
PLEIADES MICROWARE

--SORCERER--

A DISASSEMBLER PROGRAM FOR 8080 BASED MICROCOMPUTERS

COPYRIGHT © BY GEORGE GILBERT; PLEIADES MICROWARE

1977

*****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 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, ERASED, OR HAVE HAD OTHER PROGRAMS RECORDED ON THEM.

FOR THE OPERATORS CONVENIENCE, PAGES OF THE SOURCE LISTING HAVE BEEN NUMBERED 1 THRU 20 AND PAGES OF THE OPERATORS MANUAL HAVE BEEN NUMBERED 01 THRU 09. 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 03 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 HOBBIEIST 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 4000H 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 4000H LISTING WITH A "2" FOR THE (2)000H VERSION OR A "0" FOR THE (0)000H VERSION. ONE OF THE FIRST USES OF SORCERER, SHOULD BE TO TURN THE DISASSEMBLER ON ITSELF TO PRODUCE USEFUL LISTINGS OF THE 0000H AND 2000H 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 SUITABILITY 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.

PAGE 03

1 START SYNC STREAM

9 STOP " "

14 START TONE

15½ START DATA 1

17 TONE RETURNS

20 START DATA 2

24 TONE RETURNS

26 START DATA 3

30 TONE RETURNS

36 END

MACRODIS
CASSETTE

1 START TONE

3½ START 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 IMMEDIATELY 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. 6820, PATCH ADDRESS 44C4, TO YOUR SYSTEMS
OUTPUT STATUS PORT.

AT LINE NO. 6830, PATCH ADDRESS 44C6, TO YOUR SYSTEMS
OUTPUT TRANSMITTER
BUFFER EMPTY FLAG.

AT LINE NO. 6840, 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. 6860, PATCH ADDRESS 44CC, TO YOUR SYSTEMS
OUTPUT DATA PORT.

AT LINE NO. 6870, PATCH ADDRESS 44CE, TO YOUR SYSTEMS
INPUT STATUS PORT.

AT LINE NO. 6880, PATCH ADDRESS 44D0, TO YOUR SYSTEMS
INPUT DATA AVAILABLE
FLAG.

AT LINE NO. 6890, 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. 6910, PATCH ADDRESS 44D6, TO YOUR SYSTEMS
INPUT STATUS PORT

AT LINE NO. 6920, PATCH ADDRESS 44D8, TO YOUR SYSTEMS
INPUT DATA AVAILABLE
FLAG.

AT LINE NO. 6930, 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. 6940, 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 CHARACTER (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
4000	31	LXI SP,	46BD	1	==	F	100 000	061	275	106
4003	DB	IN	01	*[	A.		100 003	333	001	
4005	CD	CALL	44F7	*M	*W	D	100 005	315	367	104
4008	21	LXI H,	4525	!	2	E	100 010	041	045	105
400B	CD	CALL	4519	*M	Y.	E	100 013	315	031	105
400E	CD	CALL	45B4	*M	*4	E	100 016	315	264	105
4011	FE	CPI	0D	*†	M.		100 021	376	015	
4013	C2	JNZ	4287	*B	*G.	B	100 023	302	207	102
4016	22	SHLD	44EE	"	*N	D	100 026	042	356	104
4019	CD	CALL	44F7	*M	*W	D	100 031	315	367	104
401C	21	LXI H,	4535	!	5	E	100 034	041	065	105
401F	CD	CALL	4519	*M	Y.	E	100 037	315	031	105
4022	CD	CALL	45B4	*M	*4	E	100 042	315	264	105
4025	FE	CPI	0D	*†	M.		100 045	376	015	
4027	C2	JNZ	4287	*B	*G.	B	100 047	302	207	102
402A	C3	JMP	4684	*C	*D.	F	100 052	303	204	106
402D	21	LXI H,	4543	!	C	E	100 055	041	103	105
4030	CD	CALL	44F7	*M	*W	D	100 060	315	367	104
4033	CD	CALL	44F7	*M	*W	D	100 063	315	367	104
4036	CD	CALL	4519	*M	Y.	E	100 066	315	031	105
4039	CD	CALL	44F7	*M	*W	D	100 071	315	367	104
403C	21	LXI H,	4579	!	Y	E	100 074	041	171	105
403F	CD	CALL	4519	*M	Y.	E	100 077	315	031	105
4042	CD	CALL	44F7	*M	*W	D	100 102	315	367	104
4045	C3	JMP	464C	*C	L	F	100 105	303	114	106
4048	2A	LHLD	44EE	*	*N	D	100 110	052	356	104
404B	7E	MOV A,M		†			100 113	176		
404C	23	INX H		#			100 114	043		
404D	22	SHLD	44EE	"	*N	D	100 115	042	356	104
4050	C9	RET		*I			100 120	311		
4051	3C	INR A		<			100 121	074		
4052	E6	ANI	07	*F	G.		100 122	346	007	
4054	FE	CPI	06	*†	F.		100 124	376	006	
4056	DA	JC	405B	*Z	[	0	100 126	332	133	100
4059	C6	ADI	03	*F	C.		100 131	306	003	
405B	FE	CPI	05	*†	E.		100 133	376	005	
405D	DA	JC	4062	*Z	B	0	100 135	332	142	100
4060	C6	ADI	02	*F	B.		100 140	306	002	
4062	C6	ADI	41	*F	A		100 142	306	101	
4064	12	STAX D		R.			100 144	022		
4065	C9	RET		*I			100 145	311		
4066	06	MVI B,	04	F.	D.		100 146	006	004	
4068	7E	MOV A,M		†			100 150	176		
4069	12	STAX D		R.			100 151	022		
406A	23	INX H		#			100 152	043		

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 ADDRESS & 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...

4071 3A ED 44	0500 MASK LDA SAVE	/LOAD THE CURRENT BYTE
4074 E6 38	0510 ANI 38H	/MASK FOR THE MIDDLE BITS
4076 0F	0520 RRC	/PUT THE MIDDLE BITS INTO THE-
4077 0F	0530 RRC	/THREE LEAST SIGNIFICANT-
4078 0F	0540 RRC	/BITS IN THE ACCUMULATOR
4079 C9	0550 RET	/CONTINUE...
407A CD 71 40	0560 CONDL CALL MASK	/GET BITS 3,4,5 INTO 1,2,3
407D 87	0570 ADD A	/DOUBLE CONDITION BITS
407E 4F	0580 MOV C,A	/PUT OFFSET VALUE IN REG. C
407F 21 52 44	0590 LXI H,CONDN	/POINT TO CONDITION TABLE
4082 09	0600 DAD B	/ADD OFFSET TO COND. TABLE POINTER
4083 7E	0610 MOV A,M	/FETCH CONDITION DIGIT
4084 12	0620 STAX D	/WRITE CONDITION DIGIT IN OUTBUF
4085 23	0630 INX H	/POINT TO NEXT DIGIT IN TABLE
4086 13	0640 INX D	/POINT TO NEXT SLOT IN OUTBUF
4087 7E	0650 MOV A,M	/FETCH NEXT COND. DIGIT FROM TBL.
4088 12	0660 STAX D	/WRITE IT IN OUTPUT BUFFER
4089 C9	0670 RET	/CONTINUE...
408A CD 71 40	0680 LXICD CALL MASK	/GET BITS 3,4,5 INTO 0,1,2
408D E6 06	0690 ANI 06	/MASK FOR BITS 1,2
408F FE 06	0700 CPI 06	/ARE BOTH BITS 1 & 2 SET?
4091 C2 51 40	0710 JNZ REGLD	/NO, GO LOAD A REGISTER NAME
4094 3E 53	0720 MVI A,'S'	/YES, MUST BE LXI SP. LOAD AN S
4096 12	0730 STAX D	/WRITE AN 'S' IN THE OUTPUT BUFF.
4097 13	0740 INX D	/POINT TO NEXT SPACE IN OUTBUF
4098 3E 50	0750 MVI A,'P'	/LOAD AN ASCII 'P'
409A 12	0760 STAX D	/WRITE A 'P' IN THE OUTPUT BUFF.
409B C9	0770 RET	/CONTINUE...
409C CD F7 44	0780 DISAS CALL CRLF	/PRINT A CRLF
409F CD F4 42	0790 CALL CLRBUF	/CLEAR THE OUTPUT BUFFER
40A2 2A EE 44	0800 LHL D PGMCT	/GET CURRENT PROGRAM COUNTER
40A5 22 F5 44	0810 SHLD OSAVE	/SAVE PGMCT FOR OCTAL ROUTINE
40A8 CD 96 42	0820 CALL HEXOT	/WRITE HEX ADDRESS IN OUTBUF
40AB CD 48 40	0830 CALL FETCH	/GET REFERENCED BYTE FROM MEM.
40AE 32 F2 44	0840 STA CHARI	/SAVE BYTE IN 1ST. ASCII BUFFER
40B1 32 ED 44	0850 STA SAVE	/SAVE BYTE FOR LATER USE
40B4 67	0860 MOV H,A	/PUT BYTE IN REGISTER H
40B5 11 68 44	0870 LXI D,OUTBUF+2	/POINT TO DATA COLUMN
40B8 CD E7 42	0880 CALL XCODE	/WRITE HEX BYTE IN OUTBUF
40BB 21 55 43	0890 LXI H,OPCODES	/POINT TO OPCODE TABLE
40BE 01 11 00	0900 LXI B,17	/CLR B AND SET C TO NO. OF CODES
40C1 BE	0910 ONE CMP M	/DOES DATA BYTE MATCH TABLE?
40C2 CA 76 42	0920 JZ BYTE1	/YES, GO PROCESS ONE BYTE INST.
40C5 23	0930 INX H	/NO, POINT TO NEXT INST. IN TABLE
40C6 0D	0940 DCR C	/DECREMENT ONE BYTE TABLE COUNT
40C7 C2 C1 40	0950 JNZ ONE	/IF MORE ONE BYTE INSTS GO. TEST
40CA 0E 0A	0960 MVI C,0AH	/SET C TO NO. OF TWO BYTE INSTS
40CC BE	0970 TWO CMP M	/DOES DATA BYTE MATCH TABLE?
40CD CA 59 42	0980 JZ BYTE2	/YES, GO PROCESS A 2 BYTE INST.
40D0 23	0990 INX H	/NO, POINT TO NEXT INST. IN TABLE

40D1 0D	1000 DCR C	/DECREMENT TWO BYTE TABLE COUNT
40D2 C2 CC 40	1010 JNZ TWO	/IF MORE 2 BYTE INSTS. GO TEST
00D5 0E 06	1020 MVI C,6	/SET C TO NO. OF 3 BYTE INSTS.
40D7 BE	1030 THREE CMP M	/DOES DATA BYTE MATCH TABLE?
40D8 CA 2D 42	1040 JZ BYTE3	/YES, GO PROCESS A 3 BYTE INST?
40DB 23	1050 INX H	/POINT TO NEXT INST. IN TABLE
40DC 0D	1060 DCR C	/DECREMENT 3 BYTE TABLE COUNT
40DD C2 D7 40	1070 JNZ THREE	/IF MORE 3 BYTE INSTS. GO TEST
40E0 E6 C0	1080 ANI 0C0H	/MASK FOR BITS 6 & 7
40E2 FE 40	1090 CPI 40H	/WAS BIT 6 SET?
40E4 CA 0B 42	1100 JZ MOV /YES, GO PROCESS A MOV INSTRUCTION	
40E7 FE 80	1110 CPI 80H	/WAS BIT 7 SET?
40E9 CA F7 41	1120 JZ ADD	/YES, GO PROCESS AN ADD INST.
40EC 3A ED 44	1130 LDA SAVE	/RESTORE ORIGINAL BYTE
40EF E6 C7	1140 ANI 0C7H	/MASK FOR BITS 0,1,2,6,7
40F1 D6 04	1150 SUI 04	/IS BIT 2 SET?
40F3 CA E2 41	1160 JZ INR /YES, GO PROCESS A INR INSTRUCTION	
40F6 3D	1170 DCR A	/WERE BITS 0 AND 2 SET?
40F7 CA D9 41	1180 JZ DCR /YES, GO PROCESS A DCR INSTRUCTION	
40FA 3D	1190 DCR A	/WERE BITS 1 AND 2 SET?
40FB CA C0 41	1200 JZ MVI /YES, GO PROCESS A MVI INSTRUCTION	
40FE 3A ED 44	1210 LDA SAVE	/RESTORE ORIGINAL DATA BYTE
4101 E6 C0	1220 ANI 0C0H	/ARE BOTH BITS 6 AND 7 SET?
4103 CA 7F 41	1230 JZ LXI /YES, GO PROCESS A LXI INSTRUCTION	
4106 3A ED 44	1240 LDA SAVE	/RESTORE ORIGINAL DATA BYTE
4109 E6 C7	1250 ANI 0C7H	/MASK FOR BITS 0,1,2,6,7
410B D6 C0	1260 SUI 0C0H	/WERE BITS 6 AND 7 SET?
410D CA 72 41	1270 JZ RET /YES, GO PROCESS A RET INSTRUCTION	
4110 D6 02	1280 SUI 02	/WERE BITS 1,6,3 SET?
4112 CA 65 41	1290 JZ JMP /YES, GO PROCESS A JMP INSTRUCTION	
0115 D6 02	1300 SUI 02	/WERE BITS 2,6,7 SET?
4117 CA 58 41	1310 JZ CALL /YES, GO PROCESS CALL INSTRUCTION	
411A D6 03	1320 SUI 03	/WERE BITS 0,1,2,6,7 SET?
411C CA 43 41	1330 JZ RST /YES, GO PROCESS A RST INSTRUCTION	
411F 3A ED 44	1340 LDA SAVE	/RESTORE ORIGINAL DATA BYTE
4122 E6 07	1350 ANI 07	/MASK FOR BITS 0,1,2
4124 4F	1360 MOV C,A	/STORE REGISTER CODE IN C
4125 21 49 44	1370 LXI H,LPOP	/POINT TO 'POP' IN TABLE
4128 09	1380 DAD B /OFFSET, POINT TO CORRECT MNEMONIC	
4129 11 6D 44	1390 LXI D,OUTBUF+7	/POINT TO MNEMONIC COLUMN
412C CD 66 40	1400 CALL PRINT	/WRITE MNEMONIC IN OUTBUF
412F CD 71 40	1410 CALL MASK /GET BITS 3,4,5 INTO BITS 0,1,2	
4132 FE 06	1420 CPI 06	/WERE BITS 4,5 SET?
4134 C2 EE 41	1430 JNZ INR3 /YES, GO PROCESS INR INSTRUCTION	
4137 21 46 44	1440 LXI H,LPSW	/POINT TO 'PSW' IN TABLE
413A 11 72 44	1450 LXI D,OUTBUF+12	/POINT TO REGISTER AREA
413D CD 66 40	1460 CAHH PRINT	/WRITE 'PSW' IN OUTBUF
4140 C3 A5 42	1470 JMP FINISH /PROCESS ASCII & OCT. & PRINT	
4143 21 42 44	1480 RST LXI H,LRST	/POINT TO RST MNEMONIC
0146 11 6D 44	1490 LXI D,OUTBUF+7	/POINT TO MNEMONIC SLOT

4149	CD	66	40	1500	CALL PRINT	/WRITE MNEMONIC IN OUTBUF
414C	CD	71	40	1510	CALL MASK /GET BITS 3,4,5 INTO BITS 0,1,2	
414F	11	75	44	1520	LXI D,OUTBUF+15	/POINT TO RST NUMERIC
4152	CD	E7	42	1530	CALL XCODE	/WRITE HEX NUMERIC IN OUTBUF
4155	C3	A5	42	1540	JMP FINISH	/WRITE ASCII, OCTAL AND PRINT
4158	3E	43		1550	CALL MVI A,'C'	/LOAD AN ASCII 'C'
415A	11	6D	44	1560	LXI D,OUTBUF+7	/POINT TO MNEMONIC COLUMN
415D	12			1570	STAX D	/WRITE 'C' IN OUTPUT BUFFER
415E	13			1580	INX D	/POINT TO NEXT SLOT IN OUTBUF
415F	CD	7A	40	1590	CALL CONDL	/WRITE REST OF CALL CONDITION
4162	C3	3B	42	1600	JMP BYT3	/GO FINISH A THREE BYTE INST.
4165	3E	4A		1610	JMP MVI A,'J'	/LOAD AN ASCII 'J'
4167	11	6D	44	1620	LXI D,OUTBUF+7	/POINT TO MNEMONIC COLUMN
416A	12			1630	STAX D	/WRITE A 'J' IN OUTBUF
416B	13			1640	INX D	/POINT TO NEXT SLOT IN OUTBUF
416C	CD	7A	40	1650	CALL CONDL	/WRITE REST OF JMP CONDITION
416F	C3	3B	42	1660	JMP BYT3	/GO FINISH A THREE BYTE INST.
4172	3E	52		1670	RET MVI A,'R'	/LOAD AN ASCII 'R'
4174	11	6D	44	1680	LXI D,OUTBUF+7	/POINT TO MNEMONIC COLUMN
4177	12			1690	STAX D	/WRITE 'R' IN OUTPUT BUFFER
4178	13			1700	INX D	/POINT TO NEXT SLOT IN OUTBUF
4179	CD	7A	40	1710	CALL CONDL	/WRITE REST OF RET INST.
417C	C3	A5	42	1720	JMP FINISH	/WRITE ASCII, OCTAL & PRINT
417F	21	2A	44	1730	LXI LXI H,LLXI	/POINT TO LXI MNEMONIC
4182	3A	ED	44	1740	LDA SAVE	/RESTORE ORIGINAL DATA BYTE
4185	E6	0F		1750	ANI 0FH	/MASK FOR BITS 0,1,2,3
4187	3D			1760	DCR A	/WAS BIT 0 SET?
4188	CA	AD	41	1770	JZ LXIP	/YES, GO LOAD LXI MNEMONIC
418B	FE	04		1780	CPI 04	/WAS BYTE LESS THAN 5?
418D	DA	92	41	1790	JC LX2	/YES, GO JUSTIFY POINTER
4190	D6	05		1800	SUI 05	IF BYTE WAS >5 SUBTRACT BIAS
4192	87			1810	LX2 ADD A	/DOUBLE MNEMONIC POINTER
4193	87			1820	ADD A	/DOUBLE MNEMONIC POINTER
4194	4F			1830	MOV C,A	/PUT CALCULATED JUSTIFY IN REG. C
4195	09			1840	DAD B	/ADD JUSTIFY TO MNEMONIC POINTER
4196	11	6D	44	1850	LXI D,OUTBUF+7	/POINT TO MNEMONIC SLOT
4199	CD	66	40	1860	CALL PRINT	/WRITE MNEMONIC IN OUTBUF
419C	11	72	44	1870	LXI D,OUTBUF+12	/POINT TO LXI REG. SLOT
419F	CD	8A	40	1880	CALL LXICD	/WRITE REG. OR 'SP' IN OUTBUF
41A2	3A	ED	44	1890	LDA SAVE	/RESTORE ORIGINAL DATA BYTE
41A5	FE	20		1900	CPI ' '	/WAS IT A 20H?
41A7	CC	39	46	1910	CZ INVAL	/YES, GO INVALIDATE MNEMONIC
41AA	C3	A5	42	1920	JMP FINISH	/WRITE ASCII, OCTAL & PRINT
41AD	11	6D	44	1930	LXIP LXI D,OUTBU+7	/POINT TO MNEMONIC SLOT
41B0	CD	66	40	1940	CALL PRINT	/WRITE MNEMONIC IN OUTBUF
41B3	11	71	44	1950	LXI D,OUTBUF+11	/POINT TO REG. SLOT
41B6	CD	8A	40	1960	CALL LXICD	/WRITE REG. NAME IN OUTBUF
41B9	3E	2C		1970	MVI A,2CH	/LOAD A COMMA
41BB	13			1980	INX D	/POINT TO NEXT SLOT IN OUTBUF
41BC	12			1990	STAX D	/WRITE A COMMA IN OUTBUF

41BD C3 3B 42	2000 JMP BYT3 /GO WRITE THE LABEL IN OUTBUF
41C0 21 26 44	2010 MVI LXI H,LMVI /POINT TO MVI MNEMONIC
41C3 11 6D 44	2020 LXI D,OUTBUF+7 /POINT TO MNEMONIC SLOT
41C6 CD 66 40	2030 CALL PRINT /WRITE MNEMONIC IN OUTBUF
41C9 CD 71 40	2040 CALL MASK /GET BITS 3,4,5 IN BITS 1,2,3
41CC 11 71 44	2050 LXI D,OUTBUF+11 /POINT TO REG. SLOT
41CF CD 51 40	2060 CALL REGLD /WRITE REGISTER IN OUTBUF
41D2 3E 2C	2070 MVI A,2CH /LOAD A COMMA
41D4 13	2080 INX D /POINT TO NEXT SLOT IN OUTBUF
41D5 12	2090 STAX D /WRITE A COMMA IN OUTBUF
41D6 C3 67 42	2100 JMP BYT2 /GO WRITE VALUE IN OUTBUF
41D9 21 22 44	2110 DCR LXI H,LDCR /POINT TO DCR MNEMONIC
41DC 11 6D 44	2120 LXI D,OUTBUF+7 /POINT TO MNEMONIC SLOT
41DF C3 E8 41	2130 JMP INR2 /GO FINISH DCR INSTRUCTION
41E2 21 1E 44	2140 INR LXI H,LINR /POINT TO INR INSTRUCTION
41E5 11 6D 44	2150 LXI D,OUTBUF+7 /POINT TO MNEMONIC SLOT
41E8 CD 66 40	2160 INR2 CALL PRINT /WRITE MNEMONIC IN OUTBUF
41EB CD 71 40	2170 CALL MASK /GET BITS 3,4,5 IN BITS 1,2,3
41EE 11 72 44	2180 INR3 LXI D,OUTBUF+12 /POINT TO REG. SLOT
41F1 CD 51 40	2190 CALL REGLD /WRITE REGISTER IN OUTBUF
41F4 C3 A5 42	2200 JMP FINISH /WRITE ASCII, OCTAL & PRINT
41F7 3A ED 44	2210 ADD LDA SAVE /RESTORE ORIGINAL BYTE
41FA E6 38	2220 ANI 38H /MASK FOR BITS 3,4,5
41FC 0F	2230 RRC /CALCULATE OFFSET
41FD 4F	2240 MOV C,A /PUT OFFSET IN REG. C
41FE 21 FE 43	2250 LXI H,LADD /POINT TO ADD MNEMONIC
4201 09	2260 DAD B /ADD OFFSET TO MNEMONIC POINTER
4202 11 6D 44	2270 LXI D,OUTBUF+7 /POINT TO MNEMONIC SLOT
4205 CD 66 40	2280 CALL PRINT /WRITE MNEMONIC IN OUTBUF
4208 C3 21 42	2290 JMP MOV2 /GO FINISH ADD INSTRUCTION
420B 21 FA 43	2300 MOV LXI H,LMOV /POINT TO MOV MNEMONIC
420E 11 6D 44	2310 LXI D,OUTBUF+7 /POINT TO MNEMONIC SLOT
4211 CD 66 40	2320 CALL PRINT /WRITE MNEMONIC IN OUTBUF
4214 CD 71 40	2330 CALL MASK /GET BITS 3,4,5 IN BITS 1,2,3
4217 11 71 44	2340 LXI D,OUTBUF+11 /POINT TO REGISTER SLOT
421A CD 51 40	2350 CALL REGLD /WRITE REGISTER IN OUTBUF
421D 3E 2C	2360 MVI A,2CH /LOAD A COMMA
421F 13	2370 INX D /POINT TO NEXT SLOT IN OUTBUF
4220 12	2380 STAX D /WRITE A COMMA IN OUTBUF
4221 3A ED 44	2390 MOV2 LDA SAVE /RESTORE ORIGINAL BYTE
4224 E6 07	2400 ANI 07 /MASK FOR BITS 0,1,2
4226 13	2410 INX D /POINT TO NEXT SLOT IN OUTBUF
4227 CD 51 40	2420 CALL REGLD /WRITE REG. NAME IN OUTBUF
422A C3 A5 42	2430 JMP FINISH /WRITE ASCII,OCTAL & PRINT
422D 79	2440 BYTE3 MOV A,C /GET MNEMONIC COUNT IN REG.A
422E 87	2450 ADD A /CALCULATE-
422F 87	2460 ADD A /OFFSET
4230 4F	2470 MOV C,A /PUT OFFSET IN REG. C
4231 21 DE 43	2480 LXI H,NEM3 /POINT TO MNEMONIC TAB 3
4234 09	2490 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 0C 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 00 EC	2880 JMP 0EC00H	/JMP TO SYSTEM MONITOR
4296 E5	2890 HEX0T 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 0E 04	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 0FFH	/INPUT THE SENSE SWITCHS

42A7 E6 01	3000 ANI 1	/WAS SENSE SWITCH 8 SET?
42A9 CA C7 42	3010 JZ OVER	/YES, SKIP THE ASCII BUFFER LOAD
42AC 11 7E 44	3020 LXI D,OUTBUF+24	/POINT TO 1ST ASCII SLOT
42AF 3A F2 44	3030 LDA CHAR1	/LOAD THE FIRST ASCII BYTE
42B2 CD 11 43	3040 CALL ASCII	/GO WRITE THE ASCII CHAR.
42B5 11 82 44	3050 LXI D,OUTBUF+28	/POINT TO 2ND ASCII SLOT
42B8 3A F3 44	3060 LDA CHAR2	/LOAD THE SECOND ASCII BYTE
42BB CD 11 43	3070 CALL ASCII	/GO WRITE THE ASCII CHAR.
42BE 11 86 44	3080 LXI D,OUTBUF+32	/POINT TO 3RD ASCII SLOT
42C1 3A F4 44	3090 LDA CHAR3	/LOAD THE THIRD ASCII BYTE
42C4 CD 11 43	3100 CALL ASCII	/GO WRITE THE ASCII CHAR.
42C7 CD E1 45	3110 OVER CALL OCTAL	/GO TRY TO WRITE OCTAL
42CA CD 28 43	3120 CALL BUFNT	/DONE. PRINT THE OUTPUT BUF.
42CD 2A F0 44	3130 LHL DONE	/LOAD THE ENDING POINTER
42D0 EB	3140 XCHG	/PUT END POINTER IN REG. D & E
42D1 CD 47 46	3150 CALL FETC2	/LOAD THE CURRENT INST ADDR.
42D4 7C	3160 MOV A,H	/PUT LOW ORDER CURRENT IN REG. A
42D5 BA	3170 CMP D	/COMPARE LOW ORDER CURRENT TO END
42D6 CA DC 42	3180 JZ LOW	/IF LOW ORDERS EQUAL, GO TEST HIGH
42D9 C3 55 46	3190 JMP PAGE	/GO PROCESS NEXT INSTRUCTION!
42DC 7D	3200 LOW MOV A,L	/PUT HIGH ORDER CURRENT IN A
42DD BB	3210 CMP E	/COMPARE HIGH CURRENT TO HIGH END
42DE D2 00 40	3220 JNC START	/IF LESS THAN BEGIN AGAIN
42E1 CA 00 40	3230 JZ START	/IF EQUAL TO BEGIN AGAIN
42E4 C3 55 46	3240 JMP PAGE	/GO PROCESS NEXT INSTRUCTION
42E7 E5	3250 XCODE PUSH H	/SAVE REGISTERS H & L
42E8 C5	3260 PUSH B	/SAVE REGISTERS B & C
42E9 F5	3270 PUSH PSW	/SAVE A AND FLAGS
42EA 0E 02	3280 MVI C,2	/SET NUMBER OF CHARS. TO LOAD
42EC 67	3290 MOV H,A	/MOVE BYTE TO ENCODE TO REG. A
42ED CD AA 44	3300 CALL XLOAD	/WRITE ASCII BYTES IN OUTBUF
42F0 F1	3310 POP PSW	/RESTORE A AND FLAGS
42F1 C1	3320 POP B	/RESTORE REGISTERS B & C
42F2 E1	3330 POP H	/RESTORE REGISTERS H & L
42F3 C9	3340 RET	/CONTINUE...
42F4 11 44 00	3350 CLRBUF LXI D,68	/SET NO. OF CHAR. TO CLEAR
42F7 21 66 44	3360 LXI H,OUTBUF	/POINT TO OUTPUT BUFFER
42FA 3E 20	3370 MVI A,' '	/LOAD A SPACE
42FC 77	3380 MORE MOV M,A	/INSERT SPACE IN OUTBUF
42FD 1D	3390 DCR E	/DECREMENT CHAR. COUNT
42FE CA 05 43	3400 JZ CLRCHAR	/IF DONE CLEAR ASCII CHAR. BUF
4301 23	3410 INX H	/POINT TO NEXT SLOT IN OUTBUF
4302 C3 FC 42	3420 JMP MORE	/GO CLEAR MORE...
4305 21 F2 44	3430 CLRCHAR LXI H,CHAR1	/POINT TO ASCII BUF1
4308 0E 03	3440 MVI C,3	/SET NUMBER OF CHARS. TO CLEAR
430A 77	3450 MRE2 MOV M,A	/WRITE A SPACE IN ASCII BUF.
430B 0D	3460 DCR C	/DECREMENT BUFFER COUNT
430C C8	3470 RZ	/IF DONE, CONTINUE...
430D 23	3480 INX H	/POINT TO NEXT ASCII BUFFER
430E C3 0A 43	3490 JMP MRE2	/CLEAR NEXT ASCII BUFFER

4311 FE 7F	3500 ASCII CPI 7FH	/TEST IF SIGN BIT IS SET
4313 D4 4D 43	3510 CNC SIGN /SET? GO WRITE "*" BEFORE CHAR.	
4316 E6 7F	3520 ANI 7FH	/STRIP SIGN BIT FROM CHAR.
4318 FE 20	3530 CPI ' '	/TEST FOR ASCII SPACE
431A C8	3540 RZ	/IF EQUAL, DON'T BOTHER WITH REST
431B FE 00	3550 CPI 0	/TEST FOR NULL CHAR.
431D C8	3560 RZ	/IF EQUAL, DON'T BOTHER WITH REST
431E FE 7F	3570 CPI 7FH	/TEST FOR RUB OUT CHAR.
4320 C8	3580 RZ	/IF EQUAL, DON'T BOTHER WITH REST
4321 FE 20	3590 CPI ' '	/TEST FOR 20H
4323 DC 43 43	3600 CC CONT /IF LESS THAN 20H DO CONTROL CHAR	
4326 12	3610 STAX D	/WRITE ASCII CHAR. IN OUTBUF
4327 C9	3620 RET	/CONTINUE...
4328 21 A9 44	3630 BUFFNT LXI H,BUFEND-1	/POINT TO END OF BUF
432B 11 49 00	3640 LXI D,73	/SET NO. OF CHAR. IN BUFFER +1
432E 1B	3650 LESS DCX D	/DECREMENT CHAR. COUNT
432F 7E	3660 MOV A,M	/FETCH CHAR. FROM END OF BUF.
4330 2B	3670 DCX H	/DECREMENT BUFFER POINTER
4331 FE 20	3680 CPI ' '	/FOUND A VALID CHAR. YET?
4333 CA 2E 43	3690 JZ LESS /NO, KEEP DECREASING BUF. LENGTH	
4336 21 62 44	3700 LXI H,ADDBUF	/OTHERWISE POINT BEG OF LINE
4339 4E	3710 GET MOV C,M	/MOVE CHAR. FROM OUTBUF TO C
433A CD C3 44	3720 CALL HISPD	/PRINT CHAR. ON OUTPUT DEVICE
433D 1D	3730 DCR E	/DECREMENT CHAR. COUNT
433E C8	3740 RZ	/WHEN DONE WITH LINE, CONTINUE...
433F 23	3750 INX H	/POINT TO NEXT CHAR. TO PRINT
4340 C3 39 43	3760 JMP GET	/GO PRINT NEXT CHAR.
4343 C6 40	3770 CONT ADI '0'	/MAKE A PRINTING CHAR. OF CONT
4345 4F	3780 MOV C,A	/SAVE THE PRINTING CHAR.
4346 13	3790 INX D	/POINT TO NEXT SLOT IN OUTBUF
4347 3E 2E	3800 MVI A,'.'	/LOAD A CONTROL CHAR. INDICATOR
4349 12	3810 STAX D	/WRITE CONTROL INDICATOR IN BUF.
434A 1B	3820 DCX D	/POINT TO ASCII SLOT IN OUTBUF
434B 79	3830 MOV A,C	/RESTORE PRINTING CHAR.
434C C9	3840 RET	/CONTINUE...
434D 4F	3850 SIGN MOV C,A	/SAVE ASCII CHAR.
434E 3E 2A	3860 MVI A,'*'	/LOAD A SIGN BIT INDICATOR
4350 1B	3870 DCX D	/POINT AHEAD OF ASCII CHAR.
4351 12	3880 STAX D	/WRITE SIGN INDICATOR IN OUTBUF
4352 13	3890 INX D	/RESTORE ASCII POINTER TO CHAR.
4353 79	3900 MOV A,C	/RESTORE ASCII CHAR.
4354 C9	3910 RET	/CONTINUE...
4355 00	3920 OPCODES NOP	/THE FOLOWING IS A /NOP
4356 07	3930 DB 007H	/IS A TABLE OF 8080 /RLC
4357 0F	3940 DB 00FH	/OPCODES USED BY- /RRC
4358 17	3950 DB 017H	/THIS PROGRAM TO- /RAL
4359 1F	3960 DB 01FH	/COMPARE AGAINST- /RAR
435A 27	3970 DB 027H	/THE BYTE FETCHED- /DAA
435B 2F	3980 DB 02FH	/FROM MEMORY /CMA
435C 37	3990 DB 037H	/STC

435D 3F	4000 DB 03FH	/CMC
435E 76	4010 DB 076H	/HLT
435F C9	4020 DB 0C9H	/RET
4360 E3	4030 DB 0E3H	/XTHL
4361 E9	4040 DB 0E9H	/PCHL
4362 EB	4050 DB 0EBH	/XCHG
4363 F3	4060 DB 0F3H	/DI
4364 F9	4070 DB 0F9H	/SPHL
4365 FB	4080 DB 0FBH	/EI
4366 C6	4090 DB 0C6H	/ADI
4367 CE	4100 DB 0CEH	/ACI
4368 D3	4110 DB 0D3H	/OUT
4369 D6	4120 DB 0D6H	/SUI
436A DB	4130 DB 0DBH	/IN
436B DE	4140 DB 0DEH	/SBI
436C E6	4150 DB 0E6H	/ANI
436D EE	4160 DB 0EEH	/XRI
436E F6	4170 DB 0F6H	/ORI
436F FE	4180 DB 0FEH	/CPI
4370 22	4190 DB 022H	/SHLD
4371 2A	4200 DB 02AH	/LHLD
4372 32	4210 NEMON DB 32H	/STA
4373 3A	4220 DB 03AH	/LDA
4374 C3	4230 DB 0C3H	/JMP
4375 CD	4240 DB 0CDH	/CALL
4376 45 49	4250 DW 'IE'	/THE FOLLOWING ARE TABLES OF-
4378 20 20	4260 DW ' '	/ASCII MNEMONICS USED BY THE-
437A 53 50	4270 DW 'PS'	/DISASSEMBLER TO LOAD THE-
437C 48 4C	4280 DW 'LH'	/OUTPUT BUFFER WITH THE-
437E 44 49	4290 DW 'ID'	/INSTRUCTION MNEMONICS
4380 20 20	4300 DW ' '	
4382 58 43	4310 DW 'CX'	
4384 48 47	4320 DW 'GH'	
4386 50 43	4330 DW 'CP'	
4388 48 4C	4340 DW 'LH'	
438A 58 54	4350 DW 'TX'	
438C 48 4C	4360 DW 'LH'	
438E 52 45	4370 DW 'ER'	
4390 54 20	4380 DW 'T'	
4392 48 4C	4390 DW 'LH'	
4394 54 20	4400 DW 'T'	
4396 43 4D	4410 DW 'MC'	
4398 43 20	4420 DW 'C'	
439A 53 54	4430 DW 'TS'	
439C 43 20	4440 DW 'C'	
439E 43 4D	4450 DW 'MC'	
43A0 41 20	4460 DW 'A'	
43A2 44 41	4470 DW 'AD'	
43A4 41 20	4480 DW 'A'	
43A6 52 41	4490 DW 'AR'	

43A8	52	20	4500	DW	'R'	
43AA	52	41	4510	DW	'AR'	
43AC	4C	20	4520	DW	'L'	
43AE	52	52	4530	DW	'RR'	
43B0	43	20	4540	DW	'C'	
43B2	52	4C	4550	DW	'LR'	
43B4	43	20	4560	DW	'C'	
43B6	4E	4F	4570	NEM2	DW	'ON'
43B8	50	20	4580	DW	'P'	
43BA	43		4590	DB	43H	/C
43BB	50		4600	DB	50H	/P
43BC	49		4610	DB	49H	/I
43BD	20		4620	DB	20H	
43BE	4F		4630	DB	4FH	/O
43BF	52		4640	DB	52H	/R
43C0	49		4650	DB	49H	/I
43C1	20		4660	DB	20H	
43C2	58		4670	DB	58H	/X
43C3	52		4680	DB	52H	/R
43C4	49		4690	DB	49H	/I
43C5	20		4700	DB	20H	
43C6	41		4710	DB	41H	/A
43C7	4E		4720	DB	4EH	/N
43C8	49		4730	DB	49H	/I
43C9	20		4740	DB	20H	
43CA	53		4750	DB	53H	/S
43CB	42		4760	DB	42H	/B
43CC	49		4770	DB	49H	/I
43CD	20		4780	DB	20H	
43CE	49		4790	DB	49H	/I
43CF	4E		4800	DB	4EH	/N
43D0	20		4810	DB	20H	
43D1	20		4820	DB	20H	
43D2	53		4830	DB	53H	/S
43D3	55		4840	DB	55H	/U
43D4	49		4850	DB	49H	/I
43D5	20		4860	DB	20H	
43D6	4F		4870	DB	4FH	/O
43D7	55		4880	DB	55H	/U
43D8	54		4890	DB	54H	/T
43D9	20		4900	DB	20H	
43DA	41		4910	DB	41H	/A
43DB	43		4920	DB	43H	/C
43DC	49		4930	DB	49H	/I
43DD	20		4940	DB	20H	
43DE	41		4950	NEM3	DB	41H
43DF	44		4960	DB	44H	/D
43E0	49		4970	DB	49H	/I
43E1	20		4980	DB	20H	
43E2	43		4990	DB	43H	/C

43E3 41	5000 DB 41H	/A
43E4 4C	5010 DB 4CH	/L
43E5 4C	5020 DB 4CH	/L
43E6 4A	5030 DB 4AH	/J
43E7 4D	5040 DB 4DH	/M
43E8 50	5050 DB 50H	/P
43E9 20	5060 DB 20H	
43EA 4C	5070 DB 4CH	/L
43EB 44	5080 DB 44H	/D
43EC 41	5090 DB 41H	/A
43ED 20	5100 DB 20H	
43EE 53	5110 DB 53H	/S
43EF 54	5120 DB 54H	/T
43F0 41	5130 DB 41H	/A
43F1 20	5140 DB 20H	
43F2 4C	5150 DB 4CH	/L
43F3 48	5160 DB 48H	/H
43F4 4C	5170 DB 4CH	/L
43F5 44	5180 DB 44H	/D
43F6 53	5190 DB 53H	/S
43F7 48	5200 DB 48H	/H
43F8 4C	5210 DB 4CH	/L
43F9 44	5220 DB 44H	/D
43FA 4D	5230 LMOV DB 4DH	/M
43FB 4F	5240 DB 4FH	/O
43FC 56	5250 DB 56H	/V
43FD 20	5260 DB 20H	
43FE 41	5270 LADD DB 41H	/A
43FF 44	5280 DB 44H	/D
4400 44	5290 DB 44H	/D
4401 20	5300 DB 20H	
4402 41	5310 DB 41H	/A
4403 44	5320 DB 44H	/D
4404 43	5330 DB 43H	/C
4405 20	5340 DB 20H	
4406 53	5350 DB 53H	/S
4407 55	5360 DB 55H	/U
4408 42	5370 DB 42H	/B
4409 20	5380 DB 20H	
440A 53	5390 DB 53H	/S
440B 42	5400 DB 42H	/B
440C 42	5410 DB 42H	/B
440D 20	5420 DB 20H	
440E 41	5430 DB 41H	/A
440F 4E	5440 DB 4EH	/N
4410 41	5450 DB 41H	/A
4411 20	5460 DB 20H	
4412 58	5470 DB 58H	/X
4413 52	5480 DB 52H	/R
4414 41	5490 DB 41H	/A

4415 20	5500 DB 20H	
4416 4F	5510 DB 4FH	/O
4417 52	5520 DB 52H	/R
4418 41	5530 DB 41H	/A
4419 20	5540 DB 20H	
441A 43	5550 DB 43H	/C
441B 4D	5560 DB 4DH	/M
441C 50	5570 DB 50H	/P
441D 20	5580 DB 20H	
441E 49	5590 LINR DB 49H	/I
441F 4E	5600 DB 4EH	/N
4420 52	5610 DB 52H	/R
4421 20	5620 DB 20H	
4422 44	5630 LDCR DB 44H	/D
4423 43	5640 DB 43H	/C
4424 52	5650 DB 52H	/R
4425 20	5660 DB 20H	
4426 4D	5670 LMVI DB 4DH	/M
4427 56	5680 DB 56H	/V
4428 49	5690 DB 49H	/I
4429 20	5700 DB 20H	
442A 4C	5710 LLXI DB 4CH	/L
442B 58	5720 DB 58H	/X
442C 49	5730 DB 49H	/I
442D 20	5740 DB 20H	
442E 53	5750 DB 53H	/S
442F 54	5760 DB 54H	/T
4430 41	5770 DB 41H	/A
4431 58	5780 DB 58H	/X
4432 49	5790 DB 49H	/I
4433 4E	5800 DB 4EH	/N
4434 58	5810 DB 58H	/X
4435 20	5820 DB 20H	
4436 44	5830 DB 44H	/D
4437 41	5840 DB 41H	/A
4438 44	5850 DB 44H	/D
4439 20	5860 DB 20H	
443A 4C	5870 DB 4CH	/L
443B 44	5880 DB 44H	/D
443C 41	5890 DB 41H	/A
443D 58	5900 DB 58H	/X
443E 44	5910 DB 44H	/D
443F 43	5920 DB 43H	/C
4440 58	5930 DB 58H	/X
4441 20	5940 DB 20H	
4442 52	5950 LRST DB 52H	/R
4443 53	5960 DB 53H	/S
4444 54	5970 DB 54H	/T
4445 20	5980 DB 20H	
4446 50	5990 LPSW DB 50H	/P

4447 53	6000 DB 53H	/S
4448 57	6010 DB 57H	/W
4449 20	6020 LPOP DB 20H	
444A 50	6030 DB 50H	/P
444B 4F	6040 DB 4FH	/O
444C 50	6050 DB 50H	/P
444D 20	6060 DB 20H	
444E 50	6070 DB 50H	/P
444F 55	6080 DB 55H	/U
4450 53	6090 DB 53H	/S
4451 48	6100 DB 48H	/H
4452 4E	6110 CONDN DB 4EH	/N
4453 5A	6120 DB 5AH	/Z
4454 5A	6130 DB 5AH	/Z
4455 20	6140 DB 20H	
4456 4E	6150 DB 4EH	/N
4457 43	6160 DB 43H	/C
4458 43	6170 DB 43H	/C
4459 20	6180 DB 20H	
445A 50	6190 DB 50H	/P
445B 4F	6200 DB 4FH	/O
445C 50	6210 DB 50H	/P
445D 45	6220 DB 45H	/E
445E 50	6230 DB 50H	/P
445F 20	6240 DB 20H	
4460 4D	6250 DB 4DH	/M
4461 20	6260 DB 20H	
4462 00 00	6270 ADDBUF DW 0	/THIS BUFFER STORES THE-
4464 00 00	6280 DW 0	/HEX ADDRESS FOR PRINTING.
4466 00 00	6290 OUTBUF DW 0	/HERE BEGINS THE OUTPUT-
4468 00 00	6300 DW 0	/BUFFER WHICH HOLDS THE-
446A 00 00	6310 DW 0	/ENTIRE LINE OF DATA TO BE-
446C 00 00	6320 DW 0	/PRINTED. AFTER THIS BUFFER-
446E 00 00	6330 DW 0	/IS LOADED WITH ASCII DATA-
4470 00 00	6340 DW 0	/THE ROUTINE CALLED BUFNT-
4472 00 00	6350 DW 0	/POINTS TO THE BEGINNING OF-
4474 00 00	6360 DW 0	/ADDBUF AND OUTPUTS THE CHAR'S-
4476 00 00	6370 DW 0	/FROM THERE TO BUFEND-1. THAT-
4478 00 00	6380 DW 0	/ROUTINE ALSO CALCULATES THE-
447A 00 00	6390 DW 0	/ACTUAL LENGTH OF THIS BUFFER-
447C 00 00	6400 DW 0	/EACH TIME A LINE IS PRINTED.
447E 00 00	6410 DW 0	
4480 00 00	6420 DW 0	
4482 00 00	6430 DW 0	
4484 00 00	6440 DW 0	
4486 00 00	6450 DW 0	
4488 00 00	6460 DW 0	
448A 00 00	6470 DW 0	
448C 00 00	6480 DW 0	
448E 00 00	6490 DW 0	

4490 00 00	6500 DW 0	
4492 00 00	6510 DW 0	
4494 00 00	6520 DW 0	
4496 00 00	6530 DW 0	
4498 00 00	6540 DW 0	
449A 00 00	6550 DW 0	
449C 00 00	6560 DW 0	
449E 00 00	6570 DW 0	
44A0 00 00	6580 DW 0	
44A2 00 00	6590 DW 0	
44A4 00 00	6600 DW 0	
44A6 00 00	6610 DW 0	
44A8 00 00	6620 DW 0	
44AA	6630 BUFEND EQU 5	
44AA AF	6640 XLOAD XRA A	/THIS ROUTINE LOADS THE-
44AB 29	6650 DAD H	/OUTPUT BUFFER WITH HEX-
44AC 17	6660 RAL	/CHAR'S. UPON ENTRY REG.-
44AD 29	6670 DAD H	/C CONTAINS THE NUMBER OF-
44AE 17	6680 RAL	/CHARACTERS TO BE LOADED.
44AF 29	6690 DAD H	/UPON ENTRY REG. D & E -
44B0 17	6700 RAL	/POINT TO THE SLOT IN THE-
44B1 29	6710 DAD H	/OUTPUT BUFFER WHICH IS TO-
44B2 17	6720 RAL	/BE LOADED WITH HEX CHAR'S.
44B3 FE 0A	6730 CPI 10	/THE ROUTINE TERMINATES WHEN-
44B5 DA BA 44	6740 JC ASCOUT	/REG. C REACHES 0.
44B8 C6 07	6750 ADI 7	
44BA C6 30	6760 ASCOUT ADI '0'	
44BC 12	6770 STAX D	
44BD 13	6780 INX D	
44BE 0D	6790 DCR C	
44BF C2 AA 44	6800 JNZ XLOAD	
44C2 C9	6810 RET	
44C3 DB 00	6820 HISPD IN 0	/INPUT THE STATUS PORT
44C5 E6 80	6830 ANI 80H	/TEST FOR TRANSMITTER BUF. EMPTY
44C7 CA C3 44	6840 JZ HISPD	/IF NOT READY, TRY AGAIN
44CA 79	6850 MOV A,C	/GET THE CHAR. TO PRINT IN ACC.
44CB D3 01	6860 OUT 1	/OUTPUT IT TO THE DATA PORT
44CD DB 00	6870 IN 0	/INPUT THE STATUS PORT
44CF E6 40	6880 ANI 40H	/IS DATA AVAILABLE FLAG SET?
44D1 C2 00 40	6890 JNZ START	/YES, THEN STOP & START OVER
44D4 C9	6900 RET	/OTHERWISE CONTINUE...
44D5 DB 00	6910 INPUT IN 0	/INPUT THE STATUS PORT
44D7 E6 40	6920 ANI 40H	/TEST FOR DAV FLAG
44D9 CA D5 44	6930 JZ INPUT	/NOT READY? TRY AGAIN
44DC DB 01	6940 IN 1	/INPUT THE DATA PORT
44DE E6 7F	6950 ANI 7FH	/STRIP PARITY
44E0 4F	6960 MOV C,A	/SAVE DATA IN REG. C
44E1 C9	6970 RET	/CONTINUE...
44E2 DB 00	6980 CPRINT IN 0	/INPUT THE STATUS PORT
44E4 E6 80	6990 ANI 80H	/TEST FOR TRANSMITTER BUF. EMPTY

44E6 CA E2 44	7000 JZ CPRINT	/NOT READY? THEN TRY AGAIN
44E9 79	7010 MOV A,C	/GET CHAR TO PRINT IN ACC.
44EA D3 01	7020 OUT I	/PRINT CHARACTER
44EC C9	7030 RET	/CONTINUE...
44ED 00	7040 SAVE DB 0	/FETCHED BYTE STORAGE
44EE 00 00	7050 PGMCT DW 0	/PROGRAM COUNTER STORAGE
44F0 00 00	7060 DONE DW 0	/ENDING PROGRAM COUNTER STORAGE
44F2 00	7070 CHAR1 DB 0	/FIRST BYTE, ASCII STORAGE
44F3 00	7080 CHAR2 DB 0	/SECOND BYTE, ASCII STORAGE
44F4 00	7090 CHAR3 DB 0	/THRID BYTE, ASCII STORAGE
44F5 00 00	7100 OSAVE DW 0	/CURRENT INSTRUCTION ADDRESS
44F7 0E 0D	7110 CRLF MVI C,0DH	/LOAD A CARRIAGE RETURN
44F9 CD C3 44	7120 CALL HISPD	/PRINT A CR
44FC 0E 0A	7130 MVI C,0AH	/LOAD A LINE FEED
44FE CD C3 44	7140 CALL HISPD	/PRINT A LINE FEED
4501 0E 00	7150 MVI C,0	/LOAD A NULL
4503 CD C3 44	7160 CALL HISPD	/PRINT A NULL
4506 CD C3 44	7170 CALL HISPD	/PRINT A NULL
4509 C3 C3 44	7180 JMP HISPD	/PRINT A NULL AND CONTINUE...
450C 4E 4F	7190 DATA DW 'ON'	/ASCII STORAGE FOR-
450E 4E 20	7200 DW 'N'	/NON HEX DATA"
4510 48 45	7210 DW 'EH'	
4512 58 20	7220 DW 'X'	
4514 44 41	7230 DW 'AD'	
4516 54 41	7240 DW 'AT'	
4518 00	7250 DB 0	/MESSAGE TERMINATOR
4519 4E	7260 MSG MOV C,M	/FETCH A STORED CHAR. FROM MEM
451A CD C3 44	7270 CALL HISPD	/PRINT THAT CHARACTER
451D 79	7280 MOV A,C	/RESTORE THE CHARACTER
451E FE 00	7290 CPI 0	/WAS IT A MESSAGE TERMINATOR?
4520 C8	7300 RZ	/YES, THEN QUIT AND CONTINUE...
4521 23	7310 INX H	/POINT TO NEXT STORED CHAR.
4522 C3 19 45	7320 JMP MSG	/PRINT MORE
4525 53 54	7330 STADD DW 'TS'	/ASCII STORAGE FOR-
4527 41 52	7340 DW 'RA'	/START ADDRESS?"
4529 54 20	7350 DW 'T'	
452B 41 44	7360 DW 'DA'	
452D 44 52	7370 DW 'RD'	
452F 45 53	7380 DW 'SE'	
4531 53 3F	7390 DW '?S'	
4533 20 00	7400 DW 0020H	/SPACE AND MESSAGE TERMINATOR
4535 45 4E	7410 ENADD DW 'NE'	/ASCII STORAGE FOR-
4537 44 20	7420 DW 'D'	/END ADDRESS?"
4539 41 44	7430 DW 'DA'	
453B 44 52	7440 DW 'RD'	
453D 45 53	7450 DW 'SE'	
453F 53 3F	7460 DW '?S'	
4541 20 00	7470 DW 0020H	/SPACE AND MESSAGE TERMINATOR
4543 48 45	7480 LNONE DW 'EH'	/ASCII STORAGE FOR-
4545 58 20	7490 DW 'X'	/LINE 1 OF HEADER MESSAGE

4547 20 20	7500 DW ' '
4549 48 45	7510 DW 'EH'
454B 58 20	7520 DW 'X'
454D 20 20	7530 DW ' '
454F 20 20	7540 DW ' '
4551 20 20	7550 DW ' '
4553 20 20	7560 DW ' '
4555 20 4C	7570 DW 'L'
4557 41 42	7580 DW 'BA'
4559 4C 45	7590 DW 'EL'
455B 20 20	7600 DW ' '
455D 20 20	7610 DW ' '
455F 41 53	7620 DW 'SA'
4561 43 49	7630 DW 'IC'
4563 49 20	7640 DW 'I'
4565 20 20	7650 DW ' '
4567 20 20	7660 DW ' '
4569 20 20	7670 DW ' '
456B 4F 43	7680 DW 'CO'
456D 54 41	7690 DW 'AT'
456F 4C 20	7700 DW 'L'
4571 20 20	7710 DW ' '
4573 20 44	7720 DW 'D'
4575 41 54	7730 DW 'TA'
4577 41 00	7740 DW 0041H
4579 41 44	7750 LNTWO DW 'DA'
457B 44 52	7760 DW 'RD'
457D 20	7770 DB ' '
457E 20 44	7780 DW 'D'
4580 41 54	7790 DW 'TA'
4582 41 20	7800 DW 'A'
4584 49 4E	7810 DW 'NI'
4586 53 54	7820 DW 'TS'
4588 20 20	7830 DW ' '
458A 20 20	7840 DW ' '
458C 48 49	7850 DW 'IH'
458E 4C 4F	7860 DW 'OL'
4590 20 20	7870 DW ' '
4592 20 20	7880 DW ' '
4594 20 31	7890 DW '1'
4596 20 20	7900 DW ' '
4598 20 32	7910 DW '2'
459A 20 20	7920 DW ' '
459C 20 33	7930 DW '3'
459E 20 20	7940 DW ' '
45A0 20 41	7950 DW 'A'
45A2 44 44	7960 DW 'DD'
45A4 52 45	7970 DW 'ER'
45A6 53 53	7980 DW 'SS'
45A8 20 20	7990 DW ' '

/ASCII STORAGE FOR-
/LINE 2 OF HEADER MESSAGE

45AA 31 20	8000 DW ' 1'	
45AC 20 20	8010 DW ' '	
45AE 32 20	8020 DW ' 2'	
45B0 20 20	8030 DW ' '	
45B2 33 00	8040 DW 0033H	
45B4 21 00 00	8050 HEX LXI H,0	/THIS ROUTINE INPUTS 4
45B7 CD D9 45	8060 NXT CALL REP	/VALID HEX CHARACTERS AND
45BA FE 30	8070 CPI '0'	/CONVERTS THEM TO A BINARY
45BC D8	8080 RC	/VALUE IN THE H & L REGISTERS
45BD FE 3A	8090 CPI ':'	/THE ROUTINE TERMINATES WITH A
45BF DA CA 45	8100 JC NUM	/CARRAGE RETURN OR IF AN INVALID
45C2 FE 41	8110 CPI 'A'	/HEX CHAR. IS ENTERED. IF THE
45C4 D8	8120 RC	/ROUTINE ENDS WITH AN ERROR,
45C5 FE 47	8130 CPI 'G'	/"NON HEX DATA" WILL BE PRINTED
45C7 D0	8140 RNC	/AND THIS PROGRAM WILL TERMINATE
45C8 C6 09	8150 ADI 9	/AND JUMP BACK TO THE SYSTEM
45CA E6 0F	8160 NUM ANI 15	/MONITOR. IF NO SYSTEM
45CC 29	8170 DAD H	/MONITOR IS AVAILABLE IN PROM
45CD D8	8180 RC	/THE JMP IN THE ROUTINE LABELED
45CE 29	8190 DAD H	/"BAD" SHOULD BE CHANGED TO 4000H
45CF D8	8200 RC	
45D0 29	8210 DAD H	
45D1 D8	8220 RC	
45D2 29	8230 DAD H	
45D3 D8	8240 RC	
45D4 B5	8250 ORA L	
45D5 6F	8260 MOV L,A	
45D6 C3 B7 45	8270 JMP NXT	
45D9 CD D5 44	8280 REP CALL INPUT	/INPUT A CHAR. FROM KBD.
45DC CD C3 44	8290 CALL HISPD	/ECHO THE CHAR.
45DF 79	8300 MOV A,C	/DUPLICATE CHAR. IN REG. A
45E0 C9	8310 RET	/CONTINUE...
45E1 DB FF	8320 OCTAL IN 0FFH	/INPUT THE SENSE SWITCHES
45E3 E6 02	8330 ANI 2	/WAS SWITCH 9 SET?
45E5 C8	8340 RZ	/NO, THEN SKIP THE OCTAL LOADING
45E6 11 8A 44	8350 LXI D,OUTBUF+36	/POINT TO HIGH OCT. ADD.
45E9 2A F5 44	8360 LHLD OSAVE	/LOAD OCTAL ADDRESS OF INST.
45EC 44	8370 MOV B,H	/PUT HIGH ORDER IN REG. B
45ED CD 21 46	8380 CALL OCTCODE	/WRITE HIGH ADD. IN OUTBUF
45F0 45	8390 MOV B,L	/PUT LOW ORDER IN REG. B
45F1 11 8E 44	8400 LXI D,OUTBUF+40	/POINT TO LOW ADD. SLOT
45F4 CD 21 46	8410 CALL OCTCODE	/WRITE LOW ADD. IN OUTBUF
45F7 2A F5 44	8420 LHLD OSAVE	/RELOAD THE INSTRUCTION ADD.
45FA 46	8430 MOV B,M	/PUT FIRST BYTE IN REG. B
45FB 11 93 44	8440 LXI D,OUTBUF+45	/POINT TO FIRST BYTE SLOT
45FE CD 21 46	8450 CALL OCTCODE	/WRITE 1ST. BYTE IN OUTBUF
4601 21 75 44	8460 LXI H,OUTBUF+15	/POINT TO LABEL IN OUTBUF
4604 3A F3 44	8470 LDA CHAR2	/LOAD THE SECOND ASCII CHAR.
4607 47	8480 MOV B,A	/SAVE 2ND ASCII CHAR IN REG. B
4608 7E	8490 MOV A,M	/FETCH CHAR. FROM LABEL SLOT

4609 FE 20	8500 CPI ' ' /WAS LABEL CHAR. A SPACE?
460B CA 14 46	8510 JZ OT3 /YES, THEN SKIP BYTE TWO
460E 11 97 44	8520 LXI D,OUTBUF+49 /POINT TO BYTE 2 SLOT
4611 CD 21 46	8530 CALL OCTCODE /WRITE BYTE 2 IN OUTBUF
4614 23	8540 OT3 INX H /POINT TO BYTE THREE OF-
4615 23	8550 INX H /THE INSTRUCTION IN MEMORY
4616 3A F4 44	8560 LDA CHAR3 /LOAD THE THIRD ASCII CHAR.
4619 47	8570 MOV B,A /SAVE 3RD. ASCII CHAR. IN REG. B
461A 7E	8580 MOV A,M /FETCH 3RD. BYTE FROM MEMORY
461B FE 20	8590 CPI ' ' /WAS THIRD BYTE A SPACE?
461D C8	8600 RZ /YES, THEN SKIP LOADING BYTE 3
461E 11 9B 44	8610 LXI D,OUTBUF+53 /POINT TO BYTE 3 SLOT
4621 0E 03	8620 OCTCODE MVI C,3 /LOAD NUMBER OF CHARS.
4623 AF	8630 XRA A /THIS ROUTINE CONVERTS THE-
4624 78	8640 MOV A,B /BINARY VALUE IN REG. B TO ASCII-
4625 C3 29 46	8650 JMP ROT /CHARACTERS AND WRITES THEM IN-
4628 F1	8660 ONXT POP PSW /THE SLOT IN THE OUTPUT-
4629 17	8670 ROT RAL /BUFFER POINTED TO BY-
462A 17	8680 RAL /REGISTERS D & E WHICH MUST BE-
462B 17	8690 RAL /SET UP BEFORE THIS ROUTINE IS-
462C F5	8700 PUSH PSW /ENTERED
462D E6 07	8710 ANI 7
462F F6 30	8720 ORI '0'
4631 12	8730 STAX D
4632 13	8740 INX D
4633 0D	8750 DCR C
4634 C2 28 46	8760 JNZ ONXT
4637 F1	8770 POP PSW
4638 C9	8780 RET
4639 11 6D 44	8790 INVAL LXI D,OUTBUF+7 /POINT-MNEMONIC SLOT
463C 0E 07	8800 MVI C,7 /LOAD NO. OF CHARS. TO ELIMINATE
463E 3E 2D	8810 MVI A,'-' /LOAD AN INVALIDATION CHAR.
4640 12	8820 INV2 STAX D /WRITE "-" IN OUTBUF
4641 13	8830 INX D /POINT TO NEXT SLOT IN OUTBUF
4642 0D	8840 DCR C /DECREMENT CHAR. COUNT
4643 C8	8850 RZ /IF DONE, CONTINUE...
4644 C3 40 46	8860 JMP INV2 /GO INVALIDATE MORE CHARS.
4647 2A EE 44	8870 FETC2 LHLD PGMCT /GET CURRENT PGM. COUNT
464A	8880 DCX H /CORRECT VALUE
464B C9	8890 RET /CONTINUE...
464C 3E 2D	8900 SET MVI A,45 /LOAD NO. OF LINES/PAGE
464E 32 54 46	8910 STA COUNT /SAVE NO. OF LINES COUNT
4651 C3 9C 40	8920 JMP DISAS /GO DISASSEMBLE THE PROGRAM
4654 00	8930 COUNT DB 0 /STORAGE FOR PAGE LINE COUNT
4655 3A 54 46	8940 PAGE LDA COUNT /LOAD CURRENT LINE COUNT
4658 3D	8950 DCR A /DECREMENT LINE COUNT
4659 32 54 46	8960 STA COUNT /SAVE NEW LINE COUNT
465C FE 00	8970 CPI 0 /HAS A PAGE OF LINES BEEN PRINTED?
465E CA 64 46	8980 JZ SKIP /YES, THEN PRINT SPACE & DIV. PG.
4661 C3 9C 40	8990 JMP DISAS /CONTINUE DISASSEMBLING

4664 06 09	9000 SKIP MVI B,9	/LOAD NO. OF CR'S TO PRINT
4666 CD 7C 46	9010 CALL SK3	/PRINT 9 CRLF'S
4669 0E 2D	9020 SET2 MVI C, '-'	/LOAD PAGE DIVIDER CHAR.
466B 06 48	9030 MVI B,72	/LOAD TERMINAL WIDTH
466D CD C3 44	9040 SETA CALL HISPD	/PRINT A '-'
4670 05	9050 DCR B	/DECREMENT TERMINAL WIDTH COUNT
4671 C2 6D 46	9060 JNZ SETA	/NOT DONE THEN PRINT MORE
4674 CD 7A 46	9070 SET3 CALL SK2	/PRINT 7 CRLF'S
4677 C3 2D 40	9080 JMP HEADR	/PRINT HEADER MESSAGE
467A 06 07	9090 SK2 MVI B,7	/LOAD NO. OF CRLF'S TO PRINT
467C CD F7 44	9100 SK3 CALL CRLF	/PRINT A CRLF
467F 05	9110 DCR B	/DECREMENT CRLF COUNT
4680 C8	9120 RZ	/IF DONE CONTINUE...
4681 C3 7C 46	9130 JMP SK3	/PRINT MORE CRLF'S
4684 22 F0 44	9140 STORE SHLD DONE	/SAVE ENDING PGM. COUNT
4687 CD F7 44	9150 CALL CRLF	/PRINT A CRLF
468A C3 69 46	9160 JMP SET2	/GO PRINT SPACE AND HEADER
468D	9170 DS 30H	/ALLOW SPACE FOR STACK
46BD	9180 STACK EQU \$	/STACK POINTER SET HERE
46BD	9190 END EQU \$	/MODIFICATIONS GO HERE

SYMBOL TABLE

START	4000	HEADR	402D	FETCH	4048	REGLD	4051	REL1	405B
REL2	4062	PRINT	4066	SPRN	4068	MASK	4071	CONDL	407A
LXICD	408A	DISAS	409C	ONE	40C1	TWO	40CC	THREE	40D7
RST	4143	CALL	4158	JMP	4165	RET	4172	LXI	417F
LX2	4192	LXIP	41AD	MVI	41C0	DCR	41D9	INR	41E2
INR2	41E8	INR3	41EE	ADD	41F7	MOV	420B	MOV2	4221
BYTE3	422D	BYT3	423B	BYTE2	4259	BYT2	4267	BYTE1	4276
BAD	4287	HEXOT	4296	FINIS	42A5	OVER	42C7	LOW	42DC
XCODE	42E7	CLRBV	42F4	MORE	42FC	CLRCH	4305	MRE2	430A
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	44F0	CHAR1	44F2	CHAR2	44F3	CHAR3	44F4	OSAVE	44F5
CRLF	44F7	DATA	450C	MSG	4519	STADD	4525	ENADD	4535
LNONE	4543	LNTWO	4579	HEX	45B4	NXT	45B7	NUM	45CA
REP	45D9	OCTAL	45E1	OT3	4614	OCTC0	4621	ONXT	4628
ROT	4629	INVAL	4639	INV2	4640	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								

