

# Merlin 8/16™

The Complete Macro Assembler System  
For the Apple® IIgs, 128K IIe and IIc

Merlin Utilities

```
Q-Quit
C-Catalog
X-Copy files
T-Type files
L-Lock files
U-Unlock files
D-Delete files
E-Execute files
U-Verify files
P-Rename files
TAB-Menu change
S-Sort directory
F-Change file date
/ -Create directory
N-Show volume names
B-Toggle bell, now ON
P-Toggle prompting, now ON
```

Thursday 9-Feb-89 12:4

Merlin-16r 4.00

Copyright 1988 by Glen Bredon  
Thursday 9-FEB-89 12:36:57 PM

```
C-Catalog
L-Load source
S-Save source
A-Append file
D-Disk command
F-Full screen editor
O-O Save object code
U-Utility menu
Q-Quit
Source: A90040.L0000
```

Prefix: /HARD/MERLIN GS/



TM

Roger Wagner™  
PUBLISHING, INC.

# Merlin 8/16

**Complete Macro Assembler System  
For the Apple IIgs and 128K IIe/IIc**

by Glen Bredon

Manual by Roger Wagner and Tom Burns

Produced by:

**Roger Wagner Publishing, Inc.**

1050 Pioneer Way, Suite P  
El Cajon, California 92020

Customer Service & Technical Support:  
619/442-0522

ISBN 0-927796-28-7  
1M188

## **Customer Licensing Agreement**

The Roger Wagner Publishing, Inc. software product that you have just received from Roger Wagner Publishing, Inc., or one of its authorized dealers, is provided to you subject to the Terms and Conditions of the Software Customer Licensing Agreement. Should you decide that you cannot accept these Terms and Conditions, then you must return your product with all documentation and this License marked "REFUSED" within the 30 day examination period following the receipt of the product.

1. License. Roger Wagner Publishing, Inc. hereby grants you upon your receipt of this product, a nonexclusive license to use the enclosed Roger Wagner Publishing, Inc. product subject to the terms and restrictions set forth in this License Agreement.
2. Copyright. This software product, and its documentation, is copyrighted by Roger Wagner Publishing, Inc. You may not copy or otherwise reproduce the product or any part of it except as expressly permitted in this License.
3. Restrictions on Use and Transfer. The original and any backup copies of this product are intended for your personal use in connection with a single computer. You may not distribute copies of, or any part of, this product without the express written permission of Roger Wagner Publishing, Inc.

### **Limitations of Warranties and Liability**

Roger Wagner Publishing, Inc. and the program author shall have no liability or responsibility to purchaser or any other person or entity with respect to liability, loss or damage caused or alleged to be caused directly or indirectly by this software, including, but not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of this software. 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.

ProDOS and DOS 3.3 are copyrighted programs of Apple Computer, Inc. licensed to Roger Wagner Publishing, Inc. to distribute for use only in combination with Merlin 8/16.

APPLE COMPUTER, INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE.

## **Copyrights**

The Merlin 8/16 documentation and software are copyrighted © 1987 by Roger Wagner Publishing, Inc. This documentation and/or software, or any part thereof, may not be reproduced in any form or by any means, electronic or mechanical, including photocopying, recording, storage in an information retrieval system, or otherwise, without the prior written permission of the publisher.

We have tried our best to give you a quality product at a fair price, and made the software copyable for your personal convenience. Please recommend our product to your friends, but respect our wishes to not make copies for others. Thanks!

Apple, ProDOS, and DOS 3.3 are trademarks of Apple Computer, Inc.

Merlin 8/16 is a trademark of Roger Wagner Publishing, Inc.

## **About the Manual**

This manual was formatted using MacAuthor from Icon Technology, Ltd, 9 Jarrom Street, Leicester LE2 7DH, England, and Apple's LaserWriter printer.

## **OUR GUARANTEE**

This product carries the unconditional guarantee of satisfaction or your money back. Any product may be returned to place of purchase for complete refund or replacement within thirty (30) days of purchase if accompanied by the sales receipt or other proof of purchase.

## ABOUT THE AUTHOR

Glen Bredon is a professor at Rutgers University in New Jersey where he has taught mathematics for over twenty years. He purchased his first computer in 1979 and began exploring its internal operations because "I wanted to know more than my students." The result of this study was Merlin, the first in a series of best selling assemblers (Merlin, Merlin Pro, Merlin 64, and Merlin 128) for the Apple and Commodore personal computers. Glen has also written other utilities including Prosel, the popular ProDOS program selector. A native Californian and concerned environmentalist, Glen spends his summers away from mathematics and computing, preferring the solitude of the Sierra Nevada mountains where he has helped establish wilderness reserves.



---

# MERLIN 8/16

## Table of Contents

### INTRODUCTION

System Requirements . . . . .	1
Suggested Reading . . . . .	2
Making Back-up Copies of Merlin 8/16 . . . . .	3
Personalizing Merlin 8/16 . . . . .	4

### GETTING STARTED WITH MERLIN 8/16

Entering a Source Listing . . . . .	5
Editing a Source Listing . . . . .	8
Assembling a Source Listing . . . . .	9
Saving Programs . . . . .	12
Running Programs . . . . .	14
The Rest of This Manual . . . . .	16

### THE MAIN MENU

The Main Menu . . . . .	17
-------------------------	----

### THE MERLIN 8 EDITOR

Merlin 8 Full Screen Editor Commands . . . . .	23
Merlin 8 Control Key Commands . . . . .	24
Merlin 8 Open Apple Key Commands . . . . .	27
Merlin 8 Editor Command Summary . . . . .	33
Merlin 8 Command Mode . . . . .	36
General Guidelines for the Command Mode . . . . .	36
About the Merlin 8 Editor Documentation . . . . .	36
Command Mode Commands . . . . .	37
Assembling a Merlin 8 File . . . . .	47

### THE MERLIN 16 EDITOR

Merlin 16 Full Screen Editor Commands . . . . .	48
Merlin 16 Control Key Commands . . . . .	48
Merlin 16 Open Apple Key Commands . . . . .	52
Merlin 16 Command Box . . . . .	61
General Guidelines for the Command Box Commands . . . . .	61
About the Command Box Documentation . . . . .	61
Command Box Commands . . . . .	63
Merlin 16 Editor Command Summary . . . . .	70
Assembling a Merlin 16 File . . . . .	73

---

---

## THE ASSEMBLER

The Assembler .....	74
About the Assembler Documentation .....	75
Preliminary Definitions .....	76
Assembler Syntax Conventions .....	77
Source Code Format .....	77
Local Labels .....	78
Variables .....	79
Number Format .....	80
Expressions Allowed by the Assembler .....	81
Immediate Data Syntax .....	83
Addressing Modes .....	83
Special Forced Non-Zero Page Addressing .....	85
Sweet 16 Opcodes .....	85
65C02 and 65802 Opcodes .....	86
Assembler Pseudo Opcode Descriptions .....	87
Formatting Pseudo Ops .....	103
String Data Pseudo Ops .....	107
Data and Storage Allocation Pseudo Ops .....	110
Using Data Tables in Programs .....	113
Conditional Pseudo Ops .....	114
Using Conditional Assembly .....	116
Miscellaneous Pseudo Ops .....	119

## MACROS

Why Macros .....	127
Macro Pseudo Ops .....	127
How a Macro Works .....	128
More About Defining a Macro .....	133
Nested Macros .....	134
Macro Libraries and the USE Pseudo-Op .....	136

## THE LINKERS

The Merlin 8/16 Linkers .....	137
Why a Linker .....	137
About the Linker Documentation .....	138
Pseudo Opcodes for Use with Relocatable Code Files .....	140
The Merlin 8 Linker .....	144
Linker Name Files .....	145
The Linking Process .....	146
Using Merlin 8 Linked Files .....	147
The Merlin 16 Linkers .....	150
The Absolute Linker .....	151
The Linker Command File .....	154
Using Merlin 16 Linked Files .....	155

---



---

Multiple Output Files . . . . .	157
The GS Linker . . . . .	158
Quick Link . . . . .	159
Multiple LNK Input Files . . . . .	160
Using the GS Linker . . . . .	161
Large File GS Linker . . . . .	163

## TECHNICAL INFORMATION

Technical Information . . . . .	164
General Information (DOS 3.3 only) . . . . .	164
General Information (ProDOS and DOS 3.3) . . . . .	165
Symbol Table . . . . .	165
Ultraterm Information . . . . .	165
Memory Allocation with Merlin 8/16 . . . . .	166
Configuring Merlin 8 ProDOS . . . . .	166
Configuring Merlin 8 DOS 3.3 . . . . .	166
Data Descriptions for Merlin 8 Configurations . . . . .	166
64K Merlin and Merlin 8/16 Source Files . . . . .	167
Merlin 8/16 ProDOS Notes . . . . .	168
Transferring Source Files from DOS 3.3 to ProDOS Merlin 8/16 . . . . .	169
Merlin 8/16 and Speed Up Cards . . . . .	169
Merlin 8 and Corvus Hard Disks . . . . .	170
Configuring Merlin 16 . . . . .	170
The Merlin 16 Params File . . . . .	170
Merlin 8/16 Memory Maps . . . . .	176

## ERROR MESSAGES

Error Messages . . . . .	179
--------------------------	-----

## SOURCEROR

Using Sourceror on Merlin 16 . . . . .	184
Using Sourceror on Merlin 8 . . . . .	184
65C02 Opcodes on Older IIe/IIc Computers . . . . .	186
Disassembly Commands . . . . .	186
Merlin 16 Sourceror Notes . . . . .	188
Sourceror XL . . . . .	188
Final Processing . . . . .	190
Modifying the Finished Source . . . . .	190
The Memory Full Message . . . . .	191
Changing Sourceror's Label Tables . . . . .	191

## SOURCEROR.FP

Applesoft Source Listing . . . . .	192
Sourceror.FP . . . . .	192
Printing the Applesoft Source Listing . . . . .	193

---

---

Applesoft Source Cross Reference Listing . . . . .	195
----------------------------------------------------	-----

## UTILITY PROGRAMS

Auto Edit . . . . .	197
Classic Desk Accessories . . . . .	197
Calendar - Notepad - Rational Calculator . . . . .	197
Clock . . . . .	197
Conv Lnk Rel . . . . .	198
Converter: APW source to Merlin . . . . .	198
Using Macgen . . . . .	200
Cross Reference Programs . . . . .	205
Xref/XrefA . . . . .	205
Using Xref . . . . .	206
XrefA Notes . . . . .	208
Formatter . . . . .	208
Keyboard Macro Files . . . . .	209
Edmac - Keymac . . . . .	209
Keyboard Macro Equivalent Chart . . . . .	210
Make Dump . . . . .	211
Printfiler . . . . .	211
Printfiler Applications . . . . .	212
Using Printfiler . . . . .	212
Changing Printfiler Options . . . . .	213
Txted . . . . .	214
Type.Changer . . . . .	214
Additional Merlin 8/16 Resource Files . . . . .	215

## INDEX

Index . . . . .	216
-----------------	-----

---

## MERLIN 8/16

Merlin 8/16 is an extremely powerful, comprehensive Macro Assembler system for the Apple IIgs or 128K IIe/IIc. It consists of four main modules and numerous auxiliary and utility programs which comprise one of the most complete assembler systems available for *any* personal computer. Merlin 8/16's four main modules are:

- FILE MANAGEMENT system, for disk I/O, file management, ProDOS interpreter, etc.
- EDITOR system, for writing and editing programs with word-processor-like power.
- ASSEMBLER system, with Macros, Macro libraries, conditional assembly, linked files, etc.
- LINKER system, for generating relocatable code modules, library routines, run-time packages, etc.

However, Merlin 8/16 is more than just the sum of these four parts. Here are some of the other features offered by Merlin 8/16:

- Merlin 8/16 comes with *three* assemblers: Merlin 16 (ProDOS), Merlin 8 (ProDOS), and Merlin 8 (DOS 3.3). On the Apple IIgs, or Apple IIe or IIc computers with the 65802 or 65816 chip, Merlin 16 assembles programs written for the 6502, 65C02, 65802 and 65816 microprocessors. On the standard 128K IIe or IIc, Merlin 8 assembles programs written for the 6502, 65C02 and 65802 microprocessors.
- Merlin 8/16 is compatible with existing Merlin and Merlin Pro source files.
- Merlin 8/16 recognizes over 50 Pseudo Opcodes for extreme programming flexibility.
- Merlin 8/16 has over 70 commands for ultimate editing and assembling power.
- Merlin 8/16 produces a fully commented, disassembled source listing of Applesoft BASIC.
- Merlin 8/16 comes with Sourceror, a powerful symbolic disassembler to generate Merlin 8/16 source code from binary programs.
- Merlin 8/16 comes with many sample programs, libraries and other aids to get you going with assembly language fast.
- Merlin 8/16 is copyable and hard disk compatible.



## INTRODUCTION

Merlin 8/16 is the most comprehensive macro assembler system for the Apple IIgs or 128K IIe/IIc offering virtually every feature and function that a programmer needs, thus making it unlikely that you'll outgrow it. At the same time, Merlin 8/16's easy built-in editor and fast assemblies make it a pleasure to use whether you're writing a few lines of code or 30,000!

### SYSTEM REQUIREMENTS

To run Merlin 8/16, you'll need at least one disk drive, and one of the following:

- \* Apple IIgs or
- \* Apple IIc or
- \* 128K Apple IIe or
- \* Laser 128 or Laser 128 EX

Merlin 8 will run on all the computer systems listed above; Merlin 16 requires a 65802 or 65816 microprocessor such as in the Apple IIgs, or a modified Apple IIe, IIc or compatible.

Merlin 8/16 supports a wide variety of 80 column display devices including the Apple, the Videx standard and Ultraterm cards, the Checkmate Multi-View, the Applied Engineering Viewmaster 80, and many others.

Merlin 8/16 works with all printers, producing formatted listings with page breaks and titles.

If you're familiar with assembly language programming already, you will find it's easy to adapt to Merlin 8/16. It follows the programming standards of the 65xx family of microprocessors, and its assembler-directed commands, or pseudo-ops, are a super-set of just about every other assembler. That is, assembler directives like HEX, ASC, DS, etc. that you've used in other assemblers are used in Merlin 8/16, and better still, you'll find a new complement of functions to make programming easier. These include assembling directly to or from disk files, multiple data formats for numbers and strings, a complete set of assembler utilities such as cross-referencing and a source code generator (Sourceror), macro capabilities and more.

If you're new to assembly language programming, Merlin 8/16 is the easiest assembler there is. However, the Merlin 8/16 manual does not make any attempt to teach the techniques of assembly language programming itself. Those techniques are covered in various tutorial books available from a number of publishers, including Roger Wagner Publishing. Because everyone has different goals and objectives, you should seek out those books which best match your current needs and experience.

## SUGGESTED READING

Some of the books we recommend include:

- ✓ APPLE IIGS MACHINE LANGUAGE FOR BEGINNERS - Roger Wagner, COMPUTE! Books, Greensboro, NC 27408. Approximately 600 pages of new material, this IIGs assembly language tutorial covers a myriad of subjects such as writing your first routines, calling machine language routines from Applesoft, ProDOS 8 and 16, and concludes with a IIGs drawing program using Quickdraw, scrollable windows, and pull-down menus. The encyclopedic appendix includes examples of all machine language instructions.
- ✓ APPLE IIGS TECHNICAL REFERENCE - Michael Fischer, Osbourne-McGraw-Hill. Filled with relevant technical material, this book is an outstanding resource for the advanced programmer interested software development on the IIGs.  
  
APPLE IIGS TOOLBOX REFERENCE - Apple Computer, Inc. This 2-volume set is a highly technical reference for using the IIGs toolbox. Does not include programming examples or tutorial information.
- ✓ APPLE PRODOS - by Gary Little. Brady Communications Co., Bowie, MD 20715. A good tutorial book on how to write assembly language programs that use ProDOS.  
  
ASSEMBLY LINES: THE BOOK - by Roger Wagner. Roger Wagner Publishing, Inc. El Cajon, CA 92020. A tutorial on assembly language programming designed specifically for the novice. It gives a description of all 6502 instructions, disk access, reading and writing DOS 3.3 files, sound generation, basic math, keyboard and screen techniques and more.  
  
ASSEMBLY COOKBOOK FOR THE APPLE II/IIe - by Don Lancaster. Howard Sams & Co., Indianapolis, IN 46268. This interesting book will give you real insights into Don Lancaster's view of programming theory and practice. A good addition to any library.
- ✓ BENEATH APPLE DOS - by Don Worth and Peter Lechner. Brady Communications Co., Bowie, MD 20715.
- ✓ BENEATH APPLE PRODOS - by Don Worth and Peter Lechner. Brady Communications Co., Bowie, MD 20715. These two books are the classic reference books for learning about the inner workings of DOS 3.3 and ProDOS. A must if you intend to do any programming in this area. More of a reference than a tutorial.  
  
ENHANCING YOUR APPLE II (Volume 1) - by Don Lancaster. Howard Sams & Co., Indianapolis, IN 46268.  
  
ENHANCING YOUR APPLE II AND IIe (Volume 2) - by Don Lancaster. Howard Sams & Co., Indianapolis, IN 46268. These two books are a continuation of Don Lancaster's unique instruction in the art of assembly language programming. More tips here on different kinds of short programs.

INSIDE THE APPLE IIC - by Gary Little. Brady Communications Co., Bowie, MD 20715. An overview of the entire Apple Iic system, from the assembly language programmer's viewpoint.

INSIDE THE APPLE IIGS - Gary Bond, Sybex. Contains a lot of important information but by-passes some of the rules for producing code compatible with future system upgrades.

NOW THAT YOU KNOW APPLE ASSEMBLY LANGUAGE, WHAT CAN YOU DO WITH IT? - by Jules Guildler, Redlig Systems, Inc., 2068 79th Street, Brooklyn, NY 11214. A good applications tutorial with lots of good subroutines for input and output, printer drivers and more. An excellent follow-up book to Assembly Lines: The Book.

PROGRAMMING THE 65816 - David Eyes and Ron Lichty, Prentice Hall Press, New York, NY 10023. In-depth coverage of the differences between the 65816, 6502, 65C02, and 65802 chips and how to best utilize them from a programming standpoint. Also includes a programming tutorial, code samples and a reference section.

✓ THE APPLE IIGS TOOLBOX REVEALED - Danny Goodman, Bantam Computer Books. Details the philosophy of the toolbox and methods of accessing it, but lacks programming examples.

THE ELEMENTARY APPLE IIGS - William B. Sanders, COMPUTE! Books, Greensboro, NC 27408. An introductory book to the Iigs with information on Applesoft, Hi-Res and Super Hi-Res graphics, and sound on the Iigs.

65816/65802 ASSEMBLY LANGUAGE PROGRAMMING - by Michael Fischer. Osborne McGraw-Hill, Berkely, CA 94710. A thorough treatment of the 6502, 65C02, 65802 and 65816 microprocessors. The size of this book (nearly 700 pages) gives you an idea of why this manual for Merlin 8/16 does not attempt to cover programming on the 6500 family microprocessors itself.

## MAKING BACK-UP COPIES OF MERLIN 8/16

The Merlin 8/16 diskettes are unprotected and copies may be made using any copy utility. It is highly recommended that you use *only* the BACK-UP copy of Merlin 8/16 in your daily work, and keep the original in a safe place.

You can copy the Merlin 8/16 diskettes using:

- 1) The COPYA utility program from Apple's System Master Diskette for the DOS 3.3 version of Merlin 8/16.
- 2) The Copy Volume or Duplicate a Diskette function from the ProDOS System Disk for the ProDOS versions of Merlin 8/16, or System Utilities program on the Apple IIGS System Master diskette.

## **PERSONALIZING MERLIN 8/16**

Certain aspects of Merlin 8/16, such as printer line width, page length, default tab positions for the fields in a source listing, turning off the bell sound, etc., can be customized by changing the file **PARMS.S** on the Merlin 8/16 disk, and re-assembling the **PARMS** file. If you would like to change any of these defaults of Merlin 8/16, see the discussion of the **PARMS** file in the Technical Information section of this manual for details on making the changes. For now, though, we recommend you wait until you're more familiar with Merlin 8/16.



## GETTING STARTED WITH MERLIN 8/16

The purpose of this section is not to provide instruction in assembly language programming. Rather, it will show you the entry and running of a short assembly language program to give you an idea of how Merlin 8/16 works.

Many of the Merlin 8/16 commands and functions are very similar in operation. This section does not attempt to present demonstrations of each and every command option. The objective is to present examples of the more common operations, sufficient to get you started writing your own programs using Merlin 8/16. You should not expect to immediately use all of the various commands that Merlin 8/16 supports in your first program. The best approach is to use the Merlin 8/16 manual in an encyclopedia-like fashion, reading just those sections that provide some utility to a current programming task. We suggest that you lightly skim through the manual once, to become aware of generally what the software has to offer, and then return later to specific sections as needed.

### ENTERING A SOURCE LISTING

Now, let's try your first program with Merlin 8/16. Just follow these steps:

1. Start any of the Merlin 8/16 disks (ProDOS or DOS 3.3). A title screen appears, after which the screen changes to the Main Menu. The Main Menu is used for loading and saving files, disk operations, and of course, entering the Merlin 8/16 Editor and Assembler itself.
2. The percent (%) prompt appears at the bottom of the Main Menu. Press F if you are using Merlin 16. If you are using Merlin 8, press E to go the Editor, then A and Return to enter the Add Mode.
3. Since we are entering an entirely new program, a 1 appears at the top right corner of the screen. This number indicates the line the cursor is on in the listing. The 1 and all subsequent line numbers which appear serve roughly the same purpose as line numbers in BASIC except that in assembly source code, line numbers are not referenced for jumps to subroutines or in GOTO-like statements.
4. On line 1, type an asterisk (\*). Entering an asterisk as the first character in any line is similar to a REM statement in BASIC - it tells the assembler this is a remark line and anything after the asterisk is to be ignored. Type the title DEMO PROGRAM 1 after the asterisk and press the Return key.

```
* DEMO PROGRAM 1
```

5. After Return, the cursor moves to the beginning of line 2.
6. An asterisk is not needed to just create a blank line. To create a blank line, with no text following it, press Return again.
7. The cursor once again drops down one line, and a 3 appears at the top right corner of the screen.

8. Press the space bar once and the cursor moves to the next field. Type `ORG`, press space again, type `$8000` and then press Return.

```
* DEMO PROGRAM 1
```

```
    ORG    $8000
```

The above step instructs the assembler to create the following program so that it can run at memory location `$8000`. Merlin 8/16 almost always assembles your program in the same place in memory, but the `ORG` (for Origin) is used to tell Merlin 8/16 where you want the program to eventually be run. This is so that `JMPs`, `JSRs` and other location dependent code within your program is properly written with the final location in mind.

You'll notice that when you press the space bar, Merlin 8/16 automatically moves the cursor to the next field on the line. You'll recall that in assembly language programming, the position of text on each line determines what kind of information it is. Labels for routines and entry points are in the first position. On line 3 you skipped this field by pressing the space bar first, before entering any text. The second position is for the command itself. The command can either be a command such as `LDA`, `RTS`, etc., or it can be a directive to Merlin 8/16 itself, to be used during the assembly to write a file to disk, create a label, call up a macro, or any of Merlin 8/16's many assembler commands.

9. With the cursor at the beginning of line 4, type `BELL` and space to the next field, type `EQU` and space again, then type `$FBDD` and press Return.

```
    BELL  EQU  $FBDD
```

This defines the label `BELL` to be equal to hex `FBDD`. This use of a label is known as an equate or constant. Wherever `BELL` appears in an expression, it will be replaced with `$FBDD`. Why don't we just use `$FBDD`? For one thing, `BELL` is easier to remember than `$FBDD`. Also, if a later assembly required changing the location of `BELL`, all that needs changing is the `EQU` statement, rather than all the other `$FBDD`'s throughout the listing.

10. At the beginning of line 5, type `START` and press space, type `JSR` and space again, type `BELL` and another space, then type a semicolon (;) followed by `RING THE BELL` and then press Return.

```
    START JSR  BELL      ; RING THE BELL
```

Following the opcode is the operand, in this case `BELL`. The operand is the target information of the opcode. Where to `JSR` to, what value to load, etc.

Semicolons are like asterisks, used to mark a comment. Semicolons, however, are used to mark the start of a comment at the end of a line that contains other commands.

11. On line 6 type `DONE` and space, then type `RTS` and press Return.

12. The program has been completely entered. If you wanted to exit the Add Mode, you would press Open-Apple-Q ("Quit"). Since you are not done, *do not exit yet*.

13. The screen should now appear like this:

```
* DEMO PROGRAM 1

      ORG   $8000
BELL   EQU  $FBDD
START  JSR   BELL      ; RING THE BELL
DONE   RTS
```

Note that throughout the entry of this program, each bit of text has been moved to a specific field. Here is a summary of the fields as used so far:

Label	Opcode	Operand	; Comment
START	JSR	BELL	; RING THE BELL

Field One is reserved for labels. START is an example of a label.

Field Two is reserved for opcodes, such as the Merlin 8/16 pseudo-opcodes or directives such as ORG and EQU, and the JSR and RTS opcodes.

Field Three is for operands, such as \$8000, \$FBDD and, in this case, BELL.

Field Four contains comments which are preceded by a semi-colon (;).

It should be apparent from this exercise that it is not necessary to input extra spaces in the source file for formatting purposes, even if these spaces seem to exist in a listing you may be using.

In summary, on each line:

- 1) Do not space for a label. Space once after a label or if there is no label, once at the beginning for the opcode.
- 2) Space once after the opcode for the operand. Space once after the operand for the comment. If there is no operand, type a space and a semicolon for a comment if desired.

## EDITING A SOURCE LISTING

Assuming no errors have been made in the text entered so far, you could now assemble the program entered with Merlin 8/16. Before doing that, however, let's look at the editing abilities of Merlin 8/16.

Editing is the process of making alterations to text that you've already entered, and this ability is one of Merlin 8/16's strong points. In a sense, an assembler is just a word processor for the text that makes up a program. In that light, then, you can judge an assembler in part by how good its editing features are.

Merlin 8/16 has a powerful Full Screen Editor. Powerful in the range of operations possible and, after a little practice, remarkably easy to use. The following text describes how to use the Editor.

You can use the arrow keys to move the cursor anywhere in the listing. There are two types of cursors, the insert and the overstrike. How and when to switch between these two cursor will be explained later. For now, we'll use the insert cursor.

Inserting and deleting lines in the source code are both simple operations. The following example will insert three new lines between the existing lines 5 and 6.

1. Use the arrow key to move to the beginning of line 4 (BELL etc.). Press Return to insert a blank line.
2. Press Return.
3. Press Return again.
4. At line 7, press space once, then type TYA and press Return.
5. The listing should appear as follows:

```
* DEMO PROGRAM 1

      ORG   $8000
BELL  EQU   $FBDD

      TYA
START JSR   BELL      ; RING THE BELL
DONE  RTS
```

The three new lines (5, 6, and 7) have been inserted, and the subsequent original source lines (now lines 8 and 9) have been moved down.