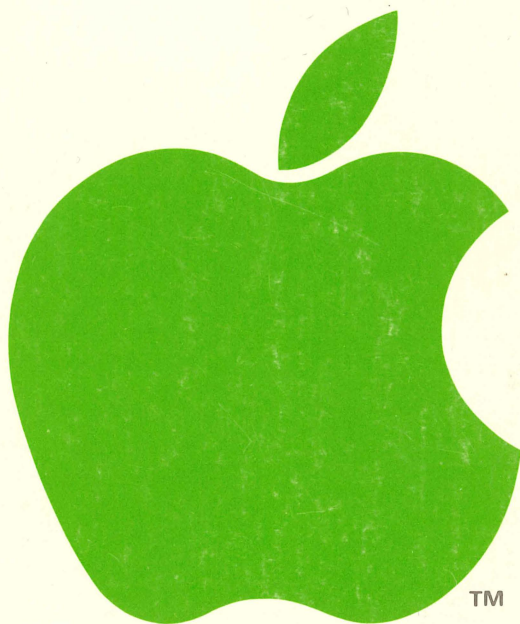


AUTOSTART ROM

INSTALLATION AND OPERATION MANUAL



Apple Intelligent Subsystems

AUTOSTART ROM

Installation and Operation Manual

NOTICE

Apple Computer Inc. reserves the right to make improvements in the product described in this manual at any time and without notice.

DISCLAIMER OF ALL WARRANTIES AND LIABILITY

APPLE COMPUTER INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL OR WITH RESPECT TO THE SOFTWARE DESCRIBED IN THIS MANUAL, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. APPLE COMPUTER INC. SOFTWARE IS SOLD OR LICENSED "AS IS". THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE IS WITH THE BUYER. SHOULD THE PROGRAMS PROVE DEFECTIVE FOLLOWING THEIR PURCHASE, THE BUYER (AND NOT APPLE COMPUTER INC., ITS DISTRIBUTOR, OR ITS RETAILER) ASSUMES THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION AND ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN NO EVENT WILL APPLE COMPUTER INC. BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE, EVEN IF APPLE COMPUTER INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

This manual is copyrighted and contains proprietary information. All rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Apple Computer Inc.

©1979 by APPLE COMPUTER INC.
10260 Bandley Drive
Cupertino, California 95014
(408) 996-1010

All rights reserved.

Reorder APPLE Product #A2L0022
(030-0038-00)

TABLE OF CONTENTS

1	Introduction to the Easy-Editing Autostart ROM
2	The AUTOSTART Features
4	The Easy-Editing Features
5	Summary
7	How to Install the Autostart ROM
13	How to Use the Autostart ROM
14	The AUTOSTART Features in Detail
15	The Easy-Editing Features in Detail
22	The Stop-List Feature
25	Summary of Features (reference chart)
27	Appendices
28	Appendix A: Subroutines Added to the Autostart ROM
30	Appendix B: Subroutines Deleted or Changed
31	Appendix C: Limitations of the Autostart ROM
32	Appendix D: Source Listing of the Autostart ROM

1	Introduction to the Easy-Editing Autostart ROM
2	The AUTOSTART Features
4	The Easy-Editing Features
5	Summary

Introduction to the Easy-Editing Autostart ROM

When you plug this small black device (called a ROM in the trade) into your Apple II, a lot of little things become even nicer. Editing becomes easier, using BASIC becomes easier, and using Disk II becomes easier. There are special features for non-disk systems as well.

The AUTOSTART Features

"It comes up running..."

First of all, this ROM helps get the Apple started. Once the ROM is installed, you only have to turn on the computer, and immediately it is in BASIC. If you have an Applesoft ROM card with the switch in the upward position, the Apple will come up in Applesoft. Otherwise, it will come up in Integer BASIC.

If you have a Disk II, then the AUTOSTART ROM does even more for you. With a disk in Drive 1, the Disk Operating System (DOS, pronounced "doss") will be booted automatically, and the Apple will execute the greeting program (commonly named "HELLO") on the disk, loading whichever BASIC the greeting program is written in. The fact that the Apple automatically executes the greeting program when it is turned on allows the Apple to act as a "turnkey" system. Say that the Apple has been programmed to keep an appointment schedule for a large office. When the receptionist starts the day's work, he or she turns on the switch to the front desk--the lights go on, the electric typewriter starts humming, and the computer automatically, without the receptionist's touching even so much as a single key on the Apple, prints the daily schedule on the screen.

A "turnkey" computer system is one that starts executing a particular program the instant it is turned on. It needs no special effort or knowledge on the part of its user to do this particular job. (Of course, the computer can also run other programs, and these may require special knowledge to use.) The word "turnkey" is used because so many systems have a lock that must be turned with a key to make them start.

The DOS 3.2 Manual has a section called "Creating a Turnkey System". That method is now even easier to use because of the AUTOSTART ROM.

(NOTE: The AUTOSTART ROM is designed to use the latest versions of DOS--Version 3.2 and higher-numbered versions--and will not work properly with earlier versions. Disks containing earlier versions of DOS should be updated via the UPDATE 3.2 program. The DOS 3.2 Manual explains how to do this.)

The Easy-Editing Features

It is very easy to move the cursor around on the screen. Look at the I, J, K, and M keys on the keyboard. They look like this:



Imagine four arrows drawn on them (or make little labels and actually stick them on).



To move the cursor in the direction of the arrows, press the key on the left of the keyboard marked "ESC" (which stands for "ESCAPE") only once. As soon as you do that, the four keys (I, J, K, and M) will, when pressed, move the cursor in the indicated direction. You can use them to move the cursor as many times as you wish. For faster cursor motion, you can hold one of these keys and then hold down the key marked "REPT" (for "REPEAT"). The cursor will glide up, down, right, or left.

To make the keyboard work normally again, press the space bar once.

Summary

The AUTOSTART ROM makes editing easier by allowing the cursor to be quickly moved around on the screen. It makes the Apple II easier to use, especially for beginners, since no special knowledge is required to get it started in BASIC. The rest of this manual explains how to install the ROM and gives more specific details about using it.

How to Install the Autostart ROM

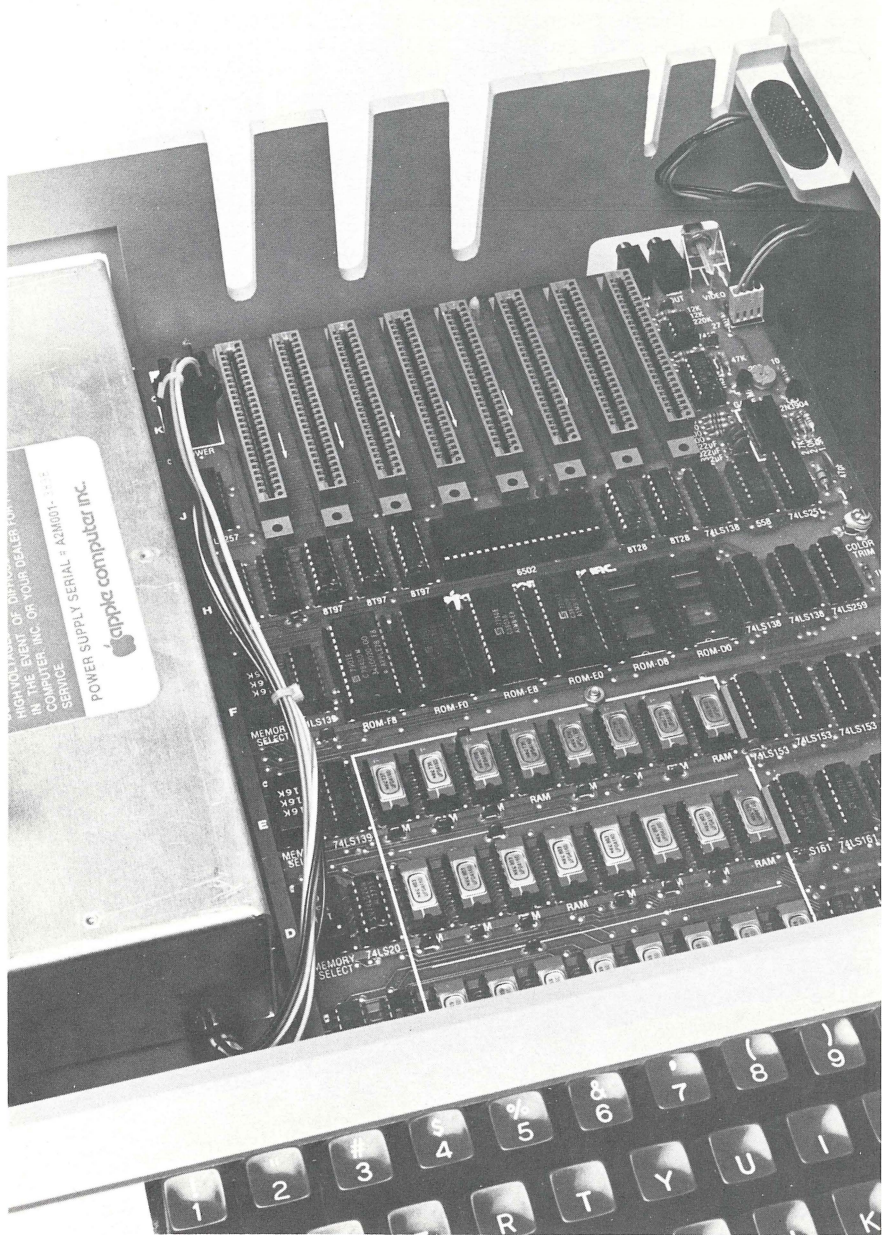
The Autostart ROM is an IC (integrated circuit, also called a "chip") that replaces a similar IC in a socket on the main circuit board of the Apple II computer. You can install the new ROM yourself if you like. Read these instructions first. If you feel uneasy working on your Apple yourself, ask your dealer to install the ROM for you. It will only take a few minutes.

1. Before doing anything else, turn off the power switch on the back of the Apple, and make sure the POWER light is off. This will prevent damage to the computer.

2. Remove the Apple's cover by pulling up on its rear edge until the two corner fasteners pop apart. Without lifting it any further, slide the cover backward (away from the keyboard).

3. Look into the open Apple, with the keyboard facing you. On the main printed-circuit board you will see many rows of small ICs. Two of these rows contain larger ICs. The nearer of these two rows (labelled "F" at its left end) has six large sockets, and the ROM you will replace is in the leftmost of these. It is labelled "ROM-F8" on the board, below the socket.

4. As you look at the old ROM, notice the small semicircular notch on the end facing the keyboard. (On some ROMs, the notch will have a different shape, but in all cases there will be a distinct mark on the end that should face the keyboard.) The ROMs should all be facing the same way.



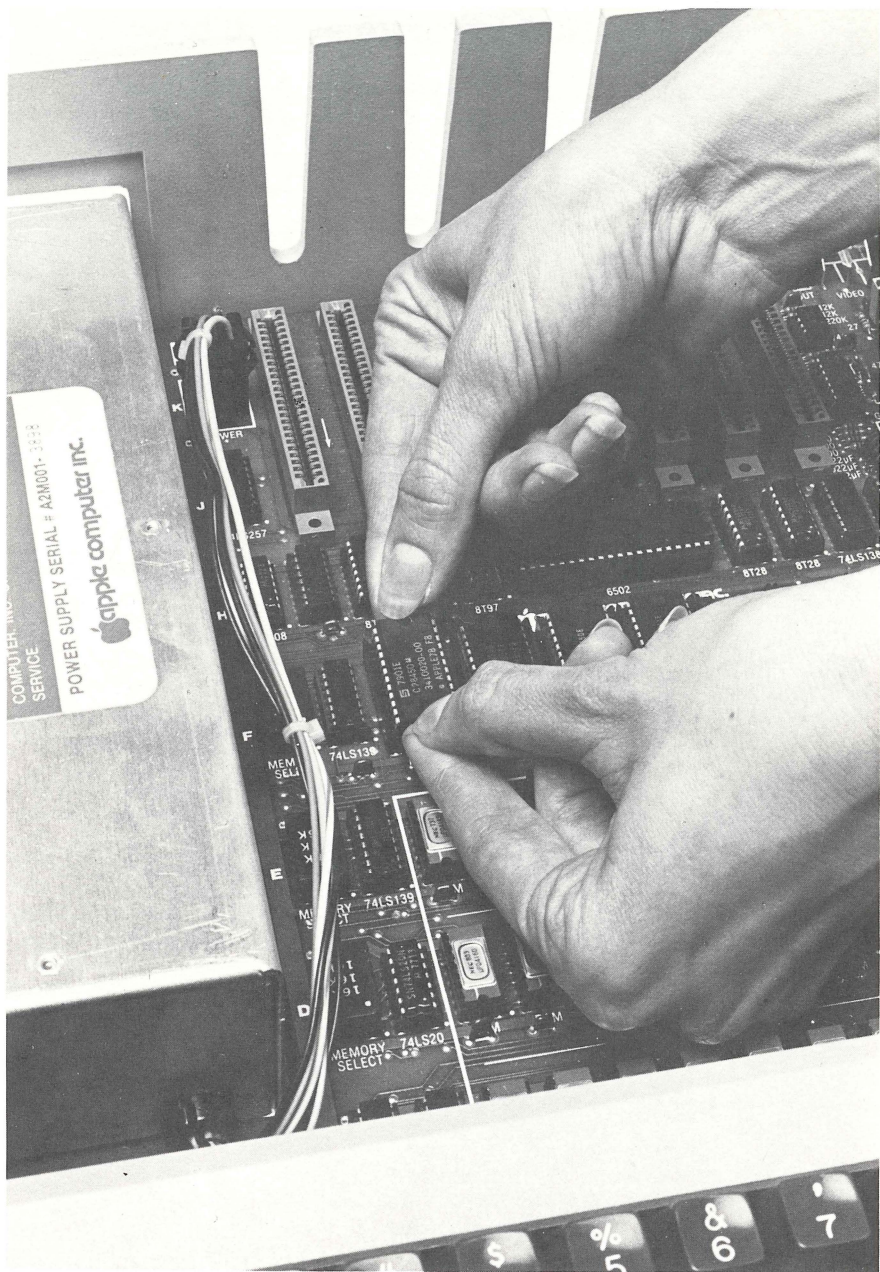
5. Before touching anything inside the Apple, discharge any static charge you may be carrying, by touching the power-supply case with your hand. (Don't worry--you won't get a shock if your Apple is properly grounded. All dangerous voltages are isolated inside the power supply, which is carefully insulated and grounded. Besides, you have already turned the Apple off, haven't you?)

6. Carefully remove the old ROM IC from its socket, using an IC puller if possible. If you can't get such a tool, there is another way to remove it, with a little care. Stand facing one side of the Apple. Place the tips of your index fingers under each end of the IC. The first knuckle of each index finger should be resting on the ICs in an adjacent row. Place your thumbs on top of the IC. By pushing down on adjacent rows, lever both fingertips up gently, turning the wrists. Control your motion with your thumbs. By pressing down with your thumb as you pry up with your finger, you can keep one end, or one side, from coming up too far. This will keep the IC fairly level as it wiggles out, so that no pins get bent. Do not try to simply yank the IC from the board--you will probably damage something. Make sure you are not trying to pull the socket from the board.

7. Line up all the pins of the new Autostart ROM IC with the holes in socket ROM-F8, making sure that there are no pins or holes left over on either end, and gently press the IC into place. If a pin bends, remove the IC from its socket. Straighten any bent pins with needlenose pliers, and press the IC into its socket again, even more carefully. Check that the IC is firmly seated and facing the same way (notch toward keyboard) as the other ROMs. If the IC is inserted backwards, it will be destroyed when you turn the Apple on.

8. Put the cover back on the Apple by first sliding the front edge of the cover into position, then pressing down on the two rear corners until they pop into place.

9. Your Autostart ROM is installed; you may now turn on your improved Apple. (If you have one or more Disk IIs, make sure Drive 1 has an initialized disk inserted and the door closed.)



13 **How to Use the Autostart ROM**

- 14 The AUTOSTART Features in Detail
- 15 The Easy-Editing Features in Detail
- 22 The Stop-List Feature

How to Use the Autostart ROM

The AUTOSTART Features in Detail

When you turn on the Apple with the new ROM in place, it will display "APPLE][]" and the prompt for whatever language you have installed in the machine: either Integer BASIC (>) or Applesoft (]). On some older Apples, you will get a display of vertical bars and then bars of characters, and a star prompt (*) will appear on the screen, followed by the blinking cursor. On these Apples, you must press RESET to continue the start-up procedure. After you do this, they will behave like the newer Apples. If you have an Applesoft ROM card, the Apple will look at the position of the switch on the back of the card.

If the switch is up, the Apple will choose Applesoft. If it is down, the Apple will choose Integer BASIC. If you have no Applesoft card, The Apple will come up in Integer BASIC.

If you have one or more disk drives, the Autostart ROM does even more for you. When you turn the Apple on, it will look for a disk controller card in the slots at the back of the main circuit board, starting with Slot 7 and moving toward Slot 1. (The first controller is normally placed in Slot 6, as many programs expect it there.) If it finds a controller, it stops searching. If the controller has no Drive 1, the Apple just sits there until you press RESET. If Drive 1 exists but has no disk in it, or has an open door, the Apple will spin the drive until the end of time, or until you press RESET, whichever comes first. If the Apple tries to boot a disk that is not initialized, it will keep spinning the disk, and will try to boot it each time you press RESET. You can stop it by pressing RESET several times in rapid succession, or by turning the Apple off. If the Apple finds no controller, or if it tries unsuccessfully to boot and you press RESET, the Apple will go into BASIC, as explained above, but will not have DOS and will act as if it had no disk drive. If the Apple successfully boots a disk, it will then try to run the greeting program on the disk booted, entering the BASIC that program is written in. If the Apple does not have the right BASIC, either in memory or on disk, it will display "LANGUAGE NOT AVAILABLE" and a prompt for the BASIC it does have, without executing the greeting program.

If you turn the Apple off and then on, it will start up as described above. Any program in memory will be lost whenever the Apple is turned off. (Turning the Apple off and on very quickly will have odd results, as the contents of memory will be only partially cleared.) If you press RESET, the Apple will stop whatever program it may be running, and display the prompt for whatever BASIC you were in, without losing your program or DOS. RESET has a function similar to CTRL-C. (CTRL-C is typed by typing C while holding down CTRL.) Both stop the program running and allow you to run it again. In addition, RESET restores the computer to the state it was in before the program was first run. CTRL-C does not restore all numbers in memory to their starting values, so when you reRUN the program it may not do the same things it did when it was first run.

(WARNING: If you have a Disk II, do not leave the Apple unattended without an initialized disk in Drive 1 of the highest numbered disk controller. If someone should inadvertently turn the Apple off and then on, or if a power failure should occur, the Apple will attempt to recover by booting the disk in this drive. If this drive has no disk or has the door open, both the drive and the Apple's power supply will get warm, but will probably suffer no damage. If the door is closed and a defective or uninitialized disk is in the drive, both the disk and the drive may be damaged.)

The Easy-Editing Features in Detail

The new Autostart ROM makes it easy to move the cursor. First you put the Apple in edit mode by pressing and releasing the ESC key, then use I to move the cursor up, M to move it down, J to move it left, and K to move it right. To move the cursor repeatedly, hold down one of the cursor direction keys (I, J, K, or M) and then hold down the REPT key at the same time. The cursor will zip along while both keys are held down. If the cursor reaches the top of the screen, it will stop. If the cursor reaches the bottom of the screen, it will stop, and the lines will start to move upward. If it reaches the right edge, the cursor will disappear and reappear at the left edge, but on the next line. If it reaches the left edge, it will reappear on the right, one line up. To leave edit mode, press the space bar. The Apple will now be in what we call "normal mode", the mode it was in before you pressed ESC. (Actually, you can press any key except A, B, C, D, I, J, K, M, E, F, @, CTRL, SHIFT or RESET, but the space bar is most convenient.)

It is also possible to move the cursor one space and then leave edit mode, by pressing ESC and then A, B, C, or D. This is to insure feature compatibility with the previous ROM.

How to clear one or more characters from a line on the screen.

In edit mode, move the cursor onto the first character. The character is now blinking. Then press the space bar to get the Apple out of edit mode, and press the space bar once again for each character you wish to clear. Each of these characters will be replaced by a blank.

How to clear to the end of a line on the screen. In edit mode, move the cursor onto the first character you wish to get rid of, then press E. The blinking character and all the characters after it on the line will disappear, and the Apple will return to normal mode.

How to clear to the end of the screen. In edit mode, move the cursor onto the first character you wish to clear, then press F. The blinking character and all the characters after it on the screen will disappear, and the Apple will return to normal mode.

How to clear the whole screen. If you are in edit mode, press @. All the characters on the screen will disappear, the cursor will return to the top left corner ("home"), and the Apple will return to normal mode. To get a BASIC prompt (> or]), press RETURN. If you are not in edit mode and wish to clear the screen, press ESC, then @.

When you press I, J, K, or M, the Apple remains in edit mode (because you may want to continue moving the cursor); but if you press A, B, C, D, E, F, or @, the Apple leaves the edit mode immediately after carrying out that command. If you press the space bar, the Apple will also return to normal mode.

All these features affect only what is displayed on the screen, not what is stored in memory. Two other keys allow you to change what is in memory, and are very useful in editing programs. These are the left- and right-pointing arrows, which can also be called the Delete and Retype keys.



These keys affect only the line currently being typed. When you press the Delete (left-pointing arrow) key, the cursor moves back (left) one space, erasing the character last entered. If the last character you typed appeared on the screen, it will be deleted. If you last typed a non-printing control character, the Apple will behave as if it had never been typed, but will not change the display. If the last character typed was a pure cursor move (I, J, K, or M in edit mode), it would not be the last character entered, and would never have been part of the current line.

When you press the Retype (right-pointing arrow) key, the cursor moves forward (right), copying the character it just passed over. If you copy a line with the Retype key, then press the RETURN key, the Apple behaves exactly as if you had retyped the line by hand.

You can "fast-retype" by holding down the Retype key, then the REPT key. The Apple will retype characters until one of the two keys is released. Any time you use the REPT key, remember that it repeats the last key typed, not the next key typed. For example, if you press the space bar to leave edit mode, press and hold the REPT key, and then press the Retype key, the Apple will delete the blinking character (as if you had pressed the space bar again). It will not retype the blinking character as you may have intended. This can be disconcerting if you do not understand what is happening.

You can use these features to replace, insert, or delete lines or parts of lines in your program. For example, let's look at a simple program in Applesoft:

```
10 PRINT "JABBERWOCK"
```

To replace 'JABBER' with 'JIBBER', you can do as follows:

1. LIST the program on the screen.

```
JLIST
10 PRINT "JABBERWOCK"
J■
```

2. Press ESC, then move the cursor onto the very first character (the digit '1') of line 10, using the J and I keys.

```
10 PRINT "JABBERWOCK"
```

3. Press the space bar to leave edit mode, then fast-retype the line up through the 'J' of 'JABBERWOCK', using the Retype and REPEAT keys. The blinking cursor should cover the 'A'.

```
10 PRINT "JABBERWOCK"
```

- 3a. If the '1' disappeared when you tried to fast-retype, you pressed the REPT key before the Retype key. To restore it, back up the cursor to the beginning of the line with the Delete key, type in a '1', then fast-retype through "J".

4. Type 'I', then fast-retype the rest of the line.

```
10 PRINT "JIBBERWOCK"■
```

5. Press RETURN, to tell the machine to replace the line in memory.

```
10 PRINT "JIBBERWOCK"
J■
```

6. LIST the program to see what you have done:

```
JLIST
10 PRINT "JIBBERWOCK"
J■
```

To insert 'JABBER' after 'JIBBER', do this:

1. List the program:

```
JLIST
10 PRINT "JIBBERWOCK"
J■
```

2. Fast-retype it through 'JIBBER'.

```
10 PRINT "JIBBERWOCK"
```

3. Press ESC, then move the cursor back to 'J', using the edit-mode keys.

```
10 PRINT "JIBBERWOCK"
```

4. Leave edit mode by pressing the space bar, then type 'JABBER' where 'JIBBER' was, using fast-retype if desired. This will delete 'JIBBER' from the screen, but not from the current line.

```
10 PRINT "JABBERWOCK"
```

5. Fast-retype the rest of the line and press RETURN.

```
10 PRINT "JABBERWOCK"
J■
```

6. LIST the program to see what you have done:

```
JLIST
10 PRINT "JIBBERJABBERWOCK"
J■
```


To delete 'JABBER' from the current line, do this:

1. LIST the program:

```
JLIST
10 PRINT "JIBBERJABBERWOCK"
J
```

2. Fast-retype it through 'JIBBER'.

```
10 PRINT "JIBBERJABBERWOCK"
```

3. Press ESC, then pass the cursor over 'JABBER'. This will delete 'JABBER' from the current line, but not from the screen.

```
10 PRINT "JIBBERJABBERWOCK"
```

4. Press the space bar to leave edit mode, then fast-retype the rest of the line and press RETURN.

```
10 PRINT "JIBBERJABBERWOCK"
J
```

5. LIST the program to see what you have done:

```
JLIST
10 PRINT "JIBBERWOCK"
J
```

Experiment with these methods, watching carefully what happens on the screen, then listing the line to see what happened to it in memory. After some practice you will become more familiar the relationship between these two processes.

Here is a trick that may be helpful in editing programs. If you have tried using the fast-retype feature on statements that took up more than one line on the screen, you may have noticed that unwanted blanks crept in. To avoid picking them up, do this:

1. Clear the screen by pressing ESC, then @.

2. Type

```
JPOKE 33, 33
```

3. Press RETURN.

4. LIST the program.

This makes the text window on the screen 30 characters wide and ensures that a statement longer than 30 characters will "wrap around" to the very beginning of the next screen line, without indentation. Consequently, if you fast-retype the statement, the cursor will go the end of one screen line and immediately jump to the beginning of the next, without picking up any extra blanks.

When you have done all the fast-retyping you need to do, type

```
JTEXT
```

and then press RETURN to restore the text window to normal. (Do not POKE a number larger than 40 into location 33 or you will clobber BASIC and your program. The Applesoft Reference Manual explains this in detail.)

The Stop-List Feature

The new Autostart ROM has another useful feature: it lets you stop and start a program listing at will. You do this by typing CTRL-S when you wish to interrupt a listing, and typing CTRL-S again to continue. (CTRL-S is typed by pressing S while holding down CTRL.) Let's try it.

1. LOAD a program of twenty lines or more from disk or cassette, or type one in at the keyboard.
2. Type 'LIST', but don't press RETURN yet.
3. Press the CTRL key and hold it down. Place one finger on the S key, ready to press it.
4. Press RETURN. The program will begin listing on the screen.
5. Press the S key, still holding CTRL down. The listing will stop.
6. Type CTRL-S to resume listing.
7. Press RESET to quit listing entirely.

This procedure takes a little practice to get down, as the Apple lists a program quickly, making it easy to go past the line you want. This is why it's a good idea to hold the CTRL key down while you are waiting for the right line.

This feature will also suspend execution of any program that outputs text to the screen. Let's try it on a simple program:

```
10 PRINT "JABBERWOCK"  
20 GOTO 10
```

1. RUN the program.
2. Type CTRL-S. When the program sends a carriage return to the screen, to start a new line, it looks at the keyboard, and if it sees CTRL-S, it suspends execution of the program and waits for further instructions.
3. Type CTRL-S again. The program will resume execution.
4. To stop the program and get back to BASIC, type CTRL-C.

For some programs, this will not work: for example,

```
10 PRINT "JABBERWOCK";  
20 GOTO 10
```

If you try to suspend this program by typing CTRL-S, The Apple will not respond. This is because this program sends a continuous stream of characters to the screen, rather than separate lines. It does not send any carriage returns to the screen, so the Apple never looks at the keyboard.

Summary of Features (reference chart)

Summary of Features (reference chart)

Start, enter BASIC, boot DOS	Turn on power
Restart, keeping program	Press RESET
Enter edit mode	Press ESC
Enter normal mode	Press space bar
Move cursor	Press I, J, K, or M
Delete a character	Press left arrow
Retype a character	Press right arrow
Clear line from cursor on	Press ESC, then E
Clear screen from cursor on	Press ESC, then F
Clear entire screen	Press ESC, then @
Stop listing	Type CTRL-S
Resume listing	Type CTRL-S

Appendices

- 28 Appendix A: Subroutines Added to the Autostart ROM
- 30 Appendix B: Subroutines Deleted or Changed
- 31 Appendix C: Limitations of the Autostart ROM
- 32 Appendix D: Source Listing of the Autostart ROM

APPENDIX A: Subroutines Added to the Autostart ROM

NAME: APPLE II (\$FB60) CALL -1184

PURPOSE: This routine clears the screen and POKes the string 'APPLE][' into line 1 of the text buffer, so that the screen will display 'APPLE][' on RESET.

ENTRY: Before CALLing this routine, make sure the scrolling window parameters have been set for maximum screen limits, by using the TEXT command or

```
10 POKE 32, 0
20 POKE 33, 40
30 POKE 34, 0
40 POKE 35, 24
```

EXIT: A=\$C1; Y=0; X=entry

STATUS: N=0, Z=1, C=X; V=X

NAME: SETPWRC (\$FB6F)

PURPOSE: This routine lets the user make the Apple go into monitor mode when RESET is pressed. It sets location \$3F4 to the exclusive OR of the contents of location \$3F3 with \$A5.

ENTRY: No conditions; location \$3F3 (POKE 1011) should be set with the high-order address of the RESET vector for this call to be meaningful.

EXIT: A=EOR#\$A5, \$3F3; Y=entry; X=entry
STATUS: N=X, Z=X, V=entry, C=entry

COMMENTS: To get the Apple to go into monitor mode upon RESET, do a CALL -155. This will get you into monitor mode. After the monitor prompt (*), type

3F2:69 FF 5A

and press RETURN.

After you have done this, the Apple will go into monitor mode whenever you press RESET.

Monitor mode is described in the Apple II Reference Manual.

APPENDIX B: Subroutines Deleted or Changed

STEP \$FA40-\$FA85
 \$FAA5-\$FAD6
 \$FAFD-\$FB18

MULPM, DIVPM \$FB60-\$FBC0

The following routines have been moved:

IRQ/BREAK was at \$FA86, is now at \$FA40

PAGE 3 MEMORY USAGE:

LOCATION	USAGE	DEFAULT VALUE
\$3F0, \$3F1	BREAK VECTOR	\$59, \$FA
\$3F2	RESET VECTOR LOW	\$03*
\$3F3	RESET VECTOR HIGH	\$E0*
\$3F4	POWERED-UP MASK	\$45*

*For non-disk systems after power-up reset.

APPENDIX C: Limitations of the Autostart ROM

1. Any programs using the 16-bit integer multiply/divide routines of the old monitor will fail, as these are not in the Autostart ROM.
2. Any programs using any part of the single-step simulator code will fail for the same reason.

APPENDIX D: Source Listing of the Autostart ROM

```

0000:          1          PAGE
0000:          2 *****
0000:          3 *
0000:          4 * APPLE II
0000:          5 * MONITOR II
0000:          6 *
0000:          7 * COPYRIGHT 1978 BY
0000:          8 * APPLE COMPUTER, INC.
0000:          9 *
0000:         10 * ALL RIGHTS RESERVED
0000:         11 *
0000:         12 * STEVE WOZNIAK
0000:         13 *
0000:         14 *****
0000:         15 *
0000:         16 * MODIFIED NOV 1978
0000:         17 * BY JOHN A
0000:         18 *
0000:         19 *****
FB00:         20          ORG $FB00
FB00:         21          OBJ $2000
FB00:         22 *****
FB00:         23 LOCO      EQU $00
FB00:         24 LOC1      EQU $01
FB00:         25 WNDLFT    EQU $20
FB00:         26 WNDWDTH   EQU $21
FB00:         27 WNDTDP    EQU $22
FB00:         28 WNDBTM   EQU $23
FB00:         29 CH       EQU $24
FB00:         30 CV       EQU $25
FB00:         31 GBASL   EQU $26
FB00:         32 GBASH   EQU $27
FB00:         33 BASL    EQU $28
FB00:         34 BASH    EQU $29
FB00:         35 BAS2L   EQU $2A
FB00:         36 BAS2H   EQU $2B
FB00:         37 H2     EQU $2C
FB00:         38 LMNEM   EQU $2C
FB00:         39 V2     EQU $2D
FB00:         40 RMNEM   EQU $2D
FB00:         41 MASK    EQU $2E
FB00:         42 CHKSUM   EQU $2E
FB00:         43 FORMAT   EQU $2E
FB00:         44 LASTIN   EQU $2F
FB00:         45 LENGTH   EQU $2F
FB00:         46 SIGN     EQU $2F
FB00:         47 COLOR   EQU $30
FB00:         48 MODE     EQU $31
FB00:         49 INVFLG   EQU $32
FB00:         50 PROMPT   EQU $33
FB00:         51 YSAV     EQU $34
FB00:         52 YSAV1    EQU $35
FB00:         53 CSWL    EQU $36
FB00:         54 CSWH     EQU $37
FB00:         55 KSWL    EQU $38
FB00:         56 KSWH     EQU $39
FB00:         57 PCL     EQU $3A
FB00:         58 PCH     EQU $3B
FB00:         59 A1L     EQU $3C
FB00:         60 A1H     EQU $3D
FB00:         61 A2L     EQU $3E
FB00:         62 A2H     EQU $3F
FB00:         63 A3L     EQU $40

```

```

F800:      64 A3H      EQU  $41
F800:      65 A4L      EQU  $42
F800:      66 A4H      EQU  $43
F800:      67 A5L      EQU  $44
F800:      68 A5H      EQU  $45
F800:      69 ACC      EQU  $45      ; NOTE OVERLAP WITH A5H!
F800:      70 XREG     EQU  $46
F800:      71 YREG     EQU  $47
F800:      72 STATUS   EQU  $48
F800:      73 SPNT     EQU  $49
F800:      74 RNDL     EQU  $4E
F800:      75 RNDH     EQU  $4F
F800:      76 PICK     EQU  $95
F800:      77 IN       EQU  $0200
F800:      78 BRKV     EQU  $3F0      ; NEW VECTOR FOR BRK
F800:      79 SOFTEV   EQU  $3F2      ; VECTOR FOR WARM START
F800:      80 PWREDUP  EQU  $3F4      ; THIS MUST = EOR ##A5 OF SOFTEV+1
F800:      81 AMPERV   EQU  $3F5      ; APPLESOFT & EXIT VECTOR
F800:      82 USRADR   EQU  $03F8
F800:      83 NMI      EQU  $03FB
F800:      84 IRQLOC   EQU  $3FE
F800:      85 LINE1    EQU  $400
F800:      86 MSL0T    EQU  $07FB
F800:      87 IOADR    EQU  $C000
F800:      88 KBD      EQU  $C000
F800:      89 KBDSTRB  EQU  $C010
F800:      90 TAPEOUT  EQU  $C020
F800:      91 SPKR     EQU  $C030
F800:      92 TXTCLR   EQU  $C050
F800:      93 TXTSET   EQU  $C051
F800:      94 MIXCLR   EQU  $C052
F800:      95 MIXSET   EQU  $C053
F800:      96 LOWSCR   EQU  $C054
F800:      97 HISCR   EQU  $C055
F800:      98 LORES   EQU  $C056
F800:      99 HIRES   EQU  $C057
F800:     100 SETANO   EQU  $C058
F800:     101 CLRANO   EQU  $C059
F800:     102 SETAN1   EQU  $C05A
F800:     103 CLRAN1   EQU  $C05B
F800:     104 SETAN2   EQU  $C05C
F800:     105 CLRAN2   EQU  $C05D
F800:     106 SETAN3   EQU  $C05E
F800:     107 CLRAN3   EQU  $C05F
F800:     108 TAPEIN   EQU  $C060
F800:     109 PADDLO   EQU  $C064
F800:     110 PTRIG    EQU  $C070
F800:     111 CLRROM   EQU  $CFFF
F800:     112 BASIC    EQU  $E000
F800:     113 BASIC2   EQU  $E003

```

FB00:		114	PAGE
FB00:	4A	115	PLOT LSR A
FB01:	08	116	PHP
FB02:	20 47 FB	117	JSR GBASCALC
FB05:	28	118	PLP
FB06:	A9 0F	119	LDA ##0F
FB08:	90 02	120	BCC RTMASK
FB0A:	69 E0	121	ADC ##E0
FB0C:	85 2E	122	RTMASK STA MASK
FB0E:	B1 26	123	PLOT1 LDA (GBASL), Y
FB10:	45 30	124	EDR COLOR
FB12:	25 2E	125	AND MASK
FB14:	51 26	126	EDR (GBASL), Y
FB16:	91 26	127	STA (GBASL), Y
FB18:	60	128	RTS
FB19:	20 00 FB	129	HLINE JSR PLOT
FB1C:	C4 2C	130	HLINE1 CPY H2
FB1E:	B0 11	131	BCS RTS1
FB20:	C8	132	INY
FB21:	20 0E FB	133	JSR PLOT1
FB24:	90 F6	134	BCC HLINE1
FB26:	69 01	135	VLINEZ ADC ##01
FB28:	48	136	VLINE PHA
FB29:	20 00 FB	137	JSR PLOT
FB2C:	68	138	PLA
FB2D:	C5 2D	139	CMP V2
FB2F:	90 F5	140	BCC VLINEZ
FB31:	60	141	RTS1 RTS
FB32:	A0 2F	142	CLRSCR LDY ##2F
FB34:	D0 02	143	BNE CLRSC2
FB36:	A0 27	144	CLRTOP LDY ##27
FB38:	84 2D	145	CLRSC2 STY V2
FB3A:	A0 27	146	LDY ##27
FB3C:	A9 00	147	CLRSC3 LDA ##00
FB3E:	85 30	148	STA COLOR
FB40:	20 28 FB	149	JSR VLINE
FB43:	88	150	DEY
FB44:	10 F6	151	BPL CLRSC3
FB46:	60	152	RTS

F847:		153	PAGE
F847:	48	154	GBASCALC PHA
F848:	4A	155	LSR A
F849:	29 03	156	AND ##03
F84B:	09 04	157	ORA ##04
F84D:	85 27	158	STA GBASH
F84F:	68	159	PLA
F850:	29 18	160	AND ##18
F852:	90 02	161	BCC GBCALC
F854:	69 7F	162	ADC ##7F
F856:	85 26	163	GBCALC STA GBASL
F858:	0A	164	ASL A
F859:	0A	165	ASL A
F85A:	05 26	166	ORA GBASL
F85C:	85 26	167	STA GBASL
F85E:	60	168	RTS
F85F:	A5 30	169	LDA COLOR
F861:	18	170	CLC
F862:	69 03	171	ADC ##03
F864:	29 0F	172	SETCOL AND ##0F
F866:	85 30	173	STA COLOR
F868:	0A	174	ASL A
F869:	0A	175	ASL A
F86A:	0A	176	ASL A
F86B:	0A	177	ASL A
F86C:	05 30	178	ORA COLOR
F86E:	85 30	179	STA COLOR
F870:	60	180	RTS
F871:	4A	181	SCRN LSR A
F872:	08	182	PHP
F873:	20 47 F8	183	JSR GBASCALC
F876:	B1 26	184	LDA (GBASL), Y
F878:	28	185	PLP
F879:	90 04	186	SCRN2 BCC RTMSKZ
F87B:	4A	187	LSR A
F87C:	4A	188	LSR A
F87D:	4A	189	LSR A
F87E:	4A	190	LSR A
F87F:	29 0F	191	RTMSKZ AND ##0F
F881:	60	192	RTS

F882:		193	PAGE
F882:	A6 3A	194	INSDS1 LDX PCL
F884:	A4 3B	195	LDY PCH
F886:	20 96 FD	196	JSR PRYX2
F889:	20 48 F9	197	JSR PRBLNK
F88C:	A1 3A	198	INSDS2 LDA (PCL, X)
F88E:	A8	199	TAX
F88F:	4A	200	LSR A
F890:	90 09	201	BCC IEVEN
F892:	6A	202	ROR A
F893:	B0 10	203	BCS ERR
F895:	C9 A2	204	CMP ##A2
F897:	F0 0C	205	BEG ERR
F899:	29 B7	206	AND ##B7
F89B:	4A	207	IEVEN LSR A
F89C:	AA	208	TAX
F89D:	BD 62 F9	209	LDA FMT1, X
F8A0:	20 79 FB	210	JSR SCRNM2
F8A3:	D0 04	211	BNE GETFMT
F8A5:	A0 80	212	ERR LDY ##80
F8A7:	A9 00	213	LDA ##00
F8A9:	AA	214	GETFMT TAX
F8AA:	BD A6 F9	215	LDA FMT2, X
F8AD:	85 2E	216	STA FORMAT
F8AF:	29 03	217	AND ##03
F8B1:	85 2F	218	STA LENGTH
F8B3:	98	219	TYA
F8B4:	29 8F	220	AND ##8F
F8B6:	AA	221	TAX
F8B7:	98	222	TYA
F8B8:	A0 03	223	LDY ##03
F8BA:	E0 8A	224	CPX ##8A
F8BC:	F0 08	225	BEG MNNDX3
F8BE:	4A	226	MNNDX1 LSR A
F8BF:	90 08	227	BCC MNNDX3
F8C1:	4A	228	LSR A
F8C2:	4A	229	MNNDX2 LSR A
F8C3:	09 20	230	ORA ##20
F8C5:	88	231	DEY
F8C6:	D0 FA	232	BNE MNNDX2
F8C8:	C8	233	INY
F8C9:	88	234	MNNDX3 DEY
F8CA:	D0 F2	235	BNE MNNDX1
F8CC:	60	236	RTS
F8CD:	FF FF FF	237	DFB \$FF, \$FF, \$FF

F8D0:		238	PAGE
F8D0:	20 82 F8	239	INSTDSP JSR INSDS1
F8D3:	48	240	PHA
F8D4:	B1 3A	241	PRNTOP LDA (PCL), Y
F8D6:	20 DA FD	242	JSR PRBYTE
F8D9:	A2 01	243	LDX ##01
F8DB:	20 4A F9	244	PRNTBL JSR PRBL2
F8DE:	C4 2F	245	CPY LENGTH
F8E0:	C8	246	INY
F8E1:	90 F1	247	BCC PRNTOP
F8E3:	A2 03	248	LDX ##03
F8E5:	C0 04	249	CPY ##04
F8E7:	90 F2	250	BCC PRNTBL
F8E9:	68	251	PLA
F8EA:	AB	252	TAY
F8EB:	B9 C0 F9	253	LDA MNEM1, Y
F8EE:	85 2C	254	STA LMNEM
F8F0:	B9 00 FA	255	LDA MNEMR, Y
F8F3:	85 2D	256	STA RMNEM
F8F5:	A9 00	257	NXTCOL LDA ##00
F8F7:	A0 05	258	LDY ##05
F8F9:	06 2D	259	PRMN2 ASL RMNEM
F8FB:	26 2C	260	ROL LMNEM
F8FD:	2A	261	ROL A
F8FE:	88	262	DEY
F8FF:	D0 F8	263	BNE PRMN2
F901:	69 BF	264	ADC ##BF
F903:	20 ED FD	265	JSR COUT
F906:	CA	266	DEX
F907:	D0 EC	267	BNE NXTCOL
F909:	20 48 F9	268	JSR PRBLNK
F90C:	A4 2F	269	LDY LENGTH
F90E:	A2 06	270	LDX ##06
F910:	E0 03	271	PRADR1 CPX ##03
F912:	F0 1C	272	BEG PRADR5
F914:	06 2E	273	PRADR2 ASL FORMAT
F916:	90 0E	274	BCC PRADR3
F918:	BD B3 F9	275	LDA CHAR1-1, X
F91B:	20 ED FD	276	JSR COUT
F91E:	BD B9 F9	277	LDA CHAR2-1, X
F921:	F0 03	278	BEG PRADR3
F923:	20 ED FD	279	JSR COUT
F926:	CA	280	PRADR3 DEX
F927:	D0 E7	281	BNE PRADR1
F929:	60	282	RTS
F92A:	88	283	PRADR4 DEY
F92B:	30 E7	284	BMI PRADR2
F92D:	20 DA FD	285	JSR PRBYTE
F930:	A5 2E	286	PRADR5 LDA FORMAT
F932:	C9 E8	287	CMP ##E8
F934:	B1 3A	288	LDA (PCL), Y
F936:	90 F2	289	BCC PRADR4

F938:		290	PAGE	
F938:	20 56 F9	291	RELADR JSR	PCADJ3
F938:	AA	292	TAX	
F93C:	E8	293	INX	
F93D:	D0 01	294	BNE	PRNTYX
F93F:	C8	295	INX	
F940:	98	296	PRNTYX	TYA
F941:	20 DA FD	297	PRNTAX	JSR
F944:	8A	298	PRNTX	TXA
F945:	4C DA FD	299	JMP	PRBYTE
F948:	A2 03	300	PRBLNK	LDX ##03
F94A:	A9 A0	301	PRBL2	LDA ##A0
F94C:	20 ED FD	302	PRBL3	JSR
F94F:	CA	303	DEX	
F950:	D0 FB	304	BNE	PRBL2
F952:	60	305	RTS	
F953:	38	306	PCADJ	SEC
F954:	A5 2F	307	PCADJ2	LDA
F956:	A4 3B	308	PCADJ3	LDY
F958:	AA	309	TAX	
F959:	10 01	310	BPL	PCADJ4
F95B:	88	311	DEY	
F95C:	65 3A	312	PCADJ4	ADC
F95E:	90 01	313	BCC	RTS2
F960:	C8	314	INX	
F961:	60	315	RTS2	RTS
F962:	04	316	FMT1	DFB \$04
F963:	20	317		DFB \$20
F964:	54	318		DFB \$54
F965:	30	319		DFB \$30
F966:	0D	320		DFB \$0D
F967:	80	321		DFB \$80
F968:	04	322		DFB \$04
F969:	90	323		DFB \$90
F96A:	03	324		DFB \$03
F96B:	22	325		DFB \$22
F96C:	54	326		DFB \$54
F96D:	33	327		DFB \$33
F96E:	0D	328		DFB \$0D
F96F:	80	329		DFB \$80
F970:	04	330		DFB \$04
F971:	90	331		DFB \$90
F972:	04	332		DFB \$04
F973:	20	333		DFB \$20
F974:	54	334		DFB \$54
F975:	33	335		DFB \$33
F976:	0D	336		DFB \$0D
F977:	80	337		DFB \$80
F978:	04	338		DFB \$04
F979:	90	339		DFB \$90
F97A:	04	340		DFB \$04
F97B:	20	341		DFB \$20
F97C:	54	342		DFB \$54
F97D:	38	343		DFB \$38
F97E:	0D	344		DFB \$0D
F97F:	80	345		DFB \$80
F980:	04	346		DFB \$04
F981:	90	347		DFB \$90
F982:	00	348		DFB \$00
F983:	22	349		DFB \$22
F984:	44	350		DFB \$44
F985:	33	351		DFB \$33
F986:	0D	352		DFB \$0D

F987:	C8	353		DFB	\$C8
F988:	44	354		DFB	\$44
F989:	00	355		DFB	\$00
F98A:	11	356		DFB	\$11
F98B:	22	357		DFB	\$22
F98C:	44	358		DFB	\$44
F98D:	33	359		DFB	\$33
F98E:	0D	360		DFB	\$0D
F98F:	C8	361		DFB	\$C8
F990:	44	362		DFB	\$44
F991:	A9	363		DFB	\$A9
F992:	01	364		DFB	\$01
F993:	22	365		DFB	\$22
F994:	44	366		DFB	\$44
F995:	33	367		DFB	\$33
F996:	0D	368		DFB	\$0D
F997:	80	369		DFB	\$80
F998:	04	370		DFB	\$04
F999:	90	371		DFB	\$90
F99A:	01	372		DFB	\$01
F99B:	22	373		DFB	\$22
F99C:	44	374		DFB	\$44
F99D:	33	375		DFB	\$33
F99E:	0D	376		DFB	\$0D
F99F:	80	377		DFB	\$80
F9A0:	04	378		DFB	\$04
F9A1:	90	379		DFB	\$90
F9A2:	26	380		DFB	\$26
F9A3:	31	381		DFB	\$31
F9A4:	87	382		DFB	\$87
F9A5:	9A	383		DFB	\$9A
F9A6:	00	384	FMT2	DFB	\$00
F9A7:	21	385		DFB	\$21
F9A8:	81	386		DFB	\$81
F9A9:	82	387		DFB	\$82
F9AA:	00	388		DFB	\$00
F9AB:	00	389		DFB	\$00
F9AC:	59	390		DFB	\$59
F9AD:	4D	391		DFB	\$4D
F9AE:	91	392		DFB	\$91
F9AF:	92	393		DFB	\$92
F9B0:	86	394		DFB	\$86
F9B1:	4A	395		DFB	\$4A
F9B2:	85	396		DFB	\$85
F9B3:	9D	397		DFB	\$9D
F9B4:	AC	398	CHAR1	DFB	\$AC
F9B5:	A9	399		DFB	\$A9
F9B6:	AC	400		DFB	\$AC
F9B7:	A3	401		DFB	\$A3
F9B8:	A8	402		DFB	\$A8
F9B9:	A4	403		DFB	\$A4
F9BA:	D9	404	CHAR2	DFB	\$D9
F9BB:	00	405		DFB	\$00
F9BC:	D8	406		DFB	\$D8
F9BD:	A4	407		DFB	\$A4
F9BE:	A4	408		DFB	\$A4
F9BF:	00	409		DFB	\$00
F9C0:	1C	410	MNEML	DFB	\$1C
F9C1:	8A	411		DFB	\$8A
F9C2:	1C	412		DFB	\$1C
F9C3:	23	413		DFB	\$23
F9C4:	5D	414		DFB	\$5D
F9C5:	8B	415		DFB	\$8B

F9C6:	1B	416	DFB	\$1B
F9C7:	A1	417	DFB	\$A1
F9C8:	9D	418	DFB	\$9D
F9C9:	8A	419	DFB	\$8A
F9CA:	1D	420	DFB	\$1D
F9CB:	23	421	DFB	\$23
F9CC:	9D	422	DFB	\$9D
F9CD:	8B	423	DFB	\$8B
F9CE:	1D	424	DFB	\$1D
F9CF:	A1	425	DFB	\$A1
F9D0:	00	426	DFB	\$00
F9D1:	29	427	DFB	\$29
F9D2:	19	428	DFB	\$19
F9D3:	AE	429	DFB	\$AE
F9D4:	69	430	DFB	\$69
F9D5:	A8	431	DFB	\$A8
F9D6:	19	432	DFB	\$19
F9D7:	23	433	DFB	\$23
F9D8:	24	434	DFB	\$24
F9D9:	53	435	DFB	\$53
F9DA:	1B	436	DFB	\$1B
F9DB:	23	437	DFB	\$23
F9DC:	24	438	DFB	\$24
F9DD:	53	439	DFB	\$53
F9DE:	19	440	DFB	\$19
F9DF:	A1	441	DFB	\$A1
F9E0:	00	442	DFB	\$00
F9E1:	1A	443	DFB	\$1A
F9E2:	5B	444	DFB	\$5B
F9E3:	5B	445	DFB	\$5B
F9E4:	A5	446	DFB	\$A5
F9E5:	69	447	DFB	\$69
F9E6:	24	448	DFB	\$24
F9E7:	24	449	DFB	\$24
F9E8:	AE	450	DFB	\$AE
F9E9:	AE	451	DFB	\$AE
F9EA:	A8	452	DFB	\$A8
F9EB:	AD	453	DFB	\$AD
F9EC:	29	454	DFB	\$29
F9ED:	00	455	DFB	\$00
F9EE:	7C	456	DFB	\$7C
F9EF:	00	457	DFB	\$00
F9F0:	15	458	DFB	\$15
F9F1:	9C	459	DFB	\$9C
F9F2:	6D	460	DFB	\$6D
F9F3:	9C	461	DFB	\$9C
F9F4:	A5	462	DFB	\$A5
F9F5:	69	463	DFB	\$69
F9F6:	29	464	DFB	\$29
F9F7:	53	465	DFB	\$53
F9F8:	84	466	DFB	\$84
F9F9:	13	467	DFB	\$13
F9FA:	34	468	DFB	\$34
F9FB:	11	469	DFB	\$11
F9FC:	A5	470	DFB	\$A5
F9FD:	69	471	DFB	\$69
F9FE:	23	472	DFB	\$23
F9FF:	A0	473	DFB	\$A0
FA00:	DB	474	DFB	\$DB
FA01:	62	475	DFB	\$62
FA02:	5A	476	DFB	\$5A
FA03:	48	477	DFB	\$48
FA04:	26	478	DFB	\$26

MNEMR

FA05:	62	479	DFB	\$62
FA06:	94	480	DFB	\$94
FA07:	8B	481	DFB	\$8B
FA08:	54	482	DFB	\$54
FA09:	44	483	DFB	\$44
FA0A:	C8	484	DFB	\$C8
FA0B:	54	485	DFB	\$54
FA0C:	68	486	DFB	\$68
FA0D:	44	487	DFB	\$44
FA0E:	E8	488	DFB	\$E8
FA0F:	94	489	DFB	\$94
FA10:	00	490	DFB	\$00
FA11:	B4	491	DFB	\$B4
FA12:	08	492	DFB	\$08
FA13:	84	493	DFB	\$84
FA14:	74	494	DFB	\$74
FA15:	B4	495	DFB	\$B4
FA16:	2B	496	DFB	\$2B
FA17:	6E	497	DFB	\$6E
FA18:	74	498	DFB	\$74
FA19:	F4	499	DFB	\$F4
FA1A:	CC	500	DFB	\$CC
FA1B:	4A	501	DFB	\$4A
FA1C:	72	502	DFB	\$72
FA1D:	F2	503	DFB	\$F2
FA1E:	A4	504	DFB	\$A4
FA1F:	8A	505	DFB	\$8A
FA20:	00	506	DFB	\$00
FA21:	AA	507	DFB	\$AA
FA22:	A2	508	DFB	\$A2
FA23:	A2	509	DFB	\$A2
FA24:	74	510	DFB	\$74
FA25:	74	511	DFB	\$74
FA26:	74	512	DFB	\$74
FA27:	72	513	DFB	\$72
FA28:	44	514	DFB	\$44
FA29:	68	515	DFB	\$68
FA2A:	B2	516	DFB	\$B2
FA2B:	32	517	DFB	\$32
FA2C:	B2	518	DFB	\$B2
FA2D:	00	519	DFB	\$00
FA2E:	22	520	DFB	\$22
FA2F:	00	521	DFB	\$00
FA30:	1A	522	DFB	\$1A
FA31:	1A	523	DFB	\$1A
FA32:	26	524	DFB	\$26
FA33:	26	525	DFB	\$26
FA34:	72	526	DFB	\$72
FA35:	72	527	DFB	\$72
FA36:	88	528	DFB	\$88
FA37:	C8	529	DFB	\$C8
FA38:	C4	530	DFB	\$C4
FA39:	CA	531	DFB	\$CA
FA3A:	26	532	DFB	\$26
FA3B:	48	533	DFB	\$48
FA3C:	44	534	DFB	\$44
FA3D:	44	535	DFB	\$44
FA3E:	A2	536	DFB	\$A2
FA3F:	C8	537	DFB	\$C8

```

FA40:          538      PAGE
FA40: 85 45     539  IRG   STA ACC
FA42: 68        540      PLA
FA43: 48        541      PHA
FA44: 0A        542      ASL A
FA45: 0A        543      ASL A
FA46: 0A        544      ASL A
FA47: 30 03     545      BMI BREAK
FA49: 6C FE 03  546      JMP (IRQLDC)
FA4C: 28        547  BREAK  PLP
FA4D: 20 4C FF  548      JSR SAV1
FA50: 68        549      PLA
FA51: 85 3A     550      STA PCL
FA53: 68        551      PLA
FA54: 85 38     552      STA PCH
FA56: 6C F0 03  553      JMP (BRKV)           ; BRKV WRITTEN OVER BY DISK BOOT
FA59: 20 82 FB  554  OLDBRK JSR INSDS1
FA5C: 20 DA FA  555      JSR RGDSP1
FA5F: 4C 65 FF  556      JMP MON
FA62: D8        557  RESET  CLD           ; DO THIS FIRST THIS TIME
FA63: 20 84 FE  558      JSR SETNORM
FA66: 20 2F FB  559      JSR INIT
FA69: 20 93 FE  560      JSR SETVID
FA6C: 20 89 FE  561      JSR SETKBD
FA6F: AD 58 C0  562  INITAN  LDA SETANO           ; AN0 = TTL HI
FA72: AD 5A C0  563      LDA SETAN1           ; AN1 = TTL HI
FA75: AD 5D C0  564      LDA CLRAN2           ; AN2 = TTL LO
FA78: AD 5F C0  565      LDA CLRAN3           ; AN3 = TTL LO
FA7B: AD FF CF  566      LDA CLRROM           ; TURN OFF EXTNSN ROM
FA7E: 2C 10 C0  567      BIT KBDSTRB        ; CLEAR KEYBOARD
FA81: D8        568  NEWMON  CLD
FA82: 20 3A FF  569      JSR BELL
FA85: AD F3 03  570      LDA SOFTEV+1
FAB8: 49 A5     571      EOR #A5             ; A FUNNY COMPLEMENT OF THE
FABA: CD F4 03  572      CMP PWREDUP         ; PWR UP BYTE ???
FABD: D0 17     573      BNE PWRUP           ; NO SO PWRUP
FABF: AD F2 03  574      LDA SOFTEV
FA92: D0 0F     575      BNE NOFIX           ; YES SEE IF COLD START
FA94: A9 E0     576      LDA #E0             ; HAS BEEN DONE YET?
FA96: CD F3 03  577      CMP SOFTEV+1       ; ??
FA99: D0 08     578      BNE NOFIX           ; YES SO REENTER SYSTEM
FA9B: A0 03     579  FIXSEV  LDY #3             ; NO SO POINT AT WARM START
FA9D: 8C F2 03  580      STY SOFTEV         ; FOR NEXT RESET
FAA0: 4C 00 E0  581      JMP BASIC           ; AND DO THE COLD START
FAA3: 6C F2 03  582  NOFIX  JMP (SOFTEV)       ; SOFT ENTRY VECTOR
FAA6:          583  *****
FAA6: 20 60 FB  584  PWRUP   JSR APPLEII
FAA9:          585  SETPG3 EQU *           ; SET PAGE 3 VECTORS
FAA9: A2 05     586      LDX #5
FAAB: BD FC FA  587  SETPLP  LDA PWRCON-1, X     ; WITH CNTRL B ADRS
FAAE: 9D EF 03  588      STA BRKV-1, X      ; OF CURRENT BASIC
FAB1: CA        589      DEX
FAB2: D0 F7     590      BNE SETPLP
FAB4: A9 C8     591      LDA #C8
FAB6: 86 00     592      STX LOC0           ; SETPG3 MUST RETURN X=0
FAB8: 85 01     593      STA LOC1           ; SET PTR H
FABA: A0 07     594  SLOOP   LDY #7             ; Y IS BYTE PTR
FABC: C6 01     595      DEC LOC1
FABE: A5 01     596      LDA LOC1
FAC0: C9 C0     597      CMP #C0           ; AT LAST SLOT YET?
FAC2: F0 D7     598      BEG FIXSEV        ; YES AND IT CANT BE A DISK
FAC4: 8D FB 07  599      STA MSLDT
FAC7: B1 00     600  NXTBYT  LDA (LOC0), Y      ; FETCH A SLOT BYTE

```



```

FAC9: D9 01 FB      601      CMP DISKID-1,Y ; IS IT A DISK ??
FACC: D0 EC      602      BNE SLOOP      ; NO SO NEXT SLOT DOWN
FACE: 88          603      DEY
FACF: 88          604      DEY ; YES SO CHECK NEXT BYTE
FAD0: 10 F5      605      BPL NXTBYT    ; UNTIL 4 CHECKED
FAD2: 6C 00 00   606      JMP (LOCO)
FAD5: EA          607      NOP
FAD6: EA          608      NOP
FAD7:            609 * REGDSP MUST ORG $FAD7
FAD7: 20 8E FD   610 REGDSP JSR CROUT
FADA: A9 45      611 RGDSP1 LDA #45
FADC: 85 40      612      STA A3L
FADE: A9 00      613      LDA #00
FAEO: 85 41      614      STA A3H
FAE2: A2 FB      615      LDX #FFB
FAE4: A9 A0      616 RDSP1  LDA #A0
FAE6: 20 ED FD   617      JSR COUT
FAE9: BD 1E FA   618      LDA RTBL-251,X
FAEC: 20 ED FD   619      JSR COUT
FAEF: A9 BD      620      LDA #BBD
FAF1: 20 ED FD   621      JSR COUT
FAF4:            622 * LDA ACC+5, X
FAF4: B5 4A      623      DFB #B5, #4A
FAF6: 20 DA FD   624      JSR PRBYTE
FAF9: EB          625      INX
FAFA: 30 EB      626      BMI RDSP1
FAFC: 60          627      RTS
FAFD: 59 FA      628 PWRCON DW DLDBRK
FAFF: 00 E0 45   629      DFB #00, #E0, #45
FB02: 20 FF 00
FB05: FF          630 DISKID DFB #20, #FF, #00, #FF
FB06: 03 FF 3C   631      DFB #03, #FF, #3C
FB09: C1 D0 D0   632 TITLE  DFB #C1, #D0, #D0
FB0C: CC C5 A0   633      DFB #CC, #C5, #A0
FB0F: DD DB      634      DFB #DD, #DB
FB11:            635 XLTBL  EQU *
FB11: C4 C2 C1   636      DFB #C4, #C2, #C1
FB14: FF C3      637      DFB #FF, #C3
FB16: FF FF FF   638      DFB #FF, #FF, #FF
FB19:            639 * MUST  ORG $FB19
FB19: C1 D8 D9   640 RTBL   DFB #C1, #D8, #D9
FB1C: D0 D3      641      DFB #D0, #D3
FB1E: AD 70 C0   642 PREAD  LDA PTRIG
FB21:            643      LST ON
FB21: A0 00      644      LDY #00
FB23: EA          645      NOP
FB24: EA          646      NOP
FB25: BD 64 C0   647 PREAD2 LDA PADDLO, X
FB28: 10 04      648      BPL RTS2D
FB2A: C8          649      INY
FB2B: D0 FB      650      BNE PREAD2
FB2D: 88          651      DEY
FB2E: 60          652 RTS2D  RTS

```

```

ASM2 &12
NEWMON. SRC2 PASS2

```

FB2F:		1	PAGE		
FB2F:	A9 00	2	INIT	LDA ##00	
FB31:	85 48	3		STA STATUS	
FB33:	AD 56 C0	4		LDA LORES	
FB36:	AD 54 C0	5		LDA LOWSCR	
FB39:	AD 51 C0	6	SETTXT	LDA TXTSET	
FB3C:	A9 00	7		LDA ##00	
FB3E:	F0 08	8		BEG SETWND	
FB40:	AD 50 C0	9	SETGR	LDA TXTCLR	
FB43:	AD 53 C0	10		LDA MIXSET	
FB46:	20 36 F8	11		JSR CLRRTOP	
FB49:	A9 14	12		LDA ##14	
FB4B:	85 22	13	SETWND	STA WNDTOP	
FB4D:	A9 00	14		LDA ##00	
FB4F:	85 20	15		STA WNDLFT	
FB51:	A9 28	16		LDA ##28	
FB53:	85 21	17		STA WNDWDTH	
FB55:	A9 18	18		LDA ##18	
FB57:	85 23	19		STA WNDBTM	
FB59:	A9 17	20		LDA ##17	
FB5B:	85 25	21	TABV	STA CV	
FB5D:	4C 22 FC	22		JMP VTAB	
FB60:	20 58 FC	23	APPLEII	JSR HOME	; CLEAR THE SCRNM
FB63:	A0 08	24		LDY #8	
FB65:	B9 08 FB	25	STITLE	LDA TITLE-1,Y	; GET A CHAR
FB68:	99 0E 04	26		STA LINE1+14,Y	
FB6B:	88	27		DEY	
FB6C:	D0 F7	28		BNE STITLE	
FB6E:	60	29		RTS	
FB6F:	AD F3 03	30	SETPWRC	LDA SOFTEV+1	
FB72:	49 A5	31		EOR ##A5	
FB74:	8D F4 03	32		STA PWREDUP	
FB77:	60	33		RTS	
FB78:		34	VIDWAIT	EGU *	; CHECK FOR A PAUSE
FB78:	C9 8D	35		CMP ##8D	; ONLY WHEN I HAVE A CR
FB7A:	D0 18	36		BNE NOWAIT	; NOT SO, DO REGULAR
FB7C:	AC 00 C0	37		LDY KBD	; IS KEY PRESSED?
FB7F:	10 13	38		BPL NOWAIT	; NO
FB81:	C0 93	39		CPY ##93	; IS IT CTL S ?
FB83:	D0 0F	40		BNE NOWAIT	; NO SO IGNORE
FB85:	2C 10 C0	41		BIT KBDSTRB	; CLEAR STROBE
FB88:	AC 00 C0	42	KBDWAIT	LDY KBD	; WAIT TILL NEXT KEY TO RESUME
FB8B:	10 FB	43		BPL KBDWAIT	; WAIT FOR KEYPRESS
FB8D:	C0 83	44		CPY ##83	; IS IT CONTROL C ?
FB8F:	F0 03	45		BEG NOWAIT	; YES SO LEAVE IT
FB91:	2C 10 C0	46		BIT KBDSTRB	; CLR STROBE
FB94:	4C FD FB	47	NOWAIT	JMP VIDOUT	; DO AS BEFORE

FB97:		48	PAGE	
FB97: 38		49	ESCOLD	SEC ; INSURE CARRY SET
FB98: 4C 2C FC		50	JMP	ESC1
FB9B: A8		51	ESCNDW	TAY ; USE CHAR AS INDEX
FB9C: B9 48 FA		52	LDA	XLTBL-\$C9,Y ; XLATE IJKM TO CBAD
FB9F: 20 97 FB		53	JSR	ESCOLD ; DO THIS CURSOR MOTION
FBA2: 20 0C FD		54	JSR	RDKEY ; AND GET NEXT
FBA5: C9 CE		55	ESCNEW	CMP ##CE ; IS THIS AN N ?
FBA7: B0 EE		56	BCS	ESCOLD ; N OR GREATER DO IT
FBA9: C9 C9		57	CMP	##C9 ; LESS THAN I ?
FBAB: 90 EA		58	BCC	ESCOLD ; YES SO OLD WAY
FBAD: C9 CC		59	CMP	##CC ; IS IT A L ?
FBAF: F0 E6		60	BEQ	ESCOLD ; DO NORMAL
FBB1: D0 EB		61	BNE	ESCNDW ; GO DO IT
FBB3: EA		62	NOP	
<u>FBB4: EA</u>		63	NOP	
FBB5: EA		64	NOP	
FBB6: EA		65	NOP	
FBB7: EA		66	NOP	
FBB8: EA		67	NOP	
FBB9: EA		68	NOP	
FBBA: EA		69	NOP	
FBBB: EA		70	NOP	
FBBC: EA		71	NOP	
FBBD: EA		72	NOP	
FBBE: EA		73	NOP	
FBBF: EA		74	NOP	
FBC0: EA		75	NOP	
FBC1:		76	*	MUST ORG \$FBC1
FBC1: 48		77	BASCALC	PHA
FBC2: 4A		78	LSR	A
FBC3: 29 03		79	AND	##03
FBC5: 09 04		80	DRA	##04
FBC7: 85 29		81	STA	BASH
FBC9: 68		82	PLA	
FBCA: 29 18		83	AND	##18
<u>FBCC: 90 02</u>		84	BCC	BASCLC2
FBCE: 69 7F		85	ADC	##7F
FBD0: 85 28		86	BASCLC2	STA BASL
FBD2: 0A		87	ASL	A
FBD3: 0A		88	ASL	A
FBD4: 05 28		89	DRA	BASL
FBD6: 85 28		90	STA	BASL
FBD8: 60		91	RTS	
FBD9: C9 87		92	BELL1	CMP ##87
FBD8: D0 12		93	BNE	RTS2B
FBD8: A9 40		94	LDA	##40
FBD8: 20 AB FC		95	JSR	WAIT
FBE2: A0 C0		96	LDY	##C0
FBE4: A9 0C		97	BELL2	LDA ##0C
FBE6: 20 AB FC		98	JSR	WAIT
FBE9: AD 30 C0		99	LDA	SPKR
FBEC: 88		100	DEY	
FBED: D0 F5		101	BNE	BELL2
FBEF: 60		102	RTS2B	RTS

FBF0:		103	PAGE	
FBF0:	A4 24	104	STORADV	LDY CH
FBF2:	<u>91 28</u>	105		STA (BASL), Y
FBF4:	<u>E6 24</u>	106	ADVANCE	INC CH
FBF6:	A5 24	107		LDA CH
FBF8:	C5 21	108		CMP WNDWDTH
FBFA:	B0 66	109		BCS CR
FBFC:	60	110	RTS3	RTS
FBFD:	C9 A0	111	VIDDOUT	CMP ##A0
FBFF:	B0 EF	112		BCS STORADV
FC01:	A8	113		TAY
FC02:	10 EC	114		BPL STORADV
FC04:	C9 8D	115		CMP ##8D
FC06:	F0 5A	116		BEG CR
FC08:	C9 8A	117		CMP ##8A
FC0A:	F0 5A	118		BEG LF
FC0C:	C9 88	119		CMP ##88
FC0E:	D0 C9	120		BNE BELL1
FC10:	C6 24	121	BS	DEC CH
FC12:	10 E8	122		BPL RTS3
FC14:	A5 21	123		LDA WNDWDTH
FC16:	<u>85 24</u>	124		STA CH
FC18:	C6 24	125		DEC CH
FC1A:	A5 22	126	UP	LDA WNDTOP
FC1C:	C5 25	127		CMP CV
FC1E:	B0 0B	128		BCS RTS4
FC20:	C6 25	129		DEC CV
FC22:	A5 25	130	VTAB	LDA CV
FC24:	20 C1	131	VTABZ	JSR BASCALC
FC27:	65 20	132		ADC WNDLFT
FC29:	85 28	133		STA BASL
FC2B:	60	134	RTS4	RTS
FC2C:	49 C0	135	ESC1	EOR ##C0 ; ESC @ ?
FC2E:	F0 28	136		BEG HOME ; IF SO DO HOME AND CLEAR
FC30:	69 FD	137		ADC ##FD ; ESC-A OR B CHECK
FC32:	90 C0	138		BCC ADVANCE ; A, ADVANCE
FC34:	F0 DA	139		BEG BS ; B, BACKSPACE
FC36:	69 FD	140		ADC ##FD ; ESC-C OR D CHECK
FC38:	90 2C	141		BCC LF ; C, DOWN
FC3A:	F0 DE	142		BEG UP ; D, GO UP
FC3C:	69 FD	143		ADC ##FD ; ESC-E OR F CKECK
FC3E:	90 5C	144		BCC CLREOL ; E, CLEAR TO END OF LINE
FC40:	<u>D0 E9</u>	145		BNE RTS4 ; ELSE NOT F, RETURN
FC42:	A4 24	146	CLREOP	LDY CH ; ESC F IS CLR TO END OF PAGE
FC44:	A5 25	147		LDA CV
FC46:	48	148	CLEDP1	PHA
FC47:	20 24	149	FC	JSR VTABZ
FC4A:	20 9E	150	FC	JSR CLEDLZ
FC4D:	A0 00	151		LDY ##00
FC4F:	68	152		PLA
FC50:	69 00	153		ADC ##00
FC52:	C5 23	154		CMP WNDBTM
FC54:	90 F0	155		BCC CLEDP1
FC56:	B0 CA	156		BCS VTAB
FC58:	A5 22	157	HOME	LDA WNDTOP
FC5A:	85 25	158		STA CV
FC5C:	A0 00	159		LDY ##00
FC5E:	84 24	160		STY CH
FC60:	F0 E4	161		BEG CLEDP1

FC62:		162	PAGE
FC62:	A9 00	163	CR LDA ##00
FC64:	85 24	164	STA CH
FC66:	E6 25	165	LF INC CV
FC68:	<u>A5 25</u>	166	LDA CV
FC6A:	C5 23	167	CMP WNDBTM
FC6C:	90 86	168	BCC VTABZ
FC6E:	C6 25	169	DEC CV
FC70:	A5 22	170	SCROLL LDA WNDTOP
FC72:	48	171	PHA
FC73:	20 24 FC	172	JSR VTABZ
FC76:	A5 28	173	SCRL1 LDA BASL
FC78:	85 2A	174	STA BAS2L
FC7A:	A5 29	175	LDA BASH
FC7C:	85 2B	176	STA BAS2H
FC7E:	A4 21	177	LDY WNDWDTH
FC80:	88	178	DEY
FC81:	68	179	PLA
FC82:	69 01	180	ADC ##01
FC84:	C5 23	181	CMP WNDBTM
FC86:	B0 0D	182	BCS SCRL3
FC88:	48	183	PHA
FC89:	20 24 FC	184	JSR VTABZ
FC8C:	B1 28	185	SCRL2 LDA (BASL), Y
FC8E:	91 2A	186	STA (BAS2L), Y
<u>FC90:</u>	<u>88</u>	187	DEY
FC91:	10 F9	188	BPL SCRL2
FC93:	30 E1	189	BMI SCRL1
FC95:	A0 00	190	SCRL3 LDY ##00
FC97:	20 9E FC	191	JSR CLEOLZ
FC9A:	B0 86	192	BCS VTAB
FC9C:	A4 24	193	CLREOL LDY CH
FC9E:	A9 A0	194	CLEOLZ LDA ##A0
FCA0:	91 28	195	CLEOL2 STA (BASL), Y
FCA2:	C8	196	INY
FCA3:	C4 21	197	CPY WNDWDTH
FCA5:	90 F9	198	BCC CLEOL2
FCA7:	60	199	RTS
FCA8:	38	200	WAIT SEC
FCA9:	48	201	WAIT2 PHA
FCAA:	E9 01	202	WAIT3 SBC ##01
FCAC:	D0 FC	203	BNE WAIT3
FCAE:	68	204	PLA
FCAF:	E9 01	205	SBC ##01
<u>FCB1:</u>	<u>D0 F6</u>	206	BNE WAIT2
FCB3:	60	207	RTS
FCB4:	E6 42	208	NXTA4 INC A4L
FCB6:	D0 02	209	BNE NXTA1
FCB8:	E6 43	210	INC A4H
FCBA:	A5 3C	211	NXTA1 LDA A1L
FCBC:	C5 3E	212	CMP A2L
FCBE:	A5 3D	213	LDA A1H
FCC0:	E5 3F	214	SBC A2H
FCC2:	E6 3C	215	INC A1L
FCC4:	D0 02	216	BNE RTS4B
FCC6:	E6 3D	217	INC A1H
FCC8:	60	218	RTS4B RTS

FCC9:		219	PAGE	
FCC9:	A0 4B	220	HEADR	LDY ##4B
FCCB:	20 DB FC	221		JSR ZERDLY
FCCE:	D0 F9	222		BNE HEADR
FCDO:	69 FE	223		ADC ##FE
FCD2:	B0 F5	224		BCS HEADR
FCD4:	A0 21	225		LDY ##21
FCD6:	20 DB FC	226	WRBIT	JSR ZERDLY
FCD9:	C8	227		INY
FCDA:	C8	228		INY
FCDB:	88	229	ZERDLY	DEY
FCDC:	D0 FD	230		BNE ZERDLY
FCDE:	90 05	231		BCC WRTAPE
FCE0:	A0 32	232		LDY ##32
FCE2:	88	233	ONEDLY	DEY
FCE3:	D0 FD	234		BNE ONEDLY
FCE5:	AC 20 CO	235	WRTAPE	LDY TAPEOUT
FCE8:	A0 2C	236		LDY ##2C
FCEA:	CA	237		DEX
FCEB:	60	238		RTS
FCEC:	A2 0B	239	RDBYTE	LDX ##0B
FCEE:	48	240	RDBYT2	PHA
FCEF:	20 FA FC	241		JSR RD2BIT
FCF2:	68	242		PLA
FCF3:	2A	243		ROL A
FCF4:	A0 3A	244		LDY ##3A
FCF6:	CA	245		DEX
FCF7:	D0 F5	246		BNE RDBYT2
FCF9:	60	247		RTS
FCFA:	20 FD FC	248	RD2BIT	JSR RDBIT
FCFD:	88	249	RDBIT	DEY
FCFE:	AD 60 CO	250		LDA TAPEIN
FD01:	45 2F	251		EOR LASTIN
FD03:	10 F8	252		BPL RDBIT
FD05:	45 2F	253		EOR LASTIN
FD07:	85 2F	254		STA LASTIN
FD09:	C0 80	255		CPY ##80
FD0B:	60	256		RTS
FD0C:	A4 24	257	RDKEY	LDY CH
FD0E:	B1 28	258		LDA (BASL), Y
FD10:	48	259		PHA
FD11:	29 3F	260		AND ##3F
FD13:	09 40	261		ORA ##40
FD15:	91 28	262		STA (BASL), Y
FD17:	68	263		PLA
FD18:	6C 3B 00	264		JMP (KSWL)
FD1B:	E6 4E	265	KEYIN	INC RNDL
FD1D:	D0 02	266		BNE KEYIN2
FD1F:	E6 4F	267		INC RNDH
FD21:	2C 00 CO	268	KEYIN2	BIT KBD ; READ KEYBOARD
FD24:	10 F5	269		BPL KEYIN
FD26:	91 28	270		STA (BASL), Y
FD28:	AD 00 CO	271		LDA KBD
FD2B:	2C 10 CO	272		BIT KBDSTRB
FD2E:	60	273		RTS
FD2F:	20 0C FD	274	ESC	JSR RDKEY
FD32:	20 A5 FB	275		JSR ESCNEW
FD35:	20 0C FD	276	RDCHAR	JSR RDKEY
FD38:	C9 98	277		CMP ##9B
FD3A:	F0 F3	278		BEQ ESC
FD3C:	60	279		RTS

FD3D:		280	PAGE	
FD3D:	A5 32	281	NOTCR	LDA INVFLG
FD3F:	48	282		PHA
FD40:	A9 FF	283		LDA ##FF
FD42:	85 32	284		STA INVFLG
FD44:	BD 00 02	285		LDA IN, X
FD47:	20 ED FD	286		JSR COUT
FD4A:	68	287		PLA
FD4B:	85 32	288		STA INVFLG
FD4D:	BD 00 02	289		LDA IN, X
FD50:	C9 88	290		CMP ##88
FD52:	F0 1D	291		BEQ BCKSPC
FD54:	C9 98	292		CMP ##98
FD56:	F0 0A	293		BEQ CANCEL
FD58:	E0 F8	294		CPX ##F8
FD5A:	90 03	295		BCC NOTCR1
FD5C:	20 3A FF	296		JSR BELL
FD5F:	E8	297	NOTCR1	INX
FD60:	D0 13	298		BNE NXTCHAR
FD62:	A9 DC	299	CANCEL	LDA ##DC
FD64:	20 ED FD	300		JSR COUT
FD67:	20 8E FD	301	GETLNZ	JSR CROUT
FD6A:	A5 33	302	GETLN	LDA PROMPT
FD6C:	20 ED FD	303		JSR COUT
FD6F:	A2 01	304		LDX ##01
FD71:	8A	305	BCKSPC	TXA
FD72:	F0 F3	306		BEQ GETLNZ
FD74:	CA	307		DEX
FD75:	20 35 FD	308	NXTCHAR	JSR RDCHAR
FD78:	C9 95	309		CMP ##95
FD7A:	D0 02	310		BNE CAPTST
FD7C:	B1 28	311		LDA (BASL), Y
FD7E:	C9 E0	312	CAPTST	CMP ##E0
FD80:	90 02	313		BCC ADDINP
FD82:	29 DF	314		AND ##DF
FD84:	9D 00 02	315	ADDINP	STA IN, X
FD87:	C9 8D	316		CMP ##8D
FD89:	D0 B2	317		BNE NOTCR
FD8B:	20 9C FC	318		JSR CLREOL
FD8E:	A9 8D	319	CROUT	LDA ##8D
FD90:	D0 58	320		BNE COUT
FD92:	A4 3D	321	PRA1	LDY A1H
FD94:	A6 3C	322		LDX A1L
FD96:	20 8E FD	323	PRYX2	JSR CROUT
FD99:	20 40 F9	324		JSR PRNTYX
FD9C:	A0 00	325		LDY ##00
FD9E:	A9 AD	326		LDA ##AD
FDA0:	4C ED FD	327		JMP COUT

; SHIFT TO UPPER CASE

FDA3:		328	PAGE	
<u>FDA3:</u>	<u>A5 3C</u>	329	XAMB LDA A1L	
FDA5:	09 07	330	ORA ##07	
FDA7:	85 3E	331	STA A2L	
FDA9:	A5 3D	332	LDA A1H	
FDAB:	85 3F	333	STA A2H	
FDAD:	A5 3C	334	MOD8CHK LDA A1L	
FDAF:	29 07	335	AND ##07	
FDB1:	D0 03	336	BNE DATAOUT	
FDB3:	20 92 FD	337	XAM JSR PRA1	
FDB6:	A9 A0	338	DATAOUT LDA ##A0	
FDB8:	20 ED FD	339	JSR COUT	
FDBB:	B1 3C	340	LDA (A1L), Y	
FDBD:	20 DA FD	341	JSR PRBYTE	
FDC0:	20 BA FC	342	JSR NXTA1	
FDC3:	90 E8	343	BCC MOD8CHK	
FDC5:	60	344	RTS4C RTS	
FDC6:	4A	345	XAMPM LSR A	
FDC7:	90 EA	346	BCC XAM	
FDC9:	4A	347	LSR A	
FDCA:	4A	348	LSR A	
<u>FDCB:</u>	<u>A5 3E</u>	349	LDA A2L	
FDCD:	90 02	350	BCC ADD	
FDCF:	49 FF	351	EDR ##FF	
FDD1:	65 3C	352	ADD ADC A1L	
FDD3:	48	353	PHA	
FDD4:	A9 BD	354	LDA ##BD	
FDD6:	20 ED FD	355	JSR COUT	
FDD9:	68	356	PLA	
FDDA:	48	357	PRBYTE PHA	
FDDB:	4A	358	LSR A	
FDDC:	4A	359	LSR A	
FDDD:	4A	360	LSR A	
FDDE:	4A	361	LSR A	
FDDF:	20 E5 FD	362	JSR PRHEXZ	
FDE2:	68	363	PLA	
FDE3:	29 0F	364	PRHEX AND ##0F	
FDE5:	09 B0	365	PRHEXZ ORA ##B0	
FDE7:	C9 BA	366	CMP ##BA	
FDE9:	90 02	367	BCC COUT	
FDEB:	69 06	368	ADC ##06	
<u>FDED:</u>	<u>6C 36 00</u>	369	COUT JMP (CSWL)	
FDF0:	C9 A0	370	COUT1 CMP ##A0	
FDF2:	90 02	371	BCC COUTZ	
FDF4:	25 32	372	AND INVFLG	
FDF6:	84 35	373	COUTZ STY YSAV1	
FDF8:	48	374	PHA	
FDF9:	20 78 FB	375	JSR VIDWAIT ; GO CHECK FOR PAUSE	
FDFC:	68	376	PLA	
FDFD:	A4 35	377	LDY YSAV1	
FDFE:	60	378	RTS	

FE00:		379	PAGE
FE00:	C6 34	380	BL1 DEC YSAV
FE02:	F0 9F	381	BEQ XAMB
FE04:	CA	382	BLANK DEX
FE05:	D0 16	383	BNE SETMDZ
FE07:	C9 BA	384	CMP ##BA
FE09:	D0 BB	385	BNE XAMPM
FE0B:	85 31	386	STOR STA MODE
FE0D:	A5 3E	387	LDA A2L
FE0F:	91 40	388	STA (A3L), Y
FE11:	E6 40	389	INC A3L
FE13:	D0 02	390	BNE RTS5
FE15:	E6 41	391	INC A3H
FE17:	60	392	RTS5 RTS
FE18:	A4 34	393	SETMODE LDY YSAV
FE1A:	B9 FF 01	394	LDA IN-1, Y
FE1D:	85 31	395	SETMDZ STA MODE
FE1F:	60	396	RTS
FE20:	A2 01	397	LT LDX ##01
FE22:	B5 3E	398	LT2 LDA A2L, X
FE24:	95 42	399	STA A4L, X
FE26:	95 44	400	STA A5L, X
FE28:	CA	401	DEX
FE29:	10 F7	402	BPL LT2
FE2B:	60	403	RTS
FE2C:	B1 3C	404	MOVE LDA (A1L), Y
FE2E:	91 42	405	STA (A4L), Y
FE30:	20 B4 FC	406	JSR NXTA4
FE33:	90 F7	407	BCC MOVE
FE35:	60	408	RTS
FE36:	B1 3C	409	VFY LDA (A1L), Y
FE38:	D1 42	410	CMP (A4L), Y
FE3A:	F0 1C	411	BEQ VFYOK
FE3C:	20 92 FD	412	JSR PRA1
FE3F:	B1 3C	413	LDA (A1L), Y
FE41:	20 DA FD	414	JSR PRBYTE
FE44:	A9 A0	415	LDA ##A0
FE46:	20 ED FD	416	JSR COUT
FE49:	A9 AB	417	LDA ##AB
FE4B:	20 ED FD	418	JSR COUT
FE4E:	B1 42	419	LDA (A4L), Y
FE50:	20 DA FD	420	JSR PRBYTE
FE53:	A9 A9	421	LDA ##A9
FE55:	20 ED FD	422	JSR COUT
FE58:	20 B4 FC	423	VFYOK JSR NXTA4
FE5B:	90 D9	424	BCC VFY
FE5D:	60	425	RTS
FE5E:	20 75 FE	426	LIST JSR A1PC
FE61:	A9 14	427	LDA ##14
FE63:	48	428	LIST2 PHA
FE64:	20 D0 F8	429	JSR INSTDSP
FE67:	20 53 F9	430	JSR PCADJ
FE6A:	85 3A	431	STA PCL
FE6C:	84 3B	432	STY PCH
FE6E:	68	433	PLA
FE6F:	38	434	SEC
FE70:	E9 01	435	SBC ##01
FE72:	D0 EF	436	BNE LIST2
FE74:	60	437	RTS

FE75:		438	PAGE		
FE75:	8A	439	A1PC	TXA	
FE76:	F0 07	440		BEG	A1PCRTS
FE78:	B5 3C	441	A1PCLP	LDA	A1L, X
FE7A:	95 3A	442		STA	PCL, X
FE7C:	CA	443		DEX	
FE7D:	10 F9	444		BPL	A1PCLP
FE7F:	60	445	A1PCRTS	RTS	
FE80:	A0 3F	446	SETINV	LDY	##3F
FE82:	D0 02	447		BNE	SETIFLG
FE84:	A0 FF	448	SETNORM	LDY	##FF
FE86:	84 32	449	SETIFLG	STY	INVFLG
FE88:	60	450		RTS	
FE89:	A9 00	451	SETKBD	LDA	##00
FE8B:	85 3E	452	INPORT	STA	A2L
FE8D:	A2 38	453	INPRT	LDX	#KSWL
FE8F:	A0 1B	454		LDY	#KEYIN
FE91:	D0 08	455		BNE	IOPRT
FE93:	A9 00	456	SETVID	LDA	##00
FE95:	85 3E	457	OUTPORT	STA	A2L
FE97:	A2 36	458	OUTPRT	LDX	#CSWL
FE99:	A0 F0	459		LDY	#COUT1
FE9B:	A5 3E	460	IOPRT	LDA	A2L
FE9D:	29 0F	461		AND	##0F
FE9F:	F0 06	462		BEG	IOPRT1
FEA1:	09 C0	463		ORA	#IOADR/256
FEA3:	A0 00	464		LDY	##00
FEA5:	F0 02	465		BEG	IOPRT2
FEA7:	A9 FD	466	IOPRT1	LDA	#COUT1/256
FEA9:		467	IOPRT2	EGU	*
FEA9:	94 00	468		STY	LOCO, X ; \$94, \$00
FEAB:	95 01	469		STA	LOC1, X ; \$95, \$01
FEAD:	60	470		RTS	
FEAE:	EA	471		NOP	
FEAF:	EA	472		NOP	
FEB0:	4C 00 E0	473	XBASIC	JMP	BASIC
FEB3:	4C 03 E0	474	BASCONT	JMP	BASIC2
FEB6:	20 75 FE	475	GO	JSR	A1PC
FEB9:	20 3F FF	476		JSR	RESTORE
FEBC:	6C 3A 00	477		JMP	(PCL)
FEBF:	4C D7 FA	478	REGZ	JMP	REGDSP
FEC2:	60	479	TRACE	RTS	
FEC3:		480	* TRACE	IS	GONE
FEC3:	EA	481		NOP	
FEC4:	60	482	STEPZ	RTS	; STEP IS GONE
FEC5:	EA	483		NOP	
FEC6:	EA	484		NOP	
FEC7:	EA	485		NOP	
FEC8:	EA	486		NOP	
FEC9:	EA	487		NOP	
FECA:	4C F8 03	488	USR	JMP	USRADR

FEC:	489	PAGE
FEC0: A9 40	490 WRITE	LDA ##40
FECF: 20 C9 FC	491	JSR HEADR
FED2: A0 27	492	LDY ##27
FED4: A2 00	493 WR1	LDX ##00
FED6: 41 3C	494	EOR (A1L, X)
FED8: 4B	495	PHA
FED9: A1 3C	496	LDA (A1L, X)
FEDB: 20 ED FE	497	JSR WRBYTE
FEDE: 20 BA FC	498	JSR NXTA1
FEE1: A0 1D	499	LDY ##1D
FEE3: 68	500	PLA
FEE4: 90 EE	501	BCC WR1
FEE6: A0 22	502	LDY ##22
FEE8: 20 ED FE	503	JSR WRBYTE
FEEB: F0 4D	504	BEG BELL
FEED: A2 10	505 WRBYTE	LDX ##10
FEEF: 0A	506 WRBYT2	ASL A
FEF0: 20 D6 FC	507	JSR WRBIT
FEF3: D0 FA	508	BNE WRBYT2
FEF5: 60	509	RTS
FEF6: 20 00 FE	510 CRMON	JSR BL1
FEF9: 68	511	PLA
FEFA: 68	512	PLA
FEFB: D0 6C	513	BNE MONZ
FEFD: 20 FA FC	514 READ	JSR RD2BIT
FF00: A9 16	515	LDA ##16
FF02: 20 C9 FC	516	JSR HEADR
FF05: 85 2E	517	STA CHKSUM
FF07: 20 FA FC	518	JSR RD2BIT
FF0A: A0 24	519 RD2	LDY ##24
FF0C: 20 FD FC	520	JSR RDBIT
FF0F: B0 F9	521	BCS RD2
FF11: 20 FD FC	522	JSR RDBIT
FF14: A0 3B	523	LDY ##3B
FF16: 20 EC FC	524 RD3	JSR RDBYTE
FF19: 81 3C	525	STA (A1L, X)
FF1B: 45 2E	526	EOR CHKSUM
FF1D: 85 2E	527	STA CHKSUM
FF1F: 20 BA FC	528	JSR NXTA1
FF22: A0 35	529	LDY ##35
FF24: 90 F0	530	BCC RD3
FF26: 20 EC FC	531	JSR RDBYTE
FF29: C5 2E	532	CMP CHKSUM
FF2B: F0 0D	533	BEG BELL
FF2D: A9 C5	534 PRERR	LDA ##C5
FF2F: 20 ED FD	535	JSR COUT
FF32: A9 D2	536	LDA ##D2
FF34: 20 ED FD	537	JSR COUT
FF37: 20 ED FD	538	JSR COUT
FF3A: A9 87	539 BELL	LDA ##87
FF3C: 4C ED FD	540	JMP COUT

FF3F:		541	PAGE
FF3F:	A5 48	542	RESTORE LDA STATUS
FF41:	48	543	PHA
FF42:	A5 45	544	LDA A5H
FF44:	A6 46	545	RESTR1 LDX XREG
FF46:	A4 47	546	LDY YREG
FF48:	28	547	PLP
FF49:	60	548	RTS
FF4A:	85 45	549	SAVE STA A5H
FF4C:	86 46	550	SAV1 STX XREG
FF4E:	84 47	551	STY YREG
FF50:	08	552	PHP
FF51:	68	553	PLA
FF52:	85 48	554	STA STATUS
FF54:	<u>BA</u>	555	TSX
FF55:	86 49	556	STX SPNT
FF57:	D8	557	CLD
FF58:	60	558	RTS
FF59:	20 84 FE	559	OLDRST JSR SETNORM
FF5C:	20 2F FB	560	JSR INIT
FF5F:	20 93 FE	561	JSR SETVID
FF62:	<u>20 89 FE</u>	562	JSR SETKBD

FF65:		563	PAGE
FF65:	D8	564	MON CLD
FF66:	20 3A FF	565	JSR BELL
FF69:	A9 AA	566	MONZ LDA ##AA
FF6B:	85 33	567	STA PROMPT
FF6D:	20 67 FD	568	JSR GETLNZ
FF70:	20 C7 FF	569	JSR ZMODE
FF73:	20 A7 FF	570	NXTITM JSR GETNUM
FF76:	84 34	571	STY YSAV
FF78:	A0 17	572	LDY ##17
FF7A:	88	573	CHRSRCH DEY
FF7B:	30 EB	574	BMI MON
FF7D:	D9 CC FF	575	CMP CHRTBL, Y
FF80:	D0 F8	576	BNE CHRSRCH
FF82:	20 BE FF	577	JSR TOSUB
FF85:	A4 34	578	LDY YSAV
FF87:	4C 73 FF	579	JMP NXTITM
FF8A:	A2 03	580	DIG LDX ##03
FF8C:	0A	581	ASL A
FF8D:	0A	582	ASL A
FF8E:	0A	583	ASL A
FF8F:	0A	584	ASL A
FF90:	0A	585	NXTBIT ASL A
FF91:	26 3E	586	ROL A2L
FF93:	26 3F	587	ROL A2H
FF95:	CA	588	DEX
FF96:	10 F8	589	BPL NXTBIT
FF98:	A5 31	590	NXTBAS LDA MODE
FF9A:	D0 06	591	BNE NXTBS2
FF9C:		592	*
FF9C:	B5 3F	593	LDA A2H, X
FF9E:		594	*
FF9E:	95 3D	595	STA A1H, X
FFA0:		596	*
FFA0:	95 41	597	STA A3H, X
FFA2:	E8	598	NXTBS2 INX
FFA3:	<u>F0 E3</u>	599	BEQ NXTBAS
FFA5:	D0 06	600	BNE NXTCHR
FFA7:	A2 00	601	GETNUM LDX ##00
FFA9:	86 3E	602	STX A2L
FFAB:	86 3F	603	STX A2H
FFAD:	B9 00 02	604	NXTCHR LDA IN, Y
FFB0:	C8	605	INY
FFB1:	49 B0	606	EOR ##B0
FFB3:	C9 0A	607	CMP ##0A
FFB5:	90 D3	608	BCC DIG
FFB7:	69 88	609	ADC ##88
FFB9:	C9 FA	610	CMP ##FA
FFBB:	B0 CD	611	BCS DIG
FFBD:	60	612	RTS
FFBE:	A9 FE	613	TOSUB LDA #G0/256
FFC0:	48	614	PHA
FFC1:	B9 E3 FF	615	LDA SUBTBL, Y
FFC4:	48	616	PHA
FFC5:	A5 31	617	LDA MODE
FFC7:	<u>A0 00</u>	618	ZMODE LDY ##00
FFC9:	84 31	619	STY MODE
FFCB:	60	620	RTS

FFCC:	621	PAGE	
FFCC: BC	622	CHRTBL	DFB \$BC
FFCD: B2	623		DFB \$B2
FFCE: BE	624		DFB \$BE
FFCF: B2	625		DFB \$B2 ; T CMD NOW LIKE USR
FFD0: EF	626		DFB \$EF
FFD1: C4	627		DFB \$C4
FFD2: B2	628		DFB \$B2 ; S CMD NOW LIKE USR
FFD3: A9	629		DFB \$A9
FFD4: BB	630		DFB \$BB
FFD5: A6	631		DFB \$A6
FFD6: A4	632		DFB \$A4
FFD7: 06	633		DFB \$06
FFD8: 95	634		DFB \$95
FFD9: 07	635		DFB \$07
FFDA: 02	636		DFB \$02
FFDB: 05	637		DFB \$05
FFDC: F0	638		DFB \$F0
FFDD: 00	639		DFB \$00
FFDE: EB	640		DFB \$EB
FFDF: 93	641		DFB \$93
FFE0: A7	642		DFB \$A7
FFE1: C6	643		DFB \$C6
FFE2: 99	644		DFB \$99
FFE3: B2	645	SUBTBL	DFB \$B2
FFE4: C9	646		DFB \$C9
FFE5: BE	647		DFB \$BE
FFE6: C1	648		DFB \$C1
FFE7: 35	649		DFB \$35
FFE8: 8C	650		DFB \$8C
FFE9: C4	651		DFB \$C4
FFEA: 96	652		DFB \$96
FFEB: AF	653		DFB \$AF
FFEC: 17	654		DFB \$17
FFED: 17	655		DFB \$17
FFEE: 28	656		DFB \$28
FFEF: 1F	657		DFB \$1F
FFF0: 83	658		DFB \$83
FFF1: 7F	659		DFB \$7F
FFF2: 5D	660		DFB \$5D
FFF3: CC	661		DFB \$CC
FFF4: B5	662		DFB \$B5
FFF5: FC	663		DFB \$FC
FFF6: 17	664		DFB \$17
FFF7: 17	665		DFB \$17
FFF8: F5	666		DFB \$F5
FFF9: 03	667		DFB \$03
FFFA: FB 03	668		DW NMI
FFFC: 62 FA	669		DW RESET
FFFE: 40 FA	670		DW IRQ

ENDASM

*** SUCCESSFUL ASSEMBLY: NO ERRORS

0000	LOCO	0001	LOC1	0020	WNDLFT	0021	WNDWDTH
0022	WNDTOP	0023	WNBDBTM	0024	CH	0025	CV
0026	GBASL	0027	GBASH	0028	BASL	0029	BASH
002A	BAS2L	002B	BAS2H	002C	H2	002C	LMNEM
002D	V2	002D	RMNEM	002E	MASK	002E	CHKSUM
002E	FORMAT	002F	LASTIN	002F	LENGTH	002F	SIGN
0030	COLOR	0031	MODE	0032	INVFLG	0033	PROMPT
0034	YSAV	0035	YSAV1	0036	CSWL	0037	CSWH
003B	KSWL	0039	KSWH	003A	PCL	003B	PCH
003C	A1L	003D	A1H	003E	A2L	003F	A2H
0040	A3L	0041	A3H	0042	A4L	0043	A4H
0044	A5L	0045	A5H	0045	ACC	0046	XREG
0047	YREG	0048	STATUS	0049	SPNT	004E	RNDL
004F	RNDH	0095	PICK	0200	IN	03F0	BRKV
03F2	SDFTEV	03F4	PWRREDUP	03F5	AMPERV	03F8	USRADR
03FB	NMI	03FE	IRGLLOC	0400	LINE1	07F8	MSLOT
C000	IQADR	C000	KBD	C010	KBDSTRB	C020	TAPEOUT
C030	SPKR	C050	TXTCLR	C051	TXTSET	C052	MIXCLR
C053	MIXSET	C054	LDWSCR	C055	HISCR	C056	LDRES
C057	HIRE5	C058	SETANO	C059	CLRANO	C05A	SETAN1
C05B	CLRAN1	C05C	SETAN2	C05D	CLRAN2	C05E	SETAN3
C05F	CLRAN3	C060	TAPEIN	C064	PADDLO	C070	PTRIG
CFFF	CLRRDM	E000	BASIC	E003	BASIC2	F800	PLDT
F80C	RTMASK	F80E	PLOT1	F819	HLINE	F81C	HLINE1
F826	VLINEZ	F828	VLINE	F831	RTS1	F832	CLRSCR
F836	CLRTDP	F83B	CLRSC2	F83C	CLRSC3	F847	GBASCALC
F856	GBCALC	F864	SETCDL	F871	SCRN	F879	SCRN2
F87F	RTMSKZ	F882	INSDS1	F88C	INSDS2	F89B	IEVEN
F8A5	ERR	F8A9	GETFMT	F8BE	MNNDX1	F8C2	MNNDX2
F8C9	MNNDX3	F8D0	INSTDSP	F8D4	PRNTOP	F8DB	PRNTBL
F8F5	NXTCDL	F8F9	PRM2	F910	PRADR1	F914	PRADR2
F926	PRADR3	F92A	PRADR4	F930	PRADR5	F93B	RELADR
F940	PRNTYX	F941	PRNTAX	F944	PRNTX	F948	PRBLNK
F94A	PRBL2	F94C	PRBL3	F953	PCADJ	F954	PCADJ2
F956	PCADJ3	F95C	PCADJ4	F961	RTS2	F962	FMT1
F9A6	FMT2	F9B4	CHAR1	F9BA	CHAR2	F9C0	MNEML
FA00	MNEMR	FA40	IRG	FA4C	BREAK	FA59	OLDBRK
FA62	RESET	FA6F	INITAN	FAB1	NEWMON	FA9B	FIXSEV
FAA3	NOFIX	FAA6	PWRUP	FAA9	SETPG3	FAAB	SETPLP
FABA	SLOOP	FAC7	NXTBYT	FAD7	REGDSP	FADA	RGDSP1
FAE4	RDSP1	FAFD	PWRCON	FB02	DISKID	FB09	TITLE
FB11	XLTLBL	FB19	RTBL	FB1E	PREAD	FB25	PREAD2
FB2E	RTS2D	FB2F	INIT	FB39	SETTXT	FB40	SETGR
FB48	SETWND	FB5B	TABV	FB60	APPLEII	FB65	STITLE
FB6F	SETPWRC	FB78	VIDWAIT	FB88	KBDWAIT	FB94	NDWAIT
FB97	ESCOLD	FB9B	ESCNOW	FBA5	ESCNEW	FBC1	BASCALC
FBDO	BASCLC2	FBD9	BELL1	FBE4	BELL2	FBEF	RTS2B
FBFO	STORADV	FBF4	ADVANCE	FBFC	RTS3	FBFD	VIDOUT
FC10	BS	FC1A	UP	FC22	VTAB	FC24	VTABZ
FC2B	RTS4	FC2C	ESC1	FC42	CLREOP	FC46	CLEDP1
FC5B	HOME	FC62	CR	FC66	LF	FC70	SCROLL
FC76	SCRL1	FC8C	SCRL2	FC95	SCRL3	FC9C	CLREOL
FC9E	CLEOLZ	FCA0	CLEOL2	FCA8	WAIT	FCA9	WAIT2
FCAA	WAIT3	FCB4	NXTA4	FCBA	NXTA1	FCCB	RTS4B
FCC9	HEADR	FCDE	WRBIT	FCEB	ZRDLY	FCE2	ONEDLY
FCE5	WRTAPE	FCEC	RDBYTE	FCEE	RDBYT2	FCFA	RD2BIT
FCFD	RDBIT	FD0C	RDKEY	FD1B	KEYIN	FD21	KEYIN2
FD2F	ESC	FD35	RDCHAR	FD3D	NOTCR	FD5F	NOTCR1
FD62	CANCEL	FD67	GETLNZ	FD6A	GETLN	FD71	BCKSPC
FD75	NXTCHAR	FD7E	CAPTST	FD84	ADDINP	FD8E	CRDUT
FD92	PRA1	FD96	PRYX2	FDA3	XAMB	FDAD	MDBBCHK
FDB3	XAM	FDB6	DATAOUT	FDC5	RTS4C	FDC6	XAMPM

FDD1 ADD	FDDA PRBYTE	FDE3 PRHEX	FDE5 PRHEXZ
FDED COUT	FDF0 COUT1	FDF6 COUTZ	FE00 BL1
FE04 BLANK	FE0B STOR	FE17 RTSS	FE18 SETMODE
FE1D SETMDZ	FE20 LT	FE22 LT2	FE2C MOVE
FE36 VFY	FE58 VFYOK	FE5E LIST	FE63 LIST2
FE75 A1PC	FE78 A1PCLP	FE7F A1PCRTS	FE80 SETINV
FEB4 SETNORM	FE86 SETIFLG	FE89 SETKBD	FE8B INPORT
FE8D INPRT	FE93 SETVID	FE95 OUTPORT	FE97 OUTPRT
FE9B IOPRT	FEA7 IOPRT1	FEA9 IOPRT2	FEBO XBASIC
FEB3 BASCONT	FEB6 GO	FEBF REGZ	FEC2 TRACE
FEC4 STEPZ	FECA USR	FECD WRITE	FED4 WR1
FEED WRBYTE	FEF6 WRBYT2	FEF6 CRMON	FEFD READ
FF0A RD2	FF16 RD3	FF2D PRERR	FF3A BELL
FF3F RESTORE	FF44 RESTR1	FF4A SAVE	FF4C SAV1
FF59 OLDRST	FF65 MON	FF69 MONZ	FF73 NXTITM
FF7A CHRSRCH	FF8A DIG	FF90 NXTBIT	FF98 NXTBAS
FFA2 NXTBS2	FFA7 GETNUM	FFAD NXTCHR	FFBE TOSUB
FFC7 ZMODE	FFCC CHRIBL	FFE3 SUBTBL	

SYMBOL TABLE SIZE
 2589 BYTES USED
 2531 BYTES REMAINING

ERR: SYNTAX

:SLIST4

003D A1H	003C A1L	FE78 A1PCLP	FE75 A1PC
FE7F A1PCRTS	003F A2H	003E A2L	0041 A3H
0040 A3L	0043 A4H	0042 A4L	0045 A5H
0044 A5L	0045 ACC	FD84 ADDINP	FDD1 ADD
FBF4 ADVANCE	03F5 AMPERV	FB60 APPLEII	002B BAS2H
002A BAS2L	FBC1 BASCALC	FBDO BASCLC2	FEB3 BASCONT
0029 BASH	E000 BASIC	E003 BASIC2	002B BASL
FD71 BCKSPC	FBD9 BELL1	FBE4 BELL2	FF3A BELL
FE00 BL1	FE04 BLANK	FA4C BREAK	03FO BRKV
FC10 BS	FD62 CANCEL	FD7E CAPTST	F9B4 CHAR1
F9BA CHAR2	002E CHKSUM	FF7A CHRSRCH	FFCC CRTBL
0024 CH	FCA0 CLEOL2	FC9E CLEOLZ	FC46 CLEOP1
C059 CLRANO	C05B CLRAN1	C05D CLRAN2	C05F CLRAN3
FC9C CLREDL	FC42 CLREDP	CFFF CLRROM	F83B CLRSC2
F83C CLRSC3	F832 CLRSCR	F836 CLRTOP	0030 COLOR
FDED CDUT	FDF0 CDUT1	FDF6 COUTZ	FD8E CRDUT
FC62 CR	FEF6 CRMON	0037 CSWH	0036 CSWL
0025 CV	FDB6 DATAOUT	FF8A DIG	FB02 DISKID
F8A5 ERR	FC2C ESC1	FBA5 ESCNEW	FB9B ESCNOW
FB97 ESCOLD	FD2F ESC	FA9B FIXSEV	F962 FMT1
F9A6 FMT2	002E FORMAT	F847 GBASCALC	0027 GBASH
0026 GBASL	F856 GBCALC	F8A9 GETFMT	FD67 GETLNC
FD6A GETLN	FFA7 GETNUM	FEB6 GO	002C H2
FCC9 HEADR	C057 HIRES	C055 HISCR	F81C HLINE1
F819 HLINE	FC58 HOME	F89B I EVEN	FA6F INITAN
Q200 IN	FB2F INIT	FE8B INPORT	FE8D INPRT
F882 INSDS1	F88C INSDS2	F8D0 INSTDSP	0032 INVFLG
C000 IOADR	FEA7 IOPRT1	FEA9 IOPRT2	FE9B IOPRT
03FE IRQLOC	FA40 IRQ	C010 KBDSTRB	FB8B KBDWAIT
C000 KBD	FD1B KEYIN	FD21 KEYIN2	0039 KSWH
003B KSWL	002F LASTIN	002F LENGTH	FC66 LF
0400 LINE1	FE5E LIST	FE63 LIST2	002C LMNEM
0000 LOCO	0001 LOC1	C056 LORES	C054 LOWSCR
FE22 LT2	FE20 LT	002E MASK	C052 MIXCLR
C053 MIXSET	F9C0 MNEML	FA00 MNEMR	FB8E MNNDX1
F8C2 MNNDX2	F8C9 MNNDX3	FDAD MDDBCHK	0031 MDDE
FF69 MONZ	FF65 MON	FE2C MOVE	07FB MSLDT
FAB1 NEWMON	03FB NMI	FAA3 NOFIX	FD3D NOTCR
FD5F NDTCR1	FB94 NQWAIT	FCBA NXTA1	FCB4 NXTA4
FF98 NXTBAS	FF90 NXTBIT	FFA2 NXTBS2	FACT NXTBYT
FD75 NXTCHAR	FFAD NXTCHR	F8F5 NXTCOL	FF73 NXTITM
FA59 OLDBRK	FF59 OLDRST	FCE2 ONEDLY	FE95 OUTPORT
FE97 OUTPRT	C064 PADDLO	F954 PCADJ2	F953 PCADJ
F956 PCADJ3	F95C PCADJ4	003B PCH	003A PCL
0095 PICK	F80E PLOT1	F800 PLOT	FD92 PRA1
F910 PRADR1	F914 PRADR2	F926 PRADR3	F92A PRADR4
F930 PRADR5	F94A PRBL2	F94C PRBL3	F94B PRBLNK
FDDA PRBYTE	FB1E PREAD	FB25 PREAD2	FF2D PRERR
FDE3 PRHEX	FDE5 PRHEXZ	F8F9 PRMN2	F941 PRNTAX
F8DB PRNTBL	F8D4 PRNTP	F944 PRNTX	F940 PRNTYX
0033 PROMPT	FD96 PRYX2	C070 PTRIG	FAFD PWRCON
03F4 PWREDUP	FAA6 PWRUP	FCFA RD2BIT	FF0A RD2
FF16 RD3	FCFD RDBIT	FCEE RDBYT2	FCEC RDBYTE
FD35 RDCHAR	FD0C RDKEY	FAE4 RDSP1	FEFD READ
FAD7 REGDSP	FEFB REGZ	F938 RELADR	FA62 RESET
FF3F RESTORE	FF44 RESTR1	FADA RGDSP1	002D RMNEM
004F RNDH	004E RNDL	FB19 RTBL	F80C RTMASK
F87F RTMSKZ	FB31 RTS1	FBF9 RTS2B	FB2E RTS2D
F961 RTS2	FBFC RTS3	FCC8 RTS4B	FDC5 RTS4C
FC2B RTS4	FE17 RTS5	FF4C SAV1	FF4A SAVE
FC76 SCRL1	FC8C SCRL2	FC95 SCRL3	F871 SCRN
F879 SCRNR2	FC70 SCROLL	C05B SETANO	C05A SETAN1

C05C SETAN2	C05E SETAN3	F864 SETCOL	FB40 SETGR
FE86 SETIFLG	FE80 SETINV	FE89 SETKBD	FE1D SETMDZ
FE18 SETMODE	FE84 SETNORM	FAA9 SETPG3	FAAB SETPLP
FB6F SETPWRC	FB39 SETTXT	FE93 SETVID	FB4B SETWND
002F SIGN	FABA SLOOP	03F2 SOFTEV	C030 SPKR
0049 SPNT	0048 STATUS	FEC4 STEPZ	FB65 STITLE
FE0B STOR	FBF0 STORADV	FFE3 SUBTBL	FB5B TABV
C060 TAPEIN	C020 TAPEOUT	FB09 TITLE	FFBE TDSUB
FEC2 TRACE	C050 TXTCLR	C051 TXTSET	FC1A UP
FECA USR	03F8 USRADR	002D V2	FE36 VFY
FE58 VFYOK	FBFD VIDOUT	FB78 VIDWAIT	FB26 VLINEZ
F828 VLINE	FC24 VTABZ	FC22 VTAB	FCA9 WAIT2
FCAB WAIT	FCAA WAIT3	0023 WNDBTM	0020 WNDLFT
0022 WNDTOP	0021 WNDWDTH	FED4 WR1	FCD6 WRBIT
FEF7 WRBYT2	FEED WRBYTE	FECD WRITE	FCE5 WRTAPE
FDA3 XAMB	FDC6 XAMPM	FDB3 XAM	FEBO XBASIC
FB11 XLTBL	0046 XREG	0047 YREG	0035 YSAV1
0034 YSAV	FCDB ZERDLY	FFC7 ZMODE	

SYMBOL TABLE SIZE
2589 BYTES USED
2531 BYTES REMAINING



10260 Bandley Drive,
Cupertino, California 95014,
(408) 996-1010