PLEIADES MICROWARE

··SORCERER··

A DISASSEMBLER PROGRAM FOR 8080 BASED MICROCOMPUTERS

ALTAIR 8800 MACHINE/ASSEMBLY LANGUAGE PROGRAM CODING SHEET

Program Name: BASIC Keyboard-ACR Input Routine, #3-12-763 Page 1 of

Programmer: Christopher J. Flynn Date: 3/3/76

Address:

Program Length in Bytes: 64 Language: x Machine ___ Asse

Other Information:

TAG	MNEMONIC	ADDRESS	OCTAL CODE	EXPLANATION
START	PUSHH	XXX, 300	345	Save H,L
	PUSHB	301	305	Save B,C
	LXIH	302	041	Point to "locate CR" switch
		303	334	
		304	XXX	
SENSE	IN	305	333	Read sense switch
		306	377	
	RAL	307	027	Bit 7 to Carry - test A15
	JC	310	332	A15 is set. GO TO ACRIN
		311	335	
		312	XXX	
KBDIN	MVIA	313	076	Set ACC to 1
		314	001	
	MOV	315	167	Set "locate CR" switch to 1
KBDCSW	IN	316	333	Read keyboard status word
		317	000	
	ANI	320	346	examine data ready bit
		321	002	
	JZ	322	312	Not ready. Go to KBDCSW
		323	316	

Program Name: BASIC Keyboard-ACR Input Routine #3-12-763 Page 2 of 10

TAG	MNEMONIC	ADDRESS	OCTAL CODE	EXPLANATION
		XXX, 324	XXX	
	IN	325	333	Read keyboard data
		326	001	
	ANI	327	346	Select bits 0-6
		330	177	
	POPB	331	301	Restore B,C
	POPH	332	341	Restore H,L
	RET	333	311	Pass kbd character to BASIC
SWITCH		334	001	If switch is 1, search for CR
ACRIN	CALL	335	315	Obtain data word from ACR
		336	367	
		337	XXX	
	MOVBA	340	107	Save ACR data word in Reg B
	MOVAM	341	176	Load "locate CR" switch
	RAR	342	037	Bit 0 to carry -test for seek or read
	JNC	343	332	Bit 0 is 0 go to read mode
		344	361	
		345	XXX	
ACRSEEK	MOVAB	346	170	Restore ACC from Reg B
	CPI	347	376	Does ACC contain a CR
		350	015	
	JNZ	351	302	No, keep looking for the 1st CR
		352	335	
		353	XXX	
	SUBA	354	227	Yes, set "locate CR" switch to 0
	MOVMA	355	167	
	JMP	356	303	Now go read data

rogram Name: BASIC Keyboard-ACR Input Routine #3-12-763 Page 3 of 10

TAG	MNEMONIC	ADDRESS	OCTAL CODE	EXPLANATION
		XXX, 360	XXX	
ACRREAD	MOVAB	361	170	Restore ACC from Reg B
	ANI	362	346	Select bits 0-6
		363	177	
	POPB	364	301	Restore B,C
	POPH	365	341	Restore H,L
	RET	366	311	Pass ACR character to BASIC
ACRCSW	IN	367	333	Read ACR status word
		370	006	
	RRC	371	017	Bit 0 to carry - te t for data ready
	JC	372	332	Not ready. GOTO ACRCSW
		373	367	
		374	XXX	
	IN	375	333	Ready. Read ACR character
		376	007	
	RET	377	311	Pass character to caller

ALTAIR 8800 MACHINE/ASSEMBLY LANGUAGE PROGRAM CODING SHEET

Program Name: Modified BASIC Output Routine #1 #3-12-763 Page 4 of

Programmer: Christopher J. Flynn Date: 3/3/76

Address: 2601 Claxton Drive, Herndon, VA 22070

Program Length in Bytes: 13 Language: X Machine ____ As

Other Information: Assumes output device baud rate is less than ACR baud rate.
Code applies to unmodified 88-SIO board.

TAG	MNEMONIC	ADDRESS	OCTAL CODE	EXPLANATION
TTYCSW	IN	XXX, 260	333	Read output device CSW
		261	000	
	ANI	262	346	Select device ready bit
		263	001	
	JZ	264	312	Not ready. GOTO TTYCSW
		265	260	
		266	XXX	
	POPA	267	361	Restore ACC
	OUT	270	323	Send contents of ACC to output device
		271	001	
	OUT	272	323	Send contents of ACC to ACR
		273	007	
	RET	274	311	Back to caller

ALTAIR 8800 MACHINE/ASSEMBLY LANGUAGE PROGRAM CODING SHEET

Program Name: Modified BASIC Output Routine #2 #3-12-763 Page 5 of 1

Programmer: Christopher J. Flynn Date: 3/3/76

Address: 2601 Claxton Drive, Herndon, VA 22070

Program Length in Bytes: 12 Language: X Machine ____ Assem

Other Information: Assumes output device baud rate is greater than or equal to ACR baud rate.

TAG	MNEMONIC	ADDRESS	OCTAL CODE	EXPLANATION
BEGIN	IN	XXX, 260	333	Read ACR status
		261	006	
	RLC	262	007	Move device ready bit to carry
	JC	263	332	Not ready. GOTO BEGIN
		264	260	
		265	XXX	
	POPA	266	361	Restore ACC
	OUT	267	323	Send contents of ACC to ACR
		270	007	
	OUT	271	323	Send contents of ACC to output device
		272	001	
	RET	273	311	Back to caller

Saving BASIC Programs with the ACR

Introduction

Users of the cassette versions of BASIC and users lacking paper tape equipment have only limited capability to save debugged BASIC programs for later use. 4K users are in a peculiar situation in that there is no provision at all for saving programs short of investing in a paper tape reader/punch. In 8K BASIC (cassette version) there is the capability to save progr (the CSAVE command) on cassette tape and to later reload programs (CLOAD). When programs are stored in this manner, they are written to tape in interna form rather than in source (ASCII character) form. This method may impose a release dependency on stored programs. That is, programs CSAVEd in one version of BASIC may or may not CLOAD in a different version of BASIC. Another difficulty with CSAVE and CLOAD is the potential problem in merging program segments from multiple tapes to create a composite program. Finally, BASIC compatible tapes are not easily created or listed off-line.

This article describes a method of using the ACR cassette interface together with patches to BASIC's input and output subroutines in order to si ulate a paper tape reader and punch. Thus, any information which can be displayed on the terminal (programs, subroutines, comments, DATA statements, ar so on) can be stored on cassette tape and retrieved at some point later in t

Since the system described herein stores data in source form, several advantages in operational flexibility are immediately obtained. Barring any changes in BASIC syntax, release dependency is minimized. Programs saved under 4K 3.2 BASIC should load and execute properly under future releases. BASIC statements may be loaded from several tapes to create a larger program For example, a main program may be loaded from one tape, any required subroutines may be loaded from a tape containing a subroutine library, and finally, DATA statements may be read from yet another tape. Using the propo system, it is also possible to create and list BASIC tapes without bringing BASIC. Some of the text editting systems now available can be useful for of line data preparation.

The proposed system does have a few disadvantages. At the present time it is not possible to name files stored on tape. The tape recorder index counter readings must be used to locate files. As will be shown later, the proposed system does not pack characters on the tape as closely as possible. Normally, this will not be of too great significance. The most severe criticism of the proposed method is that there is some question as to the transferability of tapes made in this manner to users with other terminal configurations.

Method

BASIC handles input and output to the terminal by means of input and ourput subroutines which are tailored to the particular I/O interface boards supporting the terminal. These subroutines provide a logical starting place for any attempt to develop effective cassette software.

One subroutine in BASIC is responsible for terminal output. Whenever BASIC attempts to print a character, the output subroutine is invoked. The output subroutine checks the output device status and outputs a character when the device is ready. A simple modification to this subroutine causes a character to be written to the ACR every time a character is printed on the terminal. Thus, if the tape recorder is in the record mode all information (whether typed by the user or printed by BASIC) will be stored on the casset

Operation

Once BASIC has been loaded and the new input and output routines established saving and retrieving programs becomes very straightforward.

A. Saving a BASIC program

To store a program on tape follow the steps enumerated below:

1. Type in the program. Test it to make sure it is fully operational.
2. Add a dummy line at the end of the program (e.g. 999 REM. . .).
3. Type LIST, but do not hit RETURN.
4. Set up the recorder in the record mode. Write down the footage meter reading.
5. Record at least 10-15 seconds of leader.
6. Now type RETURN. The program will be printed on the terminal and recorded on the tape recorder. Note that carriage return marks the beginning of the tape file.
7. After the program has finished printing, allow the ACR to write at least 10-15 seconds of trailer.
8. Stop the tape recorder.
9. Write down the final footage meter reading.

B. Retrieving a BASIC program from tape

To access a program which is stored on tape, perform the following step

1. Prepare the tape recorder for playback operation.
2. Using the footage meter, locate the desired file and stop the recorder in the leader.
3. At this point you may wish to type NEW at the terminal if you are loading a main program. Otherwise, the program statements read fro tape will be merged with the BASIC program currently in memory.
4. Turn on the A15 sense switch to signify that system input will come from the ACR.
5. Type RETURN on the keyboard - this completes the changeover from t keyboard read routine to the ACR read routine.
   The keyboard should now be insensitive to further input. Furthermo the INP console light should be on indicating that BASIC is expecting ACR input.
6. Start the tape recorder making sure that the tape is positioned in leader. BASIC will scan the tape until the first carriage return is encountered which signifies the beginning of the tape file.
7. After the beginning of the file has been located, BASIC statements will be read from tape and printed or displayed on the output terminal.
8. Watch for the dummy statement (999REM...) at the end of the program When this is encountered, turn off A15 as soon as possible in order switch BASIC back to keyboard entry.
9. If A15 is not turned off in time, BASIC will be "stuck" in the ACR read mode. If this should happen, keep A15 in the off posistion,

Saving BASIC programs is then a matter of selectively (and manually) turning on and off the tap recorder.

The modification to the output subroutine presumes no alteration of ACR adjustments; the ACR remains set at 300 baud. If the output device is teletype machine operation at less than 300 baud, the extra write instructi in the output subroutine will not substantially degrade printing speed. Indeed, the mismatch in baud rates is the reason that this method does not achieve optimum packing of characters on the cassette tape. Depending on the exact baud rates, there will be a delay of several milliseconds after t ACR character has been written and before the Teletype has finished printin the character. This delay, however, insures that during playback the ACR w not overrun the Teletype. If, on the other hand, an output device is used which is significantly faster than 300 baud (e.g.a TV typewriter using a pa I/O board), then the output routine, modified as above, will limit the data transfer rate from the computer to the output device. If the degradation i too severe, it may be possible to selectively enable and disable the ACR ou put logic in a manner similar to the input routine discussed below.

Depending on the version of BASIC, there may be several places in the interpreter where a check for terminal input is made. Only one of these routines, however, is used for accepting terminal data. The other routines Control C checks used to interrupt a running program.

BASIC's input routine is similar to the output routine. The device st is checked. If the device is ready, a character is read from the device an passed to BASIC for processing. Otherwise, BASIC waits until an input sign is sensed.

The modifications to BASIC's input routine are more involved that the output routine. Essentially, however, the modifications consist of checkin a sense switch on the CPU front panel and then reading from the keyboard or ACR depending on the sense switch setting. To retrieve data from tape then, the only action that is required is to turn on the sense switch and to start the tape recorder. Note that since the ACR has replaced the keyboard as the input device (as long as the sense switch is set) all characters stored on tape will apear on the output device as though they were input from the keyboard.

The timing considerations discussed earlier also apply during playback. Tapes recorded and played back on the same system should be processed properly. A potential problem exists, however, with trying to play back a tape created on another user's system if the other user employs a different speed terminal. For example, a tape made on parallel I/O board TV typewriter syst will most likely not have the several millisecond delay between ASCII charac Attempting to print such a tape on a slower Teletype will cause the ACR to overrun the Teletype. To remedy such a situation where there is a timing mismatch, simply NOP the output device status checking code, read in the pro tape, ignore the gibberish that is printed, and restore the output routine. Most probably, BASIC will have read the tape properly even though the charac could not be printed.

rewind the tape back slightly into the data, and play the tape again. As soon as one character is read from the tape, BASIC will revert back t the keyboard entry mode.

Modifying BASIC

The modification required to BASIC consist of adding a new input subroutine and a new output subroutine and modifying BASIC's existing I/O routine to CALL these new routines. Accompanying sheets contain the machine langua code for the new routines and patches for the cassette version of 3.2 4K BASIC.

Refer to the code for new I/O routines. The sections of code labelled KBDCSW and TTYCSW handle the terminal input and output devices respectively. In the example shown, keyboard input is accepted via an 88-PIO parallel I/O board. Terminal output, on the other hand, is performed via an early versio serial board. The important point is that the KBDCSW and TTYCSW routines must be tailored to the specific devices being used. Any doubt about the I/ programming can be resolved by loading BASIC and examining its terminal I/O routines.

Note that two output routines have been included in the documentation. Choose one of them according to the baud rate of the output terminal device. The new output routine is designed to capitalize on speed difference betwee the ACR and terminal. By outputting to the slower device first and by performing status checking on the slower device, the assumption can be made tha the faster device will always be ready to output. Therefore, status checkin code for the faster device can be eliminated. If, for some reason, satisfactory results are not achieved, modify the new output routine to check the status of both the ACR and the terminal before writing.

As shown on the accompanying documentation, BASIC's I/O routines are replaced with CALL instructions to the new routines. Teh locations shown are applicable to 4K BASIC Version 3.2. A recent issue of "Computer Notes" suggested a method for locating these I/O routines. An easy way to find BASIC's I/O routines consists of loading BASIC and then stopping BASIC while it is printing and stopping it again while it is waiting for terminal input. In each case, note the locations and memory contents when BASIC is stopped. Then, using the EXAMINE switch, find the device status checking IN instructi for each routine. These are the locations that will be replaced by CALLs to the new routines.

Listed below are stops to be followed in order to bring up BASIC and apply the necessary modifications:

1. Toggle in or load from tape the new I/O routines. Locate these routines in a high page of memory and above the area used by the bootstrap loader.
2. Load BASIC according to normal procedure.
3. Stop BASIC as soon as the initialization dialogue begins. Note the location where BASIC was stopped.
4. Replace BASIC's I/O routines with CALLs to the new I/O routines just loaded.
5. Restart BASIC from the location where it was stopped. If BASIC wa in the old output routine, restart it from the newly inserted CALL

statement.

6. Complete the initialization dialogue. Do not allocate all of the memory to BASIC or the new I/O routines will be overlaid.

Conclusion

This article has described a simple software interface to BASIC which effectively simulates a paper tape reader and punch with the result that BAS capability in the area of off=line data storage is greatly enhanced.

Although the system was originally intended to provide a source program storage facility, other applications suggest themselves since any data that can be entered via a keyboard can also be entered via tape. Consider the following BASIC program.

```
10   FOR I = 1 to 10
15   PRINT 900 + I; "DATA"; 3.14159*I
20   NEXT I
```

This program prints a series of DATA statements. If the DATA statement are stored on cassette tape, they can be accessed later by another BASIC pro The ACR, then, may serve as a convenient work file for communicating tempora results between programs.

An advanced user may carry the work file principle a step further. Wit the string capabilities of 8K BASIC, it is possible to write a single compiler. Instead of generating machine code, the compiler could generate BASI statements and save them on tape for later execution.

There are, in the end, a potentially unlimited number of uses for the ACR data storage system presented in this article.

MODIFICATIONS TO 4K 3.2 BASIC

The following patches to BASIC are made after BASIC has been loaded and started and before the initialization dialogue has been completed. Do not apply these patches and then start BASIC from location zero or the patches will be overlaid.

Output Routine

Location	Old Contents	New Contents	
003, 167	333	315	Call new output rt
, 170	000	260	
, 171	346	XXX	
, 172	001	311	Back to BASIC

Input Routine

Location	Old Contents	New Contents	
003, 202	333	315	Call new input rt
, 203	000	300	
, 204	346	XXX	
205	002	311	Back to BASIC

*************************INTRODUCTION****************************

SORCERER IS A DISASSEMBLER WHICH RUNS ON 8080 BASED MICROCOMPUTERS, IS WRITTEN IN ASSEMBLY LANGUAGE, COMES COMPLETE WITH A FULLY COMMENTED ASSEMBLED SOURCE LISTING AND AN OBJECT TAPE RECORDED IN "TARBELL FORMAT". A UNIQUE FEATURE OF SORCERER IS THAT IT COMES WITH THREE DIFFERENT OBJECT ASSEMBLIES ON THE CASSETTE TAPE. THIS MEANS THE PROGRAM CAN BE LOADED AND RUN AT 0000 HEX, 2000 HEX OR 4000 HEX. THE USER CAN SELECT A VERSION OF THE DISASSEMBLER (WHICH WILL LEAVE THE AREA OF MEMORY THAT WILL CONTAIN THE PROGRAM TO BE DISASSEMBLED FREE) THAT BEST SUITS HIS NEEDS. BINARY PUNCHED PAPER TAPES OF THE DISASSEMBLER ARE AVAILABLE FROM PLEIADES MICROWARE AND CAN BE ORDERED BY USING THE FORM ON THE LAST PAGE OF THE OPERATORS MANUAL.

SORCERER WILL TAKE ANOTHER PROGRAM WHICH IS LOADED IN MEMORY AND PRODUCE A "DISASSEMBLY" OF THE CODE RESIDING THERE, STARTING AND ENDING AT THE ADDRESSES THE OPERATOR SPECIFIES. THE DISASSEMBLY WILL INCLUDE IN ORDER FROM LEFT TO RIGHT, THE THE HEX ADDRESS OF THE CODE, THE HEX DATA CONTAINED AT THAT ADDRESS, THE INSTRUCTION MNEMONIC AND THE ASSOCIATED HEX LABEL OR REGISTER NAME IF APPLICAPABLE. ALSO BY PROPER SETTING OF THE SENSE SWITCHES, THE USER CAN SELECT PRINTING OF THE ASCII CHARACTER EQUIVALENTS AND THE OCTAL ADDRESSES AND DATA CONTAINED IN THE MEMORY LOCATIONS PREVIOUSLY REFERENCED IN HEX. THE ASCII DATA IS USEFUL FOR DETERMINING IF CHARACTER MESSAGES OR DATA ARE EMBEDDED IN THE PROGRAM BEING DISASSEMBLED. DURING PRINTING OF THE ASCII DATA, AN ASTERISK WILL BE PRINTED PRECEDING THE CHARACTER IF THE SIGN BIT IS SET AT THE REFERENCED LOCATION AND A PERIOD WILL BE PRINTED AFTER THE CHARACTER IF IT IS A CONTROL CHARACTER. THE SIGN BIT IS OFTEN USED TO "MARK" THE END OF AN ASCII CHARACTER STRING IN SOME PROGRAMS.

THE SORCERER OBJECT PROGRAM IS RECORDED IN TARBELL FORMAT ON CASSETTE TAPE AND THE RECORDED LOCATION OF THE THREE VERSIONS ARE GIVEN ON THE TAPE LABEL. HOWEVER, DUE TO VARIATIONS IN COUNTERS OF DIFFERENT RECORDERS, IT IS RECOMMENDED THAT THE TONES ON THE TAPE BE USED TO VERIFY THE ACTUAL LOCATION. IF THE USER IS NOT FAMILIAR WITH THESE SOUNDS, LISTEN FOR A PURE SYNC TONE WHICH PRECEDES EACH RECORD ON THE TAPE. WHEN THE DATA BEGINS A VERY DISTINCT CHANGE CAN BE HEARD IN THE TONE. THE SOUND WILL CHANGE FROM A RATHER PURE CONTINUOUS TONE TO A SOUND WHICH CONSISTS OF ALMOST PURE NOISE. THIS "NOISE" TONE MARKS THE BEGINNING OF THE DATA FOR EACH VERSION OF THE DISASSEMBLER. AS SOON AS THIS TONE IS HEARD, IMMEDIATLY STOP THE RECORDER AND REWIND JUST FAR ENOUGH TO GET BACK INTO THE "PURE SYNC TONE" AREA. THE USER WILL THEN BE READY TO LOAD THE DATA FROM THE CASSETTE TAPE INTO THE COMPUTER. USE THIS PROCEDURE TO FIND THE BEGINNING OF EACH OF THE THREE VERSIONS OF THE DISASSEMBLER WHICH ARE RECORDED ON THE TAPE.

****

********************INTRODUCTION CONTINUED**********************

IT IS SUGGESTED THAT THE ACTUAL LOCATIONS AS SHOWN ON YOUR TAPE COUNTER BE MARKED ON THE TAPE LABEL FOR SUBSEQUENT USE SO THAT THE ABOVE PROCEDURE NEED NOT BE REPEATED EACH TIME THE DISASSEMBLER IS RELOADED. IT IS ALSO HIGHLY RECOMMENDED THAT THE SMALL PLASTIC TABS ON THE CASSETTE TAPE (LOOK ON THE OPPOSITE EDGE FROM THE EDGE WHERE THE TAPE IS EXPOSED) BE BROKEN OUT USING A SMALL TIPPED SCREW DRIVER. THIS WILL PREVENT ACCIDENTAL ERASURE OF THE TAPE SINCE MOST RECORDERS DISABLE ENGAGEMENT OF THE RECORD HEAD WHEN THESE TABS ARE REMOVED. PLEIADES MICROWARE CAN NOT BE RESPONSIBLE FOR REPLACING TAPES WHICH HAVE BEEN ACCIDENTALLY RECORDED OVER, ERRASED, OR HAVE HAD OTHER PROGRAMS RECORDED ON THEM.

FOR THE OPERATORS CONVENIENCE, PAGES OF THE SOURCE LISTING HAVE BEEN NUMBERED 1 THRU 2Ø AND PAGES OF THE OPERATORS MANUAL HAVE BEEN NUMBERED Ø1 THRU Ø9. ON THE FOLLOWING PAGE, BRIEF INSTRUCTIONS ARE GIVEN ON HOW TO USE SORCERER. EACH STEP MAY REFER THE USER TO ANOTHER PAGE WHICH WILL GIVE MORE DETAILS PERTAINING TO THAT STEP. AFTER READING THE INDICATED PAGE, ALWAYS RETURN TO PAGE Ø3 AND PROCEED WITH THE NEXT STEP. SEVERAL REFERENCES ARE MADE ON PATCHING THE INPUT/OUTPUT ROUTINES TO CONFORM TO YOUR SYSTEMS CONFIGURATION. TERMS LIKE "TRANSMITTER BUFFER EMPTY" AND "DATA AVAILABLE FLAG" ARE USED IN THESE REFERENCES. IF THE USER IS UNFAMILIAR WITH THESE TERMS, HE SHOULD CONTACT THE STORE WHERE HIS SYSTEM WAS PURCHASED OR A LOCAL HOBBIEST GROUP TO FIND OUT EXACTLY HOW THEY APPLY TO HIS SYSTEM. IT CAN ONLY BENEFIT THE USER TO BECOME INTIMATELY FAMILIAR WITH THESE COMMON I/O TERMS.

ONLY ONE SOURCE LISTING IS PROVIDED, ASSEMBLED TO 4ØØØH ALTHOUGH THREE OBJECT ASSEMBLIES ARE RECORDED ON THE TAPE. TO APPLY THE SOURCE LISTING TO THE OTHER VERSIONS OF THE DISASSEMBLER, MENTALLY REPLACE THE FIRST NUMBER OF EACH ADDRESS GIVEN ON THE 4ØØØH LISTING WITH A "2" FOR THE (2)ØØØH VERSION OR A "Ø" FOR THE (Ø)ØØØH VERSION. ONE OF THE FIRST USES OF SORCERER, SHOULD BE TO TURN THE DISASSEMBLER ON ITSELF TO PRODUCE USEFUL LISTINGS OF THE ØØØØH AND 2ØØØH VERSIONS.

THIS SOFTWARE HAS BEEN THOROUGHLY TESTED AND IS BELIEVED TO BE FREE OF ERRORS. HOWEVER, NO WARRANTIES ARE EXPRESSED OR IMPLIED AND THE USER MUST DETERMINE THE SUITABLITY OF THIS PRODUCT FOR ITS INTENDED USE. PLEIADES MICROWARE RESERVES THE RIGHT TO MAKE CHANGES, CORRECTIONS AND IMPROVEMENTS IN FUTURE EDITIONS.

THE AUTHOR WISHES TO THANK THE FOLLOWING PERSONS;

DAN MAC LEAN: FOR INSPIRATION, SUGGESTIONS, AND ASSISTANCE IN THE FINAL PREPARATION OF THIS DOCUMENT.

CORINNE BROEKER: FOR PROOF READING THE OPERATORS MANUAL AND FOR "ASSISTANCE" WHICH WILL ALWAYS BE REMEMBERED.

DENNIS BURKE: FOR ISOLATING A SOFTWARE BUG.

****

********************USING THE DISASSEMBLER***********************

1. KEY IN THE TARBELL FORMAT LOADER SHOWN ON PAGE 04.
   (IF YOU HAVE A TARBELL LOADER IN PROM OR ON PAPER
   TAPE OMIT STEP 1)

2. LOAD THE DISASSEMBLER FOLLOWING THE INSTRUCTIONS ON PAGE 05.

3. THE PROGRAM IS DELIVERED WITH PROCESSOR TECH. STANDARD
   TELETYPE I/O, THAT IS;

PORT	0	FOR THE STATUS PORT.
BIT	6	FOR THE DATA AVAILABLE FLAG.
BIT	7	FOR THE TRANSMITTER BUFFER EMPTY FLAG.
PORT	1	FOR THE DATA PORT.

   BOTH OF THE FLAG BITS ARE ACTIVE HIGH, IE. USE A JZ IN A WAIT
   LOOP UNTIL READY.

   IF YOUR SYSTEM USES A DIFFERENT I/O FORMAT THAN THE ONE
   DESCRIBED ABOVE THEN MAKE THE CHANGES SHOWN ON PAGE 06
   TO THE DISASSEMBLER USING THE SOURCE LISTING TO AID YOU.

4. MAKE THE FOLLOWING PROGRAM TERMINATION PATCH ONLY IF
   YOUR SYSTEM HAS A PERMANENT MONITOR IN ROM OR PROM.
   OTHERWISE, LEAVE ADDRESS 4294 AND 4295 AS THEY ARE.
   SEE THE RUNNING INSTRUCTIONS ON PAGE 07 FOR DETAILS.

AT LINE NO. 2880, PATCH ADDRESS 4294,	TO THE LOW ADDRESS OF YOUR SYSTEM PROM MONITOR.
AT LINE NO. 2880, PATCH ADDRESS 4295,	TO THE HIGH ADDRESS OF YOUR SYSTEM PROM MONITOR.

5. THE DISASSEMBLER RUNS FROM 0000H, 2000H OR 4000H (DEPENDING
   ON WHICH VERSION YOU LOAD) EXAMINE THAT ADDRESS.

6. DECIDE WHAT DATA YOU WANT PRINTED ON THE DISASSEMBLY.
   SENSE SWITCH 8 CONTROLS PRINTING OF THE ASCII DATA AND
   SENSE SWITCH 9 CONTROLS PRINTING OF THE OCTAL DATA.

SET SENSE SWITCH 8:	UP	TO PRINT ASCII DATA
	DOWN	TO OMIT PRINTING OF THE ASCII DATA
SET SENSE SWITCH 9:	UP	TO PRINT OCTAL DATA
	DOWN	TO OMIT PRINTING OF THE OCTAL DATA

7. FOLLOW THE RUNNING INSTRUCTIONS ON PAGE 07.

****

1	START SYNCH STREAM
9	STOP " "
14	START TONE
15½	START DATA 1
1[illegible]	TONE RETORNS
20	START DATA 2
24	TINE RETURNS
26	START DATA 3
30	TONE RETURNS
36	END

MACRO DIS
CASSETTE

1	START TONE
3½	STAART DATA
5	TONE RETURNS
6½	START DATA
8½	TONE RETURNS
10	START DATA
12	TONE RETURNS
13½	END

SORCERER
CASSETTE

********************LOADING THE PROGRAM**************************

THE DISASSEMBLER IS SUPPLIED ON A PHILLIPS TYPE CASSETTE TAPE AND IS RECORDED IN STANDARD "TARBELL" FORMAT. IT IS ASSUMED THAT THE COMPUTER IS EQUIPPED WITH A "TARBELL CASSETTE INTERFACE" AND THAT THE INTERFACE IS CONNECTED TO A RECORDER CAPABLE OF RECOVERING AUDIO ENCODED DIGITAL DATA.

PROPER OPERATION OF THE TARBELL INTERFACE SHOULD HAVE ALREADY BEEN VERIFIED. IF DIFFICULTIES ARE ENCOUNTERED, CHECK THAT THE RECORDER VOLUME IS SET AT APPROXIMATELY MID-RANGE AND THAT THE TONE IS SET TO FULL TREBLE FOR MAXIMUM FREQUENCY RESPONSE. TURN THE COMPUTER ON AND PLAY THE SYNC STREAM TAPE PROVIDED WITH THE CASSETTE INTERFACE, VERIFY THAT THE SYNC LIGHT (ON THE TARBELL) IS ON CONTINUOUSLY DURING PLAYBACK OF THE SYNC STREAM. IF THE SYNC LIGHT FLICKERS OR DOESN'T LIGHT, AND CAN'T BE ADJUSTED TO REMAIN ON, THE TARBELL INTERFACE SHOULD BE SERVICED. IF IT IS CERTAIN THAT THE INTERFACE IS WORKING PROPERLY AND DATA STILL CAN NOT BE RECOVERED FROM THE TAPE, RETURN IT TO PLEIADES MICROWARE FOR A CHECK READ TEST.

A CASSETTE LOADER PROGRAM MUST BE USED TO RECOVER THE DISASSEMBLER PROGRAM FROM THE CASSETTE. IF A "STANDARD TARBELL LOADER" WITH PROVISIONS FOR KEYBOARD ENTRY IS NOT AVAILABLE IN PROM OR ROM IN YOUR SYSTEM, THEN THE LOADER SHOWN ON PAGE 02 OF THIS MANUAL MUST BE KEYED IN. IT MUST BE RE-EMPHASIZED THAT THIS IS A STANDARD FORMAT WHICH REQUIRES THE H & L REGISTERS TO POINT TO THE START ADDRESS OF THE LOAD AND THE D & E REGISTERS TO CONTAIN THE BLOCK LENGTH OF THE LOAD.
IF YOU NORMALLY USE A SPECIAL PREAMBLE FORMAT FOR DATA RECOVERY THEN YOU MUST LOAD THE "STANDARD LOADER SHOWN ON PAGE 02", LOAD THE PROGRAM INTO MEMORY AND RE-SAVE THE PROGRAM ON ANOTHER BLANK TAPE IN YOUR OWN FORMAT.

THE DISASSEMBLER CAN BE LOADED INTO THREE DIFFERENT LOCATIONS 0000H, 2000H OR 4000H. AFTER KEYING IN THE LOADER ON PAGE 02, BE CERTAIN TO CHECK THAT ADDRESS 0909H IS PATCHED TO CONTAIN THE CORRECT STARTING ADDRESS OF THE VERSION OF THE DISASSEMBLER TO BE LOADED. EXAMINE ADDRESS 0000H, 2000H OR 4000H (DEPENDING ON WHICH VERSION IS TO BE LOADED) AND VERIFY THAT THE MEMORY THERE IS UNPROTECTED. AT LEAST 2K OF MEMORY SHOULD BE AVAILABLE FOR THE DISASSEMBLER TO RESIDE IN STARTING AT ADDRESS 0000H, 2000H OR 4000H (DEPENDING ON WHICH VERSION).

EXAMINE THE START ADDRESS OF THE LOADER, 0900H. LOAD THE DISASSEMBLER CASSETTE INTO THE RECORDER AND REWIND IT TO ZERO. RESET THE TAPE COUNTER ON THE RECORDER TO ZERO. IF THE 2000H OR 4000H VERSION IS TO BE LOADED, FAST FORWARD TO THE COUNTER LOCATION RECORDED ON THE TAPE LABEL FOR THE VERSION YOU WISH TO LOAD OR VERIFY THE RECORDED LOCATION AS DESCRIBED ON PAGE 01. PUT THE RECORDER IN THE PLAY MODE AND IMMEDIATLY PRESS "RUN" ON THE COMPUTER. WHEN THE PROGRAM FINISHES LOADING, A "G" SHOULD BE PRINTED ON THE TELETYPE OR VIDEO DISPLAY INDICATING THAT A GOOD LOAD HAS OCCURRED WITHOUT A CHECKSUM ERROR. IF AN "E" IS PRINTED INSTEAD, IT INDICATES THAT A CHECKSUM ERROR OCCURRED DURING LOADING AND THE PROGRAM SHOULD BE RELOADED.

****

```
***************I/O PATCHES TO THE DISASSEMBLER******************

    AT LINE NO. 682Ø, PATCH ADDRESS 44C4,  TO YOUR SYSTEMS
                                           OUTPUT STATUS PORT.

    AT LINE NO. 683Ø, PATCH ADDRESS 44C6,  TO YOUR SYSTEMS
                                           OUTPUT TRANSMITTER
                                           BUFFER EMPTY FLAG.

    AT LINE NO. 684Ø, PATCH ADDRESS 44C7,  TO A JNZ, (C2)
                                           ONLY IF YOUR SYSTEM
                                           TRANSMITTER BUFFER
                                           EMPTY FLAG IS ACTIVE
                                           LOW.  OTHERWISE LEAVE
                                           THIS ADDRESS AS IT IS

    AT LINE NO. 686Ø, PATCH ADDRESS 44CC,  TO YOUR SYSTEMS
                                           OUTPUT DATA PORT.

    AT LINE NO. 687Ø, PATCH ADDRESS 44CE,  TO YOUR SYSTEMS
                                           INPUT STATUS PORT.

    AT LINE NO. 688Ø, PATCH ADDRESS 44DØ,  TO YOUR SYSTEMS
                                           INPUT DATA AVAILABLE
                                           FLAG.

    AT LINE NO. 689Ø, PATCH ADDRESS 44D1,  TO A JZ (CA)
                                           ONLY IF YOUR SYSTEMS
                                           INPUT DATA AVAILABLE
                                           FLAG IS ACTIVE LOW,
                                           OTHERWISE LEAVE
                                           ADDRESS 44D1 AS IT IS

    AT LINE NO. 691Ø, PATCH ADDRESS 44D6,  TO YOUR SYSTEMS
                                           INPUT STATUS PORT

    AT LINE NO. 692Ø, PATCH ADDRESS 44D8,  TO YOUR SYSTEMS
                                           INPUT DATA AVAILABLE
                                           FLAG.

    AT LINE NO. 693Ø, PATCH ADDRESS 44D9,  TO A JNZ (C2)
                                           ONLY IF YOUR SYSTEM
                                           INPUT DATA AVAILABLE
                                           FLAG IS ACTIVE LOW,
                                           OTHERWISE LEAVE
                                           ADDRESS 44D9 AS IT IS

    AT LINE NO. 694Ø, PATCH ADDRESS 44DD,  TO YOUR SYSTEMS
                                           INPUT DATA PORT.

****************************************************************
```

********************RUNNING INSTRUCTIONS************************

PRESS "RUN" ON THE COMPUTER, THE DISASSEMBLER WILL PROMPT BY PRINTING "START ADDRESS?". AT THIS POINT TYPE IN THE HEXADECIMAL START ADDRESS OF THE PROGRAM TO BE DISASSEMBLED. TYPE IN THE ABSOLUTE ADDRESS, HIGH ORDER FIRST, IE; TO DISASSEMBLE A PROGRAM AT 5000H, TYPE 5000 (CARRIAGE RETURN). IF AT ANY TIME DURING ENTRY YOU TYPE ANY CHARACTER OTHER THAN A VALID HEXADECIMAL CHARACTER, THE DISASSEMBLER WILL PRINT "NON HEX DATA" AND JMP TO YOUR SYSTEM MONITOR (PROVIDED THE PATCHES LISTED IN STEP 4 ON PAGE 03 HAVE BEEN MADE) OTHERWISE THE PROGRAM WILL JMP TO THE START OF THE DISASSEMBLER AND ASK FOR "START ADDRESS?" AGAIN.

AFTER PRESSING CARRIAGE RETURN, THE DISASSEMBLER WILL ASK FOR "END ADDRESS?". NOW TYPE IN THE HEXIDECIMAL END ADDRESS OF THE PROGRAM TO BE DISASSEMBLED. DO NOT PRESS CARRIAGE RETURN YET. THE SAME CONVENTION FOR HEX DATA ENTRY APPLIES AS DESCRIBED ABOVE AND THE SAME ERROR ROUTINE WILL BE EXECUTED IF AN INVALID HEXIDECIMAL CHARACTER IS TYPED. IF THE SENSE SWITCHES HAVEN'T ALREADY BEEN SET AS DESCRIBED ON PAGE 03 STEP 6, DO SO NOW. PRESS CARRIAGE RETURN. THE DISASSEMBLER WILL TYPE A LINE OF DASHES SEPARATING THE FIRST PAGE OF THE DISASSEMBLY, FOLLOWED BY SEVERAL CARRIAGE RETURN/LINE FEEDS AND TWO LINES OF HEADER MESSAGES. THE PROGRAM WILL THEN BEGIN PRINTING THE DISASSEMBLY, PRINTING ONLY THE DATA SELECTED BY THE SENSE SWITCHES. AFTER 45 LINES OF DISASSEMBLY HAVE BEEN PRINTED, THE DISASSEMBLER WILL PAGE, SEPARATE THE NEXT PAGE WITH ANOTHER LINE OF DASHES, PRINT NEW HEADER LINES AND CONTINUE THE DISASSEMBLY AS BEFORE.

IF AT ANY TIME DURING THE DISASSEMBLY, IT BECOMES NECESSARY TO CHANGE WHAT DATA IS BEING PRINTED, SIMPLY CHANGE THE SENSE SWITCHES (AS DESCRIBED ON PAGE 03 STEP 6) TO THE DESIRED DATA. IF AT ANY TIME DURING THE DISASSEMBLY IT IS DESIRED TO STOP THE DISASSEMBLY, PRESS ANY PRINTING CHARACTER ON THE TELETYPE OR SYSTEM KEYBOARD. THE DISASSEMBLER WILL THEN START OVER AGAIN AND ASK FOR "START ADDRESS?"

AT THIS POINT, THE DISASSEMBLY CAN BE STARTED AGAIN, BY TYPING IN A NEW START ADDRESS AND END ADDRESS EXACTLY AS ABOVE. ALTERNATELY TYPE AN INVALID HEXIDECIMAL CHARACTER, FOR EXAMPLE THE SPACE BAR, AND THE DISASSEMBLER WILL PRINT "NON HEX DATA" AND JMP TO YOUR SYSTEM MONITOR (PROVIDED THE PATCHES HAVE BEEN MADE), THIS PROVIDES A CONVENIENT MEANS OF EXITING THE DISASSEMBLER PROGRAM AND ENTERING A PROGRAM WHICH HAS DIRECTED CONTROL OF PROGRAM EXECUTION SUCH AS A SYSTEM MONITOR.

BE CAREFUL WHILE THE DISASSEMBLER IS RUNNING NOT TO ACCIDENTALLY TYPE SOMETHING ON THE SYSTEM KEYBOARD AS THIS WILL BE INTERPRETED AS A DISASSEMBLY STOP COMMAND. ALSO BE CAREFUL NOT TO ENTER AN INVALID HEX CARACTER (SUCH AS PRESSING THE SPACE BAR TWICE ACCIDENTALLY DURING A STOP COMMAND) AS THIS WILL CAUSE THE DISASSEMBLER TO JMP OUT TO THE MONITOR. CHANGING THE SENSE SWITCHES DURING ACTIVE DISASSEMBLY WILL NOT HAVE AN IMMEDIATE EFFECT, (SINCE THE OUTPUT BUFFER IS ALREADY FULL DURING PRINTING), THE CHANGE WILL BE REFLECTED IN THE NEXT LINE OF THE DISASSEMBLY.

****

```
-------------------------------------------------------------
        THIS IS A SAMPLE OF AN ACTUAL DISASSEMBLY PAGE
        WITH THE EXCEPTION THAT THE DASHED LINES ARE
        SPACED CLOSER HERE TO ALLOW THIS PAGE TO FIT IN
        11 INCHES.  THE ACTUAL PAGES ARE SEPARATED EVERY
        11 INCHES TO ALLOW THEM TO BE CUT APART AND PUT
        IN AN 8 AND 1/2 BY 11 INCH NOTEBOOK.

HEX   HEX             LABEL     ASCII        OCTAL     DATA
ADDR  DATA INST       HILO      1   2   3    ADDRESS   1   2   3

4ØØØ  31   LXI SP,    46BD      1   *=  F    1ØØ ØØØ   Ø61 275 1Ø6
4ØØ3  DB   IN         Ø1       *[   A.       1ØØ ØØ3   333 ØØ1
4ØØ5  CD   CALL       44F7     *M  *W   D    1ØØ ØØ5   315 367 1Ø4
4ØØ8  21   LXI H,     4525      !   %   E    1ØØ Ø1Ø   Ø41 Ø45 1Ø5
4ØØB  CD   CALL       4519     *M   Y.  E    1ØØ Ø13   315 Ø31 1Ø5
4ØØE  CD   CALL       45B4     *M  *4   E    1ØØ Ø16   315 264 1Ø5
4Ø11  FE   CPI        ØD       *↑   M.       1ØØ Ø21   376 Ø15
4Ø13  C2   JNZ        4287     *B  *G.  B    1ØØ Ø23   3Ø2 2Ø7 1Ø2
4Ø16  22   SHLD       44EE      "  *N   D    1ØØ Ø26   Ø42 356 1Ø4
4Ø19  CD   CALL       44F7     *M  *W   D    1ØØ Ø31   315 367 1Ø4
4Ø1C  21   LXI H,     4535      !   5   E    1ØØ Ø34   Ø41 Ø65 1Ø5
4Ø1F  CD   CALL       4519     *M   Y.  E    1ØØ Ø37   315 Ø31 1Ø5
4Ø22  CD   CALL       45B4     *M  *4   E    1ØØ Ø42   315 264 1Ø5
4Ø25  FE   CPI        ØD       *↑   M.       1ØØ Ø45   376 Ø15
4Ø27  C2   JNZ        4287     *B  *G.  B    1ØØ Ø47   3Ø2 2Ø7 1Ø2
4Ø2A  C3   JMP        4684     *C  *D.  F    1ØØ Ø52   3Ø3 2Ø4 1Ø6
4Ø2D  21   LXI H,     4543      !   C   E    1ØØ Ø55   Ø41 1Ø3 1Ø5
4Ø3Ø  CD   CALL       44F7     *M  *W   D    1ØØ Ø6Ø   315 367 1Ø4
4Ø33  CD   CALL       44F7     *M  *W   D    1ØØ Ø63   315 367 1Ø4
4Ø36  CD   CALL       4519     *M   Y.  E    1ØØ Ø66   315 Ø31 1Ø5
4Ø39  CD   CALL       44F7     *M  *W   D    1ØØ Ø71   315 367 1Ø4
4Ø3C  21   LXI H,     4579      !   Y   E    1ØØ Ø74   Ø41 171 1Ø5
4Ø3F  CD   CALL       4519     *M   Y.  E    1ØØ Ø77   315 Ø31 1Ø5
4Ø42  CD   CALL       44F7     *M  *W   D    1ØØ 1Ø2   315 367 1Ø4
4Ø45  C3   JMP        464C     *C   L   F    1ØØ 1Ø5   3Ø3 114 1Ø6
4Ø48  2A   LHLD       44EE      *  *N   D    1ØØ 11Ø   Ø52 356 1Ø4
4Ø4B  7E   MOV A,M              ↑            1ØØ 113   176
4Ø4C  23   INX  H               #            1ØØ 114   Ø43
4Ø4D  22   SHLD       44EE      "  *N   D    1ØØ 115   Ø42 356 1Ø4
4Ø5Ø  C9   RET                 *I            1ØØ 12Ø   311
4Ø51  3C   INR  A               <            1ØØ 121   Ø74
4Ø52  E6   ANI        Ø7       *F   G.       1ØØ 122   346 ØØ7
4Ø54  FE   CPI        Ø6       *↑   F.       1ØØ 124   376 ØØ6
4Ø56  DA   JC         4Ø5B     *Z   [   @    1ØØ 126   332 133 1ØØ
4Ø59  C6   ADI        Ø3       *F   C.       1ØØ 131   3Ø6 ØØ3
4Ø5B  FE   CPI        Ø5       *↑   E.       1ØØ 133   376 ØØ5
4Ø5D  DA   JC         4Ø62     *Z   B   @    1ØØ 135   332 142 1ØØ
4Ø6Ø  C6   ADI        Ø2       *F   B.       1ØØ 14Ø   3Ø6 ØØ2
4Ø62  C6   ADI        41       *F   A        1ØØ 142   3Ø6 1Ø1
4Ø64  12   STAX D               R.           1ØØ 144   Ø22
4Ø65  C9   RET                 *I            1ØØ 145   311
4Ø66  Ø6   MVI B,     Ø4        F.  D.       1ØØ 146   ØØ6 ØØ4
4Ø68  7E   MOV A,M              ↑            1ØØ 15Ø   176
4Ø69  12   STAX D               R.           1ØØ 151   Ø22
4Ø6A  23   INX  H               #            1ØØ 152   Ø43

-------------------------------------------------------------
                                                      PAGE Ø8
```

```
4000 31 BD 46     0010 START LXI SP,STACK      /SET STACK POINTER
4003 DB 01        0020  IN 1                /CLEAR TTY DAV FLAG
4005 CD F7 44     0030  CALL CRLF                   /PRINT A CRLF
4008 21 25 45     0040  LXI H,STADD          /POINT TO MESSAGE ONE
400B CD 19 45     0050  CALL MSG         /PRINT "START ADDRESS?"
400E CD B4 45     0060  CALL HEX    /INPUT 4 VALID HEX CHARACTERS
4011 FE 0D        0070  CPI 13               /WAS LAST CHAR. A CR?
4013 C2 87 42     0080  JNZ BAD  /NO, "NON HEX DATA", JMP MONITOR
4016 22 EE 44     0090  SHLD PGMCT /SAVE THE BEG. PROGRAM COUNTER
4019 CD F7 44     0100  CALL CRLF                   /PRINT A CRLF
401C 21 35 45     0110  LXI H,ENADD          /POINT TO MESSAGE TWO
401F CD 19 45     0120  CALL MSG           /PRINT "END ADDRESS?"
4022 CD B4 45     0130  CALL HEX    /INPUT 4 VALID HEX CHARACTERS
4025 FE 0D        0140  CPI 13     /WAS LAST CHAR. ENTERED A CR?
4027 C2 87 42     0150  JNZ BAD  /NO, "NON HEX DATA", JMP MONITOR
402A C3 84 46     0160  JMP STORE   /SAVE END ADDRES & PG. DIVIDE
402D 21 43 45     0170 HEADR LXI H,LNONE  /POINT TO MESSAGE THREE
4030 CD F7 44     0180  CALL CRLF                   /PRINT A CRLF
4033 CD F7 44     0190  CALL CRLF                   /PRINT A CRLF
4036 CD 19 45     0200  CALL MSG  /PRINT LINE 1 OF HEADER MESSAGE
4039 CD F7 44     0210  CALL CRLF                   /PRINT A CRLF
403C 21 79 45     0220  LXI H,LNTWO         /POINT TO MESSAGE FOUR
403F CD 19 45     0230  CALL MSG  /PRINT LINE 2 OF HEADER MESSAGE
4042 CD F7 44     0240  CALL CRLF                   /PRINT A CRLF
4045 C3 4C 46     0250  JMP SET                /GO SET PAGE LENGTH
4048 2A EE 44     0260 FETCH LHLD PGMCT /LOAD THE CURRENT ADDRESS
404B 7E           0270  MOV A,M         /FETCH THE BYTE REFERENCED
404C 23           0280  INX H              /POINT TO THE NEXT BYTE
404D 22 EE 44     0290  SHLD PGMCT       /STORE THE NEW REFERENCE
4050 C9           0300  RET                          /CONTINUE...
4051 3C           0310 REGLD INR A     /INCREMENT THE MASKED BYTE
4052 E6 07        0320  ANI 07    /STRIP OFF THE CARRY INTO BIT 3
4054 FE 06        0330  CPI 06        /WAS VALUE IN MASKED BYTE 5
4056 DA 5B 40     0340  JC REL1      /IF <5 SKIP NEXT INSTRUCTION
4059 C6 03        0350  ADI 03             /ADD THREE TO THE VALUE
405B FE 05        0360 REL1 CPI 5         /IS NEW VALUE EQUAL TO 5
405D DA 62 40     0370  JC REL2      /IF <5 SKIP NEXT INSTRUCTION
4060 C6 02        0380  ADI 02   /ADD TWO TO THE CALCULATED VALUE
4062 C6 41        0390  REL2 ADI 'A'       /ADD 41H TO CALC. VALUE
4064 12           0400  STAX D   /WRITE CALC. REGISTER IN OUTBUF
4065 C9           0410  RET                          /CONTINUE...
4066 06 04        0420 PRINT MVI B,4   /SET NO. OF CHARS. TO LOAD
4068 7E           0430 SPRN MOV A,M        /FETCH CHAR. FROM TABLE
4069 12           0440  STAX D      /WRITE CHAR. IN OUTPUT BUFFER
406A 23           0450  INX H       /POINT TO NEXT CHAR. IN TABLE
406B 13           0460  INX D       /POINT TO NEXT SLOT IN OUTBUF
406C 05           0470  DCR B              /DECREMENT CHAR. COUNT
406D C2 68 40     0480  JNZ SPRN   /IF MORE CHARS.  CONT. LOADING
4070 C9           0490  RET                          /CONTINUE...
```

```
4Ø71 3A ED 44     Ø5ØØ MASK LDA SAVE       /LOAD THE CURRENT BYTE
4Ø74 E6 38        Ø51Ø  ANI 38H          /MASK FOR THE MIDDLE BITS
4Ø76 ØF           Ø52Ø  RRC        /PUT THE MIDDLE BITS INTO THE-
4Ø77 ØF           Ø53Ø  RRC              /THREE LEAST SIGNIFICANT-
4Ø78 ØF           Ø54Ø  RRC               /BITS IN THE ACCUMULATOR
4Ø79 C9           Ø55Ø  RET                           /CONTINUE...
4Ø7A CD 71 4Ø     Ø56Ø CONDL CALL MASK /GET BITS 3,4,5 INTO 1,2,3
4Ø7D 87           Ø57Ø  ADD A               /DOUBLE CONDITION BITS
4Ø7E 4F           Ø58Ø  MOV C,A       /PUT OFFSET VALUE IN REG. C
4Ø7F 21 52 44     Ø59Ø  LXI H,CONDN      /POINT TO CONDITION TABLE
4Ø82 Ø9           Ø6ØØ  DAD B  /ADD OFFSET TO COND. TABLE POINTER
4Ø83 7E           Ø61Ø  MOV A,M             /FETCH CONDITION DIGIT
4Ø84 12           Ø62Ø  STAX D   /WRITE CONDITION DIGIT IN OUTBUF
4Ø85 23           Ø63Ø  INX H        /POINT TO NEXT DIGIT IN TABLE
4Ø86 13           Ø64Ø  INX D        /POINT TO NEXT SLOT IN OUTBUF
4Ø87 7E           Ø65Ø  MOV A,M /FETCH NEXT COND. DIGIT FROM TBL.
4Ø88 12           Ø66Ø  STAX D            /WRITE IT IN OUTPUT BUFFER
4Ø89 C9           Ø67Ø  RET                           /CONTINUE...
4Ø8A CD 71 4Ø     Ø68Ø LXICD CALL MASK /GET BITS 3,4,5 INTO Ø,1,2
4Ø8D E6 Ø6        Ø69Ø  ANI Ø6                  /MASK FOR BITS 1,2
4Ø8F FE Ø6        Ø7ØØ  CPI Ø6            /ARE BOTH BITS 1 & 2 SET?
4Ø91 C2 51 4Ø     Ø71Ø  JNZ REGLD     /NO, GO LOAD A REGISTER NAME
4Ø94 3E 53        Ø72Ø  MVI A,'S' /YES, MUST BE LXI SP. LOAD AN S
4Ø96 12           Ø73Ø  STAX D  /WRITE AN 'S' IN THE OUTPUT BUFF.
4Ø97 13           Ø74Ø  INX D       /POINT TO NEXT SPACE IN OUTBUF
4Ø98 3E 5Ø        Ø75Ø  MVI A,'P'               /LOAD AN ASCII 'P'
4Ø9A 12           Ø76Ø  STAX D   /WRITE A 'P' IN THE OUTPUT BUFF.
4Ø9B C9           Ø77Ø  RET                           /CONTINUE...
4Ø9C CD F7 44     Ø78Ø DISAS CALL CRLF                /PRINT A CRLF
4Ø9F CD F4 42     Ø79Ø  CALL CLRBUF       /CLEAR THE OUTPUT BUFFER
4ØA2 2A EE 44     Ø8ØØ  LHLD PGMCT   /GET CURRENT PROGRAM COUNTER
4ØA5 22 F5 44     Ø81Ø  SHLD OSAVE  /SAVE PGMCT FOR OCTAL ROUTINE
4ØA8 CD 96 42     Ø82Ø  CALL HEXOT   /WRITE HEX ADDRESS IN OUTBUF
4ØAB CD 48 4Ø     Ø83Ø  CALL FETCH /GET REFERENCED BYTE FROM MEM.
4ØAE 32 F2 44     Ø84Ø  STA CHAR1 /SAVE BYTE IN 1ST. ASCII BUFFER
4ØB1 32 ED 44     Ø85Ø  STA SAVE          /SAVE BYTE FOR LATER USE
4ØB4 67           Ø86Ø  MOV H,A            /PUT BYTE IN REGISTER H
4ØB5 11 68 44     Ø87Ø  LXI D,OUTBUF+2        /POINT TO DATA COLUMN
4ØB8 CD E7 42     Ø88Ø  CALL XCODE       /WRITE HEX BYTE IN OUTBUF
4ØBB 21 55 43     Ø89Ø  LXI H,OPCODES        /POINT TO OPCODE TABLE
4ØBE Ø1 11 ØØ     Ø9ØØ  LXI B,17 /CLR B AND SET C TO NO. OF CODES
4ØC1 BE           Ø91Ø ONE CMP M       /DOES DATA BYTE MATCH TABLE?
4ØC2 CA 76 42     Ø92Ø  JZ BYTE1  /YES, GO PROCESS ONE BYTE INST.
4ØC5 23           Ø93Ø  INX H   /NO, POINT TO NEXT INST. IN TABLE
4ØC6 ØD           Ø94Ø  DCR C       /DECREMENT ONE BYTE TABLE COUNT
4ØC7 C2 C1 4Ø     Ø95Ø  JNZ ONE  /IF MORE ONE BYTE INSTS GO. TEST
4ØCA ØE ØA        Ø96Ø  MVI C,ØAH /SET C TO NO. OF TWO BYTE INSTS
4ØCC BE           Ø97Ø TWO CMP M       /DOES DATA BYTE MATCH TABLE?
4ØCD CA 59 42     Ø98Ø  JZ BYTE2  /YES, GO PROCESS A 2 BYTE INST.
4ØDØ 23           Ø99Ø  INX H   /NO, POINT TO NEXT INST. IN TABLE
```

```
40D1 ØD          1ØØØ  DCR C      /DECREMENT TWO BYTE TABLE COUNT
40D2 C2 CC 4Ø    1Ø1Ø  JNZ TWO    /IF MORE 2 BYTE INSTS. GO TEST
ØØD5 ØE Ø6       1Ø2Ø  MVI C,6    /SET C TO NO. OF 3 BYTE INSTS.
4ØD7 BE          1Ø3Ø THREE CMP M   /DOES DATA BYTE MATCH TABLE?
4ØD8 CA 2D 42    1Ø4Ø  JZ BYTE3  /YES, GO PROCESS A 3 BYTE INST?
4ØDB 23          1Ø5Ø  INX H        /POINT TO NEXT INST. IN TABLE
4ØDC ØD          1Ø6Ø  DCR C        /DECREMENT 3 BYTE TABLE COUNT
4ØDD C2 D7 4Ø    1Ø7Ø  JNZ THREE  /IF MORE 3 BYTE INSTS. GO TEST
4ØEØ E6 CØ       1Ø8Ø  ANI ØCØH               /MASK FOR BITS 6 & 7
4ØE2 FE 4Ø       1Ø9Ø  CPI 4ØH                     /WAS BIT 6 SET?
4ØE4 CA ØB 42    11ØØ  JZ MOV /YES, GO PROCESS A MOV INSTRUCTION
4ØE7 FE 8Ø       111Ø  CPI 8ØH                     /WAS BIT 7 SET?
4ØE9 CA F7 41    112Ø  JZ ADD       /YES, GO PROCESS AN ADD INST.
4ØEC 3A ED 44    113Ø  LDA SAVE              /RESTORE ORIGINAL BYTE
4ØEF E6 C7       114Ø  ANI ØC7H            /MASK FOR BITS Ø,1,2,6,7
4ØF1 D6 Ø4       115Ø  SUI Ø4                       /IS BIT 2 SET?
4ØF3 CA E2 41    116Ø  JZ INR /YES, GO PROCESS A INR INSTRUCTION
4ØF6 3D          117Ø  DCR A               /WERE BITS Ø AND 2 SET?
4ØF7 CA D9 41    118Ø  JZ DCR /YES, GO PROCESS A DCR INSTRUCTION
4ØFA 3D          119Ø  DCR A               /WERE BITS 1 AND 2 SET?
4ØFB CA CØ 41    12ØØ  JZ MVI /YES, GO PROCESS A MVI INSTRUCTION
4ØFE 3A ED 44    121Ø  LDA SAVE        /RESTORE ORIGINAL DATA BYTE
41Ø1 E6 CØ       122Ø  ANI ØCØH        /ARE BOTH BITS 6 AND 7 SET?
41Ø3 CA 7F 41    123Ø  JZ LXI /YES, GO PROCESS A LXI INSTRUCTION
41Ø6 3A ED 44    124Ø  LDA SAVE        /RESTORE ORIGINAL DATA BYTE
41Ø9 E6 C7       125Ø  ANI ØC7H            /MASK FOR BITS Ø,1,2,6,7
41ØB D6 CØ       126Ø  SUI ØCØH            /WERE BITS 6 AND 7 SET?
41ØD CA 72 41    127Ø  JZ RET /YES, GO PROCESS A RET INSTRUCTION
411Ø D6 Ø2       128Ø  SUI Ø2                 /WERE BITS 1,6,3 SET?
4112 CA 65 41    129Ø  JZ JMP /YES, GO PROCESS A JMP INSTRUCTION
Ø115 D6 Ø2       13ØØ  SUI Ø2                 /WERE BITS 2,6,7 SET?
4117 CA 58 41    131Ø  JZ CALL /YES, GO PROCESS CALL INSTRUCTION
411A D6 Ø3       132Ø  SUI Ø3             /WERE BITS Ø,1,2,6,7 SET?
411C CA 43 41    133Ø  JZ RST /YES, GO PROCESS A RST INSTRUCTION
411F 3A ED 44    134Ø  LDA SAVE        /RESTORE ORIGINAL DATA BYTE
4122 E6 Ø7       135Ø  ANI Ø7                    /MASK FOR BITS Ø,1,2
4124 4F          136Ø  MOV C,A          /STORE REGISTER CODE IN C
4125 21 49 44    137Ø  LXI H,LPOP         /POINT TO 'POP' IN TABLE
4128 Ø9          138Ø  DAD B  /OFFSET, POINT TO CORRECT MNEMONIC
4129 11 6D 44    139Ø  LXI D,OUTBUF+7  /POINT TO MNEMONIC COLUMN
412C CD 66 4Ø    14ØØ  CALL PRINT       /WRITE MNEMONIC IN OUTBUF
412F CD 71 4Ø    141Ø  CALL MASK /GET BITS 3,4,5 INTO BITS Ø,1,2
4132 FE Ø6       142Ø  CPI Ø6                    /WERE BITS 4,5 SET?
4134 C2 EE 41    143Ø  JNZ INR3 /YES, GO PROCESS INR INSTRUCTION
4137 21 46 44    144Ø  LXI H,LPSW         /POINT TO 'PSW' IN TABLE
413A 11 72 44    145Ø  LXI D,OUTBUF+12   /POINT TO REGISTER AREA
413D CD 66 4Ø    146Ø  CAHH PRINT          /WRITE 'PSW' IN OUTBUF
414Ø C3 A5 42    147Ø  JMP FINISH  /PROCESS ASCII & OCT. & PRINT
4143 21 42 44    148Ø RST LXI H,LRST        /POINT TO RST MNEMONIC
Ø146 11 6D 44    149Ø  LXI D,OUTBUF+7    /POINT TO MNEMONIC SLOT
```

```
4149 CD 66 4Ø     15ØØ  CALL PRINT      /WRITE MNEMONIC IN OUTBUF
414C CD 71 4Ø     151Ø  CALL MASK /GET BITS 3,4,5 INTO BITS Ø,1,2
414F 11 75 44     152Ø  LXI D,OUTBUF+15     /POINT TO RST NUMERIC
4152 CD E7 42     153Ø  CALL XCODE   /WRITE HEX NUMERIC IN OUTBUF
4155 C3 A5 42     154Ø  JMP FINISH  /WRITE ASCII, OCTAL AND PRINT
4158 3E 43        155Ø CALL MVI A,'C'          /LOAD AN ASCII 'C'
415A 11 6D 44     156Ø  LXI D,OUTBUF+7  /POINT TO MNEMONIC COLUMN
415D 12           157Ø  STAX D        /WRITE 'C' IN OUTPUT BUFFER
415E 13           158Ø  INX D       /POINT TO NEXT SLOT IN OUTBUF
415F CD 7A 4Ø     159Ø  CALL CONDL  /WRITE REST OF CALL CONDITION
4162 C3 3B 42     16ØØ  JMP BYT3    /GO FINISH A THREE BYTE INST.
4165 3E 4A        161Ø JMP MVI A,'J'           /LOAD AN ASCII 'J'
4167 11 6D 44     162Ø  LXI D,OUTBUF+7  /POINT TO MNEMONIC COLUMN
416A 12           163Ø  STAX D              /WRITE A 'J' IN OUTBUF
416B 13           164Ø  INX D       /POINT TO NEXT SLOT IN OUTBUF
416C CD 7A 4Ø     165Ø  CALL CONDL   /WRITE REST OF JMP CONDITION
416F C3 3B 42     166Ø  JMP BYT3    /GO FINISH A THREE BYTE INST.
4172 3E 52        167Ø RET MVI A,'R'           /LOAD AN ASCII 'R'
4174 11 6D 44     168Ø  LXI D,OUTBUF+7  /POINT TO MNEMONIC COLUMN
4177 12           169Ø  STAX D        /WRITE 'R' IN OUTPUT BUFFER
4178 13           17ØØ  INX D       /POINT TO NEXT SLOT IN OUTBUF
4179 CD 7A 4Ø     171Ø  CALL CONDL       /WRITE REST OF RET INST.
417C C3 A5 42     172Ø  JMP FINISH    /WRITE ASCII, OCTAL & PRINT
417F 21 2A 44     173Ø LXI LXI H,LLXI       /POINT TO LXI MNEMONIC
4182 3A ED 44     174Ø  LDA SAVE      /RESTORE ORIGINAL DATA BYTE
4185 E6 ØF        175Ø  ANI ØFH             /MASK FOR BITS Ø,1,2,3
4187 3D           176Ø  DCR A                         /WAS BIT Ø SET?
4188 CA AD 41     177Ø  JZ LXIP        /YES, GO LOAD LXI MNEMONIC
418B FE Ø4        178Ø  CPI Ø4              /WAS BYTE LESS THAN 5?
418D DA 92 41     179Ø  JC LX2             /YES, GO JUSTIFY POINTER
419Ø D6 Ø5        18ØØ  SUI Ø5       IF BYTE WAS >5 SUBTRACT BIAS
4192 87           181Ø LX2 ADD A          /DOUBLE MNEMONIC POINTER
4193 87           182Ø  ADD A              /DOUBLE MNEMONIC POINTER
4194 4F           183Ø  MOV C,A /PUT CALCULATED JUSTIFY IN REG. C
4195 Ø9           184Ø  DAD B    /ADD JUSTIFY TO MNEMONIC POINTER
4196 11 6D 44     185Ø  LXI D,OUTBUF+7    /POINT TO MNEMONIC SLOT
4199 CD 66 4Ø     186Ø  CALL PRINT      /WRITE MNEMONIC IN OUTBUF
419C 11 72 44     187Ø  LXI D,OUTBUF+12   /POINT TO LXI REG. SLOT
419F CD 8A 4Ø     188Ø  CALL LXICD  /WRITE REG. OR 'SP' IN OUTBUF
41A2 3A ED 44     189Ø  LDA SAVE      /RESTORE ORIGINAL DATA BYTE
41A5 FE 2Ø        19ØØ  CPI ' '                     /WAS IT A 2ØH?
41A7 CC 39 46     191Ø  CZ INVAL     /YES, GO INVALIDATE MNEMONIC
41AA C3 A5 42     192Ø  JMP FINISH    /WRITE ASCII, OCTAL & PRINT
41AD 11 6D 44     193Ø LXIP LXI D,OUTBU+7 /POINT TO MNEMONIC SLOT
41BØ CD 66 4Ø     194Ø  CALL PRINT      /WRITE MNEMONIC IN OUTBUF
41B3 11 71 44     195Ø  LXI D,OUTBUF+11        /POINT TO REG. SLOT
41B6 CD 8A 4Ø     196Ø  CALL LXICD     /WRITE REG. NAME IN OUTBUF
41B9 3E 2C        197Ø  MVI A,2CH                   /LOAD A COMMA
41BB 13           198Ø  INX D       /POINT TO NEXT SLOT IN OUTBUF
41BC 12           199Ø  STAX D          /WRITE A COMMA IN OUTBUF
```

```
41BD C3 3B 42      2ØØØ  JMP BYT3    /GO WRITE THE LABEL IN OUTBUF
41CØ 21 26 44      2Ø1Ø MVI LXI H,LMVI       /POINT TO MVI MNEMONIC
41C3 11 6D 44      2Ø2Ø  LXI D,OUTBUF+7    /POINT TO MNEMONIC SLOT
41C6 CD 66 4Ø      2Ø3Ø  CALL PRINT      /WRITE MNEMONIC IN OUTBUF
41C9 CD 71 4Ø      2Ø4Ø  CALL MASK    /GET BITS 3,4,5 IN BITS 1,2,3
41CC 11 71 44      2Ø5Ø  LXI D,OUTBUF+11        /POINT TO REG. SLOT
41CF CD 51 4Ø      2Ø6Ø  CALL REGLD      /WRITE REGISTER IN OUTBUF
41D2 3E 2C         2Ø7Ø  MVI A,2CH                   /LOAD A COMMA
41D4 13            2Ø8Ø  INX D       /POINT TO NEXT SLOT IN OUTBUF
41D5 12            2Ø9Ø  STAX D          /WRITE A COMMA IN OUTBUF
41D6 C3 67 42      21ØØ  JMP BYT2      /GO WRITE VALUE IN OUTBUF
41D9 21 22 44      211Ø DCR LXI H,LDCR       /POINT TO DCR MNEMONIC
41DC 11 6D 44      212Ø  LXI D,OUTBUF+7    /POINT TO MNEMONIC SLOT
41DF C3 E8 41      213Ø  JMP INR2     /GO FINISH DCR INSTRUCTION
41E2 21 1E 44      214Ø INR LXI H,LINR   /POINT TO INR INSTRUCTION
41E5 11 6D 44      215Ø  LXI D,OUTBUF+7    /POINT TO MNEMONIC SLOT
41E8 CD 66 4Ø      216Ø INR2 CALL PRINT  /WRITE MNEMONIC IN OUTBUF
41EB CD 71 4Ø      217Ø  CALL MASK    /GET BITS 3,4,5 IN BITS 1,2,3
41EE 11 72 44      218Ø INR3 LXI D,OUTBUF+12   /POINT TO REG. SLOT
41F1 CD 51 4Ø      219Ø  CALL REGLD      /WRITE REGISTER IN OUTBUF
41F4 C3 A5 42      22ØØ  JMP FINISH    /WRITE ASCII, OCTAL & PRINT
41F7 3A ED 44      221Ø ADD LDA SAVE         /RESTORE ORIGINAL BYTE
41FA E6 38         222Ø  ANI 38H                /MASK FOR BITS 3,4,5
41FC ØF            223Ø  RRC                      /CALCULATE OFFSET
41FD 4F            224Ø  MOV C,A               /PUT OFFSET IN REG. C
41FE 21 FE 43      225Ø  LXI H,LADD          /POINT TO ADD MNEMONIC
42Ø1 Ø9            226Ø  DAD B     /ADD OFFSET TO MNEMONIC POINTER
42Ø2 11 6D 44      227Ø  LXI D,OUTBUF+7    /POINT TO MNEMONIC SLOT
42Ø5 CD 66 4Ø      228Ø  CALL PRINT      /WRITE MNEMONIC IN OUTBUF
42Ø8 C3 21 42      229Ø  JMP MOV2     /GO FINISH ADD INSTRUCTION
42ØB 21 FA 43      23ØØ MOV LXI H,LMOV       /POINT TO MOV MNEMONIC
42ØE 11 6D 44      231Ø  LXI D,OUTBUF+7    /POINT TO MNEMONIC SLOT
4211 CD 66 4Ø      232Ø  CALL PRINT      /WRITE MNEMONIC IN OUTBUF
4214 CD 71 4Ø      233Ø  CALL MASK    /GET BITS 3,4,5 IN BITS 1,2,3
4217 11 71 44      234Ø  LXI D,OUTBUF+11   /POINT TO REGISTER SLOT
421A CD 51 4Ø      235Ø  CALL REGLD      /WRITE REGISTER IN OUTBUF
421D 3E 2C         236Ø  MVI A,2CH                   /LOAD A COMMA
421F 13            237Ø  INX D       /POINT TO NEXT SLOT IN OUTBUF
422Ø 12            238Ø  STAX D          /WRITE A COMMA IN OUTBUF
4221 3A ED 44      239Ø MOV2 LDA SAVE        /RESTORE ORIGINAL BYTE
4224 E6 Ø7         24ØØ  ANI Ø7                 /MASK FOR BITS Ø,1,2
4226 13            241Ø  INX D       /POINT TO NEXT SLOT IN OUTBUF
4227 CD 51 4Ø      242Ø  CALL REGLD    /WRITE REG. NAME IN OUTBUF
422A C3 A5 42      243Ø  JMP FINISH    /WRITE ASCII,OCTAL & PRINT
422D 79            244Ø BYTE3 MOV A,C /GET MNEMONIC COUNT IN REG.A
422E 87            245Ø  ADD A                          /CALCULATE-
422F 87            246Ø  ADD A                              /OFFSET
423Ø 4F            247Ø  MOV C,A               /PUT OFFSET IN REG. C
4231 21 DE 43      248Ø  LXI H,NEM3        /POINT TO MNEMONIC TAB 3
4234 Ø9            249Ø  DAD B      /ADD OFFSET TO MNEMONIC TAB 3
```

```
4235 11 6D 44      2500  LXI D,OUTBUF+7    /POINT TO MNEMONIC SLOT
4238 CD 66 40      2510  CALL PRINT      /WRITE MNEMONIC IN OUTBUF
423B CD 48 40      2520 BYT3 CALL FETCH   /GET BITS 3,4,5 IN 1,2,3
423E 32 F3 44      2530  STA CHAR2        /SAVE SECOND ASCII CHAR.
4241 32 ED 44      2540  STA SAVE                /SAVE SECOND BYTE
4244 CD 48 40      2550  CALL FETCH  /GET BITS 3,4,5 IN BITS 1,2,3
4247 32 F4 44      2560  STA CHAR3                /SAVE THIRD BYTE
424A 11 75 44      2578  LXI D,OUTBUF+15      /POINT TO LABEL SLOT
424D CD E7 42      2580  CALL XCODE    /WRITE HIGH ORDER IN OUTBUF
4250 3A ED 44      2590  LDA SAVE             /RESTORE SECOND BYTE
4253 CD E7 42      2600  CALL XCODE     /WRITE LOW ORDER IN OUTBUF
4256 C3 A5 42      2610  JMP FINISH    /WRITE ASCII, OCTAL & PRINT
4259 79            2620 BYTE2 MOV A,C /GET MNEMONIC COUNT IN REG.A
425A 87            2630  ADD A                         /CALCULATE-
425B 87            2640  ADD A                            /OFFSET
425C 4F            2650  MOV C,A             /PUT OFFSET IN REG. C
425D 21 B6 43      2660  LXI H,NEM2       /POINT TO MNEMONIC TAB 2
4260 09            2670  DAD B     /ADD OFFSET TO MNEMONIC POINTER
4261 11 6D 44      2680  LXI D,OUTBUF+7    /POINT TO MNEMONIC SLOT
4264 CD 66 40      2690  CALL PRINT      /WRITE MNEMONIC IN OUTBUF
4267 CD 48 40      2700 BYT2 CALL FETCH   /GET BITS 3,4,5 IN 1,2,3
426A 32 F3 44      2710  STA CHAR2        /SAVE SECOND ASCII CHAR.
426D 11 75 44      2720  LXI D,OUTBUF+15      /POINT TO LABEL SLOT
4270 CD E7 42      2730  CALL XCODE         /WRITE LABEL IN OUTBUF
4273 C3 A5 42      2740  JMP FINISH    /WRITE ASCII, OCTAL & PRINT
4276 79            2750 BYTE1 MOV A,C /GET MNEMONIC COUNT IN REG.A
4277 87            2760  ADD A                         /CALCULATE-
4278 87            2770  ADD A                            /OFFSET
4279 4F            2780  MOV C,A             /PUT OFFSET IN REG. C
427A 21 72 43      2790  LXI H,NEMON      /POINT TO MNEMONIC TABLE
427D 09            2800  DAD B     /ADD OFFSET TO MNEMONIC POINTER
427E 11 6D 44      2810  LXI D,OUTBUF+7    /POINT TO MNEMONIC SLOT
4281 CD 66 40      2820  CALL PRINT      /WRITE MNEMONIC IN OUTBUF
4284 C3 A5 42      2830  JMP FINISH    /WRITE ASCII, OCTAL & PRINT
4287 21 ØC 45      2840 BAD LXI H,DATA    /POINT TO "NON HEX DATA"
428A CD F7 44      2850  CALL CRLF                  /PRINT A CRLF
428D CD 19 45      2860  CALL MSG          /PRINT "NON HEX DATA"
4290 CD F7 44      2870  CALL CRLF                  /PRINT A CRLF
4293 C3 ØØ EC      2880  JMP ØECØØH          /JMP TO SYSTEM MONITOR
4296 E5            2890 HEXOT PUSH H        /SAVE REGISTERS H & L
4297 C5            2900  PUSH B             /SAVE REGISTERS B & C
4298 D5            2910  PUSH D             /SAVE REGISTERS D & E
4299 11 62 44      2920  LXI D,ADDBUF     /POINT TO ADDRESS BUFFER
429C ØE Ø4         2930  MVI C,4        /SET NO. OF CHARS. TO LOAD
429E CD AA 44      2940  CALL XLOAD /LOAD ASCII NUMERICS IN ADDBUF
42A1 D1            2950  POP D           /RESTORE REGISTERS D & E
42A2 C1            2960  POP B           /RESTORE REGISTERS B & C
42A3 E1            2970  POP H           /RESTORE REGISTERS H & L
42A4 C9            2980  RET                         /CONTINUE...
42A5 DB FF         2990 FINISH IN ØFFH    /INPUT THE SENSE SWITCHS
```

```
42A7 E6 Ø1      3ØØØ  ANI 1            /WAS SENSE SWITCH 8 SET?
42A9 CA C7 42   3Ø1Ø  JZ OVER  /YES, SKIP THE ASCII BUFFER LOAD
42AC 11 7E 44   3Ø2Ø  LXI D,OUTBUF+24  /POINT TO 1ST ASCII SLOT
42AF 3A F2 44   3Ø3Ø  LDA CHAR1      /LOAD THE FIRST ASCII BYTE
42B2 CD 11 43   3Ø4Ø  CALL ASCII      /GO WRITE THE ASCII CHAR.
42B5 11 82 44   3Ø5Ø  LXI D,OUTBUF+28  /POINT TO 2ND ASCII SLOT
42B8 3A F3 44   3Ø6Ø  LDA CHAR2     /LOAD THE SECOND ASCII BYTE
42BB CD 11 43   3Ø7Ø  CALL ASCII      /GO WRITE THE ASCII CHAR.
42BE 11 86 44   3Ø8Ø  LXI D,OUTBUF+32  /POINT TO 3RD ASCII SLOT
42C1 3A F4 44   3Ø9Ø  LDA CHAR3      /LOAD THE THIRD ASCII BYTE
42C4 CD 11 43   31ØØ  CALL ASCII      /GO WRITE THE ASCII CHAR.
42C7 CD E1 45   311Ø OVER CALL OCTAL     /GO TRY TO WRITE OCTAL
42CA CD 28 43   312Ø  CALL BUFPNT  /DONE. PRINT THE OUTPUT BUF.
42CD 2A FØ 44   313Ø  LHLD DONE        /LOAD THE ENDING POINTER
42DØ EB         314Ø  XCHG       /PUT END POINTER IN REG. D & E
42D1 CD 47 46   315Ø  CALL FETC2   /LOAD THE CURRENT INST ADDR.
42D4 7C         316Ø  MOV A,H  /PUT LOW ORDER CURRENT IN REG. A
42D5 BA         317Ø  CMP D   /COMPARE LOW ORDER CURRENT TO END
42D6 CA DC 42   318Ø  JZ LOW /IF LOW ORDERS EQUAL, GO TEST HIGH
42D9 C3 55 46   319Ø  JMP PAGE    /GO PROCESS NEXT INSTRUCTION!
42DC 7D         32ØØ LOW MOV A,L   /PUT HIGH ORDER CURRENT IN A
42DD BB         321Ø  CMP E   /COMPARE HIGH CURRENT TO HIGH END
42DE D2 ØØ 4Ø   322Ø  JNC START        /IF LESS THAN BEGIN AGAIN
42E1 CA ØØ 4Ø   323Ø  JZ START          /IF EQUAL TO BEGIN AGAIN
42E4 C3 55 46   324Ø  JMP PAGE     /GO PROCESS NEXT INSTRUCTION
42E7 E5         325Ø XCODE PUSH H          /SAVE REGISTERS H & L
42E8 C5         326Ø  PUSH B                /SAVE REGISTERS B & C
42E9 F5         327Ø  PUSH PSW                  /SAVE A AND FLAGS
42EA ØE Ø2      328Ø  MVI C,2     /SET NUMBER OF CHARS. TO LOAD
42EC 67         329Ø  MOV H,A    /MOVE BYTE TO ENCODE TO REG. A
42ED CD AA 44   33ØØ  CALL XLOAD   /WRITE ASCII BYTES IN OUTBUF
42FØ F1         331Ø  POP PSW                /RESTORE A AND FLAGS
42F1 C1         332Ø  POP B              /RESTORE REGISTERS B & C
42F2 E1         333Ø  POP H              /RESTORE REGISTERS H & L
42F3 C9         334Ø  RET                            /CONTINUE...
42F4 11 44 ØØ   335Ø CLRBUF LXI D,68 /SET NO. OF CHAR. TO CLEAR
42F7 21 66 44   336Ø  LXI H,OUTBUF        /POINT TO OUTPUT BUFFER
42FA 3E 2Ø      337Ø  MVI A,' '                      /LOAD A SPACE
42FC 77         338Ø MORE MOV M,A         /INSERT SPACE IN OUTBUF
42FD 1D         339Ø  DCR E                /DECREMENT CHAR. COUNT
42FE CA Ø5 43   34ØØ  JZ CLRCHAR /IF DONE CLEAR ASCII CHAR. BUF
43Ø1 23         341Ø  INX H      /POINT TO NEXT SLOT IN OUTBUF
43Ø2 C3 FC 42   342Ø  JMP MORE                   /GO CLEAR MORE...
43Ø5 21 F2 44   343Ø CLRCHAR LXI H,CHAR1   /POINT TO ASCII BUF1
43Ø8 ØE Ø3      344Ø  MVI C,3    /SET NUMBER OF CHARS. TO CLEAR
43ØA 77         345Ø MRE2 MOV M,A  /WRITE A SPACE IN ASCII BUF.
43ØB ØD         346Ø  DCR C              /DECREMENT BUFFER COUNT
43ØC C8         347Ø  RZ                     /IF DONE, CONTINUE...
43ØD 23         348Ø  INX H          /POINT TO NEXT ASCII BUFFER
43ØE C3 ØA 43   349Ø  JMP MRE2         /CLEAR NEXT ASCII BUFFER
```

```
4311 FE 7F        35ØØ ASCII CPI 7FH     /TEST IF SIGN BIT IS SET
4313 D4 4D 43     351Ø  CNC SIGN  /SET? GO WRITE "*" BEFORE CHAR.
4316 E6 7F        352Ø  ANI 7FH         /STRIP SIGN BIT FROM CHAR.
4318 FE 2Ø        353Ø  CPI ' '                /TEST FOR ASCII SPACE
431A C8           354Ø  RZ        /IF EQUAL, DON'T BOTHER WITH REST
431B FE ØØ        355Ø  CPI Ø                   /TEST FOR NULL CHAR.
431D C8           356Ø  RZ        /IF EQUAL, DON'T BOTHER WITH REST
431E FE 7F        357Ø  CPI 7FH            /TEST FOR RUB OUT CHAR.
432Ø C8           358Ø  RZ        /IF EQUAL, DON'T BOTHER WITH REST
4321 FE 2Ø        359Ø  CPI ' '                       /TEST FOR  2ØH
4323 DC 43 43     36ØØ  CC CONT /IF LESS THAN 2ØH DO CONTROL CHAR
4326 12           361Ø  STAX D       /WRITE ASCII CHAR. IN OUTBUF
4327 C9           362Ø  RET                           /CONTINUE...
4328 21 A9 44     363Ø BUFPNT LXI H,BUFEND-1 /POINT TO END OF BUF
432B 11 49 ØØ     364Ø  LXI D,73   /SET NO. OF CHAR. IN BUFFER +1
432E 1B           365Ø LESS DCX D             /DECREMENT CHAR. COUNT
432F 7E           366Ø  MOV A,M     /FETCH CHAR. FROM END OF BUF.
433Ø 2B           367Ø  DCX H           /DECREMENT BUFFER POINTER
4331 FE 2Ø        368Ø  CPI ' '         /FOUND A VALID CHAR. YET?
4333 CA 2E 43     369Ø  JZ LESS  /NO, KEEP DECREASING BUF. LENGTH
4336 21 62 44     37ØØ  LXI H,ADDBUF /OTHERWISE POINT BEG OF LINE
4339 4E           371Ø GET MOV C,M   /MOVE CHAR. FROM OUTBUF TO C
433A CD C3 44     372Ø  CALL HISPD  /PRINT CHAR. ON OUTPUT DEVICE
433D 1D           373Ø  DCR E                 /DECREMENT CHAR. COUNT
433E C8           374Ø  RZ        /WHEN DONE WITH LINE, CONTINUE...
433F 23           375Ø  INX H       /POINT TO NEXT CHAR. TO PRINT
434Ø C3 39 43     376Ø  JMP GET                /GO PRINT NEXT CHAR.
4343 C6 4Ø        377Ø CONT ADI '@'/MAKE A PRINTING CHAR. OF CONT
4345 4F           378Ø  MOV C,A            /SAVE THE PRINTING CHAR.
4346 13           379Ø  INX D       /POINT TO NEXT SLOT IN OUTBUF
4347 3E 2E        38ØØ  MVI A,'.' /LOAD A CONTROL CHAR. INDICATOR
4349 12           381Ø  STAX D   /WRITE CONTROL INDICATOR IN BUF.
434A 1B           382Ø  DCX D      /POINT TO ASCII SLOT IN OUTBUF
434B 79           383Ø  MOV A,C            /RESTORE PRINTING CHAR.
434C C9           384Ø  RET                           /CONTINUE...
434D 4F           385Ø SIGN MOV C,A                /SAVE ASCII CHAR.
434E 3E 2A        386Ø  MVI A,'*'      /LOAD A SIGN BIT INDICATOR
435Ø 1B           387Ø  DCX D         /POINT AHEAD OF ASCII CHAR.
4351 12           388Ø  STAX D    /WRITE SIGN INDICATOR IN OUTBUF
4352 13           389Ø  INX D     /RESTORE ASCII POINTER TO CHAR.
4353 79           39ØØ  MOV A,C                /RESTORE ASCII CHAR.
4354 C9           391Ø  RET                           /CONTINUE...
4355 ØØ           392Ø OPCODES NOP  /THE FOLOWING IS A        /NOP
4356 Ø7           393Ø  DB ØØ7H     /IS A TABLE OF 8Ø8Ø       /RLC
4357 ØF           394Ø  DB ØØFH     /OPCODES  USED  BY-       /RRC
4358 17           395Ø  DB Ø17H     /THIS  PROGRAM  TO-       /RAL
4359 1F           396Ø  DB Ø1FH     /COMPARE   AGAINST-       /RAR
435A 27           397Ø  DB Ø27H     /THE BYTE  FETCHED-       /DAA
435B 2F           398Ø  DB Ø2FH     /FROM MEMORY              /CMA
435C 37           399Ø  DB Ø37H                               /STC
```

```
435D 3F        4ØØØ  DB Ø3FH                                   /CMC
435E 76        4Ø1Ø  DB Ø76H                                   /HLT
435F C9        4Ø2Ø  DB ØC9H                                   /RET
436Ø E3        4Ø3Ø  DB ØE3H                                   /XTHL
4361 E9        4Ø4Ø  DB ØE9H                                   /PCHL
4362 EB        4Ø5Ø  DB ØEBH                                   /XCHG
4363 F3        4Ø6Ø  DB ØF3H                                   /DI
4364 F9        4Ø7Ø  DB ØF9H                                   /SPHL
4365 FB        4Ø8Ø  DB ØFBH                                   /EI
4366 C6        4Ø9Ø  DB ØC6H          /HERE STARTS THE-        /ADI
4367 CE        41ØØ  DB ØCEH          /TWO BYTE OPCODES        /ACI
4368 D3        411Ø  DB ØD3H                                   /OUT
4369 D6        412Ø  DB ØD6H                                   /SUI
436A DB        413Ø  DB ØDBH                                   /IN
436B DE        414Ø  DB ØDEH                                   /SBI
436C E6        415Ø  DB ØE6H                                   /ANI
436D EE        416Ø  DB ØEEH                                   /XRI
436E F6        417Ø  DB ØF6H                                   /ORI
436F FE        418Ø  DB ØFEH                                   /CPI
437Ø 22        419Ø  DB Ø22H          /HERE STARTS THE-        /SHLD
4371 2A        42ØØ  DB Ø2AH          /THREE BYTE OPCODES      /LHLD
4372 32        421Ø NEMON DB 32H                               /STA
4373 3A        422Ø  DB Ø3AH                                   /LDA
4374 C3        423Ø  DB ØC3H                                   /JMP
4375 CD        424Ø  DB ØCDH                                   /CALL
4376 45 49     425Ø  DW 'IE'     /THE FOLOWING ARE TABLES OF-
4378 2Ø 2Ø     426Ø  DW '  '    /ASCII MNEMONICS USED BY THE-
437A 53 5Ø     427Ø  DW 'PS'        /DISASSEMBLER TO LOAD THE-
437C 48 4C     428Ø  DW 'LH'           /OUTPUT BUFFER WITH THE-
437E 44 49     429Ø  DW 'ID'              /INSTRUCTION MNEMONICS
438Ø 2Ø 2Ø     43ØØ  DW '  '
4382 58 43     431Ø  DW 'CX'
4384 48 47     432Ø  DW 'GH'
4386 5Ø 43     433Ø  DW 'CP'
4388 48 4C     434Ø  DW 'LH'
438A 58 54     435Ø  DW 'TX'
438C 48 4C     436Ø  DW 'LH'
438E 52 45     437Ø  DW 'ER'
439Ø 54 2Ø     438Ø  DW ' T'
4392 48 4C     439Ø  DW 'LH'
4394 54 2Ø     44ØØ  DW ' T'
4396 43 4D     441Ø  DW 'MC'
4398 43 2Ø     442Ø  DW ' C'
439A 53 54     443Ø  DW 'TS'
439C 43 2Ø     444Ø  DW ' C'
439E 43 4D     445Ø  DW 'MC'
43AØ 41 2Ø     446Ø  DW ' A'
43A2 44 41     447Ø  DW 'AD'
43A4 41 2Ø     448Ø  DW ' A'
43A6 52 41     449Ø  DW 'AR'
```

```
43A8  52  2Ø         45ØØ  DW ' R'
43AA  52  41         451Ø  DW 'AR'
43AC  4C  2Ø         452Ø  DW ' L'
43AE  52  52         453Ø  DW 'RR'
43BØ  43  2Ø         454Ø  DW ' C'
43B2  52  4C         455Ø  DW 'LR'
43B4  43  2Ø         456Ø  DW ' C'
43B6  4E  4F         457Ø NEM2 DW 'ON'
43B8  5Ø  2Ø         458Ø  DW ' P'
43BA  43             459Ø  DB 43H                                  /C
43BB  5Ø             46ØØ  DB 5ØH                                  /P
43BC  49             461Ø  DB 49H                                  /I
43BD  2Ø             462Ø  DB 2ØH
43BE  4F             463Ø  DB 4FH                                  /O
43BF  52             464Ø  DB 52H                                  /R
43CØ  49             465Ø  DB 49H                                  /I
43C1  2Ø             466Ø  DB 2ØH
43C2  58             467Ø  DB 58H                                  /X
43C3  52             468Ø  DB 52H                                  /R
43C4  49             469Ø  DB 49H                                  /I
43C5  2Ø             47ØØ  DB 2ØH
43C6  41             471Ø  DB 41H                                  /A
43C7  4E             472Ø  DB 4EH                                  /N
43C8  49             473Ø  DB 49H                                  /I
43C9  2Ø             474Ø  DB 2ØH
43CA  53             475Ø  DB 53H                                  /S
43CB  42             476Ø  DB 42H                                  /B
43CC  49             477Ø  DB 49H                                  /I
43CD  2Ø             478Ø  DB 2ØH
43CE  49             479Ø  DB 49H                                  /I
43CF  4E             48ØØ  DB 4EH                                  /N
43DØ  2Ø             481Ø  DB 2ØH
43D1  2Ø             482Ø  DB 2ØH
43D2  53             483Ø  DB 53H                                  /S
43D3  55             484Ø  DB 55H                                  /U
43D4  49             485Ø  DB 49H                                  /I
43D5  2Ø             486Ø  DB 2ØH
43D6  4F             487Ø  DB 4FH                                  /O
43D7  55             488Ø  DB 55H                                  /U
43D8  54             489Ø  DB 54H                                  /T
43D9  2Ø             49ØØ  DB 2ØH
43DA  41             491Ø  DB 41H                                  /A
43DB  43             492Ø  DB 43H                                  /C
43DC  49             493Ø  DB 49H                                  /I
43DD  2Ø             494Ø  DB 2ØH
43DE  41             495Ø NEM3 DB 41H                              /A
43DF  44             496Ø  DB 44H                                  /D
43EØ  49             497Ø  DB 49H                                  /I
43E1  2Ø             498Ø  DB 2ØH
43E2  43             499Ø  DB 43H                                  /C
```

```
43E3 41          5ØØØ  DB 41H                       /A
43E4 4C          5Ø1Ø  DB 4CH                       /L
43E5 4C          5Ø2Ø  DB 4CH                       /L
43E6 4A          5Ø3Ø  DB 4AH                       /J
43E7 4D          5Ø4Ø  DB 4DH                       /M
43E8 5Ø          5Ø5Ø  DB 5ØH                       /P
43E9 2Ø          5Ø6Ø  DB 2ØH
43EA 4C          5Ø7Ø  DB 4CH                       /L
43EB 44          5Ø8Ø  DB 44H                       /D
43EC 41          5Ø9Ø  DB 41H                       /A
43ED 2Ø          51ØØ  DB 2ØH
43EE 53          511Ø  DB 53H                       /S
43EF 54          512Ø  DB 54H                       /T
43FØ 41          513Ø  DB 41H                       /A
43F1 2Ø          514Ø  DB 2ØH
43F2 4C          515Ø  DB 4CH                       /L
43F3 48          516Ø  DB 48H                       /H
43F4 4C          517Ø  DB 4CH                       /L
43F5 44          518Ø  DB 44H                       /D
43F6 53          519Ø  DB 53H                       /S
43F7 48          52ØØ  DB 48H                       /H
43F8 4C          521Ø  DB 4CH                       /L
43F9 44          522Ø  DB 44H                       /D
43FA 4D          523Ø LMOV DB 4DH                   /M
43FB 4F          524Ø  DB 4FH                       /O
43FC 56          525Ø  DB 56H                       /V
43FD 2Ø          526Ø  DB 2ØH
43FE 41          527Ø LADD DB 41H                   /A
43FF 44          528Ø  DB 44H                       /D
44ØØ 44          529Ø  DB 44H                       /D
44Ø1 2Ø          53ØØ  DB 2ØH
44Ø2 41          531Ø  DB 41H                       /A
44Ø3 44          532Ø  DB 44H                       /D
44Ø4 43          533Ø  DB 43H                       /C
44Ø5 2Ø          534Ø  DB 2ØH
44Ø6 53          535Ø  DB 53H                       /S
44Ø7 55          536Ø  DB 55H                       /U
44Ø8 42          537Ø  DB 42H                       /B
44Ø9 2Ø          538Ø  DB 2ØH
44ØA 53          539Ø  DB 53H                       /S
44ØB 42          54ØØ  DB 42H                       /B
44ØC 42          541Ø  DB 42H                       /B
44ØD 2Ø          542Ø  DB 2ØH
44ØE 41          543Ø  DB 41H                       /A
44ØF 4E          544Ø  DB 4EH                       /N
441Ø 41          545Ø  DB 41H                       /A
4411 2Ø          546Ø  DB 2ØH
4412 58          547Ø  DB 58H                       /X
4413 52          548Ø  DB 52H                       /R
4414 41          549Ø  DB 41H                       /A
```

```
4415 2Ø          55ØØ  DB 2ØH
4416 4F          551Ø  DB 4FH                         /O
4417 52          552Ø  DB 52H                         /R
4418 41          553Ø  DB 41H                         /A
4419 2Ø          554Ø  DB 2ØH
441A 43          555Ø  DB 43H                         /C
441B 4D          556Ø  DB 4DH                         /M
441C 5Ø          557Ø  DB 5ØH                         /P
441D 2Ø          558Ø  DB 2ØH
441E 49          559Ø LINR DB 49H                     /I
441F 4E          56ØØ  DB 4EH                         /N
442Ø 52          561Ø  DB 52H                         /R
4421 2Ø          562Ø  DB 2ØH
4422 44          563Ø LDCR DB 44H                     /D
4423 43          564Ø  DB 43H                         /C
4424 52          565Ø  DB 52H                         /R
4425 2Ø          566Ø  DB 2ØH
4426 4D          567Ø LMVI DB 4DH                     /M
4427 56          568Ø  DB 56H                         /V
4428 49          569Ø  DB 49H                         /I
4429 2Ø          57ØØ  DB 2ØH
442A 4C          571Ø LLXI DB 4CH                     /L
442B 58          572Ø  DB 58H                         /X
442C 49          573Ø  DB 49H                         /I
442D 2Ø          574Ø  DB 2ØH
442E 53          575Ø  DB 53H                         /S
442F 54          576Ø  DB 54H                         /T
443Ø 41          577Ø  DB 41H                         /A
4431 58          578Ø  DB 58H                         /X
4432 49          579Ø  DB 49H                         /I
4433 4E          58ØØ  DB 4EH                         /N
4434 58          581Ø  DB 58H                         /X
4435 2Ø          582Ø  DB 2ØH
4436 44          583Ø  DB 44H                         /D
4437 41          584Ø  DB 41H                         /A
4438 44          585Ø  DB 44H                         /D
4439 2Ø          586Ø  DB 2ØH
443A 4C          587Ø  DB 4CH                         /L
443B 44          588Ø  DB 44H                         /D
443C 41          589Ø  DB 41H                         /A
443D 58          59ØØ  DB 58H                         /X
443E 44          591Ø  DB 44H                         /D
443F 43          592Ø  DB 43H                         /C
444Ø 58          593Ø  DB 58H                         /X
4441 2Ø          594Ø  DB 2ØH
4442 52          595Ø LRST DB 52H                     /R
4443 53          596Ø  DB 53H                         /S
4444 54          597Ø  DB 54H                         /T
4445 2Ø          598Ø  DB 2ØH
4446 5Ø          599Ø LPSW DB 5ØH                     /P
```

```
4447 53          6ØØØ  DB 53H                                  /S
4448 57          6Ø1Ø  DB 57H                                  /W
4449 2Ø          6Ø2Ø LPOP DB 2ØH
444A 5Ø          6Ø3Ø  DB 5ØH                                  /P
444B 4F          6Ø4Ø  DB 4FH                                  /O
444C 5Ø          6Ø5Ø  DB 5ØH                                  /P
444D 2Ø          6Ø6Ø  DB 2ØH
444E 5Ø          6Ø7Ø  DB 5ØH                                  /P
444F 55          6Ø8Ø  DB 55H                                  /U
445Ø 53          6Ø9Ø  DB 53H                                  /S
4451 48          61ØØ  DB 48H                                  /H
4452 4E          611Ø CONDN DB 4EH    /THIS IS A TABLE-        /N
4453 5A          612Ø  DB 5AH         /OF ENDINGS FOR-         /Z
4454 5A          613Ø  DB 5AH         /CONDITIONAL CALLS-      /Z
4455 2Ø          614Ø  DB 2ØH         /JUMPS AND RET'S.
4456 4E          615Ø  DB 4EH                                  /N
4457 43          616Ø  DB 43H                                  /C
4458 43          617Ø  DB 43H                                  /C
4459 2Ø          618Ø  DB 2ØH
445A 5Ø          619Ø  DB 5ØH                                  /P
445B 4F          62ØØ  DB 4FH                                  /O
445C 5Ø          621Ø  DB 5ØH                                  /P
445D 45          622Ø  DB 45H                                  /E
445E 5Ø          623Ø  DB 5ØH                                  /P
445F 2Ø          624Ø  DB 2ØH
446Ø 4D          625Ø  DB 4DH                                  /M
4461 2Ø          626Ø  DB 2ØH
4462 ØØ ØØ       627Ø ADDBUF DW Ø      /THIS BUFFER STORES THE-
4464 ØØ ØØ       628Ø  DW Ø          /HEX ADDRESS FOR PRINTING.
4466 ØØ ØØ       629Ø OUTBUF DW Ø      /HERE BEGINS THE OUTPUT-
4468 ØØ ØØ       63ØØ  DW Ø             /BUFFER WHICH HOLDS THE-
446A ØØ ØØ       631Ø  DW Ø         /ENTIRE LINE OF DATA TO BE-
446C ØØ ØØ       632Ø  DW Ø       /PRINTED.  AFTER THIS BUFFER-
446E ØØ ØØ       633Ø  DW Ø          /IS LOADED WITH ASCII DATA-
447Ø ØØ ØØ       634Ø  DW Ø         /THE ROUTINE CALLED BUFPNT-
4472 ØØ ØØ       635Ø  DW Ø        /POINTS TO THE BEGINNING OF-
4474 ØØ ØØ       636Ø  DW Ø     /ADDBUF AND OUTPUTS THE CHAR'S-
4476 ØØ ØØ       637Ø  DW Ø     /FROM THERE TO BUFEND-1.  THAT-
4478 ØØ ØØ       638Ø  DW Ø       /ROUTINE ALSO CALCULATES THE-
447A ØØ ØØ       639Ø  DW Ø      /ACTUAL LENGTH OF THIS BUFFER-
447C ØØ ØØ       64ØØ  DW Ø       /EACH TIME A LINE IS PRINTED.
447E ØØ ØØ       641Ø  DW Ø
448Ø ØØ ØØ       642Ø  DW Ø
4482 ØØ ØØ       643Ø  DW Ø
4484 ØØ ØØ       644Ø  DW Ø
4486 ØØ ØØ       645Ø  DW Ø
4488 ØØ ØØ       646Ø  DW Ø
448A ØØ ØØ       647Ø  DW Ø
448C ØØ ØØ       648Ø  DW Ø
448E ØØ ØØ       649Ø  DW Ø
```

```
4490 ØØ ØØ         65ØØ  DW Ø
4492 ØØ ØØ         651Ø  DW Ø
4494 ØØ ØØ         652Ø  DW Ø
4496 ØØ ØØ         653Ø  DW Ø
4498 ØØ ØØ         654Ø  DW Ø
449A ØØ ØØ         655Ø  DW Ø
449C ØØ ØØ         656Ø  DW Ø
449E ØØ ØØ         657Ø  DW Ø
44AØ ØØ ØØ         658Ø  DW Ø
44A2 ØØ ØØ         659Ø  DW Ø
44A4 ØØ ØØ         66ØØ  DW Ø
44A6 ØØ ØØ         661Ø  DW Ø
44A8 ØØ ØØ         662Ø  DW Ø
44AA               663Ø BUFEND EQU $
44AA AF            664Ø XLOAD XRA A        /THIS ROUTINE LOADS THE-
44AB 29            665Ø  DAD H             /OUTPUT BUFFER WITH HEX-
44AC 17            666Ø  RAL              /CHAR'S.  UPON ENTRY REG.-
44AD 29            667Ø  DAD H            /C CONTAINS THE NUMBER OF-
44AE 17            668Ø  RAL               /CHARACTERS TO BE LOADED.
44AF 29            669Ø  DAD H              /UPON ENTRY REG. D & E -
44BØ 17            67ØØ  RAL              /POINT TO THE SLOT IN THE-
44B1 29            671Ø  DAD H           /OUTPUT BUFFER WHICH IS TO-
44B2 17            672Ø  RAL             /BE LOADED WITH HEX CHAR'S.
44B3 FE ØA         673Ø  CPI 1Ø      /THE ROUTINE TERMINATES WHEN-
44B5 DA BA 44      674Ø  JC ASCOUT               /REG. C REACHES Ø.
44B8 C6 Ø7         675Ø  ADI 7
44BA C6 3Ø         676Ø ASCOUT ADI 'Ø'
44BC 12            677Ø  STAX D
44BD 13            678Ø  INX D
44BE ØD            679Ø  DCR C
44BF C2 AA 44      68ØØ  JNZ XLOAD
44C2 C9            681Ø  RET
44C3 DB ØØ         682Ø HISPD IN Ø           /INPUT THE STATUS PORT
44C5 E6 8Ø         683Ø  ANI 8ØH  /TEST FOR TRANSMITTER BUF. EMPTY
44C7 CA C3 44      684Ø  JZ HISPD          /IF NOT READY, TRY AGAIN
44CA 79            685Ø  MOV A,C   /GET THE CHAR. TO PRINT IN ACC.
44CB D3 Ø1         686Ø  OUT 1          /OUTPUT IT TO THE DATA PORT
44CD DB ØØ         687Ø  IN Ø                /INPUT THE STATUS PORT
44CF E6 4Ø         688Ø  ANI 4ØH      /IS DATA AVAILABLE FLAG SET?
44D1 C2 ØØ 4Ø      689Ø  JNZ START    /YES, THEN STOP & START OVER
44D4 C9            69ØØ  RET                /OTHERWISE CONTINUE...
44D5 DB ØØ         691Ø INPUT IN Ø           /INPUT THE STATUS PORT
44D7 E6 4Ø         692Ø  ANI 4ØH                 /TEST FOR DAV FLAG
44D9 CA D5 44      693Ø  JZ INPUT              /NOT READY? TRY AGAIN
44DC DB Ø1         694Ø  IN 1                   /INPUT THE DATA PORT
44DE E6 7F         695Ø  ANI 7FH                     /STRIP PARITY
44EØ 4F            696Ø  MOV C,A              /SAVE DATA IN REG. C
44E1 C9            697Ø  RET                          /CONTINUE...
44E2 DB ØØ         698Ø CPRINT IN Ø          /INPUT THE STATUS PORT
44E4 E6 8Ø         699Ø  ANI 8ØH  /TEST FOR TRANSMITTER BUF. EMPTY
```

```
44E6 CA E2 44    7ØØØ  JZ CPRINT       /NOT READY? THEN TRY AGAIN
44E9 79          7Ø1Ø  MOV A,C        /GET CHAR TO PRINT IN ACC.
44EA D3 Ø1       7Ø2Ø  OUT 1                     /PRINT CHARACTER
44EC C9          7Ø3Ø  RET                           /CONTINUE...
44ED ØØ          7Ø4Ø SAVE DB Ø             /FETCHED BYTE STORAGE
44EE ØØ ØØ       7Ø5Ø PGMCT DW Ø        /PROGRAM COUNTER STORAGE
44FØ ØØ ØØ       7Ø6Ø DONE DW Ø  /ENDING PROGRAM COUNTER STORAGE
44F2 ØØ          7Ø7Ø CHAR1 DB Ø      /FIRST BYTE, ASCII STORAGE
44F3 ØØ          7Ø8Ø CHAR2 DB Ø     /SECOND BYTE, ASCII STORAGE
44F4 ØØ          7Ø9Ø CHAR3 DB Ø      /THRID BYTE, ASCII STORAGE
44F5 ØØ ØØ       71ØØ OSAVE DW Ø   /CURRENT INSTRUCTION ADDRESS
44F7 ØE ØD       711Ø CRLF MVI C,ØDH     /LOAD A CARRIAGE RETURN
44F9 CD C3 44    712Ø  CALL HISPD                     /PRINT A CR
44FC ØE ØA       713Ø  MVI C,ØAH                /LOAD A LINE FEED
44FE CD C3 44    714Ø  CALL HISPD              /PRINT A LINE FEED
45Ø1 ØE ØØ       715Ø  MVI C,Ø                      /LOAD A NULL
45Ø3 CD C3 44    716Ø  CALL HISPD                   /PRINT A NULL
45Ø6 CD C3 44    717Ø  CALL HISPD                   /PRINT A NULL
45Ø9 C3 C3 44    718Ø  JMP HISPD   /PRINT A NULL AND CONTINUE...
45ØC 4E 4F       719Ø DATA DW 'ON'           /ASCII STORAGE FOR-
45ØE 4E 2Ø       72ØØ  DW ' N'                  /"NON HEX DATA"
451Ø 48 45       721Ø  DW 'EH'
4512 58 2Ø       722Ø  DW ' X'
4514 44 41       723Ø  DW 'AD'
4516 54 41       724Ø  DW 'AT'
4518 ØØ          725Ø  DB Ø                  /MESSAGE TERMINATOR
4519 4E          726Ø MSG MOV C,M /FETCH A STORED CHAR. FROM MEM
451A CD C3 44    727Ø  CALL HISPD           /PRINT THAT CHARACTER
451D 79          728Ø  MOV A,C            /RESTORE THE CHARACTER
451E FE ØØ       729Ø  CPI Ø       /WAS IT A MESSAGE TERMINATOR?
452Ø C8          73ØØ  RZ          /YES, THEN QUIT AND CONTINUE...
4521 23          731Ø  INX H           /POINT TO NEXT STORED CHAR.
4522 C3 19 45    732Ø  JMP MSG                        /PRINT MORE
4525 53 54       733Ø STADD DW 'TS'          /ASCII STORAGE FOR-
4527 41 52       734Ø  DW 'RA'                 /"START ADDRESS?"
4529 54 2Ø       735Ø  DW ' T'
452B 41 44       736Ø  DW 'DA'
452D 44 52       737Ø  DW 'RD'
452F 45 53       738Ø  DW 'SE'
4531 53 3F       739Ø  DW '?S'
4533 2Ø ØØ       74ØØ  DW ØØ2ØH    /SPACE AND MESSAGE TERMINATOR
4535 45 4E       741Ø ENADD DW 'NE'          /ASCII STORAGE FOR-
4537 44 2Ø       742Ø  DW ' D'                    /"END ADDRESS?"
4539 41 44       743Ø  DW 'DA'
453B 44 52       744Ø  DW 'RD'
453D 45 53       745Ø  DW 'SE'
453F 53 3F       746Ø  DW '?S'
4541 2Ø ØØ       747Ø  DW ØØ2ØH    /SPACE AND MESSAGE TERMINATOR
4543 48 45       748Ø LNONE DW 'EH'          /ASCII STORAGE FOR-
4545 58 2Ø       749Ø  DW ' X'        /LINE 1 OF HEADER MESSAGE
```

```
4547 2Ø 2Ø     75ØØ  DW '  '
4549 48 45     751Ø  DW 'EH'
454B 58 2Ø     752Ø  DW ' X'
454D 2Ø 2Ø     753Ø  DW '  '
454F 2Ø 2Ø     754Ø  DW '  '
4551 2Ø 2Ø     755Ø  DW '  '
4553 2Ø 2Ø     756Ø  DW '  '
4555 2Ø 4C     757Ø  DW 'L '
4557 41 42     758Ø  DW 'BA'
4559 4C 45     759Ø  DW 'EL'
455B 2Ø 2Ø     76ØØ  DW '  '
455D 2Ø 2Ø     761Ø  DW '  '
455F 41 53     762Ø  DW 'SA'
4561 43 49     763Ø  DW 'IC'
4563 49 2Ø     764Ø  DW ' I'
4565 2Ø 2Ø     765Ø  DW '  '
4567 2Ø 2Ø     766Ø  DW '  '
4569 2Ø 2Ø     767Ø  DW '  '
456B 4F 43     768Ø  DW 'CO'
456D 54 41     769Ø  DW 'AT'
456F 4C 2Ø     77ØØ  DW ' L'
4571 2Ø 2Ø     771Ø  DW '  '
4573 2Ø 44     772Ø  DW 'D '
4575 41 54     773Ø  DW 'TA'
4577 41 ØØ     774Ø  DW ØØ41H
4579 41 44     775Ø LNTWO DW 'DA'         /ASCII STORAGE FOR-
457B 44 52     776Ø  DW 'RD'        /LINE 2 OF HEADER MESSAGE
457D 2Ø        777Ø  DB ' '
457E 2Ø 44     778Ø  DW 'D '
458Ø 41 54     779Ø  DW 'TA'
4582 41 2Ø     78ØØ  DW ' A'
4584 49 4E     781Ø  DW 'NI'
4586 53 54     782Ø  DW 'TS'
4588 2Ø 2Ø     783Ø  DW '  '
458A 2Ø 2Ø     784Ø  DW '  '
458C 48 49     785Ø  DW 'IH'
458E 4C 4F     786Ø  DW 'OL'
459Ø 2Ø 2Ø     787Ø  DW '  '
4592 2Ø 2Ø     788Ø  DW '  '
4594 2Ø 31     789Ø  DW '1 '
4596 2Ø 2Ø     79ØØ  DW '  '
4598 2Ø 32     791Ø  DW '2 '
459A 2Ø 2Ø     792Ø  DW '  '
459C 2Ø 33     793Ø  DW '3 '
459E 2Ø 2Ø     794Ø  DW '  '
45AØ 2Ø 41     795Ø  DW 'A '
45A2 44 44     796Ø  DW 'DD'
45A4 52 45     797Ø  DW 'ER'
45A6 53 53     798Ø  DW 'SS'
45A8 2Ø 2Ø     799Ø  DW '  '
```

```
4664 Ø6 Ø9         9ØØØ SKIP MVI B,9     /LOAD NO. OF CR'S TO PRINT
4666 CD 7C 46      9Ø1Ø  CALL SK3                  /PRINT 9 CRLF'S
4669 ØE 2D         9Ø2Ø SET2 MVI C,'-'    /LOAD PAGE DIVIDER CHAR.
466B Ø6 48         9Ø3Ø  MVI B,72             /LOAD TERMINAL WIDTH
466D CD C3 44      9Ø4Ø SETA CALL HISPD                /PRINT A '-'
467Ø Ø5            9Ø5Ø  DCR B      /DECREMENT TERMINAL WIDTH COUNT
4671 C2 6D 46      9Ø6Ø  JNZ SETA        /NOT DONE THEN PRINT MORE
4674 CD 7A 46      9Ø7Ø SET3 CALL SK2              /PRINT 7 CRLF'S
4677 C3 2D 4Ø      9Ø8Ø  JMP HEADR           /PRINT HEADER MESSAGE
467A Ø6 Ø7         9Ø9Ø SK2 MVI B,7   /LOAD NO. OF CRLF'S TO PRINT
467C CD F7 44      91ØØ SK3 CALL CRLF                /PRINT A CRLF
467F Ø5            911Ø  DCR B               /DECREMENT CRLF COUNT
468Ø C8            912Ø  RZ                   /IF DONE CONTINUE...
4681 C3 7C 46      913Ø  JMP SK3               /PRINT MORE CRLF'S
4684 22 FØ 44      914Ø STORE SHLD DONE    /SAVE ENDING PGM. COUNT
4687 CD F7 44      915Ø  CALL CRLF                   /PRINT A CRLF
468A C3 69 46      916Ø  JMP SET2       /GO PRINT SPACE AND HEADER
468D               917Ø  DS 3ØH            /ALLOW SPACE FOR STACK
46BD               918Ø STACK EQU $        /STACK POINTER SET HERE
46BD               919Ø END EQU $           /MODIFICATIONS GO HERE
```

```
SYMBOL TABLE


START 4ØØØ    HEADR 4Ø2D    FETCH 4Ø48    REGLD 4Ø51    REL1  4Ø5B
REL2  4Ø62    PRINT 4Ø66    SPRN  4Ø68    MASK  4Ø71    CONDL 4Ø7A
LXICD 4Ø8A    DISAS 4Ø9C    ONE   4ØC1    TWO   4ØCC    THREE 4ØD7
RST   4143    CALL  4158    JMP   4165    RET   4172    LXI   417F
LX2   4192    LXIP  41AD    MVI   41CØ    DCR   41D9    INR   41E2
INR2  41E8    INR3  41EE    ADD   41F7    MOV   42ØB    MOV2  4221
BYTE3 422D    BYT3  423B    BYTE2 4259    BYT2  4267    BYTE1 4276
BAD   4287    HEXOT 4296    FINIS 42A5    OVER  42C7    LOW   42DC
XCODE 42E7    CLRBU 42F4    MORE  42FC    CLRCH 43Ø5    MRE2  43ØA
ASCII 4311    BUFPN 4328    LESS  432E    GET   4339    CONT  4343
SIGN  434D    OPCOD 4355    NEMON 4372    NEM2  43B6    NEM3  43DE
LMOV  43FA    LADD  43FE    LINR  441E    LDCR  4422    LMVI  4426
LLXI  442A    LRST  4442    LPSW  4446    LPOP  4449    CONDN 4452
ADDBU 4462    OUTBU 4466    BUFEN 44AA    XLOAD 44AA    ASCOU 44BA
HISPD 44C3    INPUT 44D5    CPRIN 44E2    SAVE  44ED    PGMCT 44EE
DONE  44FØ    CHAR1 44F2    CHAR2 44F3    CHAR3 44F4    OSAVE 44F5
CRLF  44F7    DATA  45ØC    MSG   4519    STADD 4525    ENADD 4535
LNONE 4543    LNTWO 4579    HEX   45B4    NXT   45B7    NUM   45CA
REP   45D9    OCTAL 45E1    OT3   4614    OCTCO 4621    ONXT  4628
ROT   4629    INVAL 4639    INV2  464Ø    FETC2 4647    SET   464C
COUNT 4654    PAGE  4655    SKIP  4664    SET2  4669    SETA  466D
SET3  4674    SK2   467A    SK3   467C    STORE 4684    STACK 46BD
END   46BD
```