

START

```

C000          0010 *** VECTOR ONE MONITOR - VERSION 2.0
C000          0020 * R. S. HARP 10/15/77
C000          0030 * OPTION C FOR BITSTREAMER I/O
C000          0040 *
C000          0050 * SYSTEM EQUATES
C000          0060 CONS EQU 3          CONSOLE STAT PORT
C000          0070 COND EQU 2        CONSOLE DATA PORT
C000          0072 TBE EQU 1         TRANS BUFF EMPTY
C000          0074 RDA EQU 2         REC DATA AVAIL
C000          0076 STPOL EQU 0       STATUS POLARITY
C000          0080 CASD EQU 6FH      CASS DATA PORT
C000          0090 CASC EQU 6EH      CASS STAT PORT
C000          0100 SPTR EQU 0CFD0H   STACK POINTER
C000          0110 BOOT EQU 0E900H   DISK BOOT LOADER
C000          0120 DOS EQU 02028H    DOS RESTART
C000          0130 *
C000          0140 ***** COMMAND FORMAT *****
C000          0150 * A SSSS FFFF ASCII DUMP
C000          0160 * B JUMP TO BOOTSTRAP LOADER
C000          0170 * C SSSS FFFF CCCC COMPARE BLOCKS
C000          0180 * D SSSS FFFF DUMP IN HEX
C000          0185 * E JUMP TO EXTENSION PROM
C000          0187 * F SSSS FFFF DD DD TWO BYTE SEARCH
C000          0190 * G SSSS GO TO AND EXECUTE
C000          0195 * I PP INPUT FROM PORT
C000          0200 * J JUMP TO DOS
C000          0210 * K JUMP TO 0000
C000          0220 * L SSSS FFFF LOAD AND GO
C000          0230 * M SSSS FFFF DDDD MOVE BLOCK
C000          0240 * N NON DESTRUCTIVE MEMORY TEST
C000          0245 * O PP DD OUTPUT TO PORT
C000          0250 * P LLLL PROGRAM MEMORY
C000          0260 * Q SSSS FFFF COMPUTE CHECKSUM
C000          0270 * R SSSS FFFF READ CASSETTE
C000          0280 * S SSSS FFFF DD SEARCH FOR SINGLE BYTE
C000          0290 * T SSSS FFFF TEST MEMORY
C000          0300 * U GENERATES TARBELL SYNC
C000          0310 * V SSSS FFFF VERIFY TAPE
C000          0320 * W SSSS FFFF WRITE TAPE
C000          0325 * X SSSS FFFF DDDD EXCHANGE BLOCK
C000          0327 * Y SSSS FFFF RELOCATING LOADER
C000          0330 * Z SSSS FFFF DD ZERO OR FILL MEMORY
C000          0340 *****
C000          0350 *
C000 C3 03 C0  0360 MONIT JMP INIT
C003 3E CE    0370 INIT MVI A,OCEH
C005 D3 03    0372      OUT CONS
C007 3E 27    0374      MVI A,27H
C009 D3 03    0375      OUT CONS
C00B 31 D0 CF 0380 START LXI SP,SPTR
C00E CD A6 CO 0390      CALL CRLF
C011 3E 2A    0400      MVI A,'*'          PRINT "*"
C013 CD 98 CO 0410      CALL PTCN
C016 CD BD CO 0420      CALL RDCN          READ KEYBOARD

```

C019	E6	5F	0430	ANI	5FH	UPPER AND LOWER	
C01B	F5		0440	PUSH	PSW	SAVE INPUT	
C01C	CD	96 CO	0450	CALL	SPCE		
C01F	F1		0460	POP	PSW	SAVE INPUT	
C020	21	0B CO	0470	LXI	H, START		
C023	E5		0480	PUSH	H		
C024	FE	41	0490	CPI	'A'		
C026	D8		0500	RC		TOO SMALL	
C027	FE	5B	0510	CPI	05BH		
C029	D0		0520	RNC		TOO LARGE	
C02A	21	B5 CO	0530	LXI	H, CMDTB+7EH		
C02D	F5		0540	PUSH	PSW		
C02E	87		0545	ADD	A		
C02F	85		0550	ADD	L		
C030	6F		0560	MOV	L, A		
C031	5E		0570	MOV	E, M		
C032	23		0580	INX	H		
C033	56		0590	MOV	D, M		
C034	EB		0600	XCHG			
C035	F1		0605	POP	PSW		
C036	E9		0610	PCHL		AWAY WE GO	
C037			0620	* COMMAND TABLE			
C037	EE	C1	0630	CMDTB	DW	DISP	A
C039	00	E9	0640		DW	BOOT	B
C03B	AC	C2	0650		DW	COMPR	C
C03D	EE	C1	0660		DW	DISP	D
C03F	00	C4	0670		DW	MONIT+0400H	E
C041	C5	C2	0680		DW	SRCH	F
C043	6B	CO	0690		DW	EXEC	G
C045	0B	CO	0700		DW	START	H
C047	21	C3	0710		DW	PINPT	I
C049	28	20	0720		DW	DOS	J
C04B	00	00	0730		DW	0	K
C04D	28	C1	0740		DW	CINR	L
C04F	71	C2	0750		DW	MOVEB	M
C051	93	C2	0760		DW	NDMT	N
C053	0A	C3	0770		DW	POUTP	O
C055	23	C2	0780		DW	PGM	P
C057	ED	CO	0790		DW	COUTR	Q
C059	28	C1	0800		DW	CINR	R
C05B	C5	C2	0810		DW	SRCH	S
C05D	76	C1	0820		DW	TMEM	T
C05F	36	C3	0830		DW	SYNC	U
C061	28	C1	0840		DW	CINR	V
C063	ED	CO	0850		DW	COUTR	W
C065	71	C2	0860		DW	MOVEB	X
C067	42	C3	0870		DW	RLODR	Y
C069	5D	C2	0880		DW	ZEROM	Z
C06B			0890	*			
C06B			0900	***	EXECUTE THE PROGRAM AT THE ADDRESS	***	
C06B			0910	*			
C06B	CD	70 CO	0920	EXEC	CALL	AHEX	READ ADD FROM KB
C06E	EB		0930		XCHG		
C06F	E9		0960		PCHL		JUMP TO IT
C070			0970	*			
C070			0980	***	CONVERT UP TO 4 HEX DIGITS TO BIN		

C070			0990	*			
C070	0E	04	1000	AHEX	MVI	C,4	COUNT OF 4 DIGITS
C072	21	00 00	1010	AHE0	LXI	H,0	16 BIT ZERO
C075	CD	BD CO	1020	AHE1	CALL	RDCN	READ A BYTE
C078	FE	30	1021		CPI	30H	
C07A	DA	0B CO	1022		JC	START	
C07D	FE	3A	1023		CPI	':'	
C07F	D4	B0 CO	1024		CNC	ALPH	
C082	29		1030		DAD	H	SHIFT 4 LEFT
C083	29		1040		DAD	H	
C084	29		1050		DAD	H	
C085	29		1060		DAD	H	
C086	D6	30	1070		SUI	48	ASCII BIAS
C088	FE	0A	1080		CPI	10	DIGIT 0-10
C08A	DA	8F CO	1090		JC	ALF	
C08D	D6	07	1100		SUI	7	ALPHA BIAS
C08F	85		1110	ALF	ADD	L	
C090	6F		1120		MOV	L,A	
C091	0D		1130		DCR	C	4 DIGITS?
C092	C2	75 CO	1140		JNZ	AHE1	KEEP READING
C095	EB		1150		XCHG		
C096	3E	20	1160	SPCE	MVI	A,20H	PRINT SPACE
C098	F5		1170	PTCN	PUSH	PSW	
C099	DB	03	1180	PTLOP	IN	CONS	
C09B	EE	00	1181		XRI	STPOL	
C09D	E6	01	1182		ANI	TBE	
C09F	CA	99 CO	1200		JZ	PTLOP	
COA2	F1		1210		POP	PSW	THEN RECOVER A
COA3	D3	02	1220		OUT	COND	AND PRINT IT
COA5	C9		1230		RET		RETURN FROM PTCN
COA6	3E	0D	1240	CRLF	MVI	A,0DH	PRINT CR
COA8	CD	98 CO	1250		CALL	PTCN	
COAB	3E	0A	1260		MVI	A,0AH	
COAD	C3	98 CO	1270		JMP	PTCN	
COB0	FE	41	1271	ALPH	CPI	'A'	
COB2	DA	0B CO	1272		JC	START	
COB5	E6	5F	1273		ANI	5FH	
COB7	FE	47	1274		CPI	'G'	
COB9	D2	0B CO	1275		JNC	START	
COBC	C9		1276		RET		
COBD			1280	*			
COBD			1290	***	READ FROM CONSOLE TO REG A	***	
COBD			1300	*			
COBD	DB	03	1310	RDCN	IN	CONS	READ KB STAT
COBF	EE	00	1315		XRI	STPOL	
COC1	E6	02	1320		ANI	RDA	
COC3	CA	BD CO	1330		JZ	RDCN	
COC6	DB	02	1340		IN	COND	READ FROM KB
COC8	E6	7F	1350		ANI	7FH	STRIP OFF MSB
COCA	C3	98 CO	1360		JMP	PTCN	ECHO ONTO PRINTER
COCD	CD	DC CO	1361	PAUSE	CALL	CNTLC	
COD0	FE	20	1362		CPI	20H	
COD2	C0		1363		RNZ		
COD3	CD	DC CO	1364	PLOOP	CALL	CNTLC	
COD6	FE	20	1365		CPI	20H	
COD8	C2	D3 CO	1366		JNZ	PLOOP	

CODB	C9		1367		RET		
CODC	DB	03	1368	CNTLC	IN	CONS	
CODE	EE	00	1369		XRI	STPOL	
COE0	E6	02	1370		ANI	RDA	
COE2	C8		1371		RZ		
COE3	DB	02	1372		IN	COND	
COE5	E6	7F	1373		ANI	7FH	
COE7	FE	03	1374		CPI	03	
COE9	CA	0B	1375		JZ	START	
COEC	C9		1376		RET		
COED			1380	*** CASSETTE	INTERFACE	OUTPUT	ROUTINE ***
COED			1390	*			
COED	F5		1400	COUTR	PUSH	PSW	SAVE CONTROL
COEE	CD	03	1410		CALL	TAHEX	
COF1	06	00	1420		MVI	B,0	START CHECKSUM = 0
COF3	F1		1425		POP	PSW	
COF4	4F		1426		MOV	C,A	
COF5	CD	16	1430		CALL	COUT	START BYTE OUT
COF8	3E	E6	1440		MVI	A,0E6H	SEND SYNC BYTE
COFA	CD	16	1450		CALL	COUT	TO CASSETTE
COFD	7E		1460	COLOP	MOV	A,M	GET DATA FROM MEM
COFE	CD	16	1470		CALL	COUT	SEND TO CASSETTE
C101	80		1480		ADD	B	ADD TO CHECKSUM
C102	47		1490		MOV	B,A	
C103	CD	CD	1492		CALL	PAUSE	
C106	CD	54	1500		CALL	BMP	
C109	C2	FD	1510		JNZ	COLOP	REPEAT LOOP
C10C	78		1520		MOV	A,B	GET CHECKSUM
C10D	CD	16	1530		CALL	COUT	OUTPUT IT
C110	CD	D4	1540		CALL	PT2	PRINT CHECKSUM
C113	C3	0B	1550		JMP	START	GET ANOTH COMMND
C116	F5		1560	COUT	PUSH	PSW	SAVE A AND FLAGS
C117	79		1562		MOV	A,C	
C118	FE	51	1564		CPI	'Q'	
C11A	CA	24	1566		JZ	CHKSM	
C11D	DB	6E	1570	CLOP	IN	CASC	READ CASS STATUS
C11F	E6	20	1580		ANI	20H	LOOK AT BIT 5
C121	C2	1D	1590		JNZ	CLOP	TRY AGAIN?
C124	F1		1600	CHKSM	POP	PSW	
C125	D3	6F	1610		OUT	CASD	SEND DATA TO CASS
C127	C9		1620		RET	RETURN	FROM COUT
C128			1630	*			
C128			1640	*** CASSETTE	INPUT	ROUTINE	***
C128			1650	*			
C128	F5		1660	CINR	PUSH	PSW	SAVE CONTROL CHAR
C129	3E	10	1670		MVI	A,10H	USE BIT 4 IN REG A
C12B	D3	6E	1680		OUT	CASC	TO RESET CASS INT
C12D	CD	03	1690		CALL	TAHEX	READ ADDRESSES
C130	F1		1710		POP	PSW	GET CONTROL CHAR
C131	E5		1720		PUSH	H	SAVE START ADD
C132	F5		1730		PUSH	PSW	UNDER CONTROL CHAR
C133	06	00	1740		MVI	B,0	SET CHECKSUM = 0
C135	CD	6C	1750	CILOP	CALL	CIN	READ FM CONS
C138	4F		1760		MOV	C,A	SAVE IT IN REG C
C139	F1		1770		POP	PSW	GET CONTROL CHAR
C13A	F5		1780		PUSH	PSW	SAVE IT BACK

C13B	FE	56		1790		CPI	'V'	IS IT A V?
C13D	79			1800		MOV	A,C	GET BACK DATA BYTE
C13E	CA	42	C1	1810		JZ	CINO	IF C, DON'T STORE
C141	77			1820		MOV	M,A	IF NOT , STORE
C142	80			1830	CINO	ADD	B	ADD TO CHECKSUM
C143	47			1840		MOV	B,A	
C144	CD	CD	C0	1845		CALL	PAUSE	
C147	CD	54	C2	1850		CALL	BMP	
C14A	C2	35	C1	1860		JNZ	CILOP	READ MORE
C14D	CD	6C	C1	1870		CALL	CIN	READ LAST BYTE
C150	F5			1880		PUSH	PSW	
C151	CD	D4	C1	1890		CALL	PT2	PRINT CHECKSUM
C154	CD	96	C0	1900		CALL	SPCE	SPACE OVER
C157	F1			1910		POP	PSW	
C158	B8			1920		CMP	B	COMP TO CHKSUM
C159	3E	45		1930		MVI	A,'E'	PRINT E FOR ERROR
C15B	C2	66	C1	1940		JNZ	CERR	PRINT NOW IF ERROR
C15E	F1			1950		POP	PSW	RECOVER CTL CHAR
C15F	FE	4C		1960		CPI	'L'	IF NOT L
C161	C2	66	C1	1970		JNZ	CERR	DON'T EXECUTE
C164	E1			1980		POP	H	OTHERWISE, EXECJTE
C165	E9			1990		PCHL	AT	STARTING ADDRESS
C166	CD	98	C0	2000	CERR	CALL	PTCN	PRINT V,E, OR R
C169	C3	0B	C0	2010		JMP	START	
C16C	DB	6E		2020	CIN	IN	CASC	READ STATUS
C16E	E6	10		2030		ANI	10H	LOOK AT BIT 4
C170	C2	6C	C1	2040		JNZ	CIN	WAIT UNTIL LOW
C173	DB	6F		2050		IN	CASD	READ DATA FM CASS
C175	C9			2060		RET	RETURN	FROM CIN
C176				2070	*			
C176				2080	***	MEMORY TEST ROUTINE	***	
C176				2090	*			
C176	CD	03	C3	2100	TMEM	CALL	TAHEX	READ ADDRESSES
C179	01	5A	5A	2120		LXI	B,5A5AH	INI B,C
C17C	CD	A7	C1	2130	CYCL	CALL	RNDM	
C17F	C5			2140		PUSH	B	KEEP ALL REGS
C180	E5			2150		PUSH	H	
C181	D5			2160		PUSH	D	
C182	CD	A7	C1	2170	TLOP	CALL	RNDM	
C185	70			2180		MOV	M,B	WRITE IN MEM
C186	CD	54	C2	2190		CALL	BMP	
C189	C2	82	C1	2200		JNZ	TLOP	REPEAT LOOP
C18C	D1			2210		POP	D	
C18D	E1			2220		POP	H	RESTORE ORIG
C18E	C1			2230		POP	B	VALUES OF
C18F	E5			2240		PUSH	H	
C190	D5			2250		PUSH	D	
C191	CD	A7	C1	2260	RLOP	CALL	RNDM	GEN NEW SEQ
C194	7E			2270		MOV	A,M	READ MEM
C195	B8			2280		CMP	B	COMP MEM
C196	C4	C8	C1	2290		CNZ	ERR	CALL ERROR ROUT
C199	CD	54	C2	2300		CALL	BMP	
C19C	C2	91	C1	2310		JNZ	RLOP	
C19F	D1			2320		POP	D	
C1A0	E1			2330		POP	H	
C1A1	CD	CD	C0	2335		CALL	PAUSE	

C1A4	C3	7C	C1	2340	JMP	CYCL		
C1A7				2350	***	THIS ROUTINE GENERATES	RANDOM NOS ***	
C1A7	78			2360	RNDM	MOV	A,B	LOOK AT B
C1A8	E6	B4		2370		ANI	OB4H	MASK BITS
C1AA	A7			2380		ANA	A	CLEAR CY
C1AB	EA	AF	C1	2390		JPE	PEVE	JUMP IF EVEN
C1AE	37			2400		STC		
C1AF	79			2410	PEVE	MOV	A,C	LOOK AT C
C1B0	17			2420		RAL		ROTATE CY IN
C1B1	4F			2430		MOV	C,A	RESTORE C
C1B2	78			2440		MOV	A,B	LOOK AT B
C1B3	17			2450		RAL		ROTATE CY IN
C1B4	47			2460		MOV	B,A	RESTORE B
C1B5	C9			2470		RET		RETURN W NEW B,C
C1B6				2480	*			
C1B6				2490	***	ERROR PRINT OUT ROUTINE		
C1B6				2500	*			
C1B6	CD	A6	C0	2510	PTAD	CALL	CRLF	PRINT CR,LF
C1B9	CD	CD	C0	2515		CALL	PAUSE	
C1BC	7C			2520		MOV	A,H	PRINT
C1BD	CD	D4	C1	2530		CALL	PT2	ASCII
C1C0	7D			2540		MOV	A,L	CODES
C1C1	CD	D4	C1	2550		CALL	PT2	FOR
C1C4	CD	96	C0	2560		CALL	SPCE	ADDRESS
C1C7	C9			2570		RET		
C1C8	F5			2580	ERR	PUSH	PSW	SAVE ACC
C1C9	CD	B6	C1	2590		CALL	PTAD	PRINT ADD.
C1CC	78			2600		MOV	A,B	DATA
C1CD	CD	D4	C1	2610		CALL	PT2	WRITTEN
C1D0	CD	96	C0	2620		CALL	SPCE	
C1D3	F1			2630		POP	PSW	DATA READ
C1D4	F5			2640	PT2	PUSH	PSW	
C1D5	CD	DC	C1	2650		CALL	BINH	
C1D8	F1			2660		POP	PSW	
C1D9	C3	E0	C1	2670		JMP	BINL	
C1DC	1F			2680	BINH	RAR		
C1DD	1F			2690		RAR		
C1DE	1F			2700		RAR		
C1DF	1F			2710		RAR		
C1E0	E6	0F		2720	BINL	ANI	0FH	LOW 4 BITS
C1E2	C6	30		2730		ADI	48	ASCII BIAS
C1E4	FE	3A		2740		CPI	58	DIGIT 0-9
C1E6	DA	98	C0	2750		JC	PTCN	
C1E9	C6	07		2760		ADI	7	DIGIT A-F
C1EB	C3	98	C0	2770		JMP	PTCN	
C1EE				2780	*			
C1EE				2790	***	DISPLAY MEMORY CONTENTS	***	
C1EE				2800	*			
C1EE	47			2810	DISP	MOV	B,A	SAVE CONTROL
C1EF	CD	03	C3	2820		CALL	TAHEX	READ ADDRESSES
C1F2	0E	10		2840	ENT1	MVI	C,16	LOC/LINE
C1F4	CD	B6	C1	2850		CALL	PTAD	
C1F7	78			2860	LP2	MOV	A,B	
C1F8	FE	41		2870		CPI	'A'	IS IT "A"?
C1FA	7E			2880		MOV	A,M	
C1FB	CA	0F	C2	2890		JZ	ASCD	DUMP ASCII



C26A	CD	54	C2	3470	CALL	BMP	COMP ADD, INCREMENT
C26D	C8			3480	RZ		RETURN IF DONE
C26E	C3	69	C2	3490	JMP	ZLOOP	CONTINUE TIL DONE
C271				3500	* MOVE	A BLOCK OF MEMORY	
C271	47			3510	MOVEB	MOV B,A	
C272	CD	03	C3	3520	CALL	TAHEX	READ ADDRESSES
C275	E5			3530	PUSH	H	
C276	CD	70	C0	3540	CALL	AHEX	
C279	EB			3550	XCHG		
C27A	E3			3560	XTHL		BACK TO NORMAL
C27B	4E			3570	MLOOP	MOV C,M	
C27C	E3			3580	XTHL		
C27D	78			3581	MOV	A,B	
C27E	FE	4D		3582	CPI	'M'	
C280	CA	87	C2	3583	JZ	NEXCH	
C283	7E			3584	MOV	A,M	
C284	E3			3585	XTHL		
C285	77			3586	MOV	M,A	
C286	E3			3587	XTHL		
C287	71			3590	NEXCH	MOV M,C	
C288	23			3600	INX	H	
C289	E3			3610	XTHL		
C28A	CD	54	C2	3620	CALL	BMP	
C28D	CA	0B	C0	3630	JZ	START	
C290	C3	7B	C2	3640	JMP	MLOOP	
C293				3860	* NON	DESTRUCTIVE MEMORY TEST	
C293	21	00	00	3870	NDMT	LXI H,0	START AT ZERO
C296	4E			3890	NDLOP	MOV C,M	
C297	06	FF		3900	MVI	B,OFFH	
C299	70			3910	MOV	M,B	
C29A	7E			3920	MOV	A,M	
C29B	B8			3930	CMP	B	
C29C	C2	C8	C1	3940	JNZ	ERR	PRINT ERROR
C29F	06	00		3941	MVI	B,0	
C2A1	70			3942	MOV	M,B	
C2A2	7E			3943	MOV	A,M	
C2A3	B8			3944	CMP	B	
C2A4	C2	C8	C1	3945	JNZ	ERR	
C2A7	71			3950	MOV	M,C	
C2A8	23			3960	INX	H	
C2A9	C3	96	C2	3980	JMP	NDLOP	
C2AC				3990	* COMPARE	TWO BLOCKS OF MEMORY	
C2AC	CD	03	C3	4000	COMPR	CALL TAHEX	
C2AF	E5			4020	PUSH	H	
C2B0	CD	70	C0	4030	CALL	AHEX	
C2B3	EB			4040	XCHG		
C2B4	7E			4050	VMLOP	MOV A,M	
C2B5	23			4060	INX	H	
C2B6	E3			4070	XTHL		
C2B7	BE			4080	CMP	M	
C2B8	46			4090	MOV	B,M	
C2B9	C4	C8	C1	4100	CNZ	ERR	
C2BC	CD	54	C2	4110	CALL	BMP	
C2BF	E3			4130	XTHL		
C2C0	C2	B4	C2	4140	JNZ	VMLOP	
C2C3	F1			4141	POP	PSW	



C2C4	C9	4142	RET			
C2C5		4160	* SEARCH FOR	SPECIFIC CODES		
C2C5	F5	4170	SRCH	PUSH	PSW	
C2C6	CD 03 C3	4180		CALL	TAHEX	
C2C9	E5	4190		PUSH	H	SAVE H
C2CA	0E 02	4210		MVI	C,2	COUNT OF 2
C2CC	CD 72 C0	4220		CALL	AHEO	READ 2 DIGITS
C2CF	EB	4230		XCHG		H=CODE,D=F
C2D0	45	4240		MOV	B,L	PUT CODE IN B
C2D1	E1	4250		POP	H	RESTORE H
C2D2	F1	4251		POP	PSW	
C2D3	FE 53	4252		CPI	'S'	
C2D5	F5	4253		PUSH	PSW	
C2D6	CA E2 C2	4254		JZ	CONT	
C2D9	E5	4255		PUSH	H	
C2DA	0E 02	4256		MVI	C,2	
C2DC	CD 72 C0	4257		CALL	AHEO	
C2DF	EB	4258		XCHG		
C2E0	4D	4259		MOV	C,L	
C2E1	E1	4260		POP	H	
C2E2	7E	4261	CONT	MOV	A,M	READ MEMORY
C2E3	B8	4270		CMP	B	COMPARE TO CODE
C2E4	C2 FB C2	4280		JNZ	SKP	SKIP IF NO COMP
C2E7	F1	4281		POP	PSW	FETCH CONTROL
C2E8	FE 53	4282		CPI	'S'	
C2EA	F5	4283		PUSH	PSW	
C2EB	CA F5 C2	4284		JZ	OBCP	
C2EE	23	4285		INX	H	
C2EF	7E	4286		MOV	A,M	
C2F0	2B	4287		DCX	H	
C2F1	B9	4288		CMP	C	
C2F2	C2 FB C2	4289		JNZ	SKP	
C2F5	23	4290	OBCP	INX	H	
C2F6	7E	4300		MOV	A,M	READ NEXT BYTE
C2F7	2B	4310		DCX	H	DECR ADDRESS
C2F8	CD C8 C1	4320		CALL	ERR	PRINT CODES
C2FB	CD 54 C2	4330	SKP	CALL	BMP	CHECK IF DONE
C2FE	C2 E2 C2	4340		JNZ	CONT	BACK FOR MORE
C301	F1	4345		POP	PSW	
C302	C9	4350		RET		
C303		4355	* SHORT ROUTINE TO	SAVE CODE		
C303	CD 70 C0	4360	TAHEX	CALL	AHEX	
C306	CD 70 C0	4370		CALL	AHEX	
C309	C9	4380		RET		
C30A		4389	* OUTPUT DATA TO	A PORT		
C30A	0E 02	4390	POUTP	MVI	C,2	
C30C	CD 72 C0	4400		CALL	AHEO	PORT NO IN E
C30F	0E 02	4410		MVI	C,2	
C311	CD 72 C0	4420		CALL	AHEO	PORT TO L DATA E
C314	55	4430		MOV	D,L	
C315	21 A0 CF	4440		LXI	H,SPTR-30H	
C318	36 C9	4450		MVI	M,0C9H	
C31A	2B	4460		DCX	H	
C31B	72	4470		MOV	M,D	
C31C	2B	4480		DCX	H	
C31D	36 D3	4490		MVI	M,0D3H	

C31F	7B			4500	MOV	A,E	
C320	E9			4510	PCHL		
C321				4519	* INPUT DATA	FROM A PORT	
C321	0E	02		4520	PINPT MVI	C,2	
C323	CD	72	C0	4530	CALL	AHEO	PORT NO TO E
C326	21	A0	CF	4540	LXI	H,SPTR-30H	
C329	36	C9		4550	MVI	M,0C9H	RETURN
C32B	2B			4560	DCX	H	
C32C	73			4570	MOV	M,E	PORT NO
C32D	2B			4580	DCX	H	
C32E	36	DB		4590	MVI	M,0DBH	
C330	CD	9E	CF	4600	CALL	SPTR-32H	
C333	C3	D4	C1	4610	JMP	PT2	
C336				4620	* OUTPUT TARBELL SYNC		
C336	3E	E6		4630	SYNC MVI	A,0E6H	SYNC BYTE
C338	4F			4635	MOV	C,A	
C339	CD	16	C1	4640	CALL	COUT	OUTPUT TO CASS
C33C	CD	DC	C0	4650	CALL	CNTLC	
C33F	C3	36	C3	4660	JMP	SYNC	
C342				5400	* THE VECTOR 1 RELOCATING	LOADER	
C342	3E	10		5460	RLODR MVI	A,10H	RESET CASS. INT.
C344	D3	6E		5520	OUT	CASC	
C346	CD	03	C3	5640	CALL	TAHEX	
C349	E5			5700	PUSH	H	SAVE START ADD.
C34A	06	00		5760	MVI	B,0	ZERO CHECKSUM
C34C	CD	6C	C1	5820	PASS1 CALL	CIN	READ CASS.
C34F	77			5880	MOV	M,A	WRITE IN MEMORY
C350	BE			5940	CMP	M	CHECK IF THERE
C351	C2	A8	C3	6000	JNZ	MERR	MEMORY ERROR
C354	80			6060	ADD	B	ADD TO CHECKSUM
C355	47			6120	MOV	B,A	PUT IT BACK
C356	CD	54	C2	6180	CALL	BMP	ALL DONE
C359	C2	4C	C3	6240	JNZ	PASS1	LOOP IF NOT
C35C	CD	6C	C1	6300	CALL	CIN	READ LAST BYTE
C35F	B8			6360	CMP	B	CHECKSUM AGREE?
C360	C2	A2	C3	6420	JNZ	LDERR	
C363	3E	10		6480	PASS2 MVI	A,10H	
C365	D3	6E		6540	OUT	CASC	RESET CASS.
C367	E1			6600	POP	H	GET START ADD.
C368	E5			6660	PUSH	H	SAVE IT BACK
C369	06	00		6720	MVI	B,0	ZERO CHECKSUM
C36B	CD	6C	C1	6780	L2LP CALL	CIN	READ CASS.
C36E	F5			6840	PUSH	PSW	
C36F	80			6900	ADD	B	ADD TO CHECKSUM
C370	47			6960	MOV	B,A	PUT IT BACK
C371	F1			7020	POP	PSW	
C372	BE			7080	CMP	M	
C373	CA	8F	C3	7140	JZ	OK	
C376	96			7200	SUB	M	
C377	FE	17		7260	CPI	17H	IS IT LSB?
C379	C2	8A	C3	7320	JNZ	HIBYT	JUMP IF NOT
C37C	E3			7380	XTHL		
C37D	7D			7440	MOV	A,L	LO BYTE OF START
C37E	E3			7500	XTHL		
C37F	86			7560	ADD	M	
C380	77			7620	MOV	M,A	REPLACE NEW ADD.

