

8080 Apple Monitor

Apple VI.0 ECT

Copyright (c) 1979 ECT

ALL RIGHTS RESERVED

Electronic Control Technology

763 Ramsey Ave.

Hillside, NJ 07205

(201) 686-8080

Electronic Control Technology

8080 Apple Monitor

The Apple Monitor is a program for the 8080 or Z80 microprocessors with executive commands and I/O handling routines. The author of the Apple Monitor is Roger Amidon of Applezap Corp. who also authored the Zapple Monitor <Z80 only version of the Apple Monitor> for TDL/Xitan. NOTE: The Apple Monitor has nothing to do with the Apple Computer and early versions of the Apple Monitor probably existed before the Apple Computer.

The Apple Monitor can be utilized as a software equivalent of a front panel. Memory, registers and I/O can be displayed and substituted from the system terminal. Debugging of both hardware and software is possible by use of the memory test or verifying blocks of memory or use of breakpoints.

The Apple Monitor is expandable. The user may add special routines for special I/O devices and/or additional commands. All software programs may utilize the I/O routines of the Apple Monitor through the vectors at the beginning of the Apple Monitor and thereby take advantage of the dynamic assigning of the I/O ports and routines. The Apple Monitor also includes many useful subroutines that may be used by user written programs.

USER WRITTEN COMMAND ROUTINES

Three command letters are available for user written command routines - 'I', 'K' & 'O'. Apple vectors to the user jump vectors for these commands; 'I' to F812, 'K' to F815 & 'O' to F818. JMP's to the actual user written routines should be placed at these locations. A RET instruction at the end of the user written routine will return control back to the monitor displaying the prompt '>'.

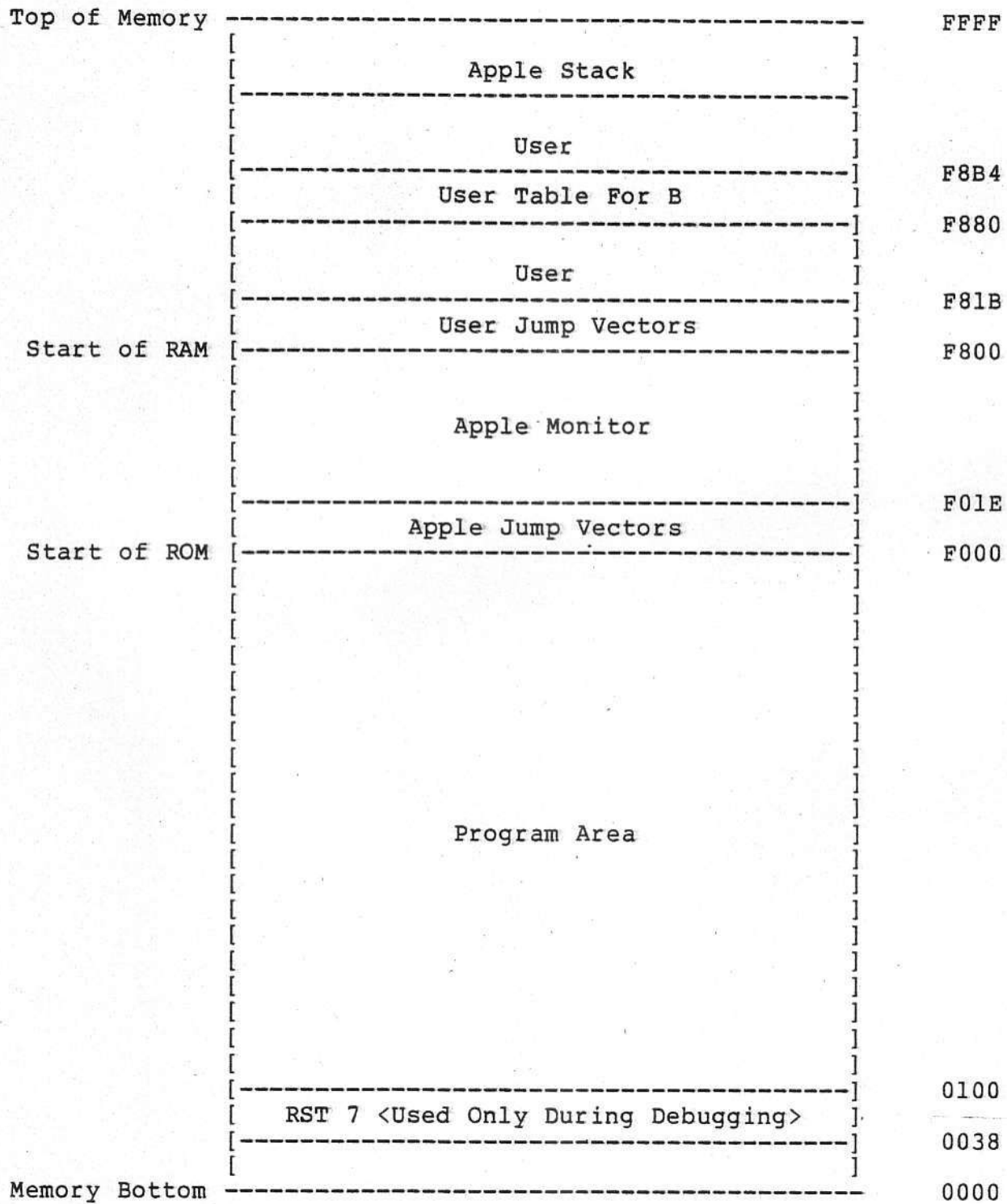
The Branch 'B' command also allows additional user written command routines with the use of a letter A - Z. Control is passed to the routine at the address found in the user table at F880 to F8B3.

USER WRITTEN I/O ROUTINES

Occasionally I/O devices require special routines. The user I/O jump vectors should be located at F800 through F811. Be careful not to modify any register except those called for and do not upset the stack pointer. PUSH and match with POP's to restore registers. Use a RET at the end of the routine to return control back to the monitor.

Electronic Control Technology

Memory Map



Electronic Control Technology

Apple Jump Vectors

F000	Begin Apple
F003	Console Input
F006	Reader Input
F009	Console Output
F00C	Punch Output
F00F	List Output
F012	Console Status
F015	I/O Assignment Check
F018	I/O Set
F01B	Memory Limit Check

User Jump Vectors

F800	Console Input
F803	Console Output
F806	Console Input Status
F809	User Defined Storage <Input>
F80C	User Defined Storage <Output>
F80F	User Defined Printer <List>
F812	I
F815	K
F818	O
F880	User Table For B
F8B3	

Assign

C	Console
C	CRT
P	Printer
B	Batch
U	User

P	Punch
D	Data Transfer Device
P	Printer
A	Alternate <Parallel>
U	User

R	Reader
D	Data Transfer Device
P	Printer
A	Alternate <Parallel>
U	User

L	List
C	CRT
P	Printer
D	Data Transfer Device
U	User

Electronic Control Technology

Apple V1.0 ECT

- A - Assign I/O
- B - Branch to user routine A-Z
- C - Undefined
- D - Display memory on console in Hex
- E - End of file tag for Hex dumps
- F - Fill memory with a constant
- G - GOTO an address with breakpoints
- H - Hex math sum & difference
- I * User defined
- J - Non-destructive memory test
- K * User defined
- L - Load a binary format file
- M - Move memory area to another address
- N - Nulls leader/trailer
- O * User defined
- P - Put ASCII into memory
- Q - Query I/O ports QI<N>=read I/O; QO<N,V>=send I/O
- R - Read a Hex file with checksum
- S - Substitute/examine memory in Hex
- T - Types the contents of memory in ASCII equivalent
- U - Unload memory in Binary format
- V - Verify memory block against another memory block
- W - Write a checksummed Hex file
- X - Examine/modify CPU registers
- Y - 'Yes there' find 'N' Bytes in memory
- Z - 'Z END' address of last R/W memory location

A - >A[cd]=[u]

Assigns a device to be a particular unit.
First letter of specifier is all that's required.

```
device:=      Console, Reader, Punch, List
unit:=
    if Console:  CRT, Printer, Batch Mode, User
    if Reader:   Data transfer, Printer, Alternate (parallel), User
    if Punch:    Data transfer, Printer, Alternate (parallel), User
    if List:     CRT, Printer, Data transfer, User
```

EXAMPLE: AC=P Assign Console = Printer
 AP=P Assign Punch = Printer

B - >B.[a-z]

Branches into address table based on letter a-z.
If no command implemented, address will
contain OFFFFH, which aborts command.

EXAMPLE: B.A

C - unused

D - >D[addr1],[addr2]<,byte>

Dumps memory from addr1 thru addr2, where <byte>
is optional depending on line width desired.
NOTE- Defaults to 16 bytes per line.

EXAMPLE: D0,1F
0000 C3 07 F7 C3 24 F7 C3 32 F5 C3 84 F5 C3 53 F5 C3
0010 65 F6 DB 76 C9 C3 CD F1 C3 DC F0 C3 38 F0 C3 38

E - >E<addr>

End of file is generated to assigned punch device.
<addr> is optional.

EXAMPLE: E
 E1234

F - >F[addr1],[addr2],[byte]

Fills from addr1 thru addr2 with byte.

EXAMPLE: F0,17FF,0

G - >G[addr1]<,<addr2><,<addr3>

Goes to addr1, and optionally set breakpoints at addr2 & addr3. If continuing from a breakpoint, the first parameter may be omitted. This will cause execution of whatever addr. is contained in the "P" register.

EXAMPLE: G1600,163E

H - >H[val1],[val2]

Hex math of: val1+val2 & val1-val2
is displayed.

EXAMPLE: H2000,102A
 302A 0FD6

I - unused

J - >J[addr1],[addr2]

Justifies memory from addr1 thru addr2. Any errors are displayed as:

addr 00100000
 where the "1" indicates a bad bit,
 in this case, bit 5, and addr is the
 location in memory the error occurred.

EXAMPLE: J800,17FF
 0F3D 00000010
 0F88 00000010
 16FD 10000000

K - unused

L - >L[addr]

Loads a binary file, starting at addr. The address following the last byte loaded will then be displayed on the console.

EXAMPLE: L800

 12A0

M - >M[addr1],[addr2],[addr3]

Moves a block of memory starting at addr1, ending at addr2, to the block starting at addr3.

EXAMPLE: M0,7FF,1000

N - >N

Null simply causes 60 blanks (00) to be sent to the currently assigned punch device.

EXAMPLE: N

O - unused

P - >P[addr]

Puts keyboard input directly into memory starting at addr. Inputting is terminated with a control-d. The address of the next byte that would have been loaded is displayed on the console.

EXAMPLE: P1000
The quick brown fox jumped over a byte.
Boy was he suprised.^D (control-d)
103D

Q - >QI[port]
>QO[port],[byte]

Q may be used to both display (QI) and send to (QO) any of the 256 I/O ports. When inputting, the results are displayed in binary; 00001101 with bit zero on the right.

When outputting, [port] will be sent [byte].

EXAMPLE: QI70 00000010
QO71,7

R - >R<addr1>,<addr2>

Read will load a normal hex file, or a relocatable hex file. Addr1 is an optional bias, which will be added to the load address, and addr2 is the optional relocation base which is used only with relocatable files.

EXAMPLE: R
R,300
R1000

S - >S[addr]

Substitutes memory, starting at addr.

EXAMPLE: S1000 54- 68- 65-79 20- 71- 75- 69- 63-
1008 6B- 20- 62- 72-

T - >T[addr1],[addr2]

Types out memory from addr1 thru addr2.

EXAMPLE: T1000,100F
 1000 Thy quick brown

 T0,1F
 0000 C.2pC.vC6vC..tCDtC+tC...qC,uC.

U - >U[addr1],[addr2]

Unloads memory from addr1 thru addr2.

EXAMPLE: U1000,17FF

V - >V[addr1],[addr2],[addr3]

Verifies the contents of memory from addr1 thru addr2
 with memory starting at addr3.

If a difference is found, first the address of
 the lower block is printed, then the byte found
 at that address, then the byte found at the address
 which would correspond relative to [addr3].

EXAMPLE: V0,402,F000

 0400 FF ED
 0401 FF 52
 0402 FF 20

W - >W[addr1],[addr2]<,byte>

Writes hex file records of the contents of memory from
 addr1 thru addr2, with the length of records
 of the optional <byte>. The default maximum
 length of records is 24 (18H).

EXAMPLE: W0,3FF
 W100,13A,FF

X - >X<*><r>

eXamines all the registers, or optionally, a single
 register. Typing X, followed by a carriage return
 displays the entire set, where X<r> followed by
 a space bar, will examine the contents of a single
 register, with the option of altering it's contents.
 The technique is similar to the 'S' command.

EXAMPLE: X
 A=18 B=AA C=28 D=A9 E=FA F=44 H=AC L=41 M=00 P=ADC2 S=AC96
 XA 18- AA- 28- A9-00 FA- 44- AC- 41- 00- ADC2-F000
 X
 A=18 B=AA C=28 D=00 E=FA F=44 H=AC L=41 M=00 P=ADC2 S=F000

Y - >Y[byte],<byte>,<byte>,<byte>.....

Y searches all of memory for a match on the series of <byte>s.
The starting address of each occurrence is displayed on
the console. Search string limit is 255 characters.

EXAMPLE: YCD,1E,F0
 0836
 0979
 1703
 231C

Z - >Z

Z alone causes the last R/W memory address to be displayed on
the console. Remember, this is the last R/W location
starting from the bottom. It is possible to have some
memory above this point, separated by either non-R/W
(ROM), or non-existent memory. In addition, the start
of R/W does not have to be at zero. The command will
first find R/W, and THEN the end of R/W.

EXAMPLE: Z
 17FF

A - Assign I/O

This command allows dynamic re-assignment of I/O configuration from within the monitor via keyboard input. Operation is straightforward with the exception of the "BATCH" mode. In this mode, the console input comes from the currently assigned reader, and any console output goes to the currently assigned list device. This allows batch processing; that is, you can store a series of commands in the reader device, and execute them automatically without keyboard intervention. The last command in such a series would normally be a re-assignment to a normal console.

B - Branch

This command allows user-defined commands to be executed directly from the monitor. The syntax requires a period (.) directly following the 'B', followed by a letter A-Z. Control is then passed to a command branch table located outside the P/ROM monitor. Routines are then written and their addresses placed in the table at the location corresponding to the appropriate letter. If the table address of any requested branch evaluates to 0FFFFH, it is considered an un-implemented command, and an error condition will occur. This would normally be the case when no memory exists in the branch table area.

C - unused

D - Display memory (in hex)

This command allows examination of memory in a hexadecimal format. An optional third parameter may be specified to allow different widths of printout. The default width is 16 bytes per line, with the address of each 16 byte block displayed on the left.

E - Eof

This generates an "end of file" string of data, and is used in conjunction with the "w" command. A 16 bit (two byte) address may be specified, which will, upon loading with the "R" command, be placed into the "P" (PC counter) storage area. This would then be used as the execution address of the object module, evoked with a simple "G<cr>".

F - Fill memory

This is used to initialize blocks of memory with a constant. Especially useful for clearing all of memory to zero after power-up.

G - Go

This command allows transferring of control to any location in memory. Also, two additional addresses may be specified as "breakpoints". If during the "fetch" cycle of operation an address that was specified as a breakpoint is encountered, system control will be transferred back to the monitor system. The contents of all processor registers may then be examined, modified, cleared, etc. Program execution may then be continued either with or without additional breakpoints. Again, the "P" counter will contain the address where the execution stopped, and a simple "G<cr>" would continue program execution.

H - Hex math

This is useful for calculating relative jump offsets, or index register pointers, etc. Overflow is ignored.

I - unused**J - Justify memory**

This command is a non-destructive memory test. It is useful to be sure that a block of memory is where you thought it was. It would also spot accidentally protected memory. Because of its quick and non destructive nature, it may not always spot "intermittent" or "nervous" memory, but any hard failures will always be detected.

K - unused

L - Load binary file

In many applications, a straight binary dump & load of memory are useful. This provides that ability, and yet does retain some degree of control. The start address is specified in the command, and the end will be determined by the file itself. This is then printed on the console. It uses the unlikely occurrence of the "OFFH" character (all bits one) appearing eight times in a row within a file. The start of file is 8 OFFH's, as is the end of file. When loading a file, when the 8 starting "OFFH"'s are found, the console bell will ring and loading begins. Loading stops when the next 8 OFFH's are read.

M - Move memory

This command will allow moving mass amounts of memory from one area to another. Care should be used so as not to crash the data during a move. When moving up, say from 100H to 800H, the amount of the move (the second address in the command) must be below 800H. If that is not the case, the block should be moved well beyond it's intended place, and then moved downwards.

N - Nulls to punch

This command is most useful when using a paper tape punch for data storage. It will send 60 blanks to the punch for use as leader/trailer. It is also useful with a cassette tape system to preface any write operations. This allows the cassette to 'synch up' quickly during playback.

O - unused

P - Place text to memory

This command allows typing from the keyboard ascii text directly into memory. Useful for modifying text in memory, etc.

Q - Query I/O

This allows direct inputting or outputting to the 256 I/O ports in the system.

R - Read a hex file.

This will read into memory an INTEL formatted hex file. A bias may be added, which would cause the program to be loaded into memory at an address other than specified in it's loading data. This monitor also has the ability to load TDL formatted relocatable files which were generated on the TDL macro assembler. In normal usage, it is mainly meant to read in files that were generated by the "W" command.

S - Substitute memory

This allows a byte by byte examination of memory with the option of altering the data there. It will print the address on the left every 8 bytes in order to keep track of the current memory location being examined. An underline (or left arrow) will back the location to the previous byte. The command is exited by a carriage return. A space bar steps to the next location.

T - Type out memory

It is sometimes useful to examine memory in an ascii format. This command provides that ability. Any non-printing characters will be converted to periods prior to printing. A third parameter is allowed in this command, which defines the maximum characters per line. The default is 64.

U - Unload memory

This routine will dump a continuous block of memory. It is a full 8-bit binary dump, and is formatted with a blank leader, followed by 8 OFFH characters, followed by the first byte of the memory location being dumped. It continues until the range requested has been satisfied, and then dumps 8 more OFFH's, followed by some blank trailer. Files generated by this command are meant to be read by the "L" command. The formatting scheme used here relies on the fact that a OFFH is not normally found in a file, at least not 8 in a row. In order that this scheme perform correctly, it is advisable to initialize memory to zero, or some other such character. This eliminates the chance of accidentally dumping 8 or more OFFH's, which would cause an early termination during read-in with the "L" command.

V - Verify memory

This is a block to block comparison of memory. Useful to see if a program is still as it was when first loaded. You would make a 'copy' first, using the "M" command, at some safe location in memory. Then, if during running of the program you wanted to see if it had altered itself, or if the memory had dropped bits, etc., you would verify the two blocks against each other. Any changes will be printed with the address on the console.

W - Write a hex file

This will do a dump of a specific memory block, similar to the "U" command. However, this is formatted with checksums, and is in 7 bit ascii, which allows transmission over modems, or use with 7 bit storage devices, etc. A third parameter is allowed here, which defines the maximum number of bytes per record. A record defaults to 24 bytes per record, but may be optionally set to a maximum of 255 bytes. Files generated by this command may not be read by the "U" command, but must be read by the "R" command. Also, after all sections of memory have been written out, an "EOF" record must be generated, using the "E" command. This terminates the "R" command.

X - eXamine registers.

The "X" command allows quick examination and modification of all 8080 8 bit temporary registers, and the 16 bit stack pointer and pc counter. The values in these registers are only valid while executing a user program via the "G" command, and are initialized to zero on powerup. If the monitor is entered by either a "BREAKPOINT" or by a "CALL TRAP", all registers will be saved, and are displayable by the "X" command. Upon continuing execution ("G<cr>"), the values are restored to the appropriate registers, and execution resumes from whence it came.

Y - Y'is there.

This is a memory search routine, used to find collections of hex bytes in memory. All of memory is searched, with every occurrence of the string printed on the console. It is desirable to look for at least 2 bytes at a time, with 3-4 the usual case. There is no limit to the length of search string, but more than 12 would be unusual.

Z - Zend

This routine looks for the "TOP" of the first continuous block of memory in the system, and prints the value on the console. This represents the last R/W location in memory.

Note that the monitor places a value in the "S" register (stack pointer). This value is to be used as the highest location a user should place his stack pointer to avoid any conflict with the monitor's stack. It is initiated on powerup, and therefore a user does not need to set the stack pointer unless he desires to do so.

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>
 COPYRIGHT 1979 BY APPLEZAP CORP.

```

; "APPLE MONITOR" COPYRIGHT 1975,1976,1977
; BY ROGER AMIDON
;
; .I8080 ;THIS MONITOR IS 8080 CODE ONLY
; .PROGID APPLE,1,0
; .IDENT APPLE
;
F000 BASE =\ "Rom starting address?"
; .PABS ;THIS MONITOR IN ABSOLUTE FORMAT
; .XLINK ;NO LINKING IN THIS PROGRAM
; .PHEX
;
;
; THIS VERSION WRITTEN FOR ELECTRONIC CONTROL TECHNOLOGY
;
; ALL RIGHTS RESERVED
;
F800 USER = BASE+800H
;
0000 IO =\ "I/O Port base?"
;
0000 CONFIG = 0 ;INITIAL CONFIGURATION
;
0038 RST7 = 38H ;RST 7 (LOCATION FOR TRAP)
;
; <I/O DEVICES>
;
; -C.R.T. SYSTEM
;
0001 CRTI = IO+1H ;DATA PORT (IN)
0000 CRTS = IO+0H ;STATUS PORT (IN)
0001 CRTD = IO+1H ;DATA PORT (OUT)
0001 CRTDA = 1 ;DATA AVAILABLE MASK
0080 CRTBE = 80H ;XMTR BUFFER EMPTY MASK
;
; -PRINTER
;
0003 TTI = IO+3H ;DATA IN PORT
0003 TTO = IO+3H ;DATA OUT PORT
0002 TTS = IO+2H ;STATUS PORT (IN)
0001 TTYDA = 1 ;DATA AVAILABLE MASK BIT
0080 TTYBE = 80H ;XMTR BUFFER EMPTY MASK
;
; -DATA TRANSFER SYSTEM
;
0005 RCSD = IO+5H ;DATA IN PORT
0004 RCSS = IO+4H ;STATUS PORT (IN)
0001 RCSDA = 1 ;DATA AVAILABLE MASK
0005 PCASD = IO+5H ;DATA PORT (OUT)
;
0080 PCSBE = 80H ;XMTR BUFFER EMPTY MASK
;
;
; PARALLEL PORT

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

;
0007 Ppdata = IO+7 ;PARALLEL DATA PORT
0006 Ppstat = IO+6 ;PARALLEL STATUS PORT
0001 Ppda = 1 ;DATA AVAILABLE
0080 Ppbe = 80H ;CLEAR TO TRANSMIT DATA
;
; <CONSTANTS>
;
0000 FALSE = 0 ;ISN'T SO
FFFF TRUE = # FALSE ;IT IS SO
000D CR = 0DH ;ASCII CARRIAGE RETURN
000A LF = 0AH ;ASCII LINE FEED
0007 BELL = 7 ;DING
00FF RUB = OFFH ;RUB OUT
0000 FIL = 00 ;FILL CHARACTERS AFTER CRLF
0007 MAX = 7 ;NUMBER OF QUES IN EOF
;
; <I/O CONFIGURATION MASKS>
;
00FC Cmsk = 11111100B ;CONSOLE DEVICE
00F3 Rmsk = 11110011B ;STORAGE DEVICE (IN)
00CF Pmsk = 11001111B ;STORAGE DEVICE (OUT)
003F Lmsk = 00111111B ;LIST DEVICE
;
;--CONSOLE CONFIGURATION
0000 CCRT = 0 ;C.R.T.
0001 CTTY = 1 ;PRINTER
0002 BATCH = 2 ;READER FOR INPUT, LIST FOR OUTPUT
0003 CUSE = 3 ;USER DEFINED
;
;--STORAGE INPUT CONFIGURATION
0000 RPTR = 0 ;DATA TRANSFER DEVICE
0004 RTTY = 4 ;PRINTER DEVICE
0008 RCAS = 8 ;PARALLEL PORT
000C RUSER = 0CH ;USER DEFINED
;
;--STORAGE OUTPUT CONFIGURATION
0000 PPTP = 0 ;DATA TRANSFER DEVICE
0010 PTTY = 10H ;PRINTER PUNCH
0020 PCAS = 20H ;PARALLEL PORT
0030 PUSER = 30H ;USER DEFINED
;
;--LIST DEVICE CONFIGURATION
0000 LTTY = 0 ;CONSOLE DEVICE
0040 LCRT = 40H ;PRINTER
0080 LINE = 80H ;DATA TRANSFER DEVICE
00C0 LUSER = 0C0H ;USER DEFINED
;
;
; VECTORS FOR USER DEFINED ROUTINES
;
F800 .LOC USER
F800 CILOC: .BLKB 3 ;CONSOLE INPUT

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F803          COLOC:  .BLKB 3 ;CONSOLE OUTPUT
F806          CSLOC:  .BLKB 3 ;CONSOLE INPUT STATUS ROUTINE
F809          RULOC:  .BLKB 3 ;USER DEFINED STORAGE (INPUT)
F80C          PULOC:  .BLKB 3 ;USER DEFINED STORAGE (OUTPUT)
F80F          LULOC:  .BLKB 3 ;USER DEFINED PRINTER (LIST)
F812          J =.
;
;
F080          UTAB    = BASE+80H
;
;          PROGRAM CODE BEGINS HERE
;
F000          .LOC    BASE
;
F000          C3 F0D8 APPLE:  JMP      BEGIN    ;GO AROUND VECTORS
;
;          <VECTORS FOR CALLING PROGRAMS>
;
;          THESE VECTORS MAY BE USED BY USER WRITTEN
;          PROGRAMS TO SIMPLIFY THE HANDLING OF I/O
;          FROM SYSTEM TO SYSTEM.  WHATEVER THE CURRENT
;          ASSIGNED DEVICE, THESE VECTORS WILL PERFORM
;          THE REQUIRED I/O OPERATION, AND RETURN TO
;          THE CALLING PROGRAM. (RET)
;
;          THE REGISTER CONVENTION USED FOLLOWS-
;
;          ANY INPUT OR OUTPUT DEVICE-
;          CHARACTER TO BE OUTPUT IN 'C' REGISTER.
;          CHARACTER WILL BE IN 'A' REGISTER UPON
;          RETURNING FROM AN INPUT OR OUTPUT.
;
;          'CSTS'-
;          RETURNS TRUE (OFFH IN 'A' REG.) IF THERE IS
;          SOMETHING WAITING, AND ZERO (00) IF NOT.
;
;          'IOCHK'-
;          RETURNS WITH THE CURRENT I/O CONFIGURATION
;          BYTE IN 'A' REGISTER.
;
;          'IOSET'-
;          ALLOWS A PROGRAM TO DYNAMICALLY ALTER THE
;          CURRENT I/O CONFIGURATION, AND REQUIRES
;          THE NEW BYTE IN 'C' REGISTER.
;
;          'MEMCK'-
;          RETURNS WITH THE HIGHEST ALLOWED USER
;          MEMORY LOCATION. 'B'=HIGH BYTE, 'A'=LOW.
;
;          'TRAP'-
;          THIS IS THE 'BREAKPOINT' ENTRY POINT,
;          BUT MAY BE 'CALLED'.  IT WILL SAVE
;          THE MACHINE STATE.  RETURN CAN BE MADE WITH
;          A SIMPLE 'GICRJ' ON THE CONSOLE.
;
;
F003          C3 F70B          JMP      CI      ;CONSOLE INPUT
F006          C3 F72F          JMP      RI      ;READER INPUT
F009          C3 F56A          JMP      CO      ;CONSOLE OUTPUT
F00C          C3 F6CC          JMP      PO      ;PUNCH OUTPUT
F00F          C3 F590          JMP      LO      ;LIST OUTPUT

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F012  C3 F5D6      JMP      CSTS      ;CONSOLE STATUS
F015  C3 F1A2      JMP      IDCHK     ;I/O ASSIGNMENT CHECK
F018  C3 F19D      JMP      IOSET     ;I/O SET
F01B  C3 F09A      JMP      MEMCK     ;MEMORY LIMIT CHECK

;
F01E  E5           TRAP:  PUSH     H          ;ASSUME A VALID STACK
F01F  D5           PUSH     D
F020  C5           PUSH     B
F021  F5           PUSH     PSW      ;SAVE MACHINE STATE
F022  11 FFEA      LXI     D,65535-(ENDX-EXIT)
F025  21 000A     ..R0:  LXI     H,10    ;GO UP 10 BYTES IN STACK
F028  39           DAD     SP
F029  0604        MVI     B,4
F02B  EB           XCHG
F02C  2B           ..R1:  DCX     H
F02D  72           MOV     M,D
F02E  2B           DCX     H
F02F  73           MOV     M,E
F030  D1           POP     D
F031  05           DCR     B
F032  C2 F02C     JNZ     ..R1
F035  C1           POP     B          ;OLD PC
F036  0B           DCX     B          ;-1
F037  F9           SPHL           ;SET MONITOR'S STACK
F038  21 0014     LXI     H,TLOC
F03B  39           DAD     SP
F03C  CD F07A     CALL    ..R5      ;TEST IF A TRAP SET
F03F  23           INX     H
F040  23           INX     H
F041  C4 F07A     CNZ     ..R5      ;TEST FOR 2nd TRAP
F044  CA F048     JZ      ..R2      ; NO
F047  03           INX     B
F048  21 000F     ..R2:  LXI     H,LLOC
F04B  39           DAD     SP
F04C  73           MOV     M,E
F04D  23           INX     H
F04E  72           MOV     M,D
F04F  23           INX     H
F050  23           INX     H
F051  71           MOV     M,C
F052  23           INX     H
F053  70           MOV     M,B
F054  C5           PUSH     B
F055  0E40        MVI     C,'@'
F057  CD F56A     CALL    CO
F05A  E1           POP     H
F05B  CD F665     CALL    LADR
F05E  21 0014     LXI     H,TLOC
F061  39           DAD     SP
F062  11 0002     ..R3:  LXI     D,2
F065  4E           MOV     C,M
F066  72           MOV     M,D
F067  23           INX     H
F068  46           MOV     B,M

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F069 72 MOV M,D
F06A 23 INX H
F06B 79 MOV A,C
F06C B0 ORA B
F06D CA F072 JZ ..R4
F070 7E MOV A,M
F071 02 STAX B
F072 23 ..R4: INX H
F073 1D DCR E
F074 C2 F065 JNZ ..R3
F077 C3 F0FC JMP START

;
;..R5:
F07A 7E ..R5: MOV A,M
F07B 91 SUB C
F07C 23 INX H
F07D C0 RNZ
F07E 7E MOV A,M
F07F 90 SUB B
F080 C9 RET

;
MEMSIZ:
F081 21 FFFF MEMSIZ: LXI H,-1 ;START AT THE BOTTOM
F084 24 ..M0: INR H ;FIRST FIND R/W MEMORY
F085 7E MOV A,M
F086 2F CMA
F087 77 MOV M,A
F088 BE CMP M
F089 2F CMA
F08A 77 MOV M,A
F08B C2 F084 JNZ ..M0
F08E 24 ..M1: INR H ;NOW FIND NON-R/W
F08F 7E MOV A,M
F090 2F CMA
F091 77 MOV M,A
F092 BE CMP M
F093 2F CMA
F094 77 MOV M,A
F095 CA F08E JZ ..M1
F098 25 DCR H
F099 C9 RET

;
MEMCK:
F09A E5 MEMCK: PUSH H
F09B CD F081 CALL MEMSIZ
F09E 44 MOV B,H ;USER'S HIGH BYTE
F09F E1 POP H
FOA0 3E C0 MVI A,0C0H ;USER'S LOW BYTE
FOA2 C9 RET

;
TOM:
FOA3 21 F0C5 TOM: LXI H,MSG
FOA6 4E TOM1: MOV C,M
FOA7 23 INX H
FOA8 CD F56A CALL CD
FOAB 05 DCR B
FOAC C2 FOA6 JNZ TOM1
FOAF CD F5D6 CALL CSTS

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

FOB2   B7           ORA   A
FOB3   C8           RZ

;
FOB4   CD F78D     CCHK:  CALL   KI
FOB7   FE03       CPI    3
FOB9   C0         RNZ

;
FOBA   31 FFE2     ERROR: LXI    SP,65535-((ENDX-EXIT)+8)
FOBD   0E2A       MVI    C,'*'
FOBF   CD F56A     CALL   CD
FOC2   C3 F0FC     JMP    START

;
;
;
;
;
ANNOUNCEMENT OF MONITOR NAME & VERSION
;
FOC5   0D0A000000 MSG:   .BYTE  CR,LF,FIL,FIL,FIL
FOCA   4170706C6320 .ASCII 'Apple V'
FOD1   312E30204543 .ASCII '1.0 ECT'
0013   MSGL        =  .-MSG

;
;
LET US BEGIN
;
;
FOD8   21 FFEA     BEGIN: LXI    H,65535-(ENDX-EXIT)
FODB   F9         SPHL   ;SET UP A STACK
FODC   0615       MVI    B,ENDX-EXIT
FODE   11 F7C1     LXI    D,EXIT
FOE1   1A         ..BG1: LDAX   D
FOE2   77         MOV    M,A
FOE3   23         INX    H
FOE4   13         INX    D
FOE5   05         DCR    B
FOE6   C2 F0E1     JNZ    ..BG1
FOE9   CD F081     CALL   MEMSIZ ;GET USER'S STACK
FOEC   E5         PUSH   H
FOED   60         MOV    H,B ;Zero out HL
FOEE   68         MOV    L,B
FOEF   E5         PUSH   H
FOF0   E5         PUSH   H
FOF1   E5         PUSH   H
FOF2   3E00       MVI    A,CONFIG
FOF4   32 FFFF     STA    -1
FOF7   0613       MVI    B,MSGL
FOF9   CD F0A3     CALL   TOM ;PRINT SIGN-ON
FOFC   11 F0FC     START: LXI    D,START
FOFF   D5         PUSH   D
F100   CD F5CC     CALL   CRLF
F103   0E3E       MVI    C,'>'
F105   CD F56A     CALL   CD
F108   21 F129     LXI    H,TBL
F10B   CD F793     STARO: CALL   TI
F10E   CA F10B     JZ    STARO
F111   FE20       CPI    ' ' ;CONTROL?

```

ELECTRONIC CONTROL TECHNOLOGY

PAGE 7

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

F113	DA F10B	JC	STARO	; IGNORE
F116	D641	SUI	'A'	
F118	D8	RC		; <A
F119	FE1A	CPI	'Z'-'A'+1	
F11B	D0	RNC		; >Z
F11C	87	ADD	A	; A*2
F11D	85	ADD	L	; +TBL
F11E	6F	MOV	L,A	
F11F	7E	MOV	A,M	
F120	23	INX	H	
F121	66	MOV	H,M	
F122	6F	MOV	L,A	
F123	A4	ANA	H	
F124	3C	INR	A	
F125	CA FOBA	JZ	ERROR	; DON'T GO TO OFFFFH
F128	E9	PCHL		

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

;
F129          TBL:
F129          F15D      .WORD  ASSIGN      ;A - ASSIGN I/O
F12B          F1A6      .WORD  BRANCH      ;B - BRANCH TO USER ROUTINE A-Z
F12D          FFFF      .WORD  OFF          ;C UNDEFINED
F12F          F1B4      .WORD  DISP        ;D - DISPLAY MEMORY ON CONSOLE IN HEX
F131          F1D4      .WORD  EOF          ;E - END OF FILE TAG FOR HEX DUMPS
F133          F1F0      .WORD  FILL        ;F - FILL MEMORY WITH CONSTANT
F135          F1FE      .WORD  GOTO        ;G - GOTO <ADDRESS>, W/BKPTS (2)
F137          F656      .WORD  HEXN        ;H - HEX MATH <SUM> <DIFFERENCE>
F139          F812      .WORD  J           ;I *** USER DEFINED
F13B          F24C      .WORD  TEST        ;J - NON-DESTRUCTIVE MEMORY TEST
F13D          F815      .WORD  J+3        ;K *** USER DEFINED
F13F          F267      .WORD  LOAD        ;L - LOAD A BINARY FORMAT FILE .
F141          F2AD      .WORD  MOVE        ;M - MOVE MASS MEMORY
F143          F702      .WORD  NULL        ;N - PUNCH LEADER/TRAILER
F145          F818      .WORD  J+6        ;O *** USER DEFINED
F147          F2B9      .WORD  PUTA        ;P - 'PUT' ASCII INTO MEMORY.
F149          F535      .WORD  QUERY       ;Q - QI(N)=READ I/O; QO(N,V)=SEND I/O
F14B          F31F      .WORD  READ        ;R - READ A HEX FILE (W/CHECKSUM)
F14D          F3F5      .WORD  SUBS        ;S - EXAMINE/SUBSTITUTE MEMORY
F14F          F420      .WORD  TYPE        ;T - DISPLAY MEMORY IN ASCII
F151          F6E9      .WORD  UNLD        ;U - DUMP MEMORY IN BINARY FORMAT
F153          F44C      .WORD  VERIFY       ;V - COMPARE MEMORY TO MEMORY
F155          F472      .WORD  WRITE       ;W - DUMP MEMORY IN HEX FILE FORMAT
F157          F4BA      .WORD  XAM         ;X - EXAMINE/MODIFY CPU REGISTERS
F159          F2DB      .WORD  WHERE       ;Y - FIND 'N' BYTES IN MEMORY
F15B          F55F      .WORD  SIZE        ;Z - ADDR OF LAST R/W MEMORY LOCATION

;
FFFF          OFF      = -1
;
F880          UTAB     = USER+80H
;

```


01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

;
;
F15D   CD F793   ;ASSIGN: CALL   TI       ;GET A DEVICE
F160   21 F7AC   LXI     H,LTBL-1 ;POINT TO TABLE
F163   01 0004   LXI     B,4       ;TO SKIP THRU TABLE
F166   CD F186   CALL    ..A3      ;GET DEVICE COUNT
F169   D5        PUSH    D         ;SAVE IN STACK
F16A   CD F793   ..A1:  CALL    TI
F16D   D63D      SUI     '='
F16F   C2 F16A   JNZ     ..A1
F172   4F        MOV     C,A       ;C=0
F173   CD F793   CALL    TI       ;GET ASSIGNMENT
F176   CD F186   CALL    ..A3
F179   F1        POP     PSW      ;A=BEVICE
F17A   6A        MOV     L,D       ;L=ASSIGNMENT
F17B   2603      MVI     H,3       ;SETUP A MASK
F17D   3D        ..A2:  DCR     A         ;ZERO=DONE
F17E   FA F195   JM      ..A5
F181   29        DAD     H
F182   29        DAD     H         ;DOUBLE SHIFT LEFT
F183   C3 F17D   JMP     ..A2
F186   11 0004   ..A3:  LXI     D,4       ;GO THRU THIS 4 TIMES
F189   23        ..A4:  INX     H         ;BUMP POINTER 1
F18A   BE        CMP     M         ;MATCH?
F18B   C8        RZ
F18C   09        DAD     B         ;BUMP HL
F18D   14        INR     D
F18E   1D        DCR     E         ;COUNT DOWN
F18F   C2 F189   JNZ     ..A4
F192   C3 FOBA   JMP     ERROR    ;CAN'T FIND IT

;
;..A5:
F195   AC        ..A5:  XRA     H         ;COMPLIMENT H
F196   67        MOV     H,A
F197   CD F1A2   CALL    IOCHK    ;GET CURRENT CONFIGURATION
F19A   A4        ANA     H         ;KILL ASSIGNMENT BITS
F19B   B5        ORA     L         ;MODIFY TO NEW DEVICE
F19C   4F        MOV     C,A       ;PUT NEW IOBYT IN C

;
;ZA=.-ASSIGN
0040
;
F19D   79        IOSET:  MOV     A,C
F19E   32 FFFF   STA     -1
F1A1   C9        RET

;
;IOCHK:
F1A2   3A FFFF   IOCHK:  LDA     -1
F1A5   C9        RET

;
;BRANCH:
F1A6   CD F793   BRANCH: CALL    TI       ;GET A '.'
F1A9   FE2E      CPI     '.'
F1AB   C2 FOBA   JNZ     ERROR
F1AE   21 F880   LXI     H,UTAB  ;POINT TO USER'S TBL
F1B1   C3 F10B   JMP     STARO   ;GOOD LUCK

;
;ZB=.-BRANCH
000E

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

0000          ;
          ZC=.-.
          ;
F1B4      0E10      DISP:      MVI      C,16      ;SET A DEFAULT
F1B6      CD F5F7          CALL      EXPC
F1B9      F5          PUSH      PSW
F1BA      CD F562      ..D0:    CALL      LFADR
F1BD      F1          POP       PSW
F1BE      F5          PUSH      PSW      ;GET SIZE
F1BF      47          MOV       B,A      ;IN B
F1C0      CD F568      ..D1:    CALL      BLK
F1C3      7E          MOV       A,M
F1C4      CD F66A          CALL      LBYTE
F1C7      CD F64C          .CALL     HILO
F1CA      DA F64A          JC       PRET
F1CD      05          DCR       B
F1CE      C2 F1C0          JNZ      ..D1
F1D1      C3 F1BA          JMP      ..D0

0020          ;
          ZD=.-DISP
          ;
F1D4      CD F623      EOF:      CALL      EXPR
F1D7      CD F6C5          CALL      PEOL
F1DA      0E3A          MVI      C,';'
F1DC      CD F6CC          CALL      PO
F1DF      AF          XRA      A
F1E0      CD F6A2          CALL      PBYTE
F1E3      E1          POP      H
F1E4      CD F69D          CALL      PADR
F1E7      21 0000          LXI      H,0
F1EA      CD F69D          CALL      PADR
F1ED      C3 F702          JMP      NULL

001C          ;
          ZE=.-EOF
          ;
F1F0      CD F5F7      FILL:    CALL      EXPC
F1F3      71          ..F1:    MOV      M,C
F1F4      CD F64C          CALL      HILO
F1F7      D2 F1F3          JNC      ..F1
F1FA      D1          POP      D
F1FB      C3 F0FC          JMP      START

000E          ;
          ZF=.-FILL
          ;
F1FE      CD F6B6      GOTO:    CALL      PCHK
F201      CA F20F          JZ       ..GO      ;DELIMITER ENTERED
F204      CD F626          CALL      EXF      ; ELSE GET A 'GO' ADDR
F207      D1          POP      D
F208      21 0015          LXI      H,PLOC
F20B      39          DAD      SP
F20C      72          MOV      M,D      ;PLACE IN EXIT TEMPLATE
F20D      2B          DCX      H
F20E      73          MOV      M,E
F20F      FE0D          ..GO:    CPI      CR      ;TEST DELIMITER

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F211    CA F243          JZ      ..G4      ;NO BREAKPOINTS, JUST GO
F214    1602           MVI      D,2        ;2 POSSIBLE BREAKPOINTS
F216    21 0016       LXI      H,TLOCX
F219    39            DAD      SP
F21A    E5            ..G1:  PUSH   H
F21B    CD F623       CALL   EXPR      ;GET AN ADDRESS
F21E    C1            POP    B          ;IN BC
F21F    E1            POP    H
F220    F5            PUSH  PSW       ;SAVE DELIMITER
F221    78            MOV   A,B       ;CAN'T ALLOW ANY
F222    B1            ORA   C          ; BREAKPOINTS AT ZERO
F223    CA F230       JZ     ..G2      ;DO NOTHING
F226    71            MOV   M,C
F227    23            INX   H
F228    70            MOV   M,B       ;ELSE SAVE BKPT ADDRESS
F229    23            INX   H
F22A    0A           LDAX  B          ;AND OPCODE THERE
F22B    77            MOV   M,A
F22C    23            INX   H
F22D    3EFF         MVI   A,OFFH   ;RST 7
F22F    02            STAX  B          ;REPLACE OPCODE
F230    F1            ..G2:  POP    PSW  ;LOOK AT DELIMITER
F231    DA F238       JC     ..G3
F234    15            DCR   D
F235    C2 F21A       JNZ   ..G1
F238    3EC3         ..G3:  MVI   A,JMP   ;SET A 'JMP' AT RST7
F23A    32 0038       STA  RST7
F23D    21 F01E       LXI  H,TRAP
F240    22 0039       SHLD RST7+1
F243    CD F5CC       ..G4:  CALL  CRLF
F246    D1            POP   D          ;THROW AWAY RETURN
F247    21 0008       LXI  H,8
F24A    39            DAD  SP
F24B    E9            PCHL

;
;ZG=-GOTO
;
F24C    CD F5F7       TEST:  CALL   EXPC
F24F    7E            ..T1:  MOV   A,M
F250    47            MOV   B,A       ;SAVE CHAR IN 'B'
F251    2F            CMA
F252    77            MOV   M,A
F253    AE            XRA   M
F254    70            MOV   M,B       ;REPLACE BYTE
F255    CA F261       JZ     ..T2
F258    D5            PUSH  D          ;SAVE END POINTER
F259    5F            MOV   E,A       ;SAVE ERROR MASK
F25A    CD F565       CALL  HLSP      ;DISPLAY BAD ADDRESS
F25D    CD F5B8       CALL  BITS+1   ;DISPLAY BAD BIT(S)
F260    D1            POP   D          ;RESTORE DE
F261    CD F646       ..T2:  CALL  HILOX
F264    C3 F24F       JMP   ..T1

;
;ZJ=-TEST
001B

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

;
F267    CD F623    LOAD:    CALL    EXPR
F26A    CD F5CC    CALL    CRLF
F26D    E1        POP    H
F26E    16FF      MVI    D,RUB
F270    01 0407   ..L0:   LXT    B,407H ;B=4 MATCHES, C=BELL
F273    CD F785   ..L1:   CALL    RIFF
F276    C2 F270   JNZ    ..L0
F279    05        DCR    B
F27A    C2 F273   JNZ    ..L1
F27D    CD F785   ..L2:   CALL    RIFF
F280    CA F27D   JZ     ..L2
F283    77        MOV    M,A
F284    CD F56A   ..L3:   CALL    CO ;TELL CONSOLE
F287    23        INX    H
F288    CD F785   CALL    RIFF
F28B    CA F292   JZ     ..L5
F28E    77        ..L4:   MOV    M,A
F28F    C3 F287   JMP    ..L3
F292    1E01      ..L5:   MVI    E,1 ;INITIALIZE
F294    CD F785   ..L6:   CALL    RIFF
F297    C2 F2A4   JNZ    ..L7
F29A    1C        INR    E
F29B    3E07      MVI    A,MAX
F29D    BB        CMP    E
F29E    C2 F294   JNZ    ..L6
F2A1    C3 F562   JMP    LFADR
F2A4    72        ..L7:   MOV    M,D
F2A5    23        INX    H
F2A6    1D        DCR    E
F2A7    C2 F2A4   JNZ    ..L7
F2AA    C3 F28E   JMP    ..L4

;
0046    ;ZL=.-LOAD
;
F2AD    CD F5F7   MOVE:   CALL    EXPC
F2B0    7E        ..M:   MOV    A,M
F2B1    02        STAX   B
F2B2    03        INX    B
F2B3    CD F646   CALL    HILOX
F2B6    C3 F2B0   JMP    ..M

;
000C    ;ZM=.-MOVE
;
F2B9    CD F623   PUTA:   CALL    EXPR
F2BC    CD F5CC   CALL    CRLF
F2BF    E1        POP    H
F2C0    CD F78D   ..P0:   CALL    KI
F2C3    FE04      CPI    4 ;EOT?
F2C5    CA F562   JZ     LFADR ;PRINT ADDRESS & QUIT
F2C8    FE7F      CPI    7FH ;RUB-OUT?
F2CA    CA F2D6   JZ     ..P2 ; YES
F2CD    77        MOV    M,A ;PUT CHARACTER INTO MEMORY
F2CE    4F        MOV    C,A ;

```

01/07/79 22:40:00

APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F2CF      23          INX      H
F2D0      CD F56A    ..P1:    CALL    CO      ;ECHO CHARACTER
F2D3      C3 F2C0    JMP      ..P0    ;& CONTINUE
F2D6      2B          ..P2:    DCX      H      ;BACK-UP POINTER
F2D7      4E          MOV      C,M     ;ECHO CANCELED CHARACTER
F2D8      C3 F2D0    JMP      ..P1

;
0022      ;ZF=-PUTA
;

F2DB      21 0000    WHERE:  LXI      H,0      ;GET STRING POINTER (SP)
F2DE      4D          MOV      C,L      ;ZERO C REG
F2DF      39          DAD      SP
F2E0      2B          DCX      H      ;SP-1
F2E1      EB          XCHG
F2E2      CD F623    ..Y1:    CALL    EXPR
F2E5      E1          POP      H      ;CONSERVE STACK USAGE
F2E6      65          MOV      H,L      ;L=SEARCH BYTE
F2E7      E5          PUSH     H      ;H=L
F2E8      33          INX      SP     ;ADJUST STACK
F2E9      0C          INR      C      ;COUNT SEARCH BYTES
F2EA      D2 F2E2    JNC      ..Y1
F2ED      EB          XCHG
F2EE      51          MOV      D,C
F2EF      E5          PUSH     H      ;HL=SEARCH STRING POINTER
F2F0      01 0000    LXI      B,0
F2F3      C5          PUSH     B      ;BC=START SEARCH (0)
F2F4      CD F5CC    ..Y2:    CALL    CRLF
F2F7      C1          ..Y3:    POP      B
F2F8      E1          POP      H
F2F9      5A          MOV      E,D
F2FA      78          MOV      A,B
F2FB      A1          ANA      C
F2FC      3C          INR      A
F2FD      C2 F303    JNZ      ..Y5
F300      23          ..Y4:    INX      H
F301      F9          SPHL
F302      C9          RET
F303      0A          ..Y5:    LDAX     B
F304      03          INX      B
F305      BE          CMP      M
F306      E5          PUSH     H
F307      C5          PUSH     B
F308      C2 F2F7    ..Y6:    JNZ      ..Y3
F30B      1D          DCR      E
F30C      CA F316    JZ       ..Y7
F30F      0A          LDAX     B
F310      03          INX      B
F311      2B          DCX      H
F312      BE          CMP      M
F313      C3 F308    JMP      ..Y6
F316      E1          ..Y7:    POP      H
F317      E5          PUSH     H
F318      2B          DCX      H
F319      CD F665    CALL    LADR

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F31C      C3 F2F4          JMP      ..Y2
0044      ;
          ZY=.-WHERE
          ;
F31F      CD F623      READ:   CALL    EXPR    ;GET 16 BIT VALUE
F322      D1          POP      D      ;DE=BIAS
F323      21 0000     LXI     H,0    ;SET-UP DEFAULT BASEC13
F326      E5          PUSH    H      ;AND DEFAULT BASEC23
F327      DA F337     JC      ..R0   ;CR
F32A      CD F623     CALL   EXPR    ;GET ACTUAL BASEC13
F32D      E1          POP      H      ;HL=BASEC13
F32E      DA F337     JC      ..R0   ;CR
F331      E3          XTHL           ;GET DEFAULT BASEC23
F332      CD F623     CALL   EXPR    ;GET ACTUAL BASEC23
F335      E1          POP      H
F336      E3          XTHL           ;(SP)=BASEC23
F337      E5          ..R0:  PUSH    H      ;HL=BASEC13
F338      D5          PUSH    D      ;DE=BIAS
F339      CD F5CC     CALL   CRLF    ;BEGIN READING FILE
F33C      CD F77F     ..R1:  CALL   RIX     ;GET READER CHARACTER
F33F      D63A       SUI     ';;'   ;GET FILE TYPE CUE
F341      47          MOV     B,A     ;SAVE CUE CLUE
F342      E6FE       ANI     OFEH    ;KILL BIT 0
F344      C2 F33C     JNZ    ..R1   ;NOT ';;' OR ';'
F347      57          MOV     D,A     ;ZERO CHECKSUM STORAGE
F348      CD F3D4     CALL   ..BYTE  ;GET FILE LENGTH
F34B      5F          MOV     E,A     ;SAVE IN E
F34C      CD F3D4     CALL   ..BYTE  ;GET LOAD MSB
F34F      F5          PUSH   PSW     ;SAVE IN STACK
F350      CD F3D4     CALL   ..BYTE  ;GET LOAD LSB
F353      E1          POP      H      ;H=MSB
F354      6F          MOV     L,A     ;HL=LOAD ADDR
F355      CD F3D4     CALL   ..BYTE  ;GET FILE TYPE
F358      B7          ORA     A      ;TEST FILE TYPE
F359      78          MOV     A,B     ;GET CUE
F35A      C1          POP      B      ;BC=BIAS
F35B      CA F365     JZ     ..R2   ;ABSOLUTE LOAD
F35E      EB          XCHG           ;RELOCATE LOAD ADDR.
F35F      E3          XTHL
F360      EB          XCHG
F361      19          DAD     D      ;DO IT
F362      EB          XCHG
F363      E3          XTHL
F364      EB          XCHG           ;HL=LOAD+BASEC13
F365      1C          ..R2:  INR     E      ;TEST LENGTH
F366      1D          DCR     E      ;ZERO?
F367      CA F3E7     JZ     ..DONE
F36A      09          DAD     B      ;ADD BIAS TO LOAD
F36B      C5          PUSH   B      ;SAVE BIAS
F36C      47          MOV     B,A     ;SET-UP B
F36D      3D          DCR     A      ;TEST CUE CLUE
F36E      CA F386     JZ     ..R6   ;Z=REL. FILE, NZ=ABS.
F371      CD F3D4     ..R3:  CALL   ..BYTE  ;GET NEXT DATA BYTE
F374      77          MOV     M,A     ;WRITE TO MEMORY

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F375 23 INX H ;BUMP UP LOAD POINT
F376 1D DCR E ;BUMP DOWN BYTE COUNT
F377 C2 F371 JNZ ..R3 ;CONTINUE
F37A CD F3D4 ..R4: CALL ..BYTE ;TEST CHECKSUM
F37D CA F33C JZ ..R1 ;OK; CONTINUE W/NEXT
F380 CD F665 ..R5: CALL LADR ; ELSE PRINT LOAD ADDR
F383 C3 F0BA JMP ERROR ; & ABORT
F386 CD F3BE ..R6: CALL ..R10 ;GET NEXT DATA BYTE
F389 77 MOV M,A ;STORE IT
F38A D2 F3B6 JNC ..R9 ;NORMAL BYTE
F38D E5 PUSH H ;CARRY=RELOCATE NEXT WORD
F38E 21 0005 LXI H,5 ;POINT TO BASEC1J
F391 39 DAD SP ;IN STACK
F392 CD F3BE ..R7: CALL ..R10 ;GET HIGH BYTE
F395 D2 F3A5 JNC ..R8 ;USE BASECnJ
F398 1D DCR E ;COUNT EXTRA BYTE
F399 E3 XTHL ;GET LOAD ADDR
F39A 35 DCR M ;TEST FOR BASEC1J
F39B 77 MOV M,A ;NEW LOW BYTE
F39C E3 XTHL ;SAVE LOAD AGAIN
F39D CA F392 JZ ..R7 ;BASEC1J
F3A0 23 INX H
F3A1 23 INX H ;POINT TO BASEC2J
F3A2 C3 F392 JMP ..R7 ;AND TRY AGAIN

;
F3A5 86 ..R8: ADD M ;ADD IN MSB
F3A6 E3 XTHL
F3A7 23 INX H ;STICK AT LOAD+1
F3A8 77 MOV M,A
F3A9 2B DCX H ;GET LOAD BYTE
F3AA 7E MOV A,M ;IN A
F3AB E3 XTHL
F3AC 2B DCX H
F3AD 86 ADD M ;RELOCATE LSB
F3AE E1 POP H ;GET LOAD ADDR
F3AF 77 MOV M,A ;STORE IT
F3B0 23 INX H ;GET MSB
F3B1 7E MOV A,M ;IN A
F3B2 CE00 ACI O ;ADJUST FOR CARRY
F3B4 77 MOV M,A ;STORE IT
F3B5 1D DCR E ;COUNT IT
F3B6 23 ..R9: INX H ;BUMP THE COUNT
F3B7 1D DCR E ;MORE?
F3B8 C2 F386 JNZ ..R6 ; & CONTINUE
F3BB C3 F37A JMP ..R4 ;TEST CHECKSUM

;
F3BE 05 ..R10: DCR B ;COUNT BITS/BYTES
F3BF C2 F3C9 JNZ ..R11 ;NEXT IS DATA BYTE
F3C2 CD F3D4 CALL ..BYTE ;GET RELOC. MAP
F3C5 1D DCR E ;BUMP DOWN BYTE COUNT
F3C6 4F MOV C,A ;MAP IN C
F3C7 0608 MVI B,8 ;RESET FOR NEXT 8
F3C9 CD F3D4 ..R11: CALL ..BYTE ;NEXT DATA BYTE
F3CC D5 PUSH D ;SAVE DE

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F3CD 57          MOV     D,A      ;SAVE DATA BYTE
F3CE 79          MOV     A,C      ;TEST FOR RELOC.
F3CF 17          RAL             ;IN CARRY FLAG
F3D0 4F          MOV     C,A      ;UPDATE C
F3D1 7A          MOV     A,D      ;RESTORE DATA BYTE
F3D2 D1          POP     D        ;RESTORE DE
F3D3 C9          RET             ;CONTINUE

;
F3D4 C5          ;..BYTE: PUSH   B        ;SAVE BC
F3D5 CD F68A     CALL   RIBBLE ;GET A CONVERTED CHAR.
F3D8 07          RLC             ;
F3D9 07          RLC             ;
F3DA 07          RLC             ;
F3DB 07          RLC             ;MOVE IT TO HIGH NIBBLE
F3DC 4F          MOV     C,A      ;SAVE IT
F3DD CD F68A     CALL   RIBBLE ;GET OTHER HALF
F3E0 B1          ORA     C        ;MAKE WHOLE
F3E1 4F          MOV     C,A      ;SAVE IN C
F3E2 82          ADD     D        ;UPDATE CHECKSUM
F3E3 57          MOV     D,A      ;NEW CHECKSUM
F3E4 79          MOV     A,C      ;RESTORE DATA BYTE
F3E5 C1          POP     B        ;RESTORE BC
F3E6 C9          RET             ;CONTINUE

;
F3E7 C1          ;..DONE: POP     B        ;BASEC11
F3E8 C1          POP     B        ;BASEC23
F3E9 7C          MOV     A,H      ;TEST EOF
F3EA B5          ORA     L        ;FOR ZERO
F3EB C8          RZ             ;
F3EC EB          XCHG            ;ELSE STORE IT IN 'P'
F3ED 21 0015     LXI     H,PLOC
F3F0 39          DAD     SP
F3F1 72          MOV     M,D      ;IN 'EXIT' TEMPLATE
F3F2 2B          DCX     H
F3F3 73          MOV     M,E
F3F4 C9          RET             ;REALLY DONE.

;
00D6           ;ZR=.-READ
;
F3F5 CD F623     ;SUBS: CALL   EXPR
F3F8 E1          POP     H
F3F9 D8          RC             ;QUIT
F3FA 7E          ;..S0: MOV     A,M
F3FB CD F66A     CALL   LBYTE
F3FE CD F6B1     CALL   COPCK
F401 D8          RC
F402 CA F412     JZ     ..S1
F405 FE5F       CPI     '_' ;BACK-UP?
F407 CA F41C     JZ     ..S3
F40A E5          PUSH   H
F40B CD F626     CALL   EXF
F40E D1          POP     D
F40F E1          POP     H
F410 73          MOV     M,E

```


01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F411    D8          RC
F412    23          ..S1: INX      H
F413    7D          ..S2: MOV      A,L
F414    E607        ANI      7
F416    CC F562     CZ      LFADR
F419    C3 F3FA     JMP      ..S0
F41C    2B          ..S3: DCX      H      ;BACK-UP
F41D    C3 F413     JMP      ..S2

;
002B    %S=.-SUBS
;
F420    0E40        TYPE:  MVI      C,64      ;SET UP A DEFAULT
F422    CD F5F7     CALL     EXPC
F425    F5          PUSH     PSW
F426    CD F562     ..T0:  CALL     LFADR
F429    F1          POP      PSW
F42A    F5          PUSH     PSW
F42B    47          MOV      B,A      ;RESET LENGTH
F42C    7E          ..T1:  MOV      A,M
F42D    E67F        ANI      7FH
F42F    FE20        CPI      ' '      ;TEST LOWER END
F431    D2 F436     JNC      ..T3
F434    3E2E        ..T2:  MVI      A,'.'  ;PRINT PERIODS INSTEAD
F436    FE7D        ..T3:  CPI      7DH      ;TEST UPPER END
F438    D2 F434     JNC      ..T2
F43B    4F          MOV      C,A      ;PUT WHATEVER INTO C
F43C    CD F56A     CALL     CO
F43F    CD F64C     CALL     HILO
F442    DA F64A     JC      FRET
F445    05          DCR      B
F446    C2 F42C     JNZ      ..T1
F449    C3 F426     JMP      ..T0

;
002C    %T=.-TYPE
;
F44C    CD F5F7     VERIFY: CALL     EXPC
F44F    0A          ..V0:  LDAX     B
F450    D5          PUSH     D      ;SAVE END POINTER
F451    5E          MOV      E,M      ;GET MEMORY DATA
F452    BB          CMP      E      ;TEST FOR MATCH
F453    CA F46A     JZ      ..V1      ;MATCHES
F456    C5          PUSH     B
F457    47          MOV      B,A
F458    CD F565     CALL     HLSP
F45B    7B          MOV      A,E      ;GET MISMATCH
F45C    CD F66A     CALL     LBYTE     ;PRINT IT
F45F    CD F568     CALL     BLK      ;SPACE OVER
F462    78          MOV      A,B      ;GET OTHER MISMATCH
F463    CD F66A     CALL     LBYTE     ;PRINT THAT TOO
F466    CD F5CC     CALL     CRLF     ;PREPARE FOR ANOTHER
F469    C1          POP      B
F46A    D1          ..V1:  POP      D      ;RESTORE END POINTER
F46B    03          INX      B
F46C    CD F646     CALL     HILOX

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F46F    C3 F44F                JMP      ..W0
;
0026    ;ZV=-VERIFY
;
F472    CD F5F7                WRITE:   CALL    EXPC
F475    CD F705                CALL    WAIT
F478    CD F6C5                ..W0:    CALL    PEOL
F47B    01 003A                LXI     B,':'
F47E    CD F6CC                CALL    PO
F481    D5                      PUSH   D
F482    E5                      PUSH   H
F483    04                      ..W1:    INR     B
F484    CD F64C                CALL    HILO
F487    DA F498                JC      ..W2
F48A    3E18                    MVI     A,24
F48C    90                      SUB     B
F48D    C2 F483                JNZ    ..W1
F490    E1                      POP     H
F491    CD F49A                CALL    ..W3
F494    D1                      POP     D
F495    C3 F478                JMP     ..W0
F498    E1                      ..W2:    POP     H
F499    D1                      POP     D
F49A    78                      ..W3:    MOV     A,B
F49B    CD F6A2                CALL    PBYTE ;PUNCH FILE SIZE
F49E    CD F69D                CALL    PADR  ;AND ADDR.
F4A1    78                      MOV     A,B  ;SET-UP CHECKSUM
F4A2    84                      ADD     H
F4A3    85                      ADD     L
F4A4    57                      MOV     D,A ;CHECKSUM IN D
F4A5    AF                      XRA     A  ;ZERO FILE TYPE
F4A6    CD F6A2                CALL    PBYTE
F4A9    7E                      ..W4:    MOV     A,M
F4AA    82                      ADD     D  ;UPDATE CHECKSUM
F4AB    57                      MOV     D,A
F4AC    7E                      MOV     A,M
F4AD    CD F6A2                CALL    PBYTE
F4B0    23                      INX     H
F4B1    05                      ICR     B
F4B2    C2 F4A9                JNZ    ..W4
F4B5    AF                      XRA     A
F4B6    92                      SUB     D
F4B7    C3 F6A2                JMP     PBYTE

0048    ;ZW=-WRITE
;
F4BA    CD F6B6                XAM:    CALL    PCHK
F4BD    21 F7D6                LXI     H,ACTBL ;POINT TO REG. TABLE
F4C0    060B                    MVI     B,ACTSZ ;SET UP B
F4C2    DA F4F9                JC      ..X6
F4C5    BE                      ..X0:    CMP     M  ;VALID REG. NAME?
F4C6    CA F4D2                JZ      ..X1 ; YES
F4C9    23                      INX     H  ;ELSE TEST NEXT ONE
F4CA    23                      INX     H  ;SKIP OFFSET

```

01/07/79 22:40:00

KAPPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>
 COPYRIGHT 1979 BY APPELZAP CORP.

F4CB	05		DCR	B	;END OF TABLE?
F4CC	CA F0BA		JZ	ERROR	; YES
F4CF	C3 F4C5		JMP	..X0	;ELSE KEEP LOOKING
F4D2	CD F568	..X1:	CALL	BLK	
F4D5	CD F511	..X2:	CALL	..X8	;GET & PRINT REG(S)
F4D8	CD F6B1	..X3:	CALL	COPCK	;MODIFY?
F4DB	CA F4F2		JZ	..X5	; NO, DELIMITER ENTERED
F4DE	E5		PUSH	H	;SAVE TABLE POINTER
F4DF	C5		PUSH	B	;SAVE FLAG TEST (B)
F4E0	CD F626		CALL	EXF	;GET NEW VALUE
F4E3	E1		POP	H	;IN HL
F4E4	C1		POP	B	;B=FLAG BYTE
F4E5	F5		PUSH	PSW	;A=DELIMITER
F4E6	7D		MOV	A,L	;L=LOW BYTE
F4E7	12		STAX	D	;STORE IT
F4E8	78		MOV	A,B	;GET FLAG
F4E9	17		RAL		;TEST BIT 7
F4EA	D2 F4F0		JNC	..X4	;SINGLE BYTE
F4ED	13		INX	D	;ELSE
F4EE	7C		MOV	A,H	; SAVE
F4EF	12		STAX	D	; HIGH BYTE
F4F0	F1	..X4:	POP	PSW	;GET DELIMITER
F4F1	E1		POP	H	;RESTORE TABLE POINTER
F4F2	D8	..X5:	RC		;CR=DONE
F4F3	7E		MOV	A,M	;END OF TABLE?
F4F4	B7		ORA	A	;TEST BIT 7
F4F5	F8		RM		;YES, DONE
F4F6	C3 F4D5		JMP	..X2	;ELSE CONTINUE
F4F9	CD F5CC	..X6:	CALL	CRLF	;FULL REGISTER DISPLAY
F4FC	CD F568	..X7:	CALL	BLK	;SPACE OVER
F4FF	7E		MOV	A,M	;GET REGISTER NAME
F500	B7		ORA	A	;END OF TABLE?
F501	F8		RM		;YES, RETURN
F502	4F		MOV	C,A	;ELSE PRINT IDENTIFIER
F503	CD F56A		CALL	CO	; ON CONSOLE
F506	0E3D		MVI	C,'='	;FOR READABILITY
F508	CD F56A		CALL	CO	
F50B	CD F511		CALL	..X8	;GET & PRINT REG(S)
F50E	C3 F4FC		JMP	..X7	
F511	23	..X8:	INX	H	;POINT TO DISPLACEMENT
F512	7E		MOV	A,M	;GET IT
F513	23		INX	H	;POINT TO NEXT IN TABLE
F514	EB		XCHG		;SAVE IN DE
F515	47		MOV	B,A	;SAVE FOR FLAGS
F516	E63F		ANI	3FH	;KILL FLAGS
F518	6F		MOV	L,A	;CALCULATE DISPLACEMENT
F519	2600		MVI	H,0	
F51B	39		DAD	SP	;UP IN STACK
F51C	23		INX	H	;ADJUST FOR RET IN STACK
F51D	23		INX	H	
F51E	78		MOV	A,B	;TEST FOR "M"
F51F	E640		ANI	40H	;BIT 6
F521	CA F528		JZ	..X9	;NO, NOT "M"

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F524 7E          MOV      A,M      ;ELSE GET "M" POINTER
F525 2B          DCX      H        ; INSTEAD
F526 6E          MOV      L,M      ; IN HL
F527 67          MOV      H,A      ; (WHERE ELSE)
F528 7E          ..X9:  MOV      A,M      ;GET THE VALUE
F529 CD F66A     CALL     LBYTE   ;AND PRINT IT
F52C EB          XCHG                     ;SWITCH POINTERS
F52D 78          MOV      A,B      ;TEST FLAG
F52E 17          RAL                     ;SINGLE OR DOUBLE?
F52F D0          RNC                     ;SINGLE
F530 1B          DCX      D        ;DOUBLE
F531 1A          LDAX   D        ;GET IT
F532 C3 F66A     JMP      LBYTE   ;PRINT IT & RETURN

;
;ZX=-XAM
;
F535 CD F793     QUERY:  CALL     TI        ;SEE IF IN OR OUT
F538 21 001D     LXI      H,QLOC   ;PRESET
F53B 39          DAD      SP        ;TO ROUTINE IN EXIT AREA
F53C E5          PUSH     H        ;FOR BOTH ROUTINES
F53D FE4F       CPI      '0'      ;OUT?
F53F C2 F54D     JNZ      ..QI      ; NO, MUST BE IN
F542 CD F5F7     CALL     EXPC     ;GET PORT & VALUE
F545 7B          MOV      A,E      ;L=PORT E=VALUE
F546 4D          MOV      C,L
F547 E1          POP      H
F548 71          MOV      M,C
F549 2B          DCX      H
F54A 36D3       MVI      M,(OUT)
F54C E9          PCHL                     ;DO IT & RETURN

;
;..QI:
F54D FE49       CPI      'I'
F54F C2 F0BA     JNZ      ERROR
F552 CD F623     CALL     EXPR
F555 C1          POP      B
F556 21 F5B7     LXI      H,BITS   ;SET-UP A RETURN
F559 E3          XTHL
F55A 71          MOV      M,C      ;SET PORT NUMBER
F55B 2B          DCX      H
F55C 36DB       MVI      M,(IN)  ;SET FOR INPUT
F55E E9          PCHL                     ;DO IT

;
;ZQ=-QUERY
;
F55F CD F081     SIZE:  CALL     MEMSIZ
;
F562 CD F5CC     LFADR:  CALL     CRLF
;
F565 CD F665     HLSP:  CALL     LADR
;
F568 0E20       BLK:   MVI      C,' '
;
F56A 3A FFFF     CO:    LDA      -1
F56D E603       ANI      # CMSK

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>
 COPYRIGHT 1979 BY APPLEZAP CORP.

```

F56F    CA F581          JZ      CRTOUT
F572    3D              DCR      A      ;
F573    C2 F58C        JNZ      COU

;
F576    DB02          TTYOUT: IN    TTS
F578    E680          ANI      TTYBE
F57A    C2 F576        JNZ      TTYOUT
F57D    79            MOV      A,C
F57E    D303          OUT      TTO
F580    C9            RET

;
F581    DB00          CRTOUT: IN    CRTS
F583    E680          ANI      CRTBE
F585    C2 F581        JNZ      CRTOUT
F588    79            MOV      A,C
F589    D301          OUT      CRT0
F58B    C9            RET

;
F58C    3D            COU:    DCR      A      ;BATCH
F58D    C2 F803        JNZ      COLOC  ;NO

;
F590    3A FFFF        LO:    LDA      -1
F593    E6C0          ANI      # LMSK
F595    CA F581        JZ      CRTOUT ;USE MAIN CONSOLE
F598    FE40          CPI      LCRT
F59A    CA F576        JZ      TTYOUT ;USE PRINTER
F59D    FE80          CPI      LINE
F59F    C2 F80F        JNZ      LULOC  ;MUST BE USER DEFINED
;ELSE USE DATA TRANSFER

;
F5A2    DB04          LNLOC:  IN    RCSS
F5A4    E680          ANI      PCSBE
F5A6    C2 F5A2        JNZ      LNLOC
F5A9    79            MOV      A,C
F5AA    D305          OUT      PCASO
F5AC    C9            RET

;
F5AD    E60F          CONV:  ANI      0FH
F5AF    C690          ADI      90H
F5B1    27            DAA
F5B2    CE40          ACI      40H
F5B4    27            DAA
F5B5    4F            MOV      C,A
F5B6    C9            RET

;
F5B7    5F            BITS:  MOV      E,A
F5B8    1608          MVI      D,8
F5BA    CD F568        CALL     BLK
F5BD    7B            ..BI:  MOV      A,E
F5BE    17            RAL
F5BF    5F            MOV      E,A
F5C0    3E00          MVI      A,0
F5C2    CE30          ACI      '0'
F5C4    4F            MOV      C,A
F5C5    CD F56A        CALL     CO

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F5C8      15          DCR      D
F5C9      C2 F5BD    JNZ      ..BI
;
F5CC      E5          CRLF:   PUSH   H
F5CD      C5          PUSH   B      ;SAVE BC
F5CE      0605       MVI    B,5
F5D0      CD F0A3    CALL   TOM
F5D3      C1          POP    B
F5D4      E1          POP    H
F5D5      C9          RET
;
F5D6      3A FFFF    CSTS:   LDA    -1
F5D9      E603       ANI    # CMSK
F5DB      CA F5EE    JZ     ..CS1  ;CRT
F5DE      3D         DCR    A
F5DF      CA F5E7    JZ     ..CS0  ;TTY
F5E2      3D         DCR    A
F5E3      C8         RZ     ;BATCH MODE
F5E4      C3 F806    JMP    CSLOC  ;USER
;
F5E7      DB02       ..CS0:  IN     TTS
F5E9      E601       ANI    TTYDA
F5EB      C3 F5F2    JMP    ..CS2
;
F5EE      DB00       ..CS1:  IN     CRTS
F5F0      E601       ANI    CRTDA
F5F2      3EFF       ..CS2:  MVI    A,TRUE
F5F4      C8         RZ
F5F5      2F         CMA
F5F6      C9          RET
;
; THIS ROUTINE WILL GET TWO PARAMETERS
; FROM THE KEYBOARD, AND RETURN WITH THE
; 'C' REGISTER IN A, & CARRY SET IF THE
; TERMINATOR WAS A CARRIAGE RETURN. OTHERWISE,
; IT WILL GET THE THIRD PARAMETER. IF THE
; THIRD PARAMETER IS NON-ZERO, IT WILL RETURN
; WITH THE THIRD PARAMETER IN 'A'. IF IT IS
; ZERO, IT WILL RETURN WITH THE DEFAULT PARAM.
; - IN EITHER CASE, IF THREE PARAMETERS WERE
; ENTERED, IT WILL RETURN WITH THE CARRY CLEAR.
;
F5F7      C5          EXPC:   PUSH   B      ;SAVE DEFAULT PARAMETER
F5F8      CD F623    CALL   EXPR  ;GET 1st.
F5FB      DA F0BA    JC     ERROR ;CR ENTERED TOO SOON
F5FE      CD F623    CALL   EXPR  ;GET 2nd. PARAMETER
F601      D1         POP    D      ;2nd. IN DE
F602      E1         POP    H      ;1st. IN HL
F603      C1         POP    B      ;REMOVE DEFAULT
F604      E5         PUSH   H      ;SAVE 1st. PARAMETER
F605      79         MOV    A,C    ;USE DEFAULT
F606      DA F615    JC     ..E1  ;NO THIRD PARAMETER
F609      C5         PUSH   B      ;SAVE DEFAULT AGAIN
F60A      CD F623    CALL   EXPR  ;GET 3rd. PARAMETER

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F60D   C1           POP     B           ;BC=TRUE 3rd. PARAMETER
F60E   79           MOV     A,C       ;TEST IT
F60F   E1           POP     H           ;HL=DEFAULT
F610   B7           ORA     A           ;TEST LOW BYTE
F611   C2 F615      JNZ     ..E1       ;OK, TAKE IT
F614   7D           MOV     A,L       ;ELSE USE DEFAULT
F615   E1           ..E1: POP     H           ;GET 1st. PARAM
F616   F5           PUSH    PSW       ;SAVE ACC & FLAGS
F617   CD F5CC      CALL   CRLF
F61A   F1           POP     PSW
F61B   C9           RET

;
; THIS ROUTINE RETURNS ONLY IF THREE PARAMETERS
; WERE ENTERED. LESS THAN THREE RESULTS IN AN
; ERROR CONDITION.
;
F61C   CD F5F7      EXP3:  CALL   EXPC   ;GET THREE PARAMETERS
F61F   DA FOBA      JC     ERROR   ;I SAID 3
F622   C9           RET

;
; EXPR: CALL TI ;GET KEYBOARD
; EXF: LXI H,0 ;INITIALIZE HL
; ..E1: MOV B,A ;SAVE KEYBOARD
F623   CD F793      ..E1: CALL NIBBLE ;CONVERT ASCII TO HEX
F626   21 0000      JC     ..E2     ;NOT LEGAL
F629   47           DAD     H        ;HL*16
F62A   CD F68D      DAD     H
F62D   DA F63C      DAD     H
F630   29           DAD     H
F631   29           DAD     H
F632   29           DAD     H
F633   29           DAD     H
F634   B5           ORA     L        ;ADD IN NIBBLE
F635   6F           MOV     L,A
F636   CD F793      CALL   TI        ;GET NEXT KEYBOARD
F639   C3 F629      JMP     ..E1     ;AND CONTINUE
F63C   E3           ..E2: XTHL     ;STICK PARAMETER IN STACK
F63D   E5           PUSH    H        ;REPLACE RETURN
F63E   78           MOV     A,B      ;TEST CHARACTER
F63F   CD F689      CALL   QCHK     ;FOR DELIMITERS
F642   C2 FOBA      JNZ     ERROR   ;ILLEGAL
F645   C9           RET

;
; HILOX: CALL HILO
; RNC ;RETURN IF OK
; POP D ;ELSE RETURN
; RET ; ONE LEVEL BACK
F646   CD F64C      HILOX: CALL HILO
F649   D0           RNC
F64A   D1           PRET: POP     D
F64B   C9           RET

;
; HILO: INX H
; MOV A,H
; ORA L
; STC
; RZ
; MOV A,E
; SUB L
; MOV A,D
; SBB H
F64C   23           HILO: INX     H
F64D   7C           MOV     A,H
F64E   B5           ORA     L
F64F   37           STC
F650   C8           RZ
F651   7B           MOV     A,E
F652   95           SUB     L
F653   7A           MOV     A,D
F654   9C           SBB     H

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F655      C9              RET
;
F656      CD F5F7      HEXN:  CALL  EXPC
F659      E5              PUSH  H
F65A      19              DAD   D
F65B      CD F565      CALL  HLSP
F65E      E1              POP   H
F65F      7D              MOV   A,L
F660      93              SUB   E
F661      6F              MOV   L,A
F662      7C              MOV   A,H
F663      9A              SBB  D
F664      67              MOV   H,A
;
000F      ;
;ZH=.-HEXN
;
F665      7C              LADR:  MOV   A,H
F666      CD F66A      CALL  LBYTE
F669      7D              MOV   A,L
;
F66A      F5              LBYTE:  PUSH  PSW
F66B      0F              RRC
F66C      0F              RRC
F66D      0F              RRC
F66E      0F              RRC
F66F      CD F673      CALL  ..L
F672      F1              POP   PSW
F673      CD F5AD      ..L:   CALL  CONV
F676      C3 F56A      JMP   CO
;
F679      01 08FF      MARK:  LXI   B,08FFH ;Preset for rub-outs
F67C      C3 F682      JMP   LEED
;
F67F      01 4800      LEAD:  LXI   B,4800H ;Preset for NULLs
F682      CD F6CC      LEED:  CALL  PO
F685      05              DCR  B
F686      C2 F682      JNZ  LEED
F689      C9              RET
;
F68A      CD F77F      RIBBLE: CALL  RIX
F68D      D630      NIBBLE: SUI  '0'
F68F      D8              RC
F690      FE17      CPI   'G'-'0'
F692      3F              CMC
F693      D8              RC
F694      FE0A      CPI   10
F696      3F              CMC
F697      D0              RNC
F698      D607      SUI  'A'-'9'-1
F69A      FE0A      CPI   10
F69C      C9              RET
;
F69D      7C              PADR:  MOV   A,H
F69E      CD F6A2      CALL  PBYTE

```


01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

COPYRIGHT 1979 BY APPLEZAP CORP.

```

F6A1      7D              MOV      A,L
;
F6A2      F5      PBYTE:  PUSH     PSW
F6A3      0F              RRC
F6A4      0F              RRC
F6A5      0F              RRC
F6A6      0F              RRC
F6A7      CD F6AB      CALL     ..L
F6AA      F1              POP      PSW
F6AB      CD F5AD      ..L:    CALL     CONV
F6AE      C3 F6CC      JMP      PO
;
F6B1      0E2D      COPCK:  MVI     C,'-'
F6B3      CD F56A      CALL     CO
;
F6B6      CD F793      PCHK:   CALL     TI
;
F6B9      FE20      QCHK:   CPI     ' '
F6BB      C8              RZ
F6BC      FE2C      CPI     ' '
F6BE      C8              RZ
F6BF      FE0D      CPI     CR
F6C1      37              STC
F6C2      C8              RZ
F6C3      3F              CMC
F6C4      C9              RET
;
F6C5      0E0D      PEOL:   MVI     C,CR
F6C7      CD F6CC      CALL     PO
F6CA      0E0A      MVI     C,LF
;
F6CC      3A FFFF      PO:     LDA     -1
F6CF      E630      ANI     # PMSK
F6D1      CA F5A2      JZ      LNLOC ;DATA XFER DEVICE
F6D4      FE10      CPI     PTTY
F6D6      CA F576      JZ      TTYOUT ;PRINTER DEVICE
F6D9      FE20      CPI     PCAS
F6DB      C2 F80C      JNZ     PULOC ;USER DEFINED
;
F6DE      DB06      PTPL:   IN     PPSTAT ;PARALLEL PORT
F6E0      E680      ANI     PPBE
F6E2      C2 F6DE      JNZ     PTPL
F6E5      79              MOV     A,C
F6E6      D307      OUT     PPDATA
F6E8      C9              RET
;
F6E9      CD F5F7      UNLD:   CALL     EXPC
F6EC      CD F705      CALL     WAIT
F6EF      CD F67F      CALL     LEAD
F6F2      CD F679      CALL     MARK
F6F5      4E              ..U1:   MOV     C,M
F6F6      CD F6CC      CALL     PO
F6F9      CD F64C      CALL     HILO
F6FC      D2 F6F5      JNC     ..U1

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

C COPYRIGHT 1979 BY APPLEZAP CORP.

```

F6FF      CD F679                CALL    MARK
;
0019      ;ZU=-UNLD
;
F702      CD F67F                NULL:   CALL    LEAD
;
0003      ;ZN=-NULL
;
F705      3A FFFF                WAIT:   LDA     -1
F708      E603                   ANI     # CMSK
F70A      C8                     RZ
;
F70B      3A FFFF                CI:     LDA     -1
F70E      E603                   ANI     # CMSK
F710      CA F721                JZ      CRTIN
F713      3D                     DCR     A
F714      C2 F72B                JNZ     CIU
;
F717      DB02                   TTYIN:  IN     TTS
F719      E601                   ANI     TTYDA
F71B      C2 F717                JNZ     TTYIN
F71E      DB03                   IN     TTI
F720      C9                     RET
;
F721      DB00                   CRTIN:  IN     CRTS
F723      E601                   ANI     CRTDA
F725      C2 F721                JNZ     CRTIN
F728      DB01                   IN     CRTI
F72A      C9                     RET
;
F72B      3D                     CIU:    DCR     A ; BATCH?
F72C      C2 F800                JNZ     CILOC ; NO, MUST BE USER
;
F72F      3A FFFF                RI:     LDA     -1
F732      E60C                   ANI     # RMSK
F734      D302                   OUT     TTS ; PULSE A PORT TO SHOW REQUEST
F736      C2 F746                JNZ     ..R3 ; NEXT
;
F739      CD F76A                ;DATA XFER
F73C      DB04                   ..R4:   CALL    ..R2 ; ABORT?
F73E      E601                   ANI     RCSS
F740      C2 F739                JNZ     ..R4
F743      DB05                   IN     RCSD
F745      C9                     RET
;
F746      FE04                   ;..R3:   CPI     RTTY ; IS IT PRINTER?
F748      C2 F758                JNZ     ..R5 ; NEXT
;
F74B      CD F76A                ;PRINTER
F74E      DB02                   ..R1:   CALL    ..R2 ; SEE IF ABORT
F750      E601                   ANI     TTS
F752      C2 F74B                JNZ     ..R1
F755      DB03                   IN     TTI
F757      C9                     RET

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

C YRIGHT 1979 BY APPLEZAP CORP.

```

;
F758 FE08 ;..R5: CPI RCAS
F75A C2 F809 JNZ RULOC ;USER DEFINED
;PARALLEL PORT
F75D CD F76A ;..R6: CALL ;..R2
F760 DB06 IN PPSTAT
F762 E601 ANI PPDA
F764 C2 F75D JNZ ;..R6
F767 DB07 IN PPDATA
F769 C9 RET

;
F76A 3A FFFF ;..R2: LDA -1 ;MAKE SURE CONSOLE=0
F76D E603 ANI # CMSK
F76F C0 RNZ
F770 CD F5D6 CALL CSTS ;ANYTHING WAITING THERE?
F773 B7 ORA A
F774 C8 RZ ;NO, CONTINUE
F775 CD F78D CALL KI ;ELSE GET IT
F778 FE03 CPI 3 ;CONTROL-C?
F77A C0 RNZ
F77B F1 POP PSW ;ELSE RETURN
F77C AF XRA A ;WITH CARRY SET
F77D 37 STC
F77E C9 RET

;
F77F CD F785 RIX: CALL RIFF
F782 E67F ANI 7FH
F784 C9 RET

;
F785 CD F72F RIFF: CALL RI
F788 DA F0BA JC ERROR
F78B BA CMP D
F78C C9 RET

;
F78D CD F70B KI: CALL CI ;GET CONSOLE CHARACTER
F790 E67F ANI 7FH ;KILL PARITY BIT
F792 C9 RET

;
F793 CD F78D TI: CALL KI
F796 C8 RZ
F797 FE7F CPI 7FH
F799 C8 RZ ;TEST FOR RUB-OUT
F79A FE0D CPI CR ;IGNORE CR'S
F79C C8 RZ
F79D C5 PUSH B
F79E 4F MOV C,A
F79F CD F56A CALL CO
F7A2 79 MOV A,C
F7A3 C1 POP B
F7A4 FE40 CPI 'A'-1 ;CONVERT TO UPPER CASE
F7A6 D8 RC
F7A7 FE7B CPI 'z'+1
F7A9 D0 RNC
F7AA E65F ANI 05FH

```

01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

C YRIGHT 1979 BY APPLEZAP CORP.

```

F7AC      C9          RET
;
;
; <SYSTEM I/O LOOK-UP TABLE>
;
; THE FIRST CHARACTER IS THE DEVICE NAME
; (ONE LETTER) AND THE NEXT FOUR ARE THE
; NAMES OF THE FOUR POSSIBLE DRIVERS TO BE
; ASSIGNED.
;
F7AD      LTBL:
;
F7AD      43          .BYTE   'C'      ; CONSOLE ASSIGNMENTS
;
F7AE      43          .BYTE   'C'      ; CRT
F7AF      50          .BYTE   'P'      ; PRINTER
F7B0      42          .BYTE   'B'      ; BATCH= COMMANDS FROM READER
F7B1      55          .BYTE   'U'      ; CUSE  USER
;
;
F7B2      52          .BYTE   'R'      ; READER ASSIGNMENTS
;
F7B3      44          .BYTE   'D'      ; DATA TRANSFER DEVICE
F7B4      50          .BYTE   'P'      ; PRINTER
F7B5      41          .BYTE   'A'      ; ALTERNATE (PARALLEL)
F7B6      55          .BYTE   'U'      ; RUSER  USER
;
;
F7B7      50          .BYTE   'P'      ; PUNCH ASSIGNMENTS
;
F7B8      44          .BYTE   'D'      ; DATA TRANSFER DEVICE
F7B9      50          .BYTE   'P'      ; PRINTER
F7BA      41          .BYTE   'A'      ; ALTERNATE (PARALLEL)
F7BB      55          .BYTE   'U'      ; PUSER  USER
;
;
F7BC      4C          .BYTE   'L'      ; LIST ASSIGNMENTS
;
F7BD      43          .BYTE   'C'      ; CRT
F7BE      50          .BYTE   'P'      ; PRINTER
F7BF      44          .BYTE   'D'      ; DATA TRANSFER DEVICE
F7C0      55          .BYTE   'U'      ; LUSER  USER
;
;
F7C1      EXIT:
F7C1      D1          POP      D
F7C2      C1          POP      B
F7C3      F1          POP      PSW
F7C4      E1          POP      H
F7C5      F9          SPHL
F7C6      00          NOP
F7C7      21 0000     LXI      H,0      ; COULD BE EI
F7C8      HLX        =     -2
F7CA      C3 0000     JMP      0
F7CB      PCX        =     -2

```


01/07/79 22:40:00

<APPLE MONITOR, *ECT ROM* V1.0 JAN 07, 1979>

+ ++ SYMBOL TABLE +++++

ACTBL	F7D6	ACTSZ	000B	ALOC	0007	APPLE	F000
ASSIGN	F15D	BASE	F000	BATCH	0002	BEGIN	F0D8
BELL	0007	BITS	F5B7	BLK	F568	BLOC	0005
BRANCH	F1A6	CCHK	F0B4	CCRT	0000	CI	F70B
CILOC	F800	CIU	F72B	CLOC	0004	CMSK	00FC
CO	F56A	COLOC	F803	CONFIG	0000	CONV	F5AD
COPCK	F6B1	COU	F58C	CR	000D	CRLF	F5CC
CRTBE	0080	CRTDA	0001	CRTI	0001	CRTIN	F721
CRTD	0001	CRTOUT	F581	CRTS	0000	CSLOC	F806
CSTS	F5D6	CTTY	0001	CUSE	0003	DISP	F1B4
DLOC	0003	ELOC	0002	ENDX	F7D6	EOF	F1D4
ERROR	F0BA	EXF	F626	EXIT	F7C1	EXP3	F61C
EXPC	F5F7	EXPR	F623	FALSE	0000	FIL	0000
FILL	F1F0	FLOC	0006	GOTO	F1FE	HEXN	F656
HILO	F64C	HILOX	F646	HLOC	0012	HLSP	F565
HLX	F7C8	IO	0000	IOCHK	F1A2	IOSET	F19D
J	F812	KI	F78D	LADR	F665	LBYTE	F66A
LCRT	0040	LEAD	F67F	LEED	F682	LF	000A
LFADR	F562	LINE	0080	LLOC	000F	LMSK	003F
LNLOC	F5A2	LO	F390	LOAD	F267	LTBL	F7AD
LTTY	0000	LULOC	F80F	LUSER	00C0	MARK	F679
MAX	0007	MEMCK	F09A	MEMSIZ	F0B1	MOVE	F2AD
MSG	F0C5	MSGL	0013	NIBBLE	F68D	NULL	F702
Q	FFFF	PADR	F69D	PBYTE	F6A2	PCAS	0020
PCASD	0005	PCASS	0004	PCHK	F6B6	PCSBE	0080
PCX	F7CB	PEOL	F6C5	PLOC	0015	PMSK	00CF
PO	F6CC	PPBE	0080			PPDA	0001
PPDATA	0007	PPSTAT	0006	PPTP	0000	PRET	F64A
PTPL	F6DE	PTTY	0010	PULOC	F80C	PUSER	0030
PUTA	F2B9	QCHK	F6B9	QIO	F7D3	QLOC	001D
QUERY	F535	RCAS	0008	RCSL	0005	RCSDA	0001
RCSS	0004	READ	F31F	RI	F72F	RIBBLE	F68A
RIFF	F785	RIX	F77F	RMSK	00F3	RPTR	0000
RST7	0038	RTTY	0004	RUB	00FF	RULOC	F809
RUSER	000C	SIZE	F55F	SLOC	0009	STARO	F10B
START	F0FC	SUBS	F3F5	T1A	F7CD	TBL	F129
TEST	F24C	TI	F793	TLOC	0014	TLOCX	0016
TOM	F0A3	TOM1	F0A6	TRAP	F01E	TRUE	FFFF
TTI	0003	TTO	0003	TTS	0002	TTYBE	0080
TTYDA	0001	TTYIN	F717	TTYOUT	F576	TYPE	F420
UNLD	F6E9	USER	F800	UTAB	F880	VERIFY	F44C
WAIT	F705	WHERE	F2DB	WRITE	F472	XAM	F4BA
Z	F7F0	ZA	0040	ZB	000E	ZC	0000
ZD	0020	ZE	001C	ZF	000E	ZG	004E
ZH	000F	ZJ	001B	ZL	0046	ZM	000C
ZN	0003	ZP	0022	ZQ	002A	ZR	0006
ZS	002B	ZT	002C	ZU	0019	ZV	0026
ZW	0048	ZX	007B	ZY	0044		

```

F000 C3 D8 F0 C3 0B F7 C3 2F F7 C3 6A F5 C3 CC F6 C3
F010 90 F5 C3 D6 F5 C3 A2 F1 C3 9D F1 C3 9A F0 E5 D5
F020 C5 F5 11 EA FF 21 0A 00 39 06 04 EB 2B 72 2B 73
F030 D1 05 C2 2C F0 C1 0B F9 21 14 00 39 CD 7A F0 23
F040 23 C4 7A F0 CA 48 F0 03 21 0F 00 39 73 23 72 23
F050 23 71 23 70 C5 0E 40 CD 6A F5 E1 CD 65 F6 21 14
F060 00 39 11 02 00 4E 72 23 46 72 23 79 B0 CA 72 F0
F070 7E 02 23 1D C2 65 F0 C3 FC F0 7E 91 23 C0 7E 90
F080 C9 21 FF FF 24 7E 2F 77 BE 2F 77 C2 84 F0 24 7E
F090 2F 77 BE 2F 77 CA 8E F0 25 C9 E5 CD 81 F0 44 E1
FOA0 3E C0 C9 21 C5 F0 4E 23 CD 6A F5 05 C2 A6 F0 CD
FOB0 D6 F5 B7 C8 CD 8D F7 FE 03 C0 31 E2 FF 0E 2A CD
FOC0 6A F5 C3 FC F0 0D 0A 00 00 00 41 70 70 6C 65 20
FOD0 56 31 2E 30 20 45 43 54 21 EA FF F9 06 15 11 C1
FOE0 F7 1A 77 23 13 05 C2 E1 F0 CD 81 F0 E5 60 68 E5
FOF0 E5 E5 3E 00 32 FF FF 06 13 CD A3 F0 11 FC F0 D5
F100 CD CC F5 0E 3E CD 6A F5 21 29 F1 CD 93 F7 CA 0B
F110 F1 FE 20 DA 0B F1 D6 41 D8 FE 1A D0 87 85 6F 7E
F120 23 66 6F A4 3C CA BA F0 E9 5D F1 A6 F1 FF FF B4
F130 F1 D4 F1 F0 F1 FE F1 56 F6 12 F8 4C F2 15 F8 67
F140 F2 AD F2 02 F7 18 F8 B9 F2 35 F5 1F F3 F5 F3 20
F150 F4 E9 F6 4C F4 72 F4 BA F4 DB F2 5F F5 CD 93 F7
F160 21 AC F7 01 04 00 CD 86 F1 D5 CD 93 F7 D6 3D C2
F170 6A F1 4F CD 93 F7 CD 86 F1 F1 6A 26 03 3D FA 95
F180 F1 29 29 C3 7D F1 11 04 00 23 BE C8 09 14 1D C2
F190 89 F1 C3 BA F0 AC 67 CD A2 F1 A4 E5 4F 79 32 FF
F1A0 FF C9 3A FF FF C9 CD 93 F7 FE 2E C2 BA F0 21 80
F1B0 F8 C3 0B F1 0E 10 CD F7 F5 F5 CD 62 F5 F1 F5 47
F1C0 CD 68 F5 7E CD 6A F6 CD 4C F6 DA 4A F6 05 C2 C0
F1D0 F1 C3 BA F1 CD 23 F6 CD C5 F6 0E 3A CD CC F6 AF
F1E0 CD A2 F6 E1 CD 9D F6 21 00 00 CD 9D F6 C3 02 F7
F1F0 CD F7 F5 71 CD 4C F6 D2 F3 F1 D1 C3 FC F0 CD E6
F200 F6 CA 0F F2 CD 26 F6 D1 21 15 00 39 72 2B 73 FE
F210 0D CA 43 F2 16 02 21 16 00 39 E5 CD 23 F6 C1 E1
F220 F5 78 B1 CA 30 F2 71 23 70 23 0A 77 23 3E FF 02
F230 F1 DA 38 F2 15 C2 1A F2 3E C3 32 38 00 21 1E F0
F240 22 39 00 CD CC F5 D1 21 08 00 39 E9 CD F7 F5 7E
F250 47 2F 77 AE 70 CA 61 F2 D5 5F CD 65 F5 CD B8 F5
F260 D1 CD 46 F6 C3 4F F2 CD 23 F6 CD CC F5 E1 16 FF
F270 01 07 04 CD 85 F7 C2 70 F2 05 C2 73 F2 CD 85 F7
F280 CA 7D F2 77 CD 6A F5 23 CD 85 F7 CA 92 F2 77 C3
F290 87 F2 1E 01 CD 85 F7 C2 A4 F2 1C 3E 07 BE C2 94
F2A0 F2 C3 62 F5 72 23 1D C2 A4 F2 C3 8E F2 CD F7 F5
F2B0 7E 02 03 CD 46 F6 C3 B0 F2 CD 23 F6 CD CC F5 E1
F2C0 CD 8D F7 FE 04 CA 62 F5 FE 7F CA D6 F2 77 4F 23
F2D0 CD 6A F5 C3 C0 F2 2B 4E C3 D0 F2 21 00 00 4D 39
F2E0 2B EB CD 23 F6 E1 65 E5 33 0C D2 E2 F2 EB 51 E5
F2F0 01 00 00 C5 CD CC F5 C1 E1 5A 78 A1 3C C2 03 F3

```

F300	23	F9	C9	0A	03	BE	E5	C5	C2	F7	F2	1D	CA	16	F3	0A
F310	03	2E	BE	C3	08	F3	E1	E5	2B	CD	65	F6	C3	F4	F2	CD
F320	23	F6	D1	21	00	00	E5	DA	37	F3	CD	23	F6	E1	DA	37
F330	F3	E3	CD	23	F6	E1	E3	E5	D5	CD	CC	F5	CD	7F	F7	D6
F340	3A	47	E6	FE	C2	3C	F3	57	CD	D4	F3	5F	CD	D4	F3	F5
F350	CD	D4	F3	E1	6F	CD	D4	F3	B7	78	C1	CA	65	F3	EB	E3
F360	EB	19	EB	E3	EB	1C	1D	CA	E7	F3	09	C5	47	3D	CA	86
F370	F3	CD	D4	F3	77	23	1D	C2	71	F3	CD	D4	F3	CA	3C	F3
F380	CD	65	F6	C3	BA	F0	CD	BE	F3	77	D2	B6	F3	E5	21	05
F390	00	39	CD	BE	F3	D2	A5	F3	1D	E3	35	77	E3	CA	92	F3
F3A0	23	23	C3	92	F3	86	E3	23	77	2B	7E	E3	2B	86	E1	77
F3B0	23	7E	CE	00	77	1D	23	1D	C2	86	F3	C3	7A	F3	05	C2
F3C0	C9	F3	CD	D4	F3	1D	4F	06	08	CD	D4	F3	D5	57	79	17
F3D0	4F	7A	D1	C9	C5	CD	8A	F6	07	07	07	07	4F	CD	8A	F6
F3E0	B1	4F	82	57	79	C1	C9	C1	C1	7C	B5	C8	EB	21	15	00
F3F0	39	72	2B	73	C9	CD	23	F6	E1	D8	7E	CD	6A	F6	CD	B1
F400	F6	D8	CA	12	F4	FE	5F	CA	1C	F4	E5	CD	26	F6	D1	E1
F410	73	D8	23	7D	E6	07	CC	62	F5	C3	FA	F3	2B	C3	13	F4
F420	0E	40	CD	F7	F5	F5	CD	62	F5	F1	F5	47	7E	E6	7F	FE
F430	20	D2	36	F4	3E	2E	FE	7D	D2	34	F4	4F	CD	6A	F5	CD
F440	4C	F6	DA	4A	F6	05	C2	2C	F4	C3	26	F4	CD	F7	F5	0A
F450	D5	5E	BB	CA	6A	F4	C5	47	CD	65	F5	7B	CD	6A	F6	CD
F460	68	F5	78	CD	6A	F6	CD	CC	F5	C1	D1	03	CD	46	F6	C3
F470	4F	F4	CD	F7	F5	CD	05	F7	CD	C5	F6	01	3A	00	CD	CC
F480	F6	D5	E5	04	CD	4C	F6	DA	98	F4	3E	18	90	C2	83	F4
F490	E1	CD	9A	F4	D1	C3	78	F4	E1	D1	78	CD	A2	F6	CD	9D
F4A0	F6	78	84	85	57	AF	CD	A2	F6	7E	82	57	7E	CD	A2	F6
F4B0	23	05	C2	A9	F4	AF	92	C3	A2	F6	CD	B6	F6	21	D6	F7
F4C0	06	0B	DA	F9	F4	BE	CA	D2	F4	23	23	05	CA	BA	F0	C3
F4D0	C5	F4	CD	68	F5	CD	11	F5	CD	B1	F6	CA	F2	F4	E5	C5
F4E0	CD	26	F6	E1	C1	F5	7D	12	78	17	D2	F0	F4	13	7C	12
F4F0	F1	E1	D8	7E	B7	F8	C3	D5	F4	CD	CC	F5	CD	68	F5	7E
F500	B7	F8	4F	CD	6A	F5	0E	3D	CD	6A	F5	CD	11	F5	C3	FC
F510	F4	23	7E	23	EB	47	E6	3F	6F	26	00	39	23	23	78	E6
F520	40	CA	28	F5	7E	2B	6E	67	7E	CD	6A	F6	EB	78	17	D0
F530	1B	1A	C3	6A	F6	CD	93	F7	21	1D	00	39	E5	FE	4F	C2
F540	4D	F5	CD	F7	F5	7B	4D	E1	71	2B	36	D3	E9	FE	49	C2
F550	BA	F0	CD	23	F6	C1	21	B7	F5	E3	71	2B	36	DB	E9	CD
F560	81	F0	CD	CC	F5	CD	65	F6	0E	20	3A	FF	FF	E6	03	CA
F570	81	F5	3D	C2	8C	F5	DB	02	E6	80	C2	76	F5	79	D3	03
F580	C9	DB	00	E6	80	C2	81	F5	79	D3	01	C9	3D	C2	03	F8
F590	3A	FF	FF	E6	C0	CA	81	F5	FE	40	CA	76	F5	FE	80	C2
F5A0	0F	F8	DB	04	E6	80	C2	A2	F5	79	D3	05	C9	E6	0F	C6
F5B0	90	27	CE	40	27	4F	C9	5F	16	08	CD	68	F5	7B	17	5F
F5C0	3E	00	CE	30	4F	CD	6A	F5	15	C2	BD	F5	E5	C5	06	05
F5D0	CD	A3	F0	C1	E1	C9	3A	FF	FF	E6	03	CA	EE	F5	3D	CA
F5E0	E7	F5	3D	C8	C3	06	F8	DB	02	E6	01	C3	F2	F5	DB	00
F5F0	E6	01	3E	FF	C8	2F	C9	C5	CD	23	F6	DA	BA	F0	CD	23

F600	F6	D1	E1	C1	E5	79	DA	15	F6	C5	CD	23	F6	C1	79	E1
F610	E7	C2	15	F6	7D	E1	F5	CD	CC	F5	F1	C9	CD	F7	F5	DA
F620	BA	F0	C9	CD	93	F7	21	00	00	47	CD	8D	F6	DA	3C	F6
F630	29	29	29	29	B5	6F	CD	93	F7	C3	29	F6	E3	E5	78	CD
F640	B9	F6	C2	BA	F0	C9	CD	4C	F6	D0	D1	C9	23	7C	B5	37
F650	C8	7B	95	7A	9C	C9	CD	F7	F5	E5	19	CD	65	F5	E1	7D
F660	93	6F	7C	9A	67	7C	CD	6A	F6	7D	F5	0F	0F	0F	0F	CD
F670	73	F6	F1	CD	AD	F5	C3	6A	F5	01	FF	08	C3	82	F6	01
F680	00	48	CD	CC	F6	05	C2	82	F6	C9	CD	7F	F7	D6	30	D8
F690	FE	17	3F	D8	FE	0A	3F	D0	D6	07	FE	0A	C9	7C	CD	A2
F6A0	F6	7D	F5	0F	0F	0F	0F	CD	AB	F6	F1	CD	AD	F5	C3	CC
F6B0	F6	0E	2D	CD	6A	F5	CD	93	F7	FE	20	C8	FE	2C	C8	FE
F6C0	0D	37	C8	3F	C9	0E	0D	CD	CC	F6	0E	0A	3A	FF	FF	E6
F6D0	30	CA	A2	F5	FE	10	CA	76	F5	FE	20	C2	0C	F8	DE	06
F6E0	E6	80	C2	DE	F6	79	D3	07	C9	CD	F7	F5	CD	05	F7	CD
F6F0	7F	F6	CD	79	F6	4E	CD	CC	F6	CD	4C	F6	D2	F5	F6	CD
F700	79	F6	CD	7F	F6	3A	FF	FF	E6	03	C8	3A	FF	FF	E6	03
F710	CA	21	F7	3D	C2	2B	F7	DB	02	E6	01	C2	17	F7	DB	03
F720	C9	DB	00	E6	01	C2	21	F7	DB	01	C9	3D	C2	00	F8	3A
F730	FF	FF	E6	0C	D3	02	C2	46	F7	CD	6A	F7	DB	04	E6	01
F740	C2	39	F7	DB	05	C9	FE	04	C2	58	F7	CD	6A	F7	DB	02
F750	E6	01	C2	4B	F7	DB	03	C9	FE	08	C2	09	F8	CD	6A	F7
F760	DE	06	E6	01	C2	5D	F7	DE	07	C9	3A	FF	FF	E6	03	C0
F770	CD	D6	F5	B7	C8	CD	8D	F7	FE	03	C0	F1	AF	37	C9	CD
F780	85	F7	E6	7F	C9	CD	2F	F7	DA	EA	F0	BA	C9	CD	0B	F7
F790	E6	7F	C9	CD	8D	F7	C8	FE	7F	C8	FE	0D	C8	C5	4F	CD
F7A0	6A	F5	79	C1	FE	40	D8	FE	7B	D0	E6	5F	C9	43	43	50
F7B0	42	55	52	44	50	41	55	50	44	50	41	55	4C	43	50	44
F7C0	55	D1	C1	F1	E1	F9	00	21	00	00	C3	00	00	00	00	00
F7D0	00	00	00	DB	00	C9	41	07	42	05	43	04	44	03	45	02
F7E0	46	06	48	12	4C	11	4D	52	50	95	53	89	FF	52	57	41
F7F0	0D	0A	28	43	29	20	31	39	37	39	20	45	43	54	0D	0A