CP/M2 ON MITS DISK
USER'S NOTES
BY
LIFEBOAT ASSOCIATES
1651 THIRD AVENUE
NEW YORK, N.Y. 10028
TELEPHONE 212 860-0300
TELEX 220501

CP/M2 ON MITS DISK

BY

LIFEBOAT ASSOCIATES
1651 THIRD AVENUE, NEW YORK,N.Y.10028

COPYRIGHT (C) 1981

This document was created by Patrick Linstruth from information
contained in "CP/M on MITS DISK - USERS NOTES", April 27, 1977,
and "CP/M2 ON NORTH STAR - DOUBLE DENSITY - QUAD CAPACITY USER'S
NOTES", December 1979, produced by Lifeboat Associates. Errors may
be reported at https://github.com/deltecent/lifeboat-cpm22.

Revision 1.1
July 2020

TABLE OF CONTENTS

INTRODUCTION.

GENERAL INPORMATION.

Thank you for buying CP/M Software from Lifeboat Associates. You have purchased the most useful software system ever written for the 8080 and Z80 microcomputers and at the same time, you have joined the largest group of disk operating system users in the world!

We have been very careful to supply you with all of the information you need to get CP/M running on your computer and to enable you to understand the system. CP/M however, like many worthwhile things will not be mastered without effort on your part. Read all of the manuals several times. Sit down at your terminal and try all of the features so that you can see what they do.

Exploring CP/M is a little like landing on the shore of a strange new land. You will first become familiar with the simple features that you will often use. Then you will penetrate the interior and use the more complex features that make this software system so powerful, but you must understand before you appreciate these features. That will come in time.

In order to help you, Lifeboat Associates has prepared this manual and maintains a staff at its New York headquarters. Call them at (212) 860-0300 if you need help with a problem. In addition, Lifeboat Associates has a huge stock of software that runs under CP/M. Refer to the sheet supplied with this manual for a listing, and regularly consult the current advertising in the various computer magazines.

CP/M AND THE ALTAIR SYSTEM.

The MITS Altair and the MITS Altair Disk System are designed to be an integrated combination of a floppy disk hardware system and the software needed to operate it. This system is expertly described in the "Altair Floppy Disk Drive and Controller Hardware Documentation" published by MITS in 1977.

The Altair DOS (Disk Operating System) and Altair BASIC are designed to run on Altair hardware and no other. Therefore, they are called "machine dependent". You will still be able to run your DOS programs by simply booting up Altair DOS at any time.

However, by booting up CP/M instead of DOS, you will have available another large body of software written on many different machines. This is possible since CP/M is designed in such a way as to be "machine independent". Programs written for CP/M communicate only with the CP/M operating system which in turn communicates with the Altair hardware. In this way, CP/M programs are completely isolated from the actual hardware in use. This is why the exact same CP/M program written for another computer will run on your

Altair in the same way. Once the user adds CP/M to the Altair system, he has made a vast amount of computer languages and applications operable on the system.

WHAT IS CP/M?

CP/M (Control Program / Microprocessors) is a software system designed to record and retrieve programs and data on floppy disks. Like all software systems, it is a collection of programs that are inter-related and are designed to accomplish specific tasks within the system. CP/M operates with 8080 and Z80 microprocessors and is largely independent of the design of the computer and floppy disk system. It has therefore been adopted for use with almost all computers using the 8080 and Z80 families of microprocessors. CP/M has become a de facto "standard" and a large group of high level languages and application software systems have been designed to run under its control.

A BRIEF HISTORY OF CP/M.

The 8080 microprocessor was designed by Intel Corporation as an improvement over their earlier 8008 microprocessor. This chip became the basis for the INTEL MDS Development System which was designed to operate with floppy disks. Gary Kildall of Digital Research designed the CP/M system to be "machine independent" as long as the microprocessor used the 8080 machine language instruction set.

In 1976, Digital Systems (later Digital Microsystems) was formed to supply floppy disk systems for use with microcomputers. The CP/M system was adopted for use with this popular floppy disk. IMSAI, one of the largest manufacturers of microcomputer equipment, also adopted a form of CP/M for its disk operating system. CROMEMCO and other manufacturers followed suit. Many computer owners selected disk systems for their computers that ran CP/M, rather than the non CP/M systems supplied by their computer manufacturers.

GETTING STARTED.

YOUR CP/M PACKAGE.

The contents of the CP/M package you have received from your dealer should contain the following:

    a. This manual "CP/M on Altair Disk User's Manual"

    b. The Lifeboat Associates edition of the Digital Research CP/M Manuals composed of the following sections:

        * CP/M 2.0 Users Guide for CP/M 1.4 Owners
        * An Introduction to CP/M Features and Facilities.
        * ED: A Context Editor For The CP/M Disk System.
        * CP/M Assembler (ASM) User's Guide
        * CP/M Dynamic Debugging Tool (DDT) User's Guide.
        * CP/M 2.0 Interface Guide.
        * CP/M 2.0 System Alteration Guide.

    c. An 8-inch Diskette containing your CP/M System.

    d. License Agreement and Registration Card.

    ACTION: Check that you have received all of the above material. If there is anything missing, contact your dealer or Lifeboat Associates. Check that the diskette is in good condition. It should not be bent or damaged in any way. Every CP/M software system has an unique serial number. It is shown on your distribution copy of the diskette and the same number is embedded in the software for identification purposes. Record this number and use it when corresponding with Lifeboat Associates regarding your software. When you copy your system disk for your own use, put the copyright notice and serial number on the copy.

    We most strongly advise that your first act upon receiving the system is to remove the write protect notch tab (thus write protecting the disk) and then make a working copy of the distribution diskette. There are various programs which have been written under MITS BASIC to do this. If you can't copy the disk immediately, then do so using the COPY.COM utility as soon as CP/M is running.

    It is very important that you NEVER WRITE on your original distribution diskette. Put it away in a safe place until (hopefully never) needed. It will provide a last resort back-up, regardless of what happens later, in the event of hardware or software malfunction. Use the disk you just created for normal work and make additional copies as needed. Remember to put the proper copyright notices on any disk you make as required by Digital Research.

MEMORY REQUIREMENTS AND ADDRESS.

In order to run CP/M, you must have at least 24K of continuous RAM memory starting at address zero. For most practical purposes between 32K and 48K will be desirable.

FIRST BOOT PROCEDURE.

The purpose of this section is to guide you in getting your CP/M system running for the first time. We suggest you read this section several times to get a "feel" for the procedure. Make sure you have at least 24K of RAM memory starting at 0. Please try to use a "working" disk which is a copy of the distribution diskette. All tracks should have been copied. If you MUST use the original, first write protect it by removing the tab. The step by step procedure now follows. Through this text, <CR> in a command means a carriage return.

Step 1. Power up computer and disk system.

Step 2. Insert your working disk in drive A.

Step 3. Use your monitor or front panel to run the computer from address 0FF00 hex, which is the location of the standard MITS boot PROM. The front panel switches need not be specially set as when booting MITS BASIC or DOS.

Step 4. You should get disk activity which will stop in about 20 seconds. The system will come to a soft halt at 103 hex and will NOT sign-on at this time.

Step 5. Use your front panel switches, monitor or other facility to deposit the correct byte from the following table at memory location 120 hex. If using a front panel, you may also examine address 0252H, set the sense switches to the correct byte value, then run.

LIST OF STANDARD TERMINALS

		STATUS / DATA PORTS
0 -	ALTAIR 88-2SIO	10H / 11H
1 -	IMSAI SIO2	03H / 02H
2 -	ALTAIR SIO (REV NON-ZERO)	00H / 01H
3 -	ALTAIR SIO (REV ZERO)	00H / 01H
4 -	PROCESSOR TECHNOLOGY 3P+S	00H / 01H
5 -	IMSAI MIO	43H / 42H
6 -	ALTAIR 88-4PIO	10H / 11H
80 -	COMPAL 80	
81 -	PROCESSOR TECHNOLOGY SOL	
82 -	XITAN SMB (ZAPPLE MONITOR)	

```
          83 - NORTH STAR HORIZON
          84 - VECTOR GRAPHIC MZ COMPUTER
```

Step 7. If you know you need a particular option, deposit the option number at location 121 hex (refer to the MODE BYTE OPTIONS section). Skip this step if you don't know what to do as the system will default to the most common case.

Step 8. Use your front panel, monitor or other facility to RUN the computer starting at memory location 100 hex. You should get a sign-on message of the form:

```
    CONFIG   Version 4.8
    (c) 1981 Lifeboat Associates

    Your CP/M console is configured in memory.
    Ready for "SAVE 24 CONFIG.COM" and "CONFIG P"
    or "SAVEUSER" to permanently patch disk.

    CP/M2 on Altair
    24K Vers 2.20
    (c) 1981 Lifeboat Associates
    A>
```

Step 9. At this point you have a working CP/M system patched in memory with your I/O. If you have been using the original diskette, make a working copy now by typing "COPY<CR>". The program will prompt you what to do, use the "A" option. After making the copy, put the original away and put the working copy into drive A.

Step 10. Permanently write the patched system onto your working disk at this time by typing "SAVEUSER<CR>". The program will prompt you.

Step 11. Test the system by removing all disks, cycling computer power (to wipe memory), re-inserting the patched disk in Drive A and doing another cold boot by going to 0FF00 hex.

Step 12. The disk should activate and you should get a sign-on message like this:

```
    CP/M2 on Altair
    24K Vers 2.20
    (c) 1981 Lifeboat Associates
    A>
```

Congratulations, your system is on the air.

Step 13. Permanently patch CONFIG.COM with your selected configuration number at 120 hex. This will cause the configurator to run automatically in the future. Follow this procedure:

```
    Type:     DDT CONFIG.COM<CR>
```

The system will respond with:

```
DDT VERS 2.2
NEXT   PC
1900 0100
-
```

Type: S120<CR>

The system will respond: 0120 FF

Type the hex byte for your system (eg 5 for IMSAI MIO) followed by a <CR>.

The system will respond: 0120 FF

Type a period '.' Followed by <CR> to exit the "S" mode.

Type a Control-C.

The system will return to the "A>" prompt.

Type: SAVE 24 CONFIG.COM<CR>

Note that the 24 was from the 1900 message of DDT.
It refers to the 24 decimal 256-byte pages occupied
by CONFIG.COM between 100H and 1900H in memory.

The system configurator program, CONFIG.COM, has now been
preset to your desired configuration.

RE-SIZING YOUR SYSTEM.

The distribution system you have received from Lifeboat Associates contains a 24K CP/M system and is designed to run in 24K of memory. This is not a large enough system to run some high level languages or large programs. You will want to use a system that takes advantage of all the memory in your computer. If you attempt to run a program that is too large for the current CP/M system, you will get an ERROR MESSAGE "LOAD ERROR". The fix for this is to extend your CP/M system to the fullest extent of your RAM memory. This may be done by running the MOVCPM.COM program provided on your system disk.

*** NOTE ***

A new system created with MOVCPM.COM is an unconfigured system. CONFIG.COM must be on the disk when attempting to boot up any unconfigured system. Additionally, CONFIG.COM should have been patched at 120 hex for your specific terminal from the table of standard configurations. For example, CONFIG.COM should have been patched with an 00 hex at address 120 hex if you have a MITS Altair computer.

This example takes you through running MOVCPM.COM to create a 48K system and putting the new system on a disk ready to boot up.

    A>MOVCPM 48<CR>

    Constructing 48K CP/M

    CP/M2 on Altair
    48K Vers 2.20
    (c) 1981 Lifeboat Associates

    New CP/M in memory at 900H (sysgen image)
    is ready for "SYSGEN" or "SAVE 40 CPM48.COM"

    A>SYSGEN<CR>

    SYSGEN Version 4.5
    Distributed by Lifeboat Associates
    for CP/M on Altair.

    Source drive NAME (or RETURN to skip) <CR>

    CP/M image in RAM at 900H is ready to write
    or reboot and "SAVE 40 CPMxx.COM"

    Destination drive NAME (or RETURN to reboot) B<CR>
    Place DESTINATION disk on B, then type RETURN <CR>
    Function complete

    Destination drive NAME (or RETURN to reboot) <CR>

        A>

     The disk in drive B is now ready to be placed in drive A and
booted up. Remember CONFIG.COM must be patched and on the disk.
Run "SAVEUSER<CR>" after the system signs on to permanently
configure the disk.

MODE BYTE OPTIONS.

There are several user selectable options available in your CP/M system. These options are:

1. Read After write.
2. Enable Interrupts after Disk Access.
3. Run AUTO after Warm or Cold Boot.
4. Set System for a Single Drive.

The choice of these functions is controlled by the state of the bits in one byte referred to as the MODE byte. The MODE byte is located at the address immediately before the USER AREA and is at 5BFF hex in the 24K running system (257F hex in the SYSGEN IMAGE). Programs may reference the MODE byte to manipulate it.

If you wish to use any of these options the best way to implement them is to change the MODE BYTE with DDT. When you are sure the option is doing what you want, run SAVEUSER and permanently save the MODE BYTE on disk.

The following paragraphs describe the controlled functions and indicate the trade-offs implied by selecting the use or suppression of each feature. Also explained are the position of each bit, the sense of the bit for invoking or disabling the function and the default value used in distribution systems. In general, the bit involved must be HIGH for the feature to be ACTIVE.

Option 1. READ AFTER WRITE.

Use of Read After Write causes the system to read and verify the data on the disk after each sector is written. Any read error would result in repeated re-tries of the write and subsequent verification until the data is correctly written onto the disk. Use of this feature will cause no degradation in the speed of disk reading, but will slow disk writing by a factor of about four times.

The position of this bit is value 40 hex (bit 6). The distribution system is shipped with this bit LOW which causes this feature to be NOT active.

Option 2. ENABLE INTERRUPTS AFTER DISK ACCESS.

The drivers of this system disable interrupts during a disk access, since the code is real time dependent. Otherwise, an interrupt during disk access would cause an error. Since the 8080 and 8085 processors are not able to read the status of the interrupt flag, the status cannot be restored to its original condition. This feature permits the user, either in general or under program

9

control, to exit from disk accesses with interrupts either enabled
or disabled.

The position of this bit is value 10 hex (bit 4). The
distribution system is shipped with this bit HIGH and thus this
feature is ACTIVE. That is, interrupts are ENABLED on exit from a
disk access.

Option 3. RUN AUTO AFTER A COLD OR WARM BOOT.

CP/M includes a feature allowing execution of a command upon
startup. This has applications in loading monitors into memory,
initializing peripheral drivers, or causing the system to
automatically load BASIC and enter an applications program for
unskilled users. Your system has the command: AUTO set into it
which will cause a file AUTO.COM to be executed if the feature is
invoked. The user must provide the program AUTO.COM or re-name a
program to that name.

Alternatively, you may insert a command line of your choice
in the CCP of up to 128 characters using the SYSGEN PROCEDURE. An
example of such a command line would be:

```
ORG CCP+7            ;Start of this patch
DB 13               ;Length of command line
DB 'MBASIC B:PROG'  ;13 character command
```

This example would load Microsoft BASIC from drive A and then cause
BASIC to load and immediately run the program PROG.BAS from drive
B.

You may activate the AUTO feature on each warm boot, cold
boot or both. The position of this bit is value 2 (bit 1) for
activation on warm boots and value 1 (bit 0) for activation on
cold boots. This distribution system is shipped with these bits
set LOW and thus this feature is NOT active.

Option 4. SET SYSTEM FOR A SINGLE DRIVE.

This option should be selected by users of a single drive
CP/M system. Its use will cause the system to prompt you to change
disks each time a different drive is selected. In this way, a four
drive system can be simulated with only a single physical drive so
the features of CP/M which use multiple drives can be used.

The position of this bit is value 4 (bit 3). The distribution
system is shipped with this bit LOW and thus this feature is NOT
ACTIVE.

When using this option do not attempt to use COPY.COM which
requires two drives. Instead, FlLECOPY.COM and PIP.COM will work
normally to allow you to transfer files between disks.

CHANGING THE I/O CONFIGURATION.

Once your CP/M is up and running, you may find the need to make changes in your I/O, to add a different printer, or you may find a need to have more than one console. There are many possible variations on the basic console I/O that you began with. There are two methods of doing this. The SAVEUSER program is provided as a quick and easy way of making short patches. The SYSGEN PROCEDURE is a comprehensive method of making system changes.

PATCHING WITH SAVEUSER.

Running the SAVEUSER program at any time will write a portion of the then running CP/M system onto the disk in drive A. The specific portion written consists of the entire USER AREA (512 bytes or two pages starting at BIOS+500 hex) and the MODE AREA which is the 128 bytes immediately before the USER AREA. In the distribution 24K system the MODE AREA starts at 5B80 hex and the USER AREA starts at 5C00 hex and ends at 5DFF hex.

A normal change in the MODE AREA would be making a permanent change in the MODE BYTE. Any changes in the USER AREA would be for the purpose of making a quick patch to your CONSOLE, LIST, PUNCH or READER routines. These patches could be made with DDT and if they prove satisfactory, permanently saved on the system disk with SAVEUSER.

To run the program, simply type "SAVEUSER". It will sign on and ask for confirmation that you really wish to write onto the disk in drive A. "A<CR>" response will begin the write. You may also abort without writing anything by entering control C.

SYSGEN PROCEDURE.

The SYSGEN PROCEDURE is an advanced method of changing your system configuration. It should only be used when your system is running properly and you have become completely familiar with the features of CP/M. This procedure uses the ED.COM editor, the ASM.COM assembler, the DDT.COM debugger, and the SYSGEN.COM program. Familiarize yourself with the use of these programs before attempting to change your system configuration in this way.

This procedure uses the facilities of CP/M to create an assembly language file containing your customized I/O. Use as a starting point USER.ASM which contains the proper equates and jump table. Modify USER.ASM using ED to suit your own purposes. Be sure to change the equate for MSIZE in the file to the desired memory size of your CP/M. Next, assemble your modified file with ASM.COM to produce USER.HEX which will be used to overlay and modify your existing system.

The existing system is then brought into memory at location 900H (hex) using the SYSGEN program. This is the standard and most convenient memory location for making changes in the CP/M system. The copy of CP/M starting at 900H is usually called the SYSGEN IMAGE. Save the SYSGEN IMAGE as a disk file by the command "SAVE 40 CPM.COM<CR>".

CPM.COM is then brought back into memory under the control of DDT with the command "DDT CPM.COM<CR>" The SYSGEN IMAGE at 900H is then overlaid with USER.HEX using the "I" insert and "R" read commands of DDT.

Note that USER.HEX is read in with an offset so that the hex file will be placed at 2580H. The offset changes for each different CP/M system size and is equal to 2580 hex minus wherever your USER AREA begins in memory (5COO hex in standard 24K system). The proper offset is automatically calculated for each MSIZE by the OFFSET equate in USER.ASM or the DDT "H" hex calculator may be used as in the example.

At this point, the new system has been patched with your custom I/O and may be directly put on the system tracks with SYSGEN or saved again as a disk file for later use.

The dialog below is an example of using the above procedure and was actually run on a CP/M on an a MITS Altair 8800 system just as you see it. We begin after you have edited USER.ASM for your specific requirements.

```
A>ASM USER.AAZ<CR>
CP/M ASSEMBLER - VER 2.0
5D81
004H USE FACIOR
END OF ASSEMBLY

A>SYSGEN<CR>

SYSGEN Version 4.5
Distributed by Lifeboat Associates
for CP/M on Altair.

Source drive NAME (or RETURN to skip) A<CR>
Place SOURCE disk on A, then type RETURN <CR>
Function complete

CP/M image in RAM at 900H is ready to write
or reboot and "SAVE 40 CPMxx.COM"

Destination drive NAME (or RETURN to reboot) <CR>

A>SAVE 40 CPM24.COM<CR>
A>DDT CPM24.COM<CR>
DDT VERS 2.2
NEXT PC
```

```
2900 0100
-H2580,5COO<CR>
8180 C980
-IUSER.HEX<CR>
-RC980<CR>
NEXT PC
2900 0000
-^C
A>SYSGEN<CR>

SYSGEN Version 4.5
Distributed by Lifeboat Associates
for CP/M on Altair.

Source drive NAME (or RETURN to skip) <CR>

CP/M image in RAM at 900H is ready to write
or reboot and "SAVE 40 CPMxx.COM"

Destination drive NAME (or RETURN to reboot) B<CR>
Place DESTINATION disk on B, then type RETURN <CR>
Function complete

Destination drive NAME (or RETURN to reboot) <CR>

A>
```

DEFINITIONS OF CP/M PROGRAMS.

MOVCPM.COM - This program permits the user to reconfigure the system for any particular memory size. CP/M on Altair is supplied with a 24k system. In most cases a user will have a larger memory capacity. If the system is not reconfigured for the larger amount of memory, it will not recognize that it exists. REFERENCE:AN INTRODUCTION TO CP/M FEATURES AND FACILITIES MANUAL PAGE 30.

SPECIAL NOTE

WITH THE LIFEBOAT VERSIONS OF CP/M, THE PROGRAM "CONFIG.COM" MUST ALWAYS BE ON THE DISK WHEN USING MOVCPM.COM AND SHOULD BE PATCHED FOR YOUR TERMINAL OR COMPUTER.

ASM.COM -- This file contains the complete CP/M two-pass assembler. The ASM command loads and executes the CP/M 8080 assembler. It is used with the name of a source file containing assembly language statements and contains parameters directing the assembler where to get the input file and direct the output files. The following example will assemble TEST.ASM taking the source from drive A, writing the output hex file TEST.HEX to drive B and the output print file TEST.PRN to drive C.

A>ASM FILENAME.ABC<CR>

REFERENCE: AN INTRODUCTION TO CP/M FEATURES AND FACILITIES MANUAL PAGE 16 and CP/M ASSEMBLER (ASM) USER'S GUIDE.

DDT.COM -- This file contains the CP/M Dynamic Debugging Tool a very powerful monitor plus a tool for analyzing software problems. It can perform common monitor program functions such as dumping memory in HEX, or ASCII. It can also disassemble 8080 machine code into assembly language mnemonics, and assemble individual instructions. It is called by typing:

A>DDT<CR>

RESULT: DDT will sign-on with its version number and then the DDT prompt, a minus sign (-).

A>DDT FILENAME.COM<CR> - A filename can also be specified which will cause DDT to automatically load FILENAME.COM at 100 hex. In this case, DDT will display a hex number under "NEXT" is the next address after the highest byte of FILENAME. This may be used to calculate the program size or the number of pages needed to SAVE the program on disk.

REFERENCE: CP/M MANUAL- CP/M DYNAMIC DEBUGGING TOOL USER'S GUIDE.

DUMP.COM -- This program types the contents of a specified disk file in hexadecimal form with the HEX address listed at the left of each 16 bytes line.

REFERENCE: AN INTRODUCTION TO CP/M FEATURES AND FACILITIES MANUAL PAGE 30.

ED.COM -- This file contains the CP/M Editor. It permits creation and alteration of text files in ASCII under CP/M. In addition, programs in the compiled versions of high-level languages such as BASIC and FORTRAN can be entered through the ED context editor.

REFERENCE: AN INTRODUCTION TO CP/M FEATURES AND FACILITIES MANUAL PAGE 25. PAR 6.5 and CP/M MANUAL- ED: A CONTEXT EDITOR FOR THE CP/M DISK SYSTEM - USER'S MANUAL.

LOAD.COM -- This program reads a file that contains HEX machine code such as produced as output by the assembler and creates a file called FILENAME.COM i.e

A>LOAD TESTFILE<CR>

REFERENCE: AN INTRODUCTION TO CP/M FEATURES AND FACILITIES MANUAL PAR 6.3 PAGE 17.

PIP.COM --This is the Peripheral Interchange Program which is used to transfer files from disk to disk. It also-does such things as media conversion necessary to load, print, punch, copy and combine disk files. Study all the uses and forms of the PIP program very carefully. You will use it a lot.

REFERENCE: AN INTRODUCTION TO CP/M FEATURES AND FACILITIES MANUAL - PAR 6.4 PAGE 18.

STAT.COM -- Another one of the important CP/M programs. This one provides statistical information about the file storage and the device assignment.

REFERENCE: AN INTRODUCTION TO CP/M FEATURES AND FACILITIES MANUAL - PAR. 6.1 PAGE 13.

SUBMIT.COM -- This program allows CP/M commands to be "batched" together for automatic processing.

XSUB.COM--Used with SUBMIT.

REFERENCE: AN INTRODUCTION TO CP/M FEATURES AND FACILITIES MANUAL PAR. 6.7 PAGE 28.

SYSGEN.COM -- This program allows the CP/M operating system to be moved from disk to disk .

REFERENCE: AN INTRODUCTION TO CP/M FEATURES AND FACILITIES MANUAL - PAR. 6.6 PAGE 27.

DUMP.ASM - This is a sample .ASM file which shows how CP/M programs are written in assembly language.

                        SPECIAL NOTE
THE FOLLOWING PROGRAMS ARE SPECIALLY DESIGNED TO BE USED WITH THE ALTAIR CP/M SYSTEM. THEY ARE ALL TUTORIAL AND WILL PROMPT YOU AT EACH STEP.

LIST.COM -- Lists the directory and number of records for each program on your disk. Its forms are:

A)LIST<CR> or
A)LIST B:<CR>

FORMAT.COM -- This program will format a raw diskette. All sectors will be filled with 0E5 hex.

COPY.COM -- This program copies the entire disk using the- A (ALL) option to a new (may be unformatted) disk in another drive. The resulting format will be the same as the source disk. It may also be used to copy a disk until it detects an empty track using the M (MOST) option. The M option saves time but should only be used on a previously formatted target disk. COPY will prompt you at each step on exactly what to do. Do not use COPY if you have a single drive system. Use FILECOPY as explained next instead.

FILECOPY.COM --This program will transfer files between disks using only a single disk drive. It loads a single file at a time from the source disk into memory, instructs the user to change disks, and writes the file onto the new disk. To transfer a file named "TEST.HEX" simply type the command:

A>FILECOPY TEST.HEX<CR>

and the program will prompt you which disks to insert at the proper time.

SAVEUSER.COM --This program quickly writes BIOS "patches" to the disk by writing the USER AREA of the running system plus 128 bytes immediately prior (containing MODE byte) on to the disk. This is a total of 640 bytes (2 1/2 pages).

CONFIG.COM --This program sets up the USER AREA for the currently used I/O. It is possible to configure the system for many I/O configurations. This program must be on the disk properly patched for your terminal when the MOVCPM.COM program is run.

USER.ASM --This is a sample USER AREA source file which may be used as a basis for modifying the I/O of your system. See the listing in Appendix B.

MEMR.COM --This program is a memory test program that performs a battery of different tests. It's possible forms are "MEMR<CR>" for

normal testing, "MEMR T<CR>" for finding the top of memory, and "MEMR <CR>" to do a random number test only. The test will cycle continuously and report the location of any errors found.

NORMAL OPERATION.

USING MASTER COM DISKS.

In "normal" day-to-day operation, using CP/M with a two drive system, you should keep one disk with your CP/M system and all of your frequently used disks with the extension ".COM", in drive A. This is referred to as the Master COM file disk, or MASCOM. You should have several of these MASCOM disks, and you should replace one when it looks worn or "glassy". Keep the old disk as you never can tell when you want a program that you haven't used in a long time.

Disks containing the ASCII or BASIC files you are working on should be placed in drive B. Generally you will not need to use SYSGEN to put CP/M systems on most of your file disks since they will only be used in drive B. You may find it most convenient to log in on drive B after doing a cold boot by typing "B:<CR>". CP/M will automatically return to B after each warm boot.

APPENDIX A - SYSTEM LAYOUT

     The CP/M system you have received is designed to run in 24K of memory and is referred to as a 24K system. The CP/M system resides on the disk on tracks 0 and 1. The directory and files start on track 2. The USER AREA which will contain your I/O drivers is a two page (512 byte) area on disk. This is the area that contains your console and other device drivers.

     The following table will list the locations of various CP/M modules in both the distribution 24K system and the "SYSGEN POSITION". The "SYSGEN POSITION" is the location of the CP/M system in memory when SYSGEN or MOVCPM is run. This position is most useful for patching the system.

<div align="center">*** DISK PARAMETERS ***</div>

	SIZE	ON DISK TRACK	SECTOR	IN SYSGEN ADDRESS	24K SYSTEM ADDRESS
BOOT	0180H	0	1 AND 3	0900H	0000H
CCP	0800H	0	4	0A80H	4100H
BDOS	0E00H	0	20	1280H	4900H
BIOS	0500H	1	16	2080H	5700H
MODE		1	25	257FH	5BFFH
USER	0100H	1	26-27	2580H	5C00H
RSVD	0300H	1	28-32	2680H	5E00H

APPENDIX B - ALTAIR USER AREA LISTING


```
                    ; USER AREA for CP/M2 on Altair.
                    ; Version 5.0 of July 23, 1981.

                    ; Copyright (C) 1981 Lifeboat Associates

                    ; This USER AREA is identical to that produced by
                    ; CONFIG.COM Ver 5.x using configurations 0 thru 8
                    ; except for port values and initialization strings.

                    ; It may be used as is or as a prototype for
                    ; your own drivers.  Two pages (512 bytes) are
                    ; available for your custom I/O routines.

                    ; The USER AREA contains a standard CONSOLE driver
                    ; and PRINTER driver with a choice of handshaking type.
                    ; Handshaking may be ETX/ACK, XON/XOFF or NONE.
                    ; Printer may send 0-256 nulls after carriage return.
                    ; The PUNCH and READER routines go to the console.

                    ; The specific console ports and initialization strings
                    ; will depend on the terminal number selected.
                    ; Terminals 0 thru 7 are for specific I/O boards.

                    ; CONFIG terminal #8 will configure for non standard
                    ; console ports when the values for equates "1" thru "8"
                    ; are placed in the data table at 130H together with
                    ; initialization string if needed at "S" and length at "L".

                    ; Type "CONFIG P<cr>" and answer console questions to
                    ; install printer equates "9" thru "J" and init string "S".


                    ; Change MSIZE to the desired CP/M memory size in K.
  0018 =            MSIZE   EQU     24              ; Distribution size


                    ; These equates are automatically changed by MSIZE.
  5700 =            BIOS    EQU (MSIZE*1024)-900H   ; Memory location of BIOS
  5C00 =            USER    EQU     BIOS+500H       ; and of this USER AREA
  C980 =            OFFSET  EQU     2580H-USER      ; To overlay SYSGEN image

                    ; Misc standard equates.
  0003 =            IOBYT   EQU     3               ; Storage location
  000D =            CR      EQU     0DH             ; Carriage return.
  000A =            LF      EQU     0AH             ; Line feed.
  0008 =            BS      EQU     08H             ; Back space.

                    ; Hardware equates for Altair 88-2SIO using
                    ; active HIGH hardware as used in CONFIG.COM Terminal #0.

                    ; Change to the appropriate values for your I/O hardware.
```

```
                    ; See the instructions accompanying your I/O board for
                    ; the correct ports, flags and initialization code.

                    ; Set hardware sense flags xxxFLG as follows:
                    ;        Active HIGH bits to 1, active LOW to 0.
                    ;        Usually xxxFLG = xxxMSK if active HIGH.
                    ;        or     xxxFLG = 0      if active LOW.


                    ; Hardware equates for console input (TTY).
0010 =              TISPT   EQU    16               ; "1" TTY input status port
0001 =              TDAMSK  EQU    1                ; "2" Data available mask
0001 =              TDAFLG  EQU    1                ; "3" Hardware active HIGH
0011 =              TDIPT   EQU    17               ; "4" TTY data input port


                    ; Hardware equates for console output (TTY).
0010 =              TOSPT   EQU    16               ; "5" TTY output status port
0002 =              TBEMSK  EQU    2                ; "6" Tx buffer empty mask
0002 =              TBEFLG  EQU    2                ; "7" Hardware active HIGH
0011 =              TDOPT   EQU    17               ; "8" TTY data output port


                    ; Hardware equates for printer output (PTR).
0012 =              POSPT   EQU    18               ; "9" PTR output status port
0002 =              PBEMSK  EQU    2                ; "A" Tx buffer empty mask
0002 =              PBEFLG  EQU    2                ; "B" Hardware active HIGH
0013 =              PDOPT   EQU    19               ; "C" PTR data output port


                    ; Hardware equates for printer handshaking (PTR input).
                    ; Usually same ports as printer output with different MSK.
0012 =              PISPT   EQU    18               ; "D" PTR input status port
0001 =              PDAMSK  EQU    1                ; "E" Data available mask
0001 =              PDAFLG  EQU    1                ; "F" Hardware active HIGH
0013 =              PDIPT   EQU    19               ; "G" PTR input data port


                    ; Handshaking equates.
0003 =              ETX     EQU    'C'-40H          ;     Send ETX after a buffer
0006 =              ACK     EQU    'F'-40H          ;     and wait for printers ACK
007F =              BUFLEN  EQU    127              ;     Buffer length for ETX/ACK
0013 =              XOFF    EQU    'S'-40H          ;     Printer says stop
0011 =              XON     EQU    'Q'-40H          ;     Printer ready for data

                    ; HAND is type of handshaking which can be 0FFH for NONE,
                    ; 6 (ACK) for ETX/ACK or 11H (XON) for XON/XOFF.
00FF =              HAND    EQU    0FFH             ; "H" Type of handshaking

                    ; DEFIOB is initial IOBYT if used.
                    ; 80H sets printer to LPT: device, 0 sets to TTY:.
                    ; Use STAT.COM can modify IOBYT in running CP/M.
0000 =              DEFIOB  EQU    0                ; "I" Default IOBYT

                    ; NULLS is number of nulls sent after carriage return
                    ; to allow printer time to return to left margin.
0000 =              NULLS   EQU    0                ; "J" Printer nulls
```

```
5C00                            ORG     USER                ; Start of USER AREA

                        ; JUMP TABLE - Jumps MUST remain here in same order.
5C00 C3605D     CINIT   JMP     CINITR              ; Cold boot init
5C03 C35C5D     WINIT   JMP     WINITR              ; Warm boot init
5C06 C3235C     CONST   JMP     UCONST              ; Console status
5C09 C3315C     CONIN   JMP     UCONIN              ; Console input
5C0C C33F5C     CONOUT  JMP     UCONOUT             ; Console output
5C0F C34D5C     LIST    JMP     ULIST               ; Printer output
5C12 C33F5C     PUNCH   JMP     UCONOUT             ; Punch output to console
5C15 C3315C     READER  JMP     UCONIN              ; Reader input to console
5C18 C35D5C     LISTST  JMP     ULISTST             ; Printer status


                        ; This 8 byte data area used externally MUST remain.
5C1B 8101       LENUA:  DW      USRLEN              ; Length of USER AREA
5C1D 00         USRIOB: DB      DEFIOB              ; "I" Initial IOBYT
5C1E FF         HSTYPE: DB      HAND                ; "H" Handshaking type
5C1F 00         NULLOC: DB      NULLS               ; "J" Printer nulls
5C20 000000             DB      0,0,0               ; Reserved


                        ; These routines use IOBYT to select CP/M CONSOLE and LIST.

                UCONST:
                        ; Select CP/M CONSOLE status routine.
5C23 3A0300             LDA     IOBYT
5C26 CD6D5C             CALL    DEVSEL              ; Select device from table.
5C29 7A5C               DW      TTYIST              ; TTY:
5C2B BE5C               DW      PTRIST              ; CRT:
5C2D 7A5C               DW      TTYIST              ; BAT:
5C2F 7A5C               DW      TTYIST              ; UC1:


                UCONIN:
                        ; Select CP/M CONSOLE input routine.
5C31 3A0300             LDA     IOBYT
5C34 CD6D5C             CALL    DEVSEL              ; Select device from table
5C37 865C               DW      TTYIN               ; TTY: is normal console.
5C39 CA5C               DW      PTRIN               ; CRT: uses printer driver.
5C3B 865C               DW      TTYIN               ; BAT:
5C3D 865C               DW      TTYIN               ; UC1:


                UCONOUT:
                        ; Select CP/M CONSOLE output routine.
5C3F 3A0300             LDA     IOBYT
5C42 CD6D5C             CALL    DEVSEL              ; Select device from table
5C45 9E5C               DW      TTYOUT              ; TTY:
5C47 B45C               DW      PTROUT              ; CRT:
5C49 9E5C               DW      TTYOUT              ; BAT:
5C4B 9E5C               DW      TTYOUT              ; UC1:


                ULIST:
                        ; Select CP/M LIST output routine.
5C4D 3A0300             LDA     IOBYT
5C50 07                 RLC                         ; Rotate LIST selection
5C51 07                 RLC                         ; bits to 0,1
```

21

```
5C52 CD6D5C            CALL    DEVSEL      ; Select device from table.
5C55 9E5C              DW      TTYOUT      ; TTY: goes to console.
5C57 B45C              DW      PTROUT      ; CRT: uses printer driver.
5C59 215D              DW      LPTOUT      ; LPT: uses handshaking.
5C5B 9E5C              DW      TTYOUT      ; UL1:

            ULISTST:
                       ; Select CP/M LIST status routine.
5C5D 3A0300            LDA     IOBYT
5C60 07                RLC                 ; Rotate LIST selection
5C61 07                RLC                 ; bits to 0,1
5C62 CD6D5C            CALL    DEVSEL      ; Select device from table.
5C65 925C              DW      TTYOST      ; TTY:
5C67 A85C              DW      PTROST      ; CRT:
5C69 EE5C              DW      LPTST       ; LPT:
5C6B 925C              DW      TTYOST      ; UL1:

            DEVSEL:
                       ; Select routine from table of caller.
5C6D E603              ANI     3           ; Mask IOBYT and
5C6F 07                RLC                 ; mult times 2.
5C70 5F                MOV     E,A         ; Put index into
5C71 1600              MVI     D,0         ; DE register.
5C73 E1                POP     H           ; Get addr of table
5C74 19                DAD     D           ; and add index.
5C75 5E                MOV     E,M         ; Get addr of routine
5C76 23                INX     H           ; into
5C77 56                MOV     D,M         ; DE first,
5C78 EB                XCHG                ; then put into HL
5C79 E9                PCHL                ; and transfer control.

            ; Console Physical Drivers

            TTYIST:
                       ; Console input status routine.
                       ; Return 0FFH if char ready, 0 if not.
5C7A AF                XRA     A
5C7B DB10              IN      TISPT       ; "1" Read status port
5C7D 2F                CMA                 ; Adjust sense
5C7E E601              ANI     TDAMSK      ; "2" Mask status bits
5C80 EE01              XRI     TDAFLG      ; "3" Hardware sense
5C82 C8                RZ                  ; No key was pressed
5C83 3EFF              MVI     A,0FFH      ; Char is ready
5C85 C9                RET

            TTYIN:
                       ; Console input char to register A.
5C86 CD7A5C            CALL    TTYIST      ; Is char ready?
5C89 CA865C            JZ      TTYIN       ; Not yet
5C8C AF                XRA     A
5C8D DB11              IN      TDIPT       ; "4" Read data port
5C8F E67F              ANI     7FH         ; Strip parity
5C91 C9                RET
```

```
              TTYOST:
                          ; Console output status routine.
                          ; Ret 0FFH if ready for output, 0 if not.
5C92 AF                   XRA     A
5C93 DB10                 IN      TOSPT           ; "5" Read status port
5C95 2F                   CMA                     ; Adjust sense
5C96 E602                 ANI     TBEMSK          ; "6" Mask status bits
5C98 EE02                 XRI     TBEFLG          ; "7" Hardware sense
5C9A C8                   RZ                      ; Not ready
5C9B 3EFF                 MVI     A,0FFH          ; Ready for output
5C9D C9                   RET

              TTYOUT:
                          ; Console output char from register C.
5C9E CD925C               CALL    TTYOST          ; Ready to output?
5CA1 CA9E5C               JZ      TTYOUT          ; Wait until not busy
5CA4 79                   MOV     A,C             ; Char into accumulator
5CA5 D311                 OUT     TDOPT           ; "8" Output char
5CA7 C9                   RET

            ; Printer Physical Drivers

              PTROST:
                          ; Printer output status routine.
                          ; Return 0FFH if ready to print, 0 if busy.
5CA8 AF                   XRA     A
5CA9 DB12                 IN      POSPT           ; "9" Read status port
5CAB 2F                   CMA                     ; Adjust sense
5CAC E602                 ANI     PBEMSK          ; "A" Mask status bits
5CAE EE02                 XRI     PBEFLG          ; "B" Hardware sense
5CB0 C8                   RZ                      ; Printer busy
5CB1 3EFF                 MVI     A,0FFH          ; Ready to print
5CB3 C9                   RET

              PTROUT:
                          ; Printer output char from register C.
5CB4 CDA85C               CALL    PTROST          ; Ready to print?
5CB7 CAB45C               JZ      PTROUT          ; Wait until not busy
5CBA 79                   MOV     A,C             ; Char into accumulator
5CBB D313                 OUT     PDOPT           ; "C" Output char
5CBD C9                   RET

              PTRIST:
                          ; Printer input (ack) status.
                          ; Return 0FFH if ready, 0 if busy.
5CBE AF                   XRA     A
5CBF DB12                 IN      PISPT           ; "D" Read status port
5CC1 2F                   CMA                     ; Adjust sense
5CC2 E601                 ANI     PDAMSK          ; "E" Mask status bits
5CC4 EE01                 XRI     PDAFLG          ; "F" Hardware sense
5CC6 C8                   RZ                      ; No char avail
5CC7 3EFF                 MVI     A,0FFH          ; Ready to get ack
5CC9 C9                   RET
```

```
                PTRIN:
                        ; Printer input ack char to register A.
5CCA CDBE5C            CALL    PTRIST          ; Is char ready?
5CCD CACA5C            JZ      PTRIN           ; Not yet
5CD0 AF                XRA     A
5CD1 DB13              IN      PDIPT           ; "G" Read data port
5CD3 E67F              ANI     7FH             ; Strip parity
5CD5 C9                RET

                NULLOUT:
                        ; Null handler for printer output.
5CD6 CDB45C            CALL    PTROUT          ; Print the char.
5CD9 FE0D              CPI     CR              ; Was it a CR?
5CDB C0                RNZ                     ; Finished if not.
5CDC 3A1F5C            LDA     NULLOC          ; Get nr of nulls to send
5CDF 47                MOV     B,A             ; into B reg to count.
5CE0 B7                ORA     A               ; We are finished
5CE1 C8                RZ                      ; if NULLS = 0.
5CE2 0E00              MVI     C,0             ; This is a null.
                NLOOP:
5CE4 CDB45C            CALL    PTROUT          ; Print a null,
5CE7 05                DCR     B               ; decrement count
5CE8 C2E45C            JNZ     NLOOP           ; and loop until 0.
5CEB 0E0D              MVI     C,CR            ; Restore CR to C.
5CED C9                RET

                        ; LPT logical printer driver does handshaking
                        ; and calls PTR physical drivers. Reg C preserved.

                LPTST:
                        ; LPT logical status routine.
                        ; Return 0FFH if ready, 0 if busy.
5CEE CDA85C            CALL    PTROST          ; Is hardware busy?
5CF1 C8                RZ                      ; Yes
5CF2 3A1E5C            LDA     HSTYPE          ; Should be 0FFH, ACK or XON
5CF5 47                MOV     B,A             ; Save type.
5CF6 FE06              CPI     ACK             ; ETX/ACK?
5CF8 CA005D            JZ      PROTO           ; Yes, on to handler
5CFB FE11              CPI     XON             ; XON/XOFF?
5CFD 3EFF              MVI     A,0FFH          ; No handshaking in use
5CFF C0                RNZ                     ; and hardware is ready.
                PROTO:
                        ; Mark ready if ACK rvcd when ETX/ACK in use
                        ; or respond to XON/XOFF.
5D00 CDBE5C            CALL    PTRIST          ; Is hs char ready?
5D03 C4CA5C            CNZ     PTRIN           ; Yes, get it.
5D06 B8                CMP     B               ; Proper go ahead char?
5D07 CA1A5D            JZ      READY           ; Yes, must be ACK or XON.
5D0A FE13              CPI     XOFF            ; XOFF rcvd?
5D0C C2155D            JNZ     NLEGAL          ; No, ignore char.
5D0F 04                INR     B               ; Make XON
5D10 04                INR     B               ; into XOFF to make sure
5D11 B8                CMP     B               ; XON/OFF in use.
5D12 CA4A5D            JZ      BUSY            ; XOFF properly rcvd.
```

24

```
                    NLEGAL:
5D15 3A5B5D                 LDA     LPTFLG      ; Not legal hs char so
5D18 B7                     ORA     A           ; return with prev status.
5D19 C9                     RET
                    READY:
5D1A 3EFF                   MVI     A,0FFH      ; Mark ready
5D1C 325B5D                 STA     LPTFLG      ; at software flag
5D1F B7                     ORA     A           ; and return NZ.
5D20 C9                     RET

                    LPTOUT:
                            ; LPT output routine from register C.
5D21 CDEE5C                 CALL    LPTST       ; Get status
5D24 CA215D                 JZ      LPTOUT      ; Wait until ready
5D27 CDD65C                 CALL    NULLOUT     ; Then print char
5D2A 3A1E5C                 LDA     HSTYPE      ; Load protocol type
5D2D FE06                   CPI     ACK         ; Using ETX/ACK?
5D2F C0                     RNZ                 ; No, exit.
                            ; Process ETX/ACK protocol here.
                            ; Check for ESCAPE sequence first.
5D30 215A5D                 LXI     H,BUFCNT
5D33 79                     MOV     A,C         ; Was last char
5D34 FE1B                   CPI     1BH         ; an ESCAPE?
5D36 C2415D                 JNZ     ETXOUT      ; No
5D39 7E                     MOV     A,M         ; Get ETX count
5D3A FE04                   CPI     4           ; If over 3 left
5D3C D2415D                 JNC     ETXOUT      ; process normally.
5D3F 3603                   MVI     M,3         ; Send 3 char before ETX.
                    ETXOUT:
                            ; Count down until BUFLEN characters sent,
                            ; then send ETX and wait for printers ACK.
5D41 35                     DCR     M           ; Count down but
5D42 C0                     RNZ                 ; do nothing until 0
5D43 367F                   MVI     M,BUFLEN    ; Then reset count
5D45 0E03                   MVI     C,ETX       ; and send ETX
5D47 CDB45C                 CALL    PTROUT      ; to printer.
                    BUSY:
5D4A AF                     XRA     A           ; Mark busy
5D4B 325B5D                 STA     LPTFLG      ; at software flag
5D4E C9                     RET                 ; and ret Z set.

                    LPTINIT:
                            ; Initialize printer driver.
5D4F 3EFF                   MVI     A,0FFH      ; Mark printer ready
5D51 325B5D                 STA     LPTFLG      ; at software flag.
5D54 3E7F                   MVI     A,BUFLEN    ; Initialize
5D56 325A5D                 STA     BUFCNT      ; buffer count.
5D59 C9                     RET

                            ; Handshaking variables
5D5A 00             BUFCNT: DB      0           ; ETX/ACK buffer count.
5D5B 00             LPTFLG: DB      0           ; LPT status flag.
                                                ; Ready=0FFH, Busy=0
```

```
                    WINITR:
                                ; Any warm boot initialization goes here.
        5D5C 000000            DB      0,0,0            ; Patch room
        5D5F C9                RET


                    CINITR:
                                ; Hardware initialization on cold boot goes
                                ; here if needed.  Make sure it ends with a RET.
        5D60 3A1D5C            LDA     USRIOB           ; Load initial IOBYT
        5D63 320300            STA     IOBYT            ; and store.
        5D66 CD4F5D            CALL    LPTINIT          ; Init printer driver.
        5D69 000000            DB      0,0,0            ; Patch room


                                ; Initialization string for terminal #8
                                ; of length "L" usually begins at "S" below.
                                ; In this case, initialization is for 88-2SIO.
                    STRING:                             ; "S"
        5D6C 3E03              MVI     A,3              ; RESET 6850
        5D6E D310              OUT     16               ; PROGRAM FOR 8 BITS
        5D70 D312              OUT     18
        5D72 3E15              MVI     A,15H            ; 1STOP,NOPARITY, 16X CLOCK
                                                        ; NOTE: 2 STOP BITS=11H
        5D74 D310              OUT     16
        5D76 D312              OUT     18
        5D78 DB11              IN      17               ; CLEAR
        5D7A DB11              IN      17               ; INPUT
        5D7C DB13              IN      19               ; BUFFER
        5D7E DB13              IN      19
        5D80 C9                RET                      ; DONE


        0015 =      STRLEN     EQU     $-STRING         ; "L"


        0181 =      USRLEN     EQU     $-USER           ; Length of USER AREA
```

```
                    LIFEBOAT ASSOCIATES
                  SOFTWARE PROBLEM REPORT


Please use this form to report errors or problems in software
supplied by Lifeboat Associates. This form is designed to act as
a transmittal sheet, and problem details can be described on
additional pages.



        Date:_____

Software Product Name:_____ Version No. _____

        Disk Format:_____ Serial Number:_____

        Computer Type:_____ CPU Type:_____

Operating System:_____ Serial Number:_____ Version:_____

        Memory Size:_____ Number of Disk Drives:_____

Please describe the problem you have encountered. Include
references to the manual if appropriate. Try to reduce the problem
to a simple test case. Enclose any appropriate listings. If you
have discovered a patch or interim solution please describe it.

This form may also be used to suggest enhancements to our software
products.

Information on product changes and current version numbers are
published in Lifelines, our software newsletter.


PROBLEM DESCRIPTION:
```

Name:_____ Phone:_____

Address:_____

City:_____ State:_____ Zip:_____

```
Return to:Lifeboat Associates
          1651 Third Avenue
          New York, New York    10028
```