIN
1386 2541 4464 PRINT /PRINT ON LQ
1387 2542 2354 ISZ CNT56B /INCREMENT CHARACTER DUMP COUNTER
1388 2543 1354 TAD CNT56B
1389 2544 7041 CIA
1390 2545 1353 TAD CNT56A
1391 2546 7640 SZA CLA

```

```

1392 2547 7410 SKP /SKIP IF NOT EQUAL
1393 2550 5326 JMP CLR56B /START NEXT LINE
1394 2551 4462 RSPACE /SPACE ONE POSITION TO THE RIGHT
1395 2552 5334 JMP LOPS6B /CONTINUE PRINTING ON CURRENT LINE
1396 2553 0000 CNT56A, 0 /STORAGE FOR BUFFER LOCATION COUNTER
1397 2554 0000 CNT56B, 0
1398 /
1399 /ROUTINE TO RAISE AND LOWER RIBBON
1400 /
1401 2555 4433 TST57, BGMTST
1402 2556 7305 CLA CLL IAC RAL /SET BIT 10
1403 2557 4431 LQLS /LIFT RIBBON
1404 2560 4443 DELAY /DELAY
1405 2561 7777 7777
1406 2562 7300 CLA CLL
1407 2563 4431 LQLS /LOWER RIBBON
1408 2564 4443 DELAY
1409 2565 7777 7777
1410 2566 4452 KRCHK /CHECK FOR OPERATOR INTERVENTION
1411 2567 5356 JMP TST57+1 /LOOP ON TEST
1412 /
1413 /
1414 /
1415 2571 0015
1416 2572 6777
1417 2573 1401
1418 2574 0072
1419 2575 0103
1420 2576 0101
1421 2577 0012
1422 2600 PAGE
1423 /
1424 /ROUTINE TO PRINT A 10 BY 10 DOT MATRIX
1425 /NOTE: THIS IS A TEST OF THE FRICTION FEED MECHANISM
1426 /AND THEREFORE THE PAPER IN THE LQ PRINTER MUST
1427 /BE FED FROM THE BACK OF THE PRINTER AND THE
1428 /TRACTOR FEED MECHANISM MUST BE REMOVED
1429 /
1429 2600 4433 TST60, BGMTST
1430 2601 7305 CLA CLL IAC RAL /SET BIT 10
1431 2602 4431 LQLS /LIFT RIBBON
1432 2603 1377 TAD (1234
1433 2604 3776 DCA RAN1 /RESTORE RANDOM NUMBER PRIME
1434 2605 1375 TAD (5670
1435 2606 3774 DCA RAN2 /RESTORE RANDOM NUMBER PRIME
1436 2607 1373 TAD (-144
1437 2610 3326 DCA CNT60A /PREPARE TO CLEAR A 100 LOC BUFFER
1438 2611 3330 DCA NUM60A /CLEAR WORK AREA
1439 2612 3331 DCA NUM60B /CLEAR WORK AREA
1440 2613 3317 DCA LIN60A /CLEAR WORK AREA
1441 2614 3322 DCA COL60A /CLEAR WORK AREA
1442 2615 1373 TAD (-144
1443 2616 3325 DCA DOTCNT /PREPARE TO PRINT A 100 DOT MATRIX
1444 2617 1372 TAD (-12
1445 2620 3327 DCA CNT60B

```

```

1446 2621 1371 TAD (BUFF-1 /GET BUFFER ADDRESS
1447 2622 3010 DCA INDX10 /SAVE IT FOR CLEARING
1448 2623 3410 DCA I INDX10 /CLEAR A BUFFER LOCATION
1449 2624 2326 ISZ CNT60A /INCREMENT COUNTER
1450 2625 5223 JMP -2 /CONTINUE CLEARING
1451 2626 1370 TAD (BUFF1-1 /GET SECOND BUFFER ADDRESS
1452 2627 3010 DCA INDX10
1453 2630 1372 TAD (-12
1454 2631 3326 DCA CNT60A /PREPARE TO LOAD SECOND BUFFER
1455 2632 3410 DCA I INDX10 /LOAD A BUFFER LOCATION
1456 2633 2330 ISZ NUM60A /INCREMENT NUMBER TO LOAD
1457 2634 1330 TAD NUM60A
1458 2635 1331 TAD NUM60B /CALCULATE NEXT NUMBER
1459 2636 2326 ISZ CNT60A /INCREMENT COUNTER
1460 2637 5232 JMP LP60B /CONTINUE LOADING BUFFER
1461 2640 7300 CLA CLL
1462 2641 1331 TAD NUM60B
1463 2642 1367 TAD (100 /CALCULATE NEXT NUMBER
1464 2643 3331 DCA NUM60B /SAVE IT
1465 2644 3330 DCA NUM60A /CLEAR INCREMENTAL COUNTER
1466 2645 1372 TAD (-12
1467 2646 3326 DCA CNT60A /SET UP COUNTER FOR NEXT LOOP
1468 2647 1330 TAD NUM60A /GET COLUMN COUNTER
1469 2650 1331 TAD NUM60B /ADD LINE COUNTER
1470 2651 2327 ISZ CNT60B /INCREMENT LOOP COUNTER
1471 2652 5232 JMP LP60B /CONTINUE LOADING BUFFER
1472 2653 4472 LP60C, RANDOM /GET A RANDOM NUMBER
1473 2654 0366 AND (177 /MASK TO 7 BITS
1474 2655 3314 DCA RANAD1 /SAVE IT
1475 2656 1314 TAD RANAD1
1476 2657 1373 TAD (-144 /TEST FOR BUFFER ADDRESS
1477 2660 7700 SMA CLA /SKIP IF WITHIN BUFFER
1478 2661 5253 JMP LP60C /GET A NEW RANDOM NUMBER
1479 2662 1314 TAD RANAD1
1480 2663 1365 TAD (BUFF
1481 2664 3315 DCA RANAD2
1482 2665 1715 TAD I RANAD2 /GET RANDOM ADDRESS USED INDICATOR
1483 2666 7640 SZA CLA /SKIP IF THIS LOCATION NOT USED
1484 2667 5253 JMP LP60C /GET A NEW ADDRESS
1485 2670 2715 ISZ I RANAD2 /REMEMBER THIS LOCATION USED
1486 2671 1314 TAD RANAD1 /GET RANDOM NUMBER
1487 2672 1364 TAD (BUFF1 /CALCULATE ADDRESS IN SECOND BUFFER
1488 2673 3316 DCA RANAD3 /SAVE IT
1489 2674 1716 TAD I RANAD3 /GET LINE AND COLUMN INFORMATION
1490 2675 0363 AND (7700 /MASK FIRST 6 BITS
1491 2676 7002 BSW /SWAP INTO OTHER BYTE
1492 2677 3320 DCA LIN60B /SAVE LINE # 0-11
1493 2700 1716 TAD I RANAD3 /GET LINE AND COLUMN INFORMATION
1494 2701 0362 AND (77 /MASK SECOND 6 BITS
1495 2702 3323 DCA COL60B /SAVE COLUMN # 0-11
1496 2703 1317 TAD LIN60A
1497 2704 7041 CIA /NEGATE IT
1498 2705 1320 TAD LIN60B /ADD NEW LINE NUMBER
1499 2706 3321 DCA LIN60C /SAVE RELATIVE LINE NUMBER
1500 2707 1322 TAD COL60A /GET OLD COLUMN NUMBER

```

1501	2710	7041	CIA	/NEGATE IT
1502	2711	1323	TAD COL60B	/ADD NEW COLUMN NUMBER
1503	2712	3324	DCA COL60C	/SAVE RELATIVE COLUMN NUMBER
1504	2713	5761*	JMP CAL60	
1505			/	
1506			/	
1507	2714	0000	RANAD1, 0	/RANDOM ADDRESS WORK AREA #1
1508	2715	0000	RANAD2, 0	/RANDOM ADDRESS WORK AREA #2
1509	2716	0000	RANAD3, 0	/RANDOM ADDRESS WORK AREA #3
1510	2717	0000	LIN60A, 0	/LINE NUMBER WORK AREA #1
1511	2720	0000	LIN60B, 0	/LINE NUMBER WORK AREA #2
1512	2721	0000	LIN60C, 0	/LINE NUMBER WORK AREA #3
1513	2722	0000	COL60A, 0	/COLUMN NUMBER WORK AREA #1
1514	2723	0000	COL60B, 0	/COLUMN NUMBER WORK AREA #2
1515	2724	0000	COL60C, 0	/COLUMN NUMBER WORK AREA #3
1516	2725	0000	DOTCNT, 0	/DOT MATRIX COUNTER
1517	2726	0000	CNT60A, 0	/COUNTER #1
1518	2727	0000	CNT60B, 0	/COUNTER #2
1519	2730	0000	NUM60A, 0	/NUMBER WORK AREA #1
1520	2731	0000	NUM60B, 0	/NUMBER WORK AREA #2
1521			/	
1522			/	
1523			/	
1524	2761	3000		
1525	2762	0077		
1526	2763	7700		
1527	2764	7145		
1528	2765	7000		
1529	2766	0177		
1530	2767	0100		
1531	2770	7144		
1532	2771	6777		
1533	2772	7766		
1534	2773	7634		
1535	2774	4145		
1536	2775	5670		
1537	2776	4144		
1538	2777	1234		
		3000	PAGE	
1539			/	
1540			/	
1541	3000	1777*	CAL60, TAD LIN60C	/GET RELATIVE LINE NUMBER
1542	3001	7450	SNA	/SKIP IF NEGATIVE OR POSITIVE
1543	3002	5215	JMP ZMOVEA	/GO TO MOVE ZERO ROUTINE
1544	3003	7510	SPA	/SKIP IF POSITIVE
1545	3004	5217	JMP RETA	
1546	3005	7041	FEEDA, CIA	/NUMBER IS POSITIVE, NEGATE IT
1547	3006	3777*	DCA LIN60C	/SAVE IT
1548	3007	1312	TAD LININ	/ADD INCREMENTS OF ONE INCH
1549	3010	2777*	ISZ LIN60C	/INCREMENT NUMBER OF INCHES
1550	3011	5207	JMP ,=2	/CONTINUE INCREMENTING
1551	3012	3256	DCA MAG60A	/SAVE PAPER MOVEMENT MAGNITUDE
1552	3013	3255	DCA DIR60A	/SAVE PAPER MOVEMENT DIRECTION
1553	3014	5226	JMP COLCAL	/GO TO COLUMN CALCULATE ROUTINE
1554	3015	3256	ZMOVEA, DCA MAG60A	/CLEAR PAPER MOVEMENT MAGNITUDE

1555	3016	5276	JMP COLCAL	/GO TO COLUMN CALCULATE ROUTINE
1556	3017	3777*	DCA LIN60C	
1557	3020	1312	TAD LININ	/ADD INCREMENTS OF INCHES
1558	3021	2777*	ISZ LIN60C	/INCREMENT INCH COUNTER
1559	3022	5220	JMP ,=2	/CONTINUE ADDING INCHES
1560	3023	3256	DCA MAG60A	/SAVE PAPER MOVEMENT MAGNITUDE
1561	3024	7313	CLA CLL IAC RTR	/SET BIT 0
1562	3025	3255	DCA DIR60A	/SAVE PAPER MOVEMENT DIRECTION
1563	3026	1776*	COLCAL, TAD COL60C	/GET RELATIVE COLUMN POSITION
1564	3027	7450	SNA	/SKIP IF NEGATIVE OR POSITIVE
1565	3030	5243	JMP ZMOVEB	/GO TO ZERO CARRIAGE MOVEMENT ROUTINE
1566	3031	7510	SPA	/SKIP IF POSITIVE
1567	3032	5245	JMP LMOVEB	/GO TO CARRIAGE LEFT ROUTINE
1568	3033	7041	RMOVEB, CIA	
1569	3034	3776*	DCA COL60C	
1570	3035	1313	TAD COLIN	/ADD INCREMENT OF ONE INCH
1571	3036	2776*	ISZ COL60C	/INCREMENT INCH COUNTER
1572	3037	5235	JMP ,=2	/CONTINUE ADDING INCHES
1573	3040	3261	DCA MAG60B	/SAVE CARRIAGE MOVEMENT MAGNITUDE
1574	3041	3260	DCA DIR60B	/SAVE CARRIAGE MOVEMENT DIRECTION=RIGHT
1575	3042	5254	JMP MOV60	/GO MOVE PAPER AND CARRIAGE
1576	3043	3261	ZMOVEB, DCA MAG60B	/ZERO CARRIAGE MOVEMENT
1577	3044	5254	JMP MOV60	/GO MOVE PAPER
1578	3045	3776*	LMOVEB, DCA COL60C	
1579	3046	1313	TAD COLIN	/ADD INCREMENT OF ONE INCH
1580	3047	2776*	ISZ COL60C	/INCREMENT INCH COUNTER
1581	3050	5246	JMP ,=2	/CONTINUE ADDING INCHES
1582	3051	3261	DCA MAG60B	/SAVE CARRIAGE MOVEMENT MAGNITUDE
1583	3052	7313	CLA CLL IAC RTR	/SET BIT ZERO FOR LEFT DIRECTION
1584	3053	3260	DCA DIR60B	
1585			/	
1586	3054	4457	MOV60, MOVPAR	/MOVE PAPER
1587	3055	0000	DIR60A, 0	/PAPER DIRECTION
1588	3056	0000	MAG60A, 0	/PAPER MOVEMENT MAGNITUDE
1589	3057	4455	MOVCAR	/MOVE CARRIAGE
1590	3060	0000	DIR60B, 0	/CARRIAGE DIRECTION
1591	3061	0000	MAG60B, 0	/CARRIAGE MOVEMENT MAGNITUDE
1592	3062	4452	KBCHK	/CHECK FOR OPERATOR INTERVENTION
1593	3063	7300	CLA CLL	
1594	3064	1775*	TAD LIN60B	/GET NEW LINE NUMBER
1595	3065	3774*	DCA LIN60A	/SAVE IT AS OLD LINE NUMBER
1596	3066	1773*	TAD COL60B	/GET NEW COLUMN NUMBER
1597	3067	3772*	DCA COL60A	/SAVE IT AS OLD COLUMN NUMBER
1598	3070	1311	TAD CSWITCH	/GET CHARACTER PRINT SWITCH
1599	3071	7650	SNA CLA	/SKIP IF NOT ZERO
1600	3072	5276	JMP ,+4	/PRINT A DOT MATRIX
1601	3073	1775*	TAD LIN60B	/GET NEW LINE NUMBER
1602	3074	1773*	TAD COL60B	/ADD NEW COLUMN NUMBER
1603	3075	1371	TAD (301-56	
1604	3076	1370	TAD (56	
1605	3077	3124	DCA CHAR	/SAVE CHARACTER TO PRINT
1606	3100	4464	PRINT	/PRINT CHARACTER AT CURRENT COORDINATES
1607	3101	2767*	ISZ DOTCNT	/INCREMENT CHARACTER COUNT
1608	3102	5766*	JMP LP60C	/GET NEXT CHARACTER COORDINATE
1609	3103	4457	MOVPAR	/MOVE PAPER

```

1610 3104 0000 0
1611 3105 1100 1100 /6 INCHES
1612 3106 4442 SWAIT /WAIT FOR READY BITS
1613 3107 4451 STATER /ALL READY BITS NOT UP
1614 3110 5765* JMP TST60 /LOOP ON TEST
1615 /
1616 3111 7777 CSWTCH, 7777 /CHARACTER SWITCH 0000=DOTS XXX1=ALPHABETICAL
1617 3112 0140 LININ, 0140
1618 3113 0170 COLIN, 0170
1619 /
1620 /
1621 /
1622 /
1623 /ROUTINE TO SEND OUT CONTINUOUS RESTORE SIGNALS
1624 /UNTIL A CONTROL CHARACTER IS TYPED BY OPERATOR
1625 /
1626 /
1627 3114 4432 RESTLP, LORE /RESTORE PRINTER
1628 3115 4452 KBCHK /CHECK FOR OPERATOR INTERVENTION
1629 3116 5314 JMP ,*-2 /CONTINUE SENDING RESTORE
1630 /
1631 /
1632 /
1633 /ROUTINE TO RAISE OR LOWER RIBBON MECHANISM
1634 /
1635 /
1636 /TYPE ANY KEY ON OPERATOR'S TERMINAL TO CHANGE POSITION
1637 /OF RIBBON (UP OR DOWN). TYPE A CONTROL CHARACTER TO
1638 /EXIT FROM THIS ROUTINE.
1639 /
1640 /
1641 3117 7305 RIBBON, CLA CLL IAC RAL /SET BIT 10
1642 3120 4431 LGLS /RAISE RIBBON
1643 3121 6030 KCF /CLEAR KEYBOARD FLAG
1644 3122 6031 KSF /WAIT FOR KEYBOARD FLAG
1645 3123 5322 JMP ,*-1 /CHECK FOR CONTROL CHARACTER
1646 3124 4452 KBCHK /CHECK FOR CONTROL CHARACTER
1647 3125 7300 CLA CLL /LOWER RIBBON
1648 3126 4431 LGLS /LOWER RIBBON
1649 3127 6031 KSF /WAIT FOR KEYBOARD FLAG
1650 3130 5322 JMP ,*-1 /CHECK FOR CONTROL CHARACTER
1651 3131 4452 KBCHK /CHECK FOR CONTROL CHARACTER
1652 3132 5317 JMP RIBBON /GO RAISE RIBBON AGAIN
1653 /
1654 3165 2600
1655 3166 2653
1656 3167 2725

1657 3170 0056
1658 3171 0223
1659 3172 2722
1660 3173 2723
1661 3174 2717
1662 3175 2720
1663 3176 2724
1664 3177 2721
    
```

```

1665 3200 PAGE
1666 /
1667 /
1668 /
1669 /*****
1670 /
1671 / BEGINNING OF SUBROUTINES
1672 /
1673 /*****
1674 /
1675 /
1676 /LQP8 IOT SUBROUTINES
1677 /
1678 /THIS ROUTINE WILL SKIP THE NEXT INSTRUCTION IF DONE FLAG IS SET
1679 /
1680 3200 0000 LQSKRT, 0
1681 3201 6500 IOT0, XLOSK /SKIP ON DONE FLAG
1682 3202 5600 JMP I LQSKRT /DID NOT SKIP, GO TO ERROR ROUTINE
1683 3203 2200 ISZ LQSKRT /SKIP SUCCESSFUL
1684 3204 5600 JMP I LQSKRT /NORMAL RETURN
1685 /
1686 /THIS ROUTINE WILL READ THE CONTENTS OF THE DATA BUFFER INTO AC
1687 /
1688 3205 0000 LQRBRT, 0
1689 3206 6501 IOT1, XLOQB /READ DATA BUFFER
1690 3207 7410 SKP /READ DATA SKIPPED
1691 3210 4434 ERROR /READ DATA SKIPPED
1692 3211 3122 DCA SAVDAT /SAVE DATA BUFFER
1693 3212 1122 TAD SAVDAT /GET DATA BUFFER
1694 3213 5605 JMP I LQRBRT /NORMAL RETURN
1695 /
1696 /THIS ROUTINE WILL ADVANCE OR RETRACT THE PAPER AND EXIT WITH AC CLEARED
1697 /
1698 3214 0000 LQMPRT, 0
1699 3215 3123 DCA SAVPR /SAVE PAPER MOTION MAGNITUDE
1700 3216 1123 TAD SAVPR /GET MAGNITUDE
1701 3217 6502 IOT2, XLOMP /MOVE PAPER
1702 3220 5614 JMP I LQMPRT /NORMAL RETURN
1703 3221 4434 ERROR /MOVE PAPER SKIPPED
1704 3222 5614 JMP I LQMPRT /ERROR RETURN
1705 /
1706 /THIS ROUTINE WILL MOVE THE CARRIAGE LEFT OR RIGHT AND EXIT WITH AC CLR
1707 /
1708 3223 0000 LQMCRT, 0
1709 3224 3120 DCA SAVMV /SAVE MOVE MAGNITUDE
1710 3225 1120 TAD SAVMV /GET MAGNITUDE
1711 3226 6503 IOT3, XLOMC /MOVE CARRIAGE
1712 3227 5623 JMP I LQMCRT /NORMAL RETURN
1713 3230 4434 ERROR /MOVE CARRIAGE SKIPPED
1714 3231 5623 JMP I LQMCRT /ERROR RETURN
1715 /
1716 /
1717 /
1718 /THIS ROUTINE WILL PRINT THE CHARACTER IN AC BITS 4-10 AND EXIT WITH AC CLR
    
```

```

1719 /
1720 3232 0000 LQPCRT, 0
1721 3233 6504 IOT4, XLQPC /PRINT CHARACTER
1722 3234 5632 JMP I LQPCRT /NORMAL RETURN
1723 3235 4434 ERROR /PRINT IOT SKIPPED
1724 3236 5632 JMP I LQPCRT /ERROR RETURN
1725 /
1726 /THIS ROUTINE WILL READ THE STATUS REGISTER INTO AC
1727 /
1728 3237 0000 LQRSRT, 0
1729 3240 6505 IOTS, XLQRS /READ STATUS REGISTER
1730 3241 7410 SKP
1731 3242 4434 ERROR /READ STATUS SKIPPED
1732 3243 3114 DCA SVSTAT /SAVE STATUS
1733 3244 1114 TAD SVSTAT /GET STATUS
1734 3245 5637 JMP I LQRSRT /NORMAL RETURN
1735 /
1736 /THIS ROUTINE LOADS STATUS REGISTER FROM AC
1737 /
1738 3246 0000 LQLSRT, 0
1739 3247 3114 DCA SVSTAT /SAVE STATUS TO SEND
1740 3250 1114 TAD SVSTAT /GET STATUS TO SEND
1741 3251 0377 AND (117 /MASK BITS 5,8,9,10,11 (READ,WRITE BITS)
1742 3252 3117 DCA XSTAT /SAVE EXPECTED STATUS
1743 3253 1114 TAD SVSTAT /GET STATUS TO SEND
1744 3254 6506 IOT6, XLQLS /LOAD STATUS REGISTER
1745 3255 5646 JMP I LQLSRT /NORMAL RETURN
1746 3256 4434 ERROR /LOAD STATUS SKIPPED
1747 3257 5646 JMP I LQLSRT /ERROR RETURN
1748 /
1749 /THIS ROUTINE RESTORES CARRIAGE TO HOME POSITION
1750 /
1751 3260 0000 LQRERT, 0
1752 3261 6507 IOT7, XLQRE /RESTORE CARRIAGE TO HOME POSITION
1753 3262 5660 JMP I LQRERT /NORMAL RETURN
1754 3263 4434 ERROR /RESTORE SKIPPED
1755 3264 5660 JMP I LQRERT /ERROR RETURN
1756 /
1757 /ROUTINE TO LOAD DATA BUFFER
1758 /CALL IS LOADDB FOLLOWED BY IOT 2,3,OR 4
1759 /RETURNS TO CALL PLUS TWO IF DATA ERROR, CALL PLUS THREE IF DATA OK
1760 /NOTE: ON THE VT78, DATA READ FROM DATA BUFFER IS THE COMPLEMENT OF DATA WRITTEN
1761 /
1762 /
1763 3265 0000 DBLOAD, 0 /ROUTINE TO LOAD AND CHECK DATA BUFFER
1764 3266 3077 DCA SVAC /SAVE NUMBER TO LOAD
1765 3267 1077 TAD SVAC
1766 /
1767 3270 7041 V78OW1, CIA/NOP /VT78/ CREATE COMPLEMENT TO CHECK ANSWER; NOP ON VT78
1768 3271 3311 DCA COMPI /SAVE COMPLEMENT
1769 3272 1665 TAD I DBLOAD /GET IOT CALL INSTRUCTION
1770 3273 3275 DCA XCT1 /STORE IOT FOR EXECUTION
1771 3274 1077 TAD SVAC /GET NUMBER TO LOAD IN DATA BUFFER
1772 3275 0000 XCT1, 0 /MODIFIED JMS TO IOT SUBROUTINE
1773 3276 7440 SZA /SKIP IF AC CLEARED
1774 3277 4434 ERROR /AC NOT CLEARED BY IOT

```

```

1774 3300 4424 LQRB /READ DATA BUFFER
1775 3301 7000 V78OW2, NOP/IAC /VT78/ IAC FOR VT78
1776 3302 3122 DCA SAVDAT /SAVE DATA
1777 3303 1122 TAD SAVDAT /GET DATA
1778 3304 1311 TAD COMPI /ADD ONE'S COMPLEMENT OF EXPECTED DATA
1779 3305 7450 SMA /SKIP IF DATA READ NOT EQUAL TO DATA EXPECTED
1780 3306 2265 ISZ DBLOAD /BUMP RETURN ADDRESS POINTER IF OK
1781 3307 2265 ISZ DBLOAD /BUMP RETURN ADDRESS
1782 3310 5665 JMP I DBLOAD
1783 3311 0000 COMPI, 0
1784 /
1785 /
1786 /ROUTINE TO CHECK READY BITS
1787 /
1788 3312 0000 CHKRDY, 0
1789 3313 4430 LQRS /READ STATUS IOTS
1790 3314 3114 DCA SVSTAT /SAVE STATUS
1791 3315 1114 TAD SVSTAT /GET STATUS
1792 3316 0376 AND (40 /TEST FOR CHECK ERROR
1793 3317 7640 SZA CLA /SKIP IF NO CHECK ERROR
1794 3320 4451 STATER /CHECK CONDITION ON LQ PRINTER
1795 3321 1114 TAD SVSTAT
1796 3322 0375 AND (7400 /MASK BITS 0,1,2,3
1797 3323 5712 JMP I CHKRDY /RETURN
1798 /
1799 /
1800 /ROUTINE TO INITIALIZE LQP8 MODULE FOR STATIC TESTS
1801 /
1802 3324 0000 TSTIN, 0 /CALLED BY "INST"
1803 3325 4452 KBCHK /CHECK FOR OPERATOR INTERVENTION
1804 3326 6002 IOF
1805 3327 6007 CAF /CLEAR ALL FLAGS
1806 3330 4432 LORE /VT78/ CAF IN VT78 DOES NOT ACTIVATE RESTORE STROBE
1807 3331 3117 DCA XSTAT /SAVE EXPECTED STATUS
1808 3332 4446 RDYCHK /CHECK READY BITS FOR CLEAR CONDITION
1809 3333 7440 SZA /SKIP IF ALL CLEAR
1810 /
1811 3334 4451 STATER /ONE OR MORE READY BITS SET
1812 3335 7340 CLA CLL CMA /SUBTRACT ONE FROM JMS FOR TST LOUP
1813 3336 1324 TAD TSTIN /GET TEST LOOP ADDRESS
1814 3337 3350 DCA TSTLOP /SAVE TEST LOOP ADDRESS
1815 3340 3116 DCA INTFLG /CLEAR SOFTWARE INTERRUPT FLAG
1816 3341 5724 JMP I TSTIN
1817 /
1818 /
1819 /ROUTINE TO INITIATE TEST LOOPING
1820 /
1821 3342 0000 TSTEND, 0
1822 3343 4465 LAS /CHECK SWITCH REGISTER
1823 3344 7006 RTL /FOR BIT 2 = 1 TEST LOOP
1824 3345 7700 SMA CLA /SKIP IF YES
1825 3346 5742 JMP I TSTEND /CONTINUE TO NEXT TEST
1826 3347 5750 JMP I TSTLOP /LOOP ON CURRENT TEST
1827 3350 0000 TSTLOP, 0
1828 /

```

```

1829
1830 /ROUTINE TO PERFORM RESTORE OPERATION
1831 /
1832 3351 0000 RESTOR, 0
1833 3352 6002 IOF
1834 3353 7300 CLA CLL
1835 3354 3140 DCA SAVPOS /CLEAR POSITION POINTER
1836 3355 4432 LQRE /RESTORE
1837 3356 1375 TAD (7400
1838 3357 3133 DCA WSTAT /SAVE WANTED STATUS
1839 3360 4442 SWAIT /WAIT FOR ALL READY BITS SET
1840 3361 4451 STATER /ALL READY BITS NOT UP
1841 3362 7300 CLA CLL
1842 3363 5751 JMP I RESTOR
1843 /
1844 3375 7400 /
1845 3376 0040 PAGE
1846 3377 0117 /
3400 /
1847 /
1848 /
1849 /ROUTINE TO SET UP TEST, INITIALIZE MODULE
1850 /SET UP SCOPE LOOPS
1851 /CALLED BY BGNST FOLLOWED BY 0 IF SUBTEST LOOP
1852 /FOLLOWED BY SUBTEST LOOP ADDRESS
1853 /
1854 3400 0000 TSTBGN, 0
1855 3401 6002 IOF /INTERRUPT OFF
1856 3402 7340 CLA CLL CMA /SUBTRACT ONE FROM JMS FOR TEST LOOP
1857 3403 1200 TAD TSTBGN /GET TEST LOOP ADDRESS
1858 3404 3777 DCA TSTLGP /SAVE TEST LOOP ADDRESS
1859 3405 3116 DCA INTFLG /CLEAR SOFTWARE PROGRAM INT FLAG
1860 3406 4452 KBCHK /CHECK FOR OPERATOR INTERVENTION
1861 3407 3115 DCA LINCNT /CLEAR LINE COUNTER
1862 3410 3136 DCA CHRCNT /CLEAR CHARACTER COUNTER
1863 3411 3140 DCA SAVPOS /CLEAR POSITION POINTER
1864 3412 1600 TAD I TSTBGN /CHECK FOR SUBTEST LOOP
1865 3413 7640 SZA CLA /IS THERE A SUBTEST LOOP
1866 3414 5221 JMP INIT /NO, INIT MODULE
1867 3415 2200 ISZ TSTRGN /BUMP JMS TO LOOP ADDRESS
1868 3416 1600 TAD I TSTBGN /GET LOOP ADDRESS
1869 3417 3227 DCA ERRLOP /SAVE IT FOR SCOPE LOOPING
1870 3420 2200 ISZ TSTRGN /BUMP JMS TO RETURN ADDRESS
1871 3421 6007 INIT, CAF /CLEAR ALL FLAGS
1872 3422 4432 LQRE /VI78/ CAF DOES NOT ACTIVATE RESTORE STROBE IN VI78
1873 3423 4442 SWAIT /WAIT FOR ALL READY BITS
1874 3424 4451 STATER /PRINTER NOT READY

1875 3425 4461 XCRLF /EXECUTE A CARRIAGE RETURN AND LINE FEED
1876 3426 5600 JMP I TSTBGN /GO TO TEST
1877 3427 0000 ERRLOP, 0
1878 /
1879 /
1880 /DATA ERROR HANDLER
1881 /
1882 3430 0000 ERRDAT, 0

```

```

1883 3431 1777 TAD TSTLGP /GET TEST LOOP ADDRESS
1884 3432 3256 DCA TLOOP1 /SAVE IT
1885 3433 4466 CRACTV /CHECK FOR ACTIVE CONSOLE
1886 3434 5257 JMP DATPRT /CONSOLE ACTIVE
1887 3435 4465 LAS /CONSOLE NOT ACTIVE, GET SWITCHES
1888 3436 7510 SPA /SKIP IF BIT 0=0
1889 3437 5251 JMP ERD1 /GO CHECK FOR ERROR LOOP
1890 3440 7340 CLA CLL CMA
1891 3441 1230 TAD ERRDAT /GET FAILING PC
1892 3442 7402 HLT /ERROR PC IN AC
1893 3443 7300 CLA CLL
1894 3444 1077 TAD SVAC /GET AC AT TIME OF ERROR
1895 3445 7402 HLT /AC AT TIME OF ERROR IN AC
1896 3446 7300 CLA CLL
1897 3447 1122 TAD SAVDAT /GET DATA READ BACK
1898 3450 7402 HLT /DATA READ BACK IN AC
1899 3451 4465 ERD1, LAS /GET SWITCHES
1900 3452 7004 RAL /ROTATE FOR BIT 1 CHECK
1901 3453 7710 SPA CLA /SKIP IF BIT 1=0
1902 3454 5656 JMP I TLOOP1 /LOOP ON TEST
1903 3455 5630 JMP I ERRDAT /CONTINUE TEST
1904 /
1905 3456 0000 TLOOP1, 0
1906 3457 4465 DATPRT, LAS /GET SWITCHES
1907 3460 0376 AND (200 /MASK BIT 4, INHIBIT ERROR MESSAGE
1908 3461 7640 SZA CLA /SKIP IF BIT 4=0
1909 3462 5304 JMP ERD2 /GO TO WAIT ROUTINE
1910 3463 4467 MESSAGE /PRINT DATA ERROR MESSAGE
1911 3464 6071 LDATER /MESSAGE POINTER
1912 3465 4467 MESSAGE /PRINT PC= MESSAGE
1913 3466 6024 PCEQL /MESSAGE POINTER
1914 3467 7340 CLA CLL CMA
1915 3470 1230 TAD ERRDAT /GET FAILING PC
1916 3471 4470 REGPRT /PRINT FAILING PC
1917 3472 4467 MESSAGE /PRINT SENT MESSAGE
1918 3473 6103 SENT /MESSAGE POINTER
1919 3474 7300 CLA CLL
1920 3475 1077 TAD SVAC /GET DATA SENT
1921 3476 4470 REGPRT /PRINT DATA SENT
1922 3477 4467 MESSAGE /PRINT RECEIVED MESSAGE
1923 3500 6107 RCVD /MESSAGE POINTER
1924 3501 7300 CLA CLL
1925 3502 1122 TAD SAVDAT /GET DATA READ BACK
1926 3503 4470 REGPRT /PRINT DATA READ BACK
1927 3504 4465 ERD2, LAS /GET SWITCHES
1928 3505 7710 SPA CLA /SKIP IF BIT 0=0
1929 3506 5251 JMP ERD1 /GO CHECK FOR TEST LOOP
1930 3507 4467 MESSAGE /PRINT WAITING MESSAGE
1931 3510 6000 WAITNG /MESSAGE POINTER
1932 3511 6030 ERD3, KCF /CLEAR KYBD FLAG
1933 3512 6031 KSF /TEST KYBD FLAG
1934 3513 5312 JMP ,=-1 /CONTINUE WAITING
1935 3514 7300 CLA CLL
1936 3515 6034 KRS /READ KEYBOARD BUFFER, DO NOT CLEAR FLAG
1937 3516 0375 AND (177 /MASK TO 7 BITS

```



```

1938 3517 1374 TAD (-7 /CHECK FOR CONTROL-G
1939 3520 7640 SZA CLA /SKIP IF CONTROL-G
1940 3521 5311 JMP ERD3 /CONTINUE WAITING FOR CONTROL-G
1941 3522 4452 KBCHK /HANDLE A CONTROL-G
1942 3523 5251 JMP ERD1 /GO CHECK FOR ERROR LOOP
1943
1944 /
1945 /
1946 3574 7771
1947 3575 0177
1948 3576 0200
1949 3577 3350
1950 PAGE
1951 /
1952 /ERROR ROUTINE
1953 /CALLED BY ERROR
1954 3600 0000 ERRORS, 0
1955 3601 3077 DCA SVAC /SAVE AC
1956 3602 7501 MQA /GET MQ
1957 3603 3102 DCA SAVMQ /SAVE IT
1958 3604 6004 GTF /GET THE FLAGS
1959 3605 3103 DCA SAVFLG /SAVE THEM
1960 3606 4466 CRACTV /CHECK FOR ACTIVE CONSOLE
1961 3607 5236 JMP ERRPRT /GO TO PRINT ROUTINE
1962 3610 4465 LAS /GET SWITCHES, CONSOLE NOT ACTIVE
1963 3611 7710 SPA CLA /SKIP IF BIT 0=0
1964 3612 5225 JMP ERRS1 /GO CHECK FOR ERROR LOOP
1965 3613 1102 TAD SAVMQ /GET MQ
1966 3614 7421 MQL /RESTORE MQ
1967 3615 1103 TAD SAVFLG /GET FLAGS
1968 3616 6005 RTF /RESTORE FLAGS
1969 3617 7340 CIA CLL CMA
1970 3620 1200 TAD ERRORS /GET FAILING PC
1971 3621 7402 HLT /FAILING PC IN AC
1972 3622 7300 CIA CLL
1973 3623 1077 TAD SVAC /GET AC AT TIME OF ERROR
1974 3624 7402 HLT /AC AT TIME OF ERROR IN AC
1975 3625 7300 ERRS1, CIA CLL
1976 3626 1777 TAD TSTLOP /GET TEST LOOP ADDRESS
1977 3627 3235 DCA TLOOP2 /SAVE IT
1978 3630 4465 LAS /GET SWITCHES
1979 3631 7004 HAL /ROTATE TO CHECK BIT 1
1980 3632 7710 SPA CLA /SKIP IF BIT 1=0
1981 3633 5635 JMP I TLOOP2
1982 3634 5600 JMP I ERRORS
1983 3635 0000 TLOOP2, 0
1984 3636 4465 ERRPRT, LAS /GET SWITCHES
1985 3637 0376 AND (200 /TEST BIT 4
1986 3640 7640 SZA CLA /SKIP IF BIT 4=0
1987 3641 5265 JMP ERRS2 /BIT 4=1, INHIBIT ERROR TYPEOUT
1988 3642 4467 MESSAGE /PRINT ERROR MESSAGE
1989 3643 6115 FAIL /MESSAGE POINTER
1990 3644 4467 MESSAGE /PRINT PC= MESSAGE
1991 3645 6024 PCEQL /MESSAGE POINTER

```

```

1992 3646 7340 CIA CLL CMA
1993 3647 1200 TAD ERRORS /GET FAILING PC
1994 3650 4470 REGPRT /PRINT FAILING PC
1995 3651 4467 MESSAGE /PRINT AC= MESSAGE
1996 3652 6030 ACEQL /MESSAGE POINTER
1997 3653 1077 IAD SVAC /GET AC AT TIME OF ERROR
1998 3654 4470 REGPRT /PRINT CONTENTS OF AC AT TIME OF ERROR
1999 3655 4467 MESSAGE /PRINT MQ= MESSAGE
2000 3656 6035 MQEQL /MESSAGE POINTER
2001 3657 1102 TAD SAVMQ /GET MQ AT TIME OF ERROR
2002 3660 4470 REGPRT /PRINT MQ AT TIME OF ERROR
2003 3661 4467 MESSAGE /PRINT FL= MESSAGE
2004 3662 6042 FLEQL /MESSAGE POINTER
2005 3663 1103 TAD SAVFLG /GET FLAGS
2006 3664 4470 REGPRT /PRINT FLAGS
2007 3665 4465 ERRS2, LAS /GET SWITCHES
2008 3666 7710 SPA CLA /SKIP IF BIT 0=0
2009 3667 5225 JMP ERRS1 /GO CHECK FOR ERROR LOOP
2010 3670 4467 MESSAGE /PRINT WAITING MESSAGE
2011 3671 6000 WAITNG /MESSAGE POINTER
2012 3672 6030 ERRS3, KCF /CLEAR KYBD FLAG
2013 3673 6031 KSF /TEST KYBD FLAG
2014 3674 5273 JMP *-1 /CONTINUE WAITING FOR KYBD FLAG
2015 3675 7300 CIA CLL
2016 3676 6034 KRS /READ KYBD BUFFER, DO NOT CLEAR FLAG
2017 3677 0375 AND (177 /MASK TO 7 BITS
2018 3700 1374 TAD (-7 /CHECK FOR CTRL-G
2019 3701 7640 SZA CLA /SKIP IF CTRL-G
2020 3702 5272 JMP ERRS3
2021 3703 4452 KBCHK /HANDLE CTRL-G
2022 3704 5225 JMP ERRS1 /GO CHECK FOR ERROR LOOP
2023
2024 /
2025 3774 7771
2026 3775 0177
2027 3776 0200
2028 3777 3350
2029 PAGE
2030 /
2031 /
2032 /INTERRUPT SERVICE ROUTINE
2033 /
2034 4000 4447 INTSVC, SAVEAC /SAVE AC AND LINK
2035 4001 6102 SPL /SKIP ON POWER LOW
2036 4002 5224 JMP FLAGS /NOT POWER LOW CHECK FLAGS
2037 4003 7501 MQA /GET MQ
2038 4004 3213 DCA MQ /SAVE IT
2039 4005 1000 TAD 0 /GET PC
2040 4006 3214 DCA PC /SAVE IT
2041 4007 1212 TAD STRTIN /GET RESTART INSTRUCTION
2042 4010 3000 DCA 0 /SAVE IT
2043 4011 7402 HLT /HALT
2044 4012 5403 STRTIN, JMP I ADDR3 /GO TO RESTART ROUTINE
2045 4013 0000 MQ, 0 /MQ STORAGE AREA

```

```

2046 4014 0000 PC, 0 /PC STORAGE AREA
2047 4015 1213 RESTR, TAD MQ /GET MQ
2048 4016 7421 MQL /RESTORE IT
2049 4017 1777 TAD TSTLOP
2050 4020 3223 DCA RESADD
2051 4021 4450 RSTRAC
2052 4022 5623 JMP I RESADD
2053 4023 0000 RESADD, 0
2054 /
2055 4024 4452 FLAGS, KBCHK /CHECK FOR KEYBOARD INTERVENTION
2056 4025 4423 LQSK /TEST LQ FLAG
2057 4026 5231 JMP .+3 /NO LQ FLAG
2058 4027 7340 CLA CLL CMA /AC=7777
2059 4030 3116 DCA INTFLG /SET SOFTWARE INTERRUPT FLAG
2060 4031 4450 RSTRAC /RESTORE AC ANC LINK
2061 4032 5400 JMP I LOCO /RETURN TO PROGRAM
2062 /
2063 /
2064 /
2065 /
2066 /
2067 /ROUTINE TO WAIT FOR INDIVIDUAL READY BITS, TEST MASK STORED
2068 /IN LOCATION WSTAT
2069 /RETURN CALL +1 IF STATUS ERROR, CALL +2 IF APPROPRIATE READY
2070 /BIT IS SET
2071 /
2072 4033 0000 AWAIT, 0
2073 4034 4447 SAVEAC /SAVE AC AND LINK
2074 4035 3261 DCA COUNT /CLEAR ISZ COUNTER
2075 4036 1376 TAD (-30 /24 LOOPS
2076 4037 3262 DCA COUNT1 /SET UP LOOP COUNTER
2077 4040 3116 DCA INTFLG /CLEAR INTERRUPT FLAG
2078 4041 1133 TAD WSTAT /GET WANTED STATUS
2079 4042 1375 TAD (4002 /ADD PRINTER READY AND RIBBON UP
2080 4043 3117 DCA XSTAT /SAVE IT
2081 4044 4446 WLOOP, RDYCHK /GET READY BIT STATUS
2082 4045 0133 AND WSTAT /TEST WANTED STATUS BITS
2083 4046 7440 SZA /SKIP IF NOT READY
2084 4047 5255 JMP WATOUT /EXIT FROM WAIT LOOP
2085 4050 2261 ISZ COUNT /INCREMENT ISZ COUNTER
2086 4051 5244 JMP WLOOP /CONTINUE WAITING
2087 4052 2262 ISZ COUNT1 /INCREMENT LOOP COUNTER
2088 4053 5244 JMP WLOOP /CONTINUE WAITING
2089 4054 5633 JMP I AWAIT /STATUS NOT READY, ERROR RETURN
2090 4055 7300 WATOUT, CLA CLL
2091 4056 4450 RSTRAC /RESTORE AC AND LINK
2092 4057 2233 ISZ AWAIT /BUMP RETURN POINTER

2093 4060 5633 JMP I AWAIT
2094 4061 0000 COUNT, 0
2095 4062 0000 COUNT1, 0
2096 /
2097 /
2098 /
2099 /ROUTINE TO WAIT FOR ALL READY BITS SET, TEST MASK STORED
2100 /IN LOCATION WSTAT
    
```

```

2101 /RETURN CALL +1 IF STATUS ERROR, CALL +2 IF APPROPRIATE READY
2102 /BITS ARE SET
2103 /
2104 4063 0000 WAITS, 0
2105 4064 4447 SAVEAC /SAVE AC AND LINK
2106 4065 3310 DCA COUNT2 /CLEAR ISZ COUNTER
2107 4066 1376 TAD (-30 /24 LOOPS
2108 4067 3311 DCA COUNT3 /SET UP LOOP COUNTER
2109 4070 3116 DCA INTFLG /CLEAR INTERRUPT FLAG
2110 4071 1374 TAD (7400 /GET WANTED STATUS
2111 4072 3117 DCA XSTAT /SAVE IT FOR EXPECTED STATUS
2112 4073 4446 SLOOP, RDYCHK /GET READY BIT STATUS
2113 4074 1373 TAD (400 /ADD TWO'S COMPLEMENT OF ALL READY
2114 4075 7450 SNA /SKIP IF NOT ALL READY
2115 4076 5304 JMP EXIT /EXIT FROM WAIT LOOP
2116 4077 2310 ISZ COUNT2 /INCREMENT ISZ COUNTER
2117 4100 5273 JMP SLOOP /CONTINUE WAITING
    
```

```

/LQP-8 PRINTER DIAGNOSTIC      PAL10  V142A  1-AUG-77      14140  PAGE 4
2118  4101  2311      ISZ COUNT3  /INCREMENT LOOP COUNTER
2119  4102  5273      JMP SLOOP  /CONTINUE WAITING
2120  4103  5663      JMP I *WAITS /STATUS NOT READY, ERROR RETURN
2121  4104  7300      EXIT,    CLA CLL

```

SEQ 0061

```

/LQP-8 PRINTER DIAGNOSTIC      PAL10  V142A  1-AUG-77      14140  PAGE 5
2122  4105  4450      RSTRAC   /RESTORE AC AND LINK
2123  4106  2263      ISZ *WAITS /BUMP RETURN POINTER
2124  4107  5663      JMP I *WAITS /RETURN, ALL READY BITS SET
2125  4110  0000      COUNT2, 0
2126  4111  0000      COUNT3, 0
2127  /
2128  /
2129  /
2130  /
2131  /GENERATE RANDOM NUMBER, EXIT WITH NUMBER IN AC
2132  /
2133  4112  0000      XRAND, 0 /CALL BY "RANDOM"
2134  4113  7301      CLA CLL IAC
2135  4114  1344      TAD RAN1
2136  4115  1345      TAD RAN2
2137  4116  7106      CLL RTL
2138  4117  3344      DCA RAN1
2139  4120  1345      TAD RAN2
2140  4121  7012      RTR
2141  4122  1344      TAD RAN1
2142  4123  3345      DCA RAN2
2143  4124  1345      TAD RAN2
2144  4125  5712      JMP I XRAND
2145  /
2146  /
2147  /SAVE RANDOM GENERATOR PRIMES
2148  /
2149  4126  0000      XSAVGN, 0 /CALL BY "SAVGEN"
2150  4127  7200      CLA
2151  4130  1344      TAD RAN1
2152  4131  3346      DCA SAV1
2153  4132  1345      TAD RAN2
2154  4133  3347      DCA SAV2
2155  4134  5726      JMP I XSAVGN
2156  /
2157  /RESTORE RANDOM GENERATOR PRIMES
2158  /
2159  4135  0000      XRESGN, 0 /CALL BY "RESGEN"
2160  4136  7200      CLA
2161  4137  1346      TAD SAV1
2162  4140  3344      DCA RAN1
2163  4141  1347      TAD SAV2
2164  4142  3345      DCA RAN2
2165  4143  5735      JMP I XRESGN
2166  /
2167  4144  1234      RAN1, 1234
2168  4145  5670      RAN2, 5670
2169  4146  0000      SAV1, 0
2170  4147  0000      SAV2, 0
2171  /
2172  /
2173  4173  0400
2174  4174  7400
2175  4175  4002
2176  4176  7750

```

SEQ 0062

```

2177 4177 3350
          4200
2178
2179 /ROUTINE TO SAVE AC AND LINK AND EXIT WITH BOTH AC AND LINK CLEARED
2180
2181 4200 0000 /ACSAVE, 0
2182 4201 3100 DCA SAVAC /SAVE AC
2183 4202 7010 RAR /GET LINK
2184 4203 3101 DCA SAVLNK /SAVE LINK
2185 4204 5600 JMP I ACSAVE
2186
2187 /ROUTINE TO RESTORE AC AND LINK
2188
2189 4205 0000 /ACRSTR, 0
2190 4206 7300 CLA CLL
2191 4207 1101 TAD SAVLNK /GET LINK
2192 4210 7004 HAL
2193 4211 1100 TAD SAVAC /GET AC
2194 4212 5605 JMP I ACRSTR
2195
2196 /STATUS ERROR HANDLER
2197
2198 4213 0000 ERSTAT, 0
2199 4214 4466 CRACTV /CHECK FOR ACTIVE CONSOLE
2200 4215 5243 JMP STPRNT /CONSOLE IS ACTIVE, GO PRINT INFO
2201 4216 4465 LAS /CONSOLE NOT ACTIVE, GET SWITCHES
2202 4217 7710 SPA CLA /SKIP IF BIT 0=0, ERROR HALT
2203 4220 5232 JMP ERS1 /GO CHECK FOR ERROR LOOP
2204 4221 7340 CLA CLL CMA
2205 4222 1213 TAD ERSTAT /GET FAILING PC
2206 4223 7402 HLT /FAILING PC IN AC
2207 4224 7300 CLA CLL
2208 4225 1117 TAD XSTAT /GET EXPECTED STATUS
2209 4226 7402 HLT /EXPECTED STATUS IN AC
2210 4227 7300 CLA CLL
2211 4230 1114 TAD SVSTAT /GET ACTUAL STATUS
2212 4231 7402 HLT /ACTUAL STATUS IN AC
2213 4232 7300 ERS1, CLA CLL
2214 4233 1777 TAD TSTLOP /GET TEST LOOP ADDRESS
2215 4234 3242 DCA TLOOP3 /SAVE IT
2216 4235 4465 LAS /GET SWITCHES
2217 4236 7004 RAL /POTATE FOR BIT 1 CHECK
2218 4237 7710 SPA CLA /SKIP IF BIT 1=0
2219 4240 5642 JMP I TLOOP3 /LOOP ON TEST
2220 4241 5613 JMP I ERSTAT /CONTINUE TEST
2221 4242 0000 TLOOP3, 0
2222
2223 4243 4465 STPRNT, LAS /GET SWITCHES
2224 4244 0376 AND (200 /TEST BIT 4, INHIBIT ERROR TYPEOUT
2225 4245 7640 SZA CLA /SKIP IF BIT 4=0
2226 4246 5266 JMP ERS2 /GO CHECK FOR ERROR HALT
2227 4247 4467 MESSAGE /PRINT A MESSAGE
2228 4250 6005 LSTERR /STATUS ERROR MESSAGE
2229 4251 4467 MESSAGE
2230 4252 6024 PCEQL

```

```

2231 4253 7340 CLA CLL CMA
2232 4254 1213 TAD ERSTAT
2233 4255 4470 REGPRT /PRINT FAILING PC
2234 4256 4467 MESSAGE
2235 4257 6054 EXPECT /EXPECTED
2236 4260 1117 TAD XSTAT /GET EXPECTED STATUS
2237 4261 4470 REGPRT /PRINT IT
2238 4262 4467 MESSAGE
2239 4263 6062 ACTUAL /ACTUAL STATUS
2240 4264 1114 TAD SVSTAT /GET ACTUAL STATUS
2241 4265 4470 REGPRT /PRINT IT
2242
2243 4266 4465 ERS2, LAS /GET SWITCHES
2244 4267 7710 SPA CLA /SKIP IF BIT 0=0, ERROR HALT
2245 4270 5232 JMP ERS1 /GO CHECK FOR ERROR LOOP
2246 4271 4467 MESSAGE /PRINT "WAITING" MESSAGE
2247 4272 6000 WAITNG /MESSAGE POINTER
2248 4273 6030 ERS3, KCF /CLEAR KYBD FLAG
2249 4274 6031 KSF /TEST KYBD FLAG
2250 4275 5274 JMP =-1 /WAIT FOR OPERATOR INTERVENTION
2251 4276 7300 CLA CLL
2252 4277 6034 KRS /READ KYBD, DO NOT CLEAR FLAG
2253 4300 0375 AND (177 /MASK TO 7 BITS
2254 4301 1374 TAD (-7 /TEST FOR CTRL-G
2255 4302 7640 SZA CLA /SKIP IF CTRL-G
2256 4303 5273 JMP ERS3 /CONTINUE WAITING FOR CTRL-G
2257 4304 4452 KRCHK /HANDLE CTRL-G
2258 4305 5232 JMP ERS1 /GO CHECK FOR ERROR LOOP
2259
2260
2261
2262 /ROUTINE TO CHECK FOR ACTIVE CONSOLE
2263
2264 4306 0000 ACTVC8, 0
2265 4307 7300 CLA CLL
2266 4310 1022 TAD OP2SEL /GET HARDWARE CONFIGURATION WORD TWO
2267 4311 0373 AND (400 /MASK BIT 3
2268 4312 7640 SZA CLA /SKIP IF NON ACTIVE CONSOLE
2269 4313 5706 JMP I ACTVC8 /RETURN CALL +1 FOR ACTIVE CONSOLE
2270 4314 2306 ISZ ACTVC8 /BUMP RETURN POINTER
2271 4315 5706 JMP I ACTVC8 /RETURN CALL +2 FOR NON ACTIVE CONSOLE
2272
2273 /MOVE PAPER ROUTINE
2274
2275 /*****
2276
2277 / CALLED BY: MOVPAR
2278 / PRADV /PAPER DIRECTION
2279 / XXXX /PAPER MOVEMENT MAGNITUDE
2280
2281 /*****
2282
2283 4316 0000 PARMOV, 0
2284 4317 7300 CLA CLL
2285 4320 1374 TAD (0400 /PAPER READY BIT 3, PRINTER READY BIT 0

```

```

2286 4321 3133 DCA WSTAT /SAVE WANTED STATUS, PAPER READY, BIT 3
2287 4322 1716 TAD I PARMOV /GET PAPER DIRECTION
2288 4323 2316 ISZ PARMOV /INCREMENT POINTER TO PAPER MOVE MAGNITUDE
2289 4324 1716 TAD I PARMOV /ADD PAPER MOVEMENT MAGNITUDE
2290 4325 4441 WAIT /WAIT FOR PAPER READY
2291 4326 4451 STATER /PAPER READY DID NOT RETURN
2292 4327 4435 LOADDB /LOAD DATA BUFFER
2293 4330 4425 LQMP /MOVE PAPER
2294 4331 4437 DATERR /DATA ERROR AFTER IOT 2
2295 4332 2316 ISZ PARMOV /BUMP RETURN POINTER
2296 /
2297 4333 5716 JMP I PARMOV
2298 /
2299 /
2300 4373 0400
2301 4374 7771
2302 4375 0177
2303 4376 0200
2304 4377 3350
2305 4400 PAGE
2306 /MOVE CARRIAGE ROUTINE
2307 /
2308 /*****
2309 /
2310 / CALLED BY: MOVCR
2311 / CARLFT /CARRIAGE DIRECTION
2312 / XXXX /CARRIAGE MOVEMENT MAGNITUDE
2313 /
2314 /*****
2315 /
2316 4400 0000 CARMOV, 0
2317 4401 7300 CLA CLL
2318 4402 1377 TAD (1000 /CARRIAGE READY BIT 2, PRINTER READY BIT 0
2319 4403 3133 DCA WSTAT /SAVE WANTED STATUS
2320 4404 1600 TAD I CARMOV /GET CARRIAGE DIRECTION
2321 4405 3126 DCA CARDIR /SAVE CARRIAGE DIRECTION
2322 4406 1126 TAD CARDIR /GET CARRIAGE DIRECTION
2323 4407 7640 SZA CLA /SKIP IF MOVE RIGHT
2324 4410 5221 JMP MOVLEFT /GO TO CARRIAGE LEFT ROUTINE
2325 4411 2200 ISZ CARMOV /INCREMENT POINTER TO MOVE MAGNITUDE
2326 4412 1600 TAD I CARMOV /GET CARRIAGE MOVE MAGNITUDE
2327 4413 3141 DCA MOV MAG /SAVE IT
2328 4414 1141 TAD MOV MAG /GET IT AGAIN
2329 4415 1140 TAD SAVPOS /ADD CURRENT POSITION
2330 4416 3140 DCA SAVPOS /UPDATE CARRIAGE POSITION POINTER
2331 4417 1141 TAD MOV MAG /GET CARRIAGE MOVE MAGNITUDE
2332 4420 5732 JMP MVWAIT /GO WAIT FOR CARRIAGE READY
2333 4421 2200 MOVLEFT, ISZ CARMOV /INCREMENT POINTER TO MOVE MAGNITUDE
2334 4422 1600 TAD I CARMOV /GET CARRIAGE MOVE MAGNITUDE
2335 4423 3141 DCA MOV MAG /SAVE IT
2336 4424 1141 TAD MOV MAG /GET IT AGAIN
2337 4425 7041 CMA IAC /FORM NEGATIVE
2338 4426 1140 TAD SAVPOS /SUBTRACT FROM CURRENT POSITION
2339 4427 3140 DCA SAVPOS /UPDATE CARRIAGE POSITION POINTER
2340 4430 1126 TAD CARDIR /GET CARRIAGE DIRECTION

```

```

2340 4431 1141 TAD MOV MAG /ADD CARRIAGE MOVE MAGNITUDE
2341 4432 4441 MVWAIT, WAIT /WAIT FOR CARRIAGE READY
2342 4433 4451 STATER /CARRIAGE READY DID NOT RETURN
2343 4434 4435 LOADDB /LOAD DATA BUFFER FOR MOVE CARRIAGE
2344 4435 4426 LQMC /IOT 3 MOVE CARRIAGE
2345 4436 4437 DATERR /DATA ERROR AFTER IOT 3
2346 4437 2200 ISZ CARMOV /BUMP RETURN POINTER
2347 4440 5600 JMP I CARMOV /RETURN
2348 /
2349 /
2350 /ROUTINE TO MOVE CARRIAGE TO ANY DESIRED PRINT POSITION
2351 /ENTER THIS SUBROUTINE WITH PRINT POSITION COORDINATE
2352 /IN AC, CALLED BY: MOVCR
2353 /
2354 /
2355 4441 0000 CRMOV, 0
2356 4442 3121 DCA SAVMOV /SAVE DESIRED POSITION
2357 4443 1140 TAD SAVPOS /GET CURRENT POSITION
2358 4444 7041 CIA /NEGATE IT
2359 4445 1121 TAD SAVMOV /ADD DESIRED POSITION
2360 4446 7700 SMA CLA /SKIP IF DESIRED POSITION IS LEFT OF
2361 /CURRENT POSITION
2362 4447 5257 JMP RITCR /DESIRED POSITION IS RIGHT OF CURRENT
2363 /POSITION
2364 4450 1121 LFTCR, TAD SAVMOV /GET DESIRED POSITION
2365 4451 7041 CIA /NEGATE IT
2366 4452 1140 TAD SAVPOS /ADD CURRENT POSITION
2367 4453 3267 DCA MAG /SAVE MOVE MAGNITUDE
2368 4454 1376 TAD (CARLFT /SET CARRIAGE LEFT BIT 0
2369 4455 3266 DCA DIRECT /SAVE IT
2370 4456 5265 JMP MOVE /GO MOVE CARRIAGE
2371 4457 1140 RITCR, TAD SAVPOS /GET CURRENT POSITION
2372 4460 7041 CIA /NEGATE IT
2373 4461 1121 TAD SAVMOV /ADD CARRIAGE MOVE MAGNITUDE
2374 4462 0375 AND (3777 /MASK MAGNITUDE
2375 4463 3267 DCA MAG /SAVE IT
2376 4464 3266 DCA DIRECT /CLEAR BIT 0 FOR MOVE CARRIAGE RIGHT
2377 4465 4455 MOVE, MOVCR /MOVE CARRIAGE
2378 4466 0000 DIRECT, 0 /SAVE AREA FOR CARRIAGE DIRECTION
2379 4467 0000 MAG, 0 /SAVE AREA FOR CARRIAGE MOVEMENT MAGNITUDE
2380 4470 5641 JMP I CRMOV /RETURN
2381 /
2382 /ROUTINE TO EXECUTE A CARRIAGE RETURN, LINE FEED FROM ANY CARRIAGE POSITION
2383 /
2384 4471 0000 CRLFRT, 0
2385 4472 7300 CLA CLL
2386 4473 1140 TAD SAVPOS /GET CURRENT POSITION
2387 4474 3277 DCA CARMAG /SAVE IT FOR CARRIAGE RETURN
2388 4475 4455 MOVCR /MOVE CARRIAGE
2389 4476 4000 CARLFT /DIRECTION = LEFT BIT 0 = 1
2390 4477 0000 CARMAG, 0 /CARRIAGE MOVEMENT MAGNITUDE
2391 4400 7300 CLA CLL
2392 4401 1130 TAD LF /GET VALUE FOR PAPER FEED
2393 4402 3305 DCA PARMAG /SAVE IT FOR LINE FEED
2394 4403 4457 MOVPAR /MOVE PAPER

```

```

2395 4504 0000 PRADV /DIRECTION = ADVANCE BIT 0 = 0
2396 4505 0000 PARMAG, 0 /PAPER MOVEMENT MAGNITUDE
2397 4506 5671 JMP I CRLFRT
2398 /
2399 /ROUTINE TO PRINT A CHARACTER ON LQ PRINTER
2400 /
2401 4507 0000 XPRINT, 0
2402 4510 7300 CLA CLL
2403 4511 1374 TAD (2000 /CHARACTER READY BIT 1
2404 4512 3133 DCA WSTAT /SAVE FOR WANTED STATUS
2405 4513 4453 ROTCHR /ROTATE CHARACTER STORED IN CHAR
2406 4514 1125 TAD CHARRT /GET ROTATED CHARACTER
2407 4515 1373 TAD (7401 /SET UNUSED BITS FOR CROSS TALK TEST
2408 4516 4441 WAIT /WAIT FOR CHARACTER READY
2409 4517 4451 STATER /CHARACTER READY DID NOT RETURN
2410 4520 4435 LOADDB /LOAD DATA BUFFER WITH CHARACTER TO PRINT
2411 /AND PRINT IT
2412 4521 4427 LQPC /IOT 4 PRINT CHARACTER
2413 4522 4437 DATERR /DATA ERROR AFTER IOT 4
2414 4523 2136 ISZ CHRCNT /INCREMENT CHARACTER COUNT
2415 4524 7000 NOP /IN CASE CHRCNT OVERFLOWS
2416 4525 5707 JMP I XPRINT /RETURN
2417 /
2418 /ROUTINE TO ROTATE ASCII CHARACTER CODE STORED IN LOCATION CHAR
2419 /AND STORE IT IN LOCATION CHARRT
2420 /
2421 4526 0000 CHRROT, 0
2422 4527 7300 CLA CLL
2423 4530 1124 TAD CHAR /GET NORMAL ASCII CHARACTER
2424 4531 7004 RAL /ROTATE LEFT ONE POSITION
2425 4532 3125 DCA CHARRT /SAVE SHIFTED ASCII CHARACTER
2426 4533 5726 JMP I CHRROT /RETURN
2427 /
2428 /
2429 /ROUTINE TO CHECK FOR END OF PRINT LINE
2430 /RETURN IS CALL PLUS ONE IF NOT END OF LINE
2431 /RETURN IS CALL PLUS TWO IF END OF LINE
2432 /
2433 4534 0000 ENDLIN, 0
2434 4535 7300 CLA CLL
2435 4536 1136 TAD CHRCNT /GET CHARACTER COUNT
2436 4537 7041 CMA IAC /FORM TWO'S COMPLEMENT
2437 4540 1131 TAD LINE /ADD NUMBER OF CHARACTERS PER LINE
2438 4541 7640 SZA CLA /SKIP IF END OF LINE
2439 4542 5734 JMP I ENDLIN /RETURN IF NOT END OF LINE
2440 4543 3136 DCA CHRCNT /CLEAR CHARACTER COUNTER
2441 4544 2334 ISZ ENDLIN /BUMP RETURN POINTER

2442 4545 5734 JMP I ENDLIN /RETURN CALL PLUS TWO
2443 /
2444 /ROUTINE TO CHECK FOR LAST LINE OF A PRINT PATTERN TEST
2445 /RETURN CALL +1 IF NOT LAST LINE
2446 /RETURN CALL +2 IF LAST LINE HAS BEEN PRINTED
2447 /
2448 4546 0000 LINLST, 0
2449 4547 1115 TAD LINCNT /GET LINE COUNT

```

```

2450 4550 7041 CIA /FORM TWO'S COMPLEMENT
2451 4551 1132 TAD LINES /ADD NUMBER OF LINES TO PRINT
2452 4552 7640 SZA CLA /SKIP IF ENOUGH LINES PRINTED
2453 4553 5746 JMP I LINLST /RETURN IF NOT ENOUGH LINES PRINTED
2454 4554 3132 DCA LINES /CLEAR AMOUNT OF LINES
2455 4555 2346 ISZ LINLST /BUMP RETURN POINTER
2456 4556 5746 JMP I LINLST /RETURN CALL PLUS TWO
2457 /
2458 /ROUTINE TO MOVE CARRIAGE ONE SPACE
2459 /CALL BY RSPACE
2460 /
2461 4557 0000 SPACER, 0
2462 4560 7300 CLA CLL
2463 4561 1127 TAD SPACE /GET SIZE OF CHARACTER SPACING
2464 4562 3365 DCA SPSIZR /SAVE IT FOR PRINT POSITIONING
2465 4563 4455 MOVCAR /MOVE CARRIAGE
2466 4564 0000 CARRGT /DIRECTION = RIGHT
2467 4565 0000 SPSIZR, 0 /SPACE SIZE
2468 4566 5757 JMP I SPACER /RETURN
2469 4573 7401
2470 4574 2000
2471 4575 3777
2472 4576 4000
2473 4577 1000
2474 4600 PAGE
2475 /
2476 /
2477 /ROUTINE TO MOVE CARRIAGE ONE SPACE TO THE LEFT
2478 /CALLED BY LSPACE
2479 /
2480 4600 0000 SPACEL, 0
2481 4601 7300 CLA CLL
2482 4602 1127 TAD SPACE /GET SIZE OF CHARACTER SPACING
2483 4603 3206 DCA SPSIZL /SAVE IT FOR PRINT POSITIONING
2484 4604 4455 MOVCAR /MOVE CARRIAGE
2485 4605 4000 CARLFT /DIRECTION = LEFT
2486 4606 0000 SPSIZL, 0 /SPACE SIZE
2487 4607 5600 JMP I SPACEL
2488 /
2489 /
2490 /ROUTINE TO CHECK FOR KEYBOARD INTERVENTION
2491 /AND TEST FOR CTRL CHARACTER
2492 /CALLED BY KBCHK
2493 /
2494 4610 0000 CHKKB, 0
2495 4611 3226 DCA ACSV /SAVE AC
2496 4612 7010 RAR /GET LINK
2497 4613 3227 DCA LINKSV /SAVE IT
2498 4614 4466 CBACTV /SKIP IF NON ACTIVE CONSOLE
2499 4615 6031 KSF /TEST KEYBOARD FLAG
2500 4616 5221 JMP KBOUT /GO TO PROGRAM RETURN ROUTINE
2501 4617 4445 CTLCHK /CHECK FOR CONTROL CHARACTER
2502 4620 4444 ECHO /NOT CONTROL CHAR, SO ECHO IT
2503 4621 7300 KBOUT, CLA CLL

```

```

2504 4622 1227 TAD LINKSV /GET LINK
2505 4623 7004 RAL /RESTORE IT
2506 4624 1226 TAD ACSV /GET AC
2507 4625 5610 JMP I CHKKB /RETURN
2508 4626 0000 ACSV, 0 /AC SAVE AREA
2509 4627 0000 LINKSV, 0 /LINK SAVE AREA
2510 //
2511 //
2512 //ROUTINE TO CHECK FOR EACH OF SIX VALID CONTROL CHARACTERS
2513 //FROM OPERATOR'S CONSOLE, THE SIX (6) VALID CONTROL CHARACTERS ARE:
2514 //
2515 // CTRL/C RETURN TO OPERATING SYSTEM MONITOR
2516 //
2517 // CTRL/G ALLOWS CHANGING OF SOFTWARE SWITCH REGISTER
2518 // RESTARTS DIAGNOSTIC IF FOLLOWED BY LINE FEED
2519 // CONTINUES DIAGNOSTIC FROM INTERRUPTION POINT IF
2520 // FOLLOWED BY CARRIAGE RETURN
2521 //
2522 // CTRL/Q CONTINUES DIAGNOSTIC
2523 //
2524 // CTRL/S STOPS DIAGNOSTIC AND WAITS FOR CTRL CHARACTER
2525 //
2526 // CTRL/K RESTORES PRINTER AND CONTINUES CURRENT TEST
2527 //
2528 // CTRL/T FOLLOWED BY CTRL-G ALLOWS DESIRED TEST NUMBER
2529 // TO BE TYPED INTO PSEUDO SWITCH REGISTER BITS
2530 // 6-11
2531 //
2532 4630 0000 CHKCTL, 0
2533 4631 4466 CSACTV /CHECK FOR CONSOLE ACTIVE
2534 4632 7410 SKP /SKIP IF CONSOLE IS ACTIVE
2535 4633 5630 JMP I CHKCTL /CONSOLE NOT ACTIVE, RETURN
2536 4634 6031 KSF /TEST TTY FLAG
2537 4635 5630 JMP I CHKCTL /RETURN IF NOT SET
2538 4636 6036 KRB /READ KEYBOARD BUFFER
2539 4637 0377 AND (177 /MASK TO 7 BIT ASCII
2540 4640 7041 CIA /NEGATE IT
2541 4641 3137 DCA KBSAV /SAVE KEYBOARD CHARACTER
2542 4642 1137 TAD KBSAV /GET CHARACTER
2543 4643 1104 TAD K003 /IS IT CONTROL-C?
2544 4644 7650 SNA CLA /SKIP IF NO
2545 4645 5776 JMP MDNITR /RETURN TO SYSTEM MONITOR
2546 4646 1137 TAD KBSAV /GET CHARACTER
2547 4647 1105 TAD K007 /IS IT CTRL-G?
2548 4650 7650 SNA CLA /SKIP IF NO
2549 4651 5775 JMP CBSWCK /GO TO SWITCH REGISTER QUESTION
2550 4652 1137 TAD KBSAV /GET CHARACTER

2551 4653 1106 TAD K021 /IS IT CTRL-Q?
2552 4654 7650 SNA CLA /SKIP IF NO
2553 4655 5774 JMP QCONT /CONTINUE DIAGNOSTIC
2554 4656 1137 TAD KBSAV /GET CHARACTER
2555 4657 1107 TAD K023 /IS IT CTRL-S?
2556 4660 7650 SNA CLA /SKIP IF NO
2557 4661 5773 JMP SSTOP /STOP TEST
2558 4662 1137 TAD KBSAV /GET CHARACTER
    
```

```

2559 4663 1372 TAD (24 /IS IT CTRL-T?
2560 4664 7650 SNA CLA /SKIP IF NOT
2561 4665 5771 JMP TECHO /GO TO TEST SELECT ROUTINE
2562 4666 1137 TAD KBSAV /GET CHARACTER
2563 4667 1370 TAD (13 /IS IT CTRL-K
2564 4670 7650 SNA CLA /SKIP IF NO
2565 4671 4767 JMS RESTOR /RESTORE PRINTER
2566 4672 5630 JMP I CHKCTL
2567 //
2568 //
2569 //ROUTINE TO ECHO A NON CTRL CHARACTER ON CONSOLE
2570 //
2571 4673 0000 XECHO, 0
2572 4674 6036 KRB
2573 4675 0377 AND (177 /MASK TO 7 BITS
2574 4676 6046 TIS /PRINT ON TTY
2575 4677 6041 TSF /TEST DONE FLAG
2576 4700 5277 JMP ,=1 /WAIT FOR DONE FLAG
2577 4701 5673 JMP I XECHO /RETURN
2578 //
2579 //
2580 //MESSAGE PRINT ROUTINE
2581 //A "*" CHARACTER CAUSES A CARRIAGE RETURN LINE FEED
2582 //
2583 4702 0000 MESPRT, 0
2584 4703 7240 CLA CMA
2585 4704 1702 TAD I MESPRT /GET MESSAGE ADDRESS
2586 4705 3010 DCA INDX10 /SAVE IT IN MESSAGE POINTER
2587 4706 2302 ISZ MESPRT /INCREMENT RETURN POINTER
2588 4707 1410 MESLP, TAD I INDX10 /GET TWO 6 BIT PACKED ASCII CHARACTERS
2589 4710 3321 DCA MSRGHT /SAVE THEM
2590 4711 1321 TAD MSRGHT /GET THEM
2591 4712 7012 KTR
2592 4713 7012 RTR
2593 4714 7012 RTR
2594 4715 4322 JMS TYPECH /PRINT FIRST CHARACTER
2595 4716 1321 TAD MSRGHT /GET SECOND CHARACTER
2596 4717 4322 JMS TYPECH /PRINT SECOND CHARACTER
2597 4720 5307 JMP MESLP /CONTINUE PRINTING
2598 4721 0000 MSRGHT, 0
2599 4722 0000 TYPECH, 0
2600 4723 0366 AND (77 /MASK 6 BIT ASCII
2601 4724 7450 SNA /SKIP IF NOT END OF MESSAGE
2602 4725 5702 JMP I MESPRT /END OF MESSAGE RETURN
2603 4726 1365 TAD (-43
2604 4727 7450 SNA /IS IT #
2605 4730 5337 JMP ,+7 /YES, DO A CR AND LF
2606 4731 1364 TAD (3
2607 4732 7510 SPA
2608 4733 1363 TAD (100
2609 4734 1362 TAD (240
2610 4735 4761 JMS PRNT1
2611 4736 5722 JMP I TYPECH
2612 4737 4760 JMS CRLF
2613 4740 5722 JMP I TYPECH
    
```

```

2614      /
2615      /
2616      4760 5007
2617      4761 5000
2618      4762 0240
2619      4763 0100
2620      4764 0003
2621      4765 7735
2622      4766 0077
2623      4767 3351
2624      4770 0013
2625      4771 5614
2626      4772 0024
2627      4773 5317
2628      4774 5340
2629      4775 5400
2630      4776 5601
2631      4777 0177
                5000
                PAGE
                /
                /
                /ROUTINE TO PRINT ONE CHARACTER
                /
2636      5000 0000      PRNT1, 0
2637      5001 6046      TLS
2638      5002 6041      TSF
2639      5003 5202      JMP ,=-1
2640      5004 6042      TCF
2641      5005 7300      CLA CLL
2642      5006 5600      JMP I PRNT1
                /
                /ROUTINE TO EXECUTE A CARRIAGE RETURN AND LINE FEED
                /
2646      5007 0000      CRLF, 0
2647      5010 7300      CLA CLL
2648      5011 1377      TAD (15
2649      5012 4200      JMS PRNT1
2650      5013 1376      TAD (12
2651      5014 4200      JMS PRNT1
2652      5015 5607      JMP I CRLF
                /
                /
                /
                /ROUTINE TO PRINT CONTENTS OF A REGISTER
                /
2659      5016 0000      PRTRREG, 0
                /
2660      5017 3244      DCA REGCON      /SAVE REGISTER CONTENTS
2661      5020 1375      TAD (=4        /SET UP PRINT COUNTER
2662      5021 3243      DCA PRTCNT
2663      5022 7100      CLL
2664      5023 1244      TAD REGCON      /GET REGISTER CONTENTS
2665      5024 7004      RAL
2666      5025 7004      REGLUP, RAL
2667      5026 7006      RTL

```

```

2668      5027 4447      SAVEAC
2669      5030 4450      RSTRAC
2670      5031 0374      AND (7
2671      5032 1373      TAD (60
2672      5033 4200      JMS PRNT1      /PRINT A DIGIT
2673      5034 4450      RSTRAC        /RESTORE AC AND LINK
2674      5035 2243      ISZ PRTCNT    /INCREMENT COUNTER
2675      5036 5225      JMP REGLOP     /CONTINUE PRINTING REGISTER CONTENTS
2676      5037 7300      CLA CLL
2677      5040 1372      TAD (240      /PRINT A SPACE
2678      5041 4200      JMS PRNT1
2679      5042 5616      JMP I PRTRREG /RETURN
                /
2681      5043 0000      PRTCNT, 0
2682      5044 0000      REGCON, 0
                /
                /
2684      5172 0240
2685      5173 0060
2686      5174 0007
2687      5175 7774
2688      5176 0012
2689      5177 0015
                5200
                PAGE
                /
                /ROUTINE TO CHECK FOR CARRIAGE RETURN OR LINE FEED TYPED
                /FOLLOWING A CTRL G
                /
2694      5200 0000      KBCRLF, 0
2695      5201 6036      KPB          /READ KYBD BUFFER
2696      5202 0377      AND (177      /MASK TO SEVEN BITS
2697      5203 3236      DCA KBSV     /SAVE CHAR
2698      5204 1236      TAD KBSV     /GET CHAR
2699      5205 1376      TAD (=15     /IS IT CARRIAGE RETURN?
2700      5206 7640      SZA CLA     /SKIP IF CARRIAGE RETURN
2701      5207 7410      SKP
2702      5210 5217      JMP CROUT    /GO TO CONTINUE WITH PROGRAM
2703      5211 1236      TAD KBSV     /GET KYBD CHAR
2704      5212 1375      TAD (=12     /IS IT LINE FEED?
2705      5213 7640      SZA CLA     /SKIP IF LINE FEED
2706      5214 7410      SKP
2707      5215 5225      JMP LFOUT    /GO TO BEGIN OF PROGRAM
2708      5216 5600      JMP I KBCRLF /NOT CR OR LF, RETURN
                /
2710      5217 1774*     CROUT, TAD PSWCH
2711      5220 7650      SNA CLA
2712      5221 5224      JMP ,+3
2713      5222 1773*     TAD PSRSV1    /GET NEW SWITCH REGISTER
2714      5223 3020      DCA PSWR     /SAVE IN PSEUDO SWITCH REGISTER
2715      5224 5772*     JMP KBOUT     /RETURN AND CONTINUE
                /
2717      5225 1774*     LFOUT, TAD PSWCH
2718      5226 7650      SNA CLA
2719      5227 5232      JMP ,+3
2720      5230 1773*     TAD PSRSV1    /GET NEW SWITCH REGISTER
2721      5231 3020      DCA PSWR     /SAVE IN PSEUDO SWITCH REGISTER

```



```

2722 5232 1771' TAD RSTART
2723 5233 3235 DCA GOBAK
2724 5234 5635 JMP I GOBAK /RETURN TO BEGINNING OF TESTS
2725 5235 0000 GOBAK, 0
2726 /
2727 5236 0000 KBSV, 0 /KYBD CHAR SAVE AREA
2728 /
2729 /ROUTINE TO HANDLE SWITCH REGISTER ERROR
2730 /
2731 5237 7300 SWREPR, CLA CLL
2732 5240 1370 TAD (277 /CODE FOR QUESTION MARK
2733 5241 4767' JMS PRNT1 /PRINT QUESTION MARK
2734 5242 4766' JMS CRLF /CR, LF
2735 5243 5765' JMP CBSWCK+6 /START SWITCH REGISTER ROUTINE AGAIN
2736 /
2737 /
2738 /
2739 /ROUTINE TO PERMIT VARIABLE DELAY AS DETERMINED BY OCTAL
2740 /NUMBER FOLLOWING DELAY CALL
2741 /
2742 5244 0000 ADELAY, 0
2743 5245 7300 CLA CLL /1,2 USEC.
2744 5246 1644 TAD I ADELAY /3,8 USEC.
2745 5247 3271 DCA DELAYN /2,6 USEC.
2746 5250 2244 ISZ ADELAY /2,6 USEC.
2747 5251 1272 DLOP, TAD M23 /2,6 USEC.
2748 5252 3273 DCA DCOUNT /2,6 USEC.
2749 5253 2273 ISZ DCOUNT /2,6 USEC. X 19 = 29,4 USEC,
2750 5254 5253 JMP ,=1 /2,6 USEC. X 28 = 21,6 USEC,
2751 5255 7000 NOP /1,2 USEC.
2752 5256 7000 NOP /1,2 USEC.
2753 5257 7000 NOP /1,2 USEC.
2754 5260 7000 NOP /1,2 USEC.
2755 5261 2271 ISZ DELAYN /2,6 USEC.
2756 5262 7610 SKP CLA /1,2 USEC.
2757 5263 5644 JMP I ADELAY /2,4 USEC.
2758 5264 0644 AND I ADELAY /3,8 USEC.
2759 5265 0644 AND I ADELAY /3,8 USEC.
2760 5266 0244 AND ADELAY /2,6 USEC.
2761 5267 0244 AND ADELAY /2,6 USEC.
2762 5270 5251 JMP DLOP /1,2 USEC.
2763 /
2764 5271 0000 DELAYN, 0
2765 5272 7755 M23, =23
2766 5273 0000 DCOUNT, 0
2767 /
2768 /
2769 /
2770 /
2771 /ROUTINE TO HANDLE LAS INSTRUCTION FROM EITHER HARDWARE OR
2772 /SOFT SWITCH REGISTER
2773 /
2774 5274 0000 LASRT, 0
2775 5275 7300 CLA CLL
2776 5276 1021 TAD OPSEL /GET HARDWARE CONTROL WORD ONE

```

```

2777 5277 7710 SPA CLA /SKIP IF RIT 0 = 0
2778 5200 7614 7614 /DO A LAS AND SKIP
2779 5201 1020 TAD PSWR /GET PSUEDO SWITCH REGISTER
2780 5202 5674 JMP I LASRT
2781 /
2782 /
2783 /ROUTINE TO CHECK FOR OCTAL DIGIT
2784 /
2785 /
2786 5203 0000 OCTCHK, 0
2787 5204 1236 TAD KBSV /GET CHARACTER
2788 5205 1364 TAD (=60 /SUBTRACT 60
2789 5206 7710 SPA CLA /SKIP IF GREATER THAN ASCII CODE 60
2790 5207 5237 JMP SWRERR /NOT AN OCTAL NUMBER
2791 5210 1236 TAD KBSV /GET CHARACTER
2792 5211 7041 CIA /NEGATE IT
2793 5212 1363 TAD (67 /ADD 70
2794 5213 7710 SPA CLA /SKIP IF LESS THAN ASCII CODE 70
2795 5214 5237 JMP SWRERR /NOT AN OCTAL NUMBER
2796 5215 1236 TAD KBSV
2797 5216 5703 JMP I OCTCHK
2798 /
2799 /
2800 /
2801 /
2802 /ROUTINE TO HANDLE A CONTROL S
2803 /
2804 5217 7300 SSTOP, CLA CLL
2805 5220 4766' JMS CRLF /EXECUTE CR AND LF
2806 5221 1362 TAD (336 /PRINT UP ARROW
2807 5222 4767' JMS PRNT1
2808 5223 1361 TAD (123 /PRINT S
2809 5224 4767' JMS PRNT1
2810 5225 4467 MESSAGE
2811 5226 6024 PCEQL
2812 5227 7340 CLA CLL CMA
2813 5230 1760' TAD CHKKB
2814 5231 4470 REGPRT
2815 5232 6030 KCF /CLEAR KEYBOARD FLAG
2816 5233 6031 KSF /WAIT FOR OPERATOR INTERVENTION
2817 5234 5333 JMP ,=1
2818 5235 4445 CTLCHK /CHECK FOR CONTROL CHARACTER
2819 5236 4444 ECHO /ECHO CHARACTER
2820 5237 5332 JMP ,=5 /CONTINUE WAITING FOR CONTROL CHARACTER
2821 /
2822 5240 7300 QCONT, CLA CLL
2823 5241 4766' JMS CRLF
2824 5242 1362 TAD (336 /PRINT UP ARROW
2825 5243 4767' JMS PRNT1
2826 5244 1357 TAD (121 /PRINT Q
2827 5245 4767' JMS PRNT1
2828 5246 6030 KCF /CLEAR KEYBOARD FLAG
2829 5247 5772' JMP KROUT /RETURN TO PROGRAM
2830 /
2831 5257 0121

```

```

2832 5160 4610
2833 5161 0123
2834 5162 0336
2835 5163 0067
2836 5164 7720
2837 5165 5406
2838 5166 5007
2839 5167 5000
2840 5170 0277
2841 5171 0232
2842 5172 4621
2843 5173 5453
2844 5174 5455
2845 5175 7766
2846 5176 7763
2847 5177 0177
2848 5178 5400
2849
2850
2851
2852 5400 6030
2853 5401 4777
2854 5402 7300
2855 5403 1376
2856 5404 4775
2857 5405 1374
2858 5406 4775
2859 5407 4467
2860 5410 6047
2861 5411 4465
2862 5412 4470
2863 5413 5214
2864
2865
2866
2867
2868
2869
2870 5414 7300
2871 5415 3253
2872 5416 3255
2873 5417 1373
2874 5420 3772
2875 5421 6031
2876 5422 5221
2877 5423 4445
2878
2879 5424 4444
2880 5425 4771
2881 5426 4770
2882 5427 0367
2883 5430 3253
2884 5431 7240
2885 5432 3255
2886 5433 6031
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940

```

PAGE
//
/ROUTINE TO HANDLE CONTROL G
//
CRS*CK, KCF /CLEAR KEYBOARD FLAG
//
JMS CRLF /CLEAR SWITCH BUILD AREA
CLA CLL /CLEAR PSUEDO SWR CHANGE SWITCH
TAD 1336 /SET UP LOOP COUNTER
JMS PRNT1 /PRINT UP ARROW
TAD 1107
JMS PRNT1 /PRINT G
MESSAGE /PRINT SR= MESSAGE
SREGI /GET SWITCHES
LAS /GET SWITCHES
REGPRT /PRINT THE SWITCH REGISTER
JMP PSUEDO
//
//
//
/ROUTINE TO CHANGE CONTENTS OF PSUEDO SWITCH REGISTER
//
PSUEDO, CIA CLL
DCA PSRSV1 /CLEAR SWITCH BUILD AREA
DCA PSWCH /CLEAR PSUEDO SWR CHANGE SWITCH
TAD (-4) /SET UP LOOP COUNTER
DCA ROTCNT
KSF
JMP -=1
CTLCHK
//
ECHO
JMS KBCRLF /CHECK FOR CR OR LF
JMS OCTCHK /CHECK FOR OCTAL NUMBER
AND 17 /MASK FIRST DIGIT
DCA PSRSV1 /SAVE IT
CLA CMA
DCA PSWCH /SET PSWR CHANGE SWITCH
PSRLOP, KSF /CHECK FOR OPERATOR INTERVENTION

```

2886 5434 5233
2887 5435 4448
2888 5436 4444
2889 5437 4771
2890 5440 4770
2891 5441 0367
2892 5442 3254
2893 5443 1253
2894 5444 7006
2895 5445 7004
2896 5446 1254
2897 5447 3253
2898 5450 2772
2899 5451 5233
2900 5452 5766
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940

```

JMP -=1
CTLCHK
ECHO
JMS KBCRLF /CHECK FOR CR OR LF
JMS OCTCHK /CHECK FOR OCTAL NUMBER
AND 17 /MASK NEXT DIGIT
DCA PSRSV2
TAD PSRSV1
RTL
RAL
TAD PSRSV2
DCA PSRSV1 /SAVE NEW SWITCH REGISTER
ISZ ROTCNT /INCREMENT ROTATE COUNTER
JMP PSRLOP /CONTINUE BUILDING NEW SWR
JMP SWRERR /TOO MANY DIGITS TYPED
//
PSRSV1, 0
PSRSV2, 0
PSWCH, 0 /PSWR CHANGE SWITCH
//
/ROUTINE TO ALLOW CHANGING OF DEVICE SELECT CODES
//
IOTCNG, CBACTV /CHECK FOR ACTIVE CONSOLE
JMP IOTMES /CONSOLE ACTIVE
CLA CLL /CONSOLE NOT ACTIVE
TAD DSCNOW /GET OLD DEVICE CODE
HLT /OLD DEVICE CODE IN AC 3-8
LAS /GET NEW DEVICE CODE
MANCNG /CHANGE IOT DEVICE CODES
TAD DSCNOW /GET NEW DEVICE CODE
HLT /NEW DEVICE CODE IN AC 3-8
JMP TST1 /GO TO LOGIC TESTS
IOTMES, LAS /GET NEW DEVICE CODE
MANCNG /CHANGE IOT DEVICE CODES
MESSAGE /PRINT MESSAGE
DSCEQL /DSC=
TAD DSCNOW /GET NEW DEVICE CODE
REGPRT /PRINT NEW DEVICE CODE
JMP TST1 /GO TO LOGIC TEST
//
//
CNGMAN, 0
AND 0770
DCA DSCNOW /SAVE NEW DSC
TAD DSCNOW /GET IT
TAD K6000 /ADD IOT OP CODE
DCA IOTSTR /SAVE IT
TAD IOTSTR
DCA I ITAB0 /SAVE IOT0
ISZ IOTSTR /INCREMENT FOR IOT1
TAD IOTSTR
DCA I ITAB1 /SAVE IOT1
ISZ IOTSTR /INCREMENT FOR IOT2
TAD IOTSTR

```

2941 5514 3740 DCA I ITAB2 /SAVE IOT2
2942 5515 2335 ISZ IOTSTR /INCREMENT FOR IOT3
2943 5516 1335 TAD IOTSTR
2944 5517 3741 DCA I ITAB3 /SAVE IOT3
2945 5520 2335 ISZ IOTSTR /INCREMENT FOR IOT4
2946 5521 1335 TAD IOTSTR
2947 5522 3742 DCA I ITAB4 /SAVE IOT4
2948 5523 2335 ISZ IOTSTR /INCREMENT FOR IOT5
2949 5524 1335 TAD IOTSTR
2950 5525 3743 DCA I ITAB5 /SAVE IOT5
2951 5526 2335 ISZ IOTSTR /INCREMENT FOR IOT6
2952 5527 1335 TAD IOTSTR
2953 5530 3744 DCA I ITAB6 /SAVE IOT6
2954 5531 2335 ISZ IOTSTR /INCREMENT FOR IOT7
2955 5532 1335 TAD IOTSTR
2956 5533 3745 DCA I ITAB7 /SAVE IOT7
2957 5534 5677 JMP I CNGMAN /RETURN
2958
2959 5535 0000 / IOTSTR, 0
2960
2961 5536 3201 ITAB0, IOT0
2962 5537 3206 ITAB1, IOT1
2963 5540 3217 ITAB2, IOT2
2964 5541 3226 ITAB3, IOT3
2965 5542 3233 ITAB4, IOT4
2966 5543 3240 ITAB5, IOT5
2967 5544 3254 ITAB6, IOT6
2968 5545 3261 ITAB7, IOT7
2969
2970 5564 0770
2971 5565 0400
2972 5566 5237
2973 5567 0007
2974 5570 5303
2975 5571 5200
2976 5572 5600
2977 5573 7774
2978 5574 0107
2979 5575 5000
2980 5576 0336
2981 5577 5007
2982 5600 PAGE
2983 5600 0000 / ROTCNT, 0
2984 /
2985 /
2986 5601 7300 MONTR, CLA CLL
2987 5602 4777* JMS CRLF
2988 5603 1376 TAD (336
2989 5604 4775* JMS PRNT1
2990 5605 1374 TAD (103
2991 5606 4775* JMS PRNT1
2992 5607 4467 MESSAGE
2993 5610 6146 MONRET
2994 5611 5612 JMP I .+1
    
```

```

2995 5612 7600 7600
2996 5613 7402 HLT
2997 /
2998 /
2999 /
3000 /ROUTINE TO ECHO A CTRL-T
3001 /
3002 /
3003 5614 4777* TECHO, JMS CRLF
3004 5615 7300 CLA CLL
3005 5616 1376 TAD (336 /GET UPARROW CODE
3006 5617 4775* JMS PRNT1
3007 5620 1373 TAD (124 /GET T
3008 5621 4775* JMS PRNT1
3009 5622 5223 JMP TSELECT
3010 /
3011 /ROUTINE TO SELECT A TEST USING PSUEDO SWITCH REGISTER
3012 /THIS ROUTINE WAITS FOR A CTRL-G, ASKS SR= QUESTION,
3013 /ACCEPTS TEST NUMBER IN BITS 6-11 OF PSUEDO SWITCH
3014 /REGISTER, AND STARTS THE DIAGNOSTIC FROM TEST NUMBER
3015 /SELECTED.
3016 /
3017 5623 4466 TSELECT, CRACTV
3018 5624 5230 JMP TPSUDD
3019 5625 4465 LAS
3020 5626 5244 JMP TSET
3021 5627 4444 ECHO
3022 5630 6030 TPSUDD, KCF /CLEAR KEYBOARD FLAG
3023 5631 6031 KSF /TEST KYBD FLAG
3024 5632 5231 JMP ,-1 /WAIT FOR A CHARACTER
3025 5633 7300 CLA CLL
3026 5634 6034 KRS /READ KYBD BUFFER, DO NOT CLEAR FLAG
3027 5635 0372 AND (177 /MASK TO 7 BITS
3028 5636 1371 TAD (-7 /IS IT CTRL-G?
3029 5637 7640 SZA CLA /SKIP IF YES
3030 5640 5227 JMP TPSUDD-1 /NO, CONTINUE WAITING
3031 5641 4452 KBCCK /HANDLE A CTRL-G
3032 5642 7300 CLA CLL
3033 5643 1020 TAD 20 /GET PSUEDO SWITCH REGISTER
3034 5644 0370 TSET, AND (77 /MASK BITS 6-11
3035 5645 1367 TAD (SIRTAB /CALCULATE STARTING ADDRESS
3036 5646 3252 DCA TPOINT /SAVE POINTER
3037 5647 1652 TAD I TPOINT
3038 5650 3253 DCA GOTEST /SAVE STARTING ADDRESS
3039 5651 5653 JMP I GOTEST /GO TO THE SELECTED TEST
3040 /
3041 5652 0000 TPOINT, 0 /TEST TABLE POINTER
3042 5653 0000 GOTEST, 0 /TEST ADDRESS POINTER
3043 /
3044 5654 0205 SIRTAB, START+1 /GO TO BEGINNING OF DIAGNOSTIC
3045 5655 0400 TST1
3046 5656 0412 TST2
3047 5657 0424 TST3
3048 5660 0437 TST4
3049 5661 0452 TST5
    
```

3050	5662	0464	TST6
3051	5663	0476	TST7
3052	5664	0510	TST10
3053	5665	0523	TST11
3054	5666	0536	TST12
3055	5667	0550	TST13
3056	5670	0600	TST14
3057	5671	0612	TST15
3058	5672	0625	TST16
3059	5673	0640	TST17
3060	5674	0654	TST20
3061	5675	0666	TST21
3062	5676	0674	TST22
3063	5677	0702	TST23
3064	5700	0710	TST24
3065	5701	0725	TST25
3066	5702	0740	TST26
3067	5703	0746	TST27
3068	5704	0760	TST30
3069	5705	1000	TST31
3070	5706	1012	TST32
3071	5707	1024	TST33
3072	5710	1036	TST34
3073	5711	1050	TST35
3074	5712	1071	TST36
3075	5713	1114	TST37
3076	5714	1137	TST40
3077	5715	1200	TST41
3078	5716	1223	TST42
3079	5717	1246	TST43
3080	5720	1264	TST44
3081	5721	1400	TST45
3082	5722	1432	TST46
3083	5723	1476	TST47
3084	5724	1600	TST50
3085	5725	2000	TST51
3086	5726	2045	TST52
3087	5727	2200	TST53
3088	5730	2255	TST54
3089	5731	2400	TST55
3090	5732	2466	TST56
3091	5733	2555	TST57
3092	5734	2600	TST60
3093	5735	0205	START+1
3094	5736	0205	START+1
3095	5737	0205	START+1
3096	5740	0205	START+1
3097	5741	0205	START+1
3098	5742	0205	START+1
3099	5743	0205	START+1
3100	5744	0205	START+1
3101	5745	0205	START+1
3102	5746	0205	START+1
3103	5747	0205	START+1
3104	5750	0205	START+1

3105	5751	0205	START+1
3106	5752	0205	START+1
3107	5753	0205	START+1
3108	5767	5654	
3109	5770	0077	
3110	5771	7771	
3111	5772	0177	
3112	5773	0124	
3113	5774	0103	
3114	5775	5000	
3115	5776	0336	
3116	5777	5007	
		6000	PAGE
3117			/
3118			/*****
3119			/
3120			/ START OF MESSAGES
3121			/
3122			/*****
3123			/
3124	6000	4327	WAITNG, TEXT "#WAITING#"
	6001	0111	
	6002	2411	
	6003	1607	
	6004	4300	
3125	6005	4314	LSTERR, TEXT "#LQP-8 STATUS ERROR#"
	6006	2120	
	6007	5570	
	6010	4023	
	6011	2401	
	6012	2425	
	6013	2340	
	6014	0522	
	6015	2217	
	6016	2243	
	6017	0000	
3126	6020	4304	DSCEQL, TEXT "#DSC= "
	6021	2303	
	6022	7540	
	6023	0000	
3127	6024	4340	PCEQL, TEXT "# PC= "
	6025	2003	
	6026	7540	
	6027	0000	
3128	6030	4040	ACEQL, TEXT " AC= "
	6031	4040	
	6032	4001	
	6033	0375	
	6034	4000	
3129	6035	4040	MQEQL, TEXT " MQ= "
	6036	4040	
	6037	4015	
	6040	2175	
	6041	4000	
3130	6042	4040	FLEQL, TEXT " FL= "

```

        6043 4040
        6044 4006
        6045 1475
        6046 4000
3131    6047 4040    SREQ, TEXT    "    SR= "
        6050 4040
        6051 4023
        6052 2275
3132    6053 4000    EXPECT, TEXT    "#EXPECTED= "
        6054 4305
        6055 3020
        6056 0503
        6057 2405
        6060 0475
        6061 4000
3133    6062 4040    ACTUAL, TEXT    "    ACTUAL= "
        6063 4040
        6064 4001
        6065 0324
        6066 2501
        6067 1475
        6070 4000
3134    6071 4314    LDATE, TEXT    "#LQP-8 DATA ERROR#"
        6072 2120
        6073 5570
        6074 4004
        6075 0124
        6076 0140
        6077 0522
        6100 2217
        6101 2243
        6102 0000
3135    6103 4023    SENT, TEXT    " SENT: "
        6104 0516
        6105 2472
        6106 4000
3136    6107 4022    RCVD, TEXT    " RECEIVED: "
        6110 0503
        6111 0511
        6112 2605
        6113 0472
        6114 4000
3137    6115 4304    FAIL, TEXT    "#DHLQA FAILED#"
        6116 1014
        6117 2101
        6120 4006
        6121 0111

        6122 1405
        6123 0443
        6124 0000
3138    6125 4305    ENDLOG, TEXT    "#END OF LOGIC TEST#"
        6126 1604
        6127 4017
        6130 0640
        6131 1417

```

```

        6132 0711
        6133 0340
        6134 2405
        6135 2374
3139    6136 4300    PASCNT, TEXT    "#PASS COUNT= "
        6137 4320
        6140 0123
        6141 2340
        6142 0317
        6143 2516
        6144 2475
3140    6145 4000    MONRET, TEXT    "#RETURN TO MONITOR#"
        6146 4322
        6147 0524
        6150 2522
        6151 1640
        6152 2417
        6153 4015
        6154 1716
        6155 1124
        6156 1722
        6157 4300
3141    6160 4314    MANDEC, TEXT    "#LQP-8 PRINTER DIAGNOSTIC MAINDEC-08=DHLQA#"
        6161 2120
        6162 5570
        6163 4020
        6164 2211
        6165 1624
        6166 0522
        6167 4004
        6170 1101
        6171 0716
        6172 1723
        6173 2411
        6174 0340
        6175 4015
        6176 0111
        6177 1604
        6200 0503
        6201 5560
        6202 7055
        6203 0410
        6204 1421
        6205 0143
        6206 0000

3142    /
3143    /START OF CHARACTER BUFFER STORAGE AREA
3144    /
3145    7000    *7000
3146    7000 0000    BUFF, 0
3147    /
3148    7001 0000    ZBLOCK 144
3149    7145 0000    BUFF1, 0
3150    /
3151    7146 0000    ZBLOCK 144

```

3152
3153

/
\$\$\$

```

0000 11110*00 11111000 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0100 11111*11 11111111 11111111 11111111 11100000 00000000 00000000 00000000 00000000

0200 11111*11 11111111 11111111 11111111 11111111 11111100 00000000 00000000 00000000
0300 00000*00 00000000 00000000 00000000 00000000 00000000 00000000 00000000 11111111

0400 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0500 11111*11 11111111 11111111 11111111 11111111 11111111 11100000 00000001 11111111

0600 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
0700 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11100001

1000 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1100 11111*11 11111111 11111111 11111111 11111111 11111111 11100111 11111111 11111111

1200 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1300 11111*11 11111111 11111111 11111111 11111111 11111111 11111110 01111111 11111111

1400 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
1500 11111*11 11111111 11111111 11111000 00000000 00000000 00000000 00111111 11111111

1600 11111*11 11111111 11111111 11111111 11111100 00000000 00000000 00000000 00000000
1700 00000*00 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000111

2000 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2100 11111*11 11111111 11111111 11111000 00000000 00000000 00000000 00111111 11111111

2200 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2300 11111*00 00000000 00000000 00000000 00000000 00000000 00000011 11111111 11111111

2400 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2500 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 01111111 11111111

2600 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
2700 11111*11 11111111 11111111 11000000 00000000 00000000 01111111 11111111 11111111

3000 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3100 11111*11 11111111 11111111 11100000 00000000 00000000 00000111 11111111 11111111

3200 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3300 11111*11 11111111 11111111 11111111 11111111 11111111 11110000 00000111 11111111

3400 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3500 11111*11 11111111 11110000 00000000 00000000 00000000 00000000 00000111 11111111

3600 11111*11 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
3700 11111*00 00000000 00000000 00000000 00000000 00000000 00000000 00000111 11111111

```

```

4000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
4100 11111111 11111111 11111111 11111111 11111111 00000000 00000000 00011111
4200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
4300 11111111 11111111 11111111 11110000 00000000 00000000 00000000 00011111
4400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
4500 11111111 11111111 11111111 11111111 11111111 11111111 11111110 00011111
4600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
4700 11111111 11111111 11111111 11111111 10000000 00000000 11111111 11111111

5000 11111111 11111111 11111111 11111111 11111000 00000000 00000000 00000000
5100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00111111

5200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
5300 11111111 11111111 11111111 11111111 11111111 00000001 11111111 11111111

5400 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
5500 11111111 11111111 11111111 11111111 11111100 00000000 00001111 11111111

5600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
5700 11111111 11111111 11111111 11111111 11111111 11110000 00000001 11111111

6000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
6100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

6200 11111110 00000000 00000000 00000000 00000000 00000000 00000000 00000000
6300 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

6400
6500

6600
6700

7000 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
7100 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

7200 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
7300 11111111 11000000 00000000 00000000 00000000 00000000 00000000 00000000

7400
7500

7600
7700
    
```

```

ACEOL 6030 CRLF 5007 INDX10 0010 LINCNT 0115
ACRSTR 4205 CRLFRT 4471 INDX11 0011 LINE 0131
ACSAVE 4200 CRMOV 4441 INDX12 0012 LINEND 4454
ACSV 4626 CPOUT 5217 INDX13 0013 LINES 0132
ACTHAL 6062 CSWTC 3111 INDX14 0014 LININ 3112
ACTVCR 4306 CTLCHK 4445 INIT 3421 LINKSV 4627
ADDS1 0001 DATERR 4437 INSWO 0237 LINLST 4546
ADDS2 0002 DATPRT 3457 INTFLG 0116 LMOVEB 3045
ADDS3 0003 DPLDAD 3265 INTST 4436 LOADDB 4435
ADFNAY 5244 DCOURT 5273 INTSVC 4000 LUCO 0000
ALRNY 7400 DELAY 4443 ITO 3201 LUOPC 1366
AWAIT 4033 DELAYN 5271 IOT1 3206 LUP56A 2504
BCN7ST 4433 DIR60A 3055 IOT2 3217 LUP56B 2534
BSW 7002 DIR60B 3060 IOT3 3226 LP60B 2632
BUF 7000 DIRECT 4466 IOT4 3233 LP60C 2653
BUFF1 7145 DLOP 5251 IOT5 3240 LPLFT 0100
C2525 0112 DTCNT 2725 IOT6 3254 LQBLK 0002
CS2K2 0111 DSCEQL 6020 IOT7 3261 LQCAR 1000
CRACTV 4466 DSCNQH 0142 IOTCNG 5456 LQCHK 0040
CRSWSCK 5400 ECHO 1645 IOTMES 5470 LQCHR 2000
CAF 6007 ELINS5 2451 IOTSTR 5535 LQDON 3400
CAL60 3000 ENDLIN 4534 ITAB0 5536 LQIEN 0001
CARDIR 0126 ENDLG 6125 ITAB1 5537 LQLS 4431
CARIFT 4000 ENDLG 6125 ITAB2 5540 LQLSRT 3246
CARMAG 4477 ENDTST 4440 ITAB3 5541 LQMC 4426
CARMOV 4400 ERD1 3451 ITAB4 5542 LQMCRT 3223
CARPGT 0000 ERD2 3504 ITAB5 5543 LQMP 4425
CHAP 0124 ERD3 3511 ITAB6 5544 LQMPRT 3214
CHARRT 0125 ERRDAT 3430 ITAB7 5545 LQOUT 0020
CHKCTL 4630 ERLDOP 3427 K003 0104 LQPAR 0400
CHKKB 4610 ERROR 4434 K007 0105 LQPC 4427
CHKRDY 3312 ERRORS 3600 K021 0106 LQPCRT 3232
CHPNT 0136 ERPRT 3636 K023 0107 LQRB 4424
CHRRT 4526 ERRS1 3625 K6000 0110 LQRBRT 3205
CVV78 0247 ERRS2 3665 KBCHK 4452 LQRDY 4000
CLRS6A 2472 ERRS3 3672 KBCRLF 5200 LQRE 4432
CLRS6B 2526 ERS1 4232 KROUT 4621 LQRED 0004
CMGVAN 5477 ERS2 4266 K88AV 0137 LQRETR 3260
CNT65A 2553 ERS3 4273 K88V 5236 LQRS 4430
CNT66B 2554 ERSTAT 4213 KCF 6030 LQRSRT 3237
CNT60A 2726 EXIT 4104 KIE 6035 LQSK 4423
CNT60B 2727 EXPECT 6054 LAS 4465 LQSKRT 3200
COL60A 2722 FAIL 6115 LASRT 5274 LSPACE 4463
COL60B 2723 FEEDA 3005 LDATER 6071 LSTART 1532
COL60C 2724 FIRST 0134 LEND 1533 LSTERR 6005
COLCAL 3026 FLAGS 4024 LF 0130 LSTLIN 4460
COLTN 3113 FLEQL 6042 LFOUT 5225 M23 5272
COMP1 3311 FSTCHR 0135 LFCR 4450 MAG 4467
COUNT 4061 GOADD 0231 LIN46 1475 MAG46 1456
COUNT1 4062 GOBK 5235 LIN60A 2717 MAG60A 3056
COUNT2 4110 GOTEST 5653 LIN60B 2720 MAG60B 3061
COUNT3 4111 GTF 6004 LIN60C 2721 MANCNG 4471
    
```

MANNEC	6160	PLP52D	2071	SAVEAC	4447	TST24	0710
MESSAGE	4467	PLP53A	2212	SAVFLG	0103	TST25	0725
MESLIP	4707	PLP53B	2227	SAVGEN	4473	TST26	0740
MESPR	4702	PLP54	2264	SAVLNK	0101	TST27	0746
MUNTR	5601	PLP55	2405	SAVMOV	0121	TST3	0424
MONNET	6146	POS45	1431	SAVMQ	0102	TST30	0760
MOV60	3054	PRADV	0000	SAVMV	0120	TST31	1000
MOVCAP	4455	PRINT	4464	SAVPOS	0140	TST32	1012
MOVCR	4456	PRNT1	5000	SAVPR	0123	TST33	1024
MOV	4465	PRRET	4000	SENT	6103	TST34	1036
MOVFT	4421	PRTCNT	5043	SLOOP	4073	TST35	1050
MOVVAG	0141	PRTREG	5016	SPACE	0127	TST36	1071
MOVVAR	4457	PSRLOP	5433	SPACEL	4600	TST37	1114
MOVVGT	4411	PSRSV1	5453	SPACER	4557	TST4	0437
MG	4013	PSRSV2	5454	SPL	6102	TST40	1137
MGA	7501	PSUEDO	5414	SPGIZL	4606	TST41	1200
MGEOL	6035	PSWCH	5455	SPSIZR	4565	TST42	1223
MGL	7421	PSWR	0020	SREGL	6047	TST43	1246
MSRHT	4721	QCONT	5340	SSTOP	5317	TST44	1264
MVWAIT	4432	RAN1	4144	START	0204	TST45	1400
NF50A	1631	RAN2	4145	STARTB	0233	TST46	1432
NF50B	1640	RANAD1	2714	STATER	4451	TST47	1476
NF51A	2026	RANAD2	2715	STMASK	0076	TST47A	1530
NF51B	2043	RANAD3	2716	STRNT	4243	TST5	0452
NF52A	2103	RANDUM	4472	STRTAB	5654	TST50	1600
NF52B	2116	RDVD	6107	STRTIN	4012	TST51	2000
NF52C	2132	RDYCHK	4446	SVAC	0077	TST52	2045
NF53A	2251	REGCON	5044	SVSTAT	0114	TST53	2200
NF53B	2253	REGLOP	5025	SWAIT	4442	TST54	2255
NF54A	2277	REGPRT	4470	SWREKR	5237	TST55	2400
NUM60A	2730	RESADD	4023	TECHO	5614	TST56	2466
NUM60B	2731	RESGEN	4474	TLOOP1	3456	TST57	2555
OCTCHK	5303	RESTLP	3114	TLOOP2	3635	TST6	0464
OF1REL	0021	RESTOR	3351	TLOOP3	4242	TST60	2600
OF2REL	0022	RESTR	4015	TPOINT	5652	TST7	0476
PARMAG	4505	RETA	3017	TPSUDD	5630	TSTBGN	3400
PARMOV	4316	RIBBON	3117	TSELC	5623	TSTEND	3342
PASCNT	6137	RIBNS2	2134	TSET	5644	TSTLN	3324
PASS	1365	RITCR	4457	TST1	0400	TSTLOP	3350
PC	4014	RMOST	1534	TST10	0510	TYPECH	4722
PCQL	6024	RMOVEB	3033	TST11	0523	V780W1	3270
PCOINT	2465	ROTCHR	4453	TST12	0536	V780W2	3301
PLP45	1413	ROICNT	5600	TST13	0550	VT78CK	4475
PLP46A	1443	RSPACE	4462	TST14	0600	WAIT	4441
PLP46B	1444	RSTART	0232	TST15	0612	WAITNG	6000
PLP47	1506	RSTRAC	4450	TST16	0625	WAITS	4063
PLP50	1610	RTF	6005	TST17	0640	WATOUT	4055
PLP*1A	2011	SAV1	4146	TST2	0412	WLOOP	4044
PLP*1B	2013	SAV2	4147	TST20	0654	WSTAT	0133
PLP*2A	2053	SAVAC	0100	TST21	0666	XCRLF	4461
PLP*2B	2057	SAVADD	0230	TST22	0674	XCT1	3275
PLP*2C	2065	SAVDAT	0122	TST23	0702	XDATA	0113

XFCO	4673
XLQTS	6506
XLQMC	6503
XLQMP	6502
XLQPC	6504
XLQPB	6501
XLQPE	6507
XLQPS	6505
XLQPK	6500
XPRYNT	4507
XPRND	4112
XPRAGN	4135
XSAVGN	4126
XSTAT	0117
ZMOVEA	3015
ZMOVEB	3043

ERRORS DETECTED: 0

LINKS GENERATED: 88

RUN-TIME: 6 SECONDS

3K CORE USED

.L2777	1432	1538#					
.L3165	1614	1654#					
.L3166	1608	1655#					
.L3167	1607	1656#					
.L3170	1604	1657#					
.L3171	1603	1658#					
.L3172	1597	1659#					
.L3173	1596	1602	1660#				
.L3174	1595	1661#					
.L3175	1594	1601	1662#				
.L3176	1563	1569	1571	1578	1580	1663#	
.L3177	1541	1547	1549	1556	1558	1664#	
.L3375	1796	1837	1844#				
.L3376	1792	1845#					
.L3377	1741	1846#					
.L3574	1938	1946#					
.L3575	1937	1947#					
.L3576	1907	1948#					
.L3577	1858	1883	1949#				
.L3774	2018	2025#					
.L3775	2017	2026#					
.L3776	1985	2027#					
.L3777	1976	2028#					
.L4173	2113	2173#					
.L4174	2110	2174#					
.L4175	2079	2175#					
.L4176	2075	2107	2176#				
.L4177	2049	2177#					
.L4373	2067	2285	2300#				
.L4374	2254	2301#					
.L4375	2253	2302#					
.L4376	2224	2303#					
.L4377	2214	2304#					
.L4573	2407	2469#					
.L4574	2403	2470#					
.L4575	2374	2471#					
.L4576	2368	2472#					
.L4577	2317	2473#					
.L4760	2612	2616#					
.L4761	2610	2617#					
.L4762	2609	2618#					
.L4763	2608	2619#					
.L4764	2606	2620#					
.L4765	2603	2621#					
.L4766	2600	2622#					
.L4767	2565	2623#					
.L4770	2563	2624#					
.L4771	2561	2625#					
.L4772	2559	2626#					
.L4773	2557	2627#					
.L4774	2553	2628#					
.L4775	2549	2629#					
.L4776	2545	2630#					

.L4777	2539	2573	2631#				
.L5172	2677	2684#					
.L5173	2671	2685#					
.L5174	2670	2686#					
.L5175	2661	2687#					
.L5176	2650	2688#					
.L5177	2648	2689#					
.L5357	2926	2831#					
.L5360	2813	2832#					
.L5361	2808	2833#					
.L5362	2806	2874	2834#				
.L5363	2793	2835#					
.L5364	2788	2836#					
.L5365	2735	2837#					
.L5366	2734	2805	2823	2838#			
.L5367	2733	2807	2809	2825	2827	2839#	
.L5370	2732	2840#					
.L5371	2722	2841#					
.L5372	2715	2829	2842#				
.L5373	2713	2720	2843#				
.L5374	2710	2717	2844#				
.L5375	2704	2845#					
.L5376	2699	2846#					
.L5377	2696	2847#					
.L5564	2929	2970#					
.L5565	2918	2925	2971#				
.L5566	2900	2972#					
.L5567	2881	2891	2973#				
.L5570	2860	2890	2974#				
.L5571	2879	2889	2975#				
.L5572	2874	2898	2976#				
.L5573	2873	2977#					
.L5574	2857	2978#					
.L5575	2856	2858	2979#				
.L5576	2855	2980#					
.L5577	2853	2981#					
.L5767	3035	3108#					
.L5770	3034	3109#					
.L5771	3028	3110#					
.L5772	3027	3111#					
.L5773	3007	3112#					
.L5774	2990	3113#					
.L5775	2989	2991	3006	3008	3114#		
.L5776	2988	3005	3115#				
.L5777	2987	3003	3116#				
.V0001	769	775	778	880#			
.V0002	746	752	755	882#			
.V0003	232	255#	2606	2620#			
.V0004	711	717	720	728#			
.V0007	2670	2686#	2881	2891	2973#		
.V0010	689	695	699	731#			
.V0012	1110	1169#	1237	1263#	1276	1421#	2650
.V0013	2563	2624#					2688#

