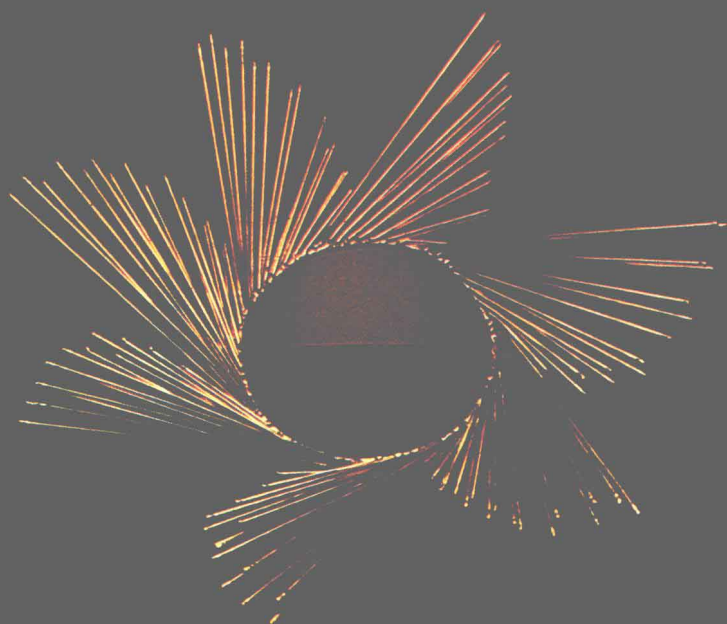


The Waite Group

88 Apple[®] Logo Programs



Michael Waite • Don Martin • Jennifer Martin

88 Apple[®] Logo Programs

by

**Donald Martin
Jennifer Ann Martin**

Howard W. Sams & Co., Inc.
4300 WEST 62ND ST., INDIANAPOLIS, INDIANA 46268 USA

88 Apple® Logo Programs

The **Waite Group, Inc.** is a San Rafael, California based producer of high-quality books on personal computing. Acknowledged as a leader in the industry, the Waite Group has written and produced over thirty titles, including such best sellers as *UNIX™ Primer Plus*, *Computer Graphics Primer*, *CP/M Primer*, *CP/M Bible* and *Soul of CP/M*. Internationally known and award winning, Waite Group books are distributed worldwide, and have been repackaged with the products of such major companies as Epson, Wang, Xerox, Tandy Radio-Shack, NCR, and Exxon Office Systems. Mr. Mitchell Waite, President of the Waite Group, has been involved in the computer industry since 1976 when he bought his first Apple I computer from Steven Jobs.



Donald Martin has served as chairman of the Physics, Astronomy and Energy Science Department at the College of Marin in Kentfield, CA. He received his B.A. from the University of California, Berkeley and his M.A. from San Jose State University. He has long been interested in the problems that students have in developing their reasoning and critical thinking skills. Recently, this interest has led him to the Logo language, a course he now teaches at the college. Mr. Martin is co-author of *UNIX™ Primer Plus*, *C Primer Plus* and *Logo Programming Primer*. His hobbies include reading, running and traveling with his wife, Kay, and family.



Jennifer Ann Martin is a graduate from California Polytechnic in San Luis Obispo, CA, with a B.A. in economics and mathematics. She is currently working with the Engineering Computer Applications Department at Pacific Gas and Electric. Her career interests include financial management and operations research. She finds programming in Logo to be programmer efficient and enjoyable. Her hobbies include reading, tennis and the beach.



Preface

Logo is the new natural computer language, widely heralded as the language most likely to replace BASIC. Logo has a good-natured, friendly structure, and exciting “turtle graphics.” These and other features have quickly allowed Logo to attain the status as today’s best language for beginning computer users of any age.

However, since Logo is so easy for beginners, Logo is often mistaken as a language just for children. Nothing could be further from the truth: Logo has such a rich, open-ended, modular structure that all kinds of programs, from business to utility, from educational to entertainment, from text to graphics, can be written. And Logo’s procedural structure automatically prepares you for more complex languages like FORTH, Pascal or C.

Here, you will find 88 fully tested, debugged, and ready to run canned Logo programs. There are common programs for business and home use, such as simple data base and graphing packages, as well as games, special turtle-graphics programs, a turtle-copter, lunar lander, and many useful utilities. Each utility is presented with a “driver” program to show how it can be used.

A wonderful educational program, called Matchmaker, is included. Worth the price of the book itself, this program lets you enter ANY question and test for ANY answer, and it’s all done interactively. You can use this program to teach children simple math, or for yourself, learn foreign languages, states and capitals, chemistry formulas, biological families or any subject where you might like help in memorizing specific information.

This book, along with its companion, *Apple Logo Programming Primer*, are the only books that take special care to provide you with these five proven learning aids:

- special focus on recursion and outputs
- formal, consistent definition of primitives
- box-chart approach to structured programming
- generous use of utilities to build complex programs
- clear, concise, diagrammatic explanation of Logo syntax

To use this book best, you should have a basic understanding of Logo's syntax and primitives. We recommend that beginning computer users examine the book, *Apple Logo Programming Primer*, which gives a complete introduction to the language.

However, computer users familiar with other languages can simply load in the program examples and run them. You will quickly gain sufficient knowledge to modify the programs to suit your tastes.

We think you will find 88 *Apple Logo Programs* a clear demonstration of the power and fun that awaits users of this new language.

DONALD MARTIN

JENNIFER MARTIN

Acknowledgments

The “spheres of influence” of friends and associates seem to expand outwards, like ripples on a pond. It is not always easy to recognize individual contributions as the cumulative effects of these wavelets wash over us daily. And it is not possible to thank professionally the hundreds of persons who have made wide-ranging contributions towards our writing efforts and the creation of this book. However, we would like to take this opportunity to thank Dr. Lois Flynn, who brought Brian Harvey to San Francisco State University in the summer of 1981 and who made it possible for Brian to bring Logo to a few enthusiastic teachers. Just as important, Brian also provided a version of Logo to run under UNIX at the College Of Marin in Kentfield, California, planting the seed that flowered into this book.

Thanks also to Jon Foreman, who keeps UNIX and Logo running smoothly at the College Of Marin and special thanks to Bob Peterson, Bernd Enders, Fred Schmitt, Nancy Zimfirescu, and Dick Rodgers, who keep everything else running smoothly, too.

We also wish to thank Greg Martin and Paul Swyers for suggesting programs and improvements, Steve Cross for assembling the Command Summary, Susan Martin for her patient typing, Dr. Stephen Prata and Dr. Marijane Axtell Paulson, co-authors of *Apple Logo Programming Primer*, for their many ideas and discussions during the development of these two books.

Many, many students contributed ideas, enthusiasm and positive feedback during the germination period of the past two years. Their contributions are reflected throughout the book. Specifically, special thanks to Martin Parlan, Toby Rein, Jason Schwager, Larry Jacobson, Jim Karas, Robert Riddel, Mike Maxwell, Tony Fardella, Ray Gruenig, Frank Adamson, Laurie McDaniel, and Cal Astrin.

Bob Johnson deserves all our appreciation for his fantastic cartoons.

A very big thank you to Judith Greisgraber, whose keen appreciation of Logo and of teaching strategies contributed greatly to the explanations used in this book. Judith also made many valuable suggestions in her review of the manuscript.

Likewise, Michael Pardee and Kim House of the Waite Group were helpful in many ways.

Finally, we'd like to publicly thank our family and friends for their many sacrifices and loving support at all times.

If any misconceptions or errors are found in this book, please do not fault our friends and associates, it is our doing.

© 1984 The Waite Group, Inc.

FIRST EDITION

SECOND PRINTING—1984

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

International Standard Book Number: 0-672-22343-0

Library of Congress Catalog Card Number: 84-50181

Edited by *Jim Rounds*

Printed in the United States of America.

Contents

CHAPTER 1

INTRODUCTION TO LOGO	13
Why This Book?—What Is Logo?—How to Read a Logo Program—The Secret to Good Program Writing—Three Stages of Programmer Development—The Magic of Recursion—How to Use This Book	

CHAPTER 2

UTILITIES FOR WRITING INTERACTIVE PROGRAMS	35
Introduction—Interactive Program Writing—Error Trapping— Summary of Utilities—How to Assemble Your Utility Disk— 1. GETA.WORD—2. GETA.LIST—3. GET.KEY— 4. GETB.NUMBER—5. GETB.INTEGER—6. GETB.WORD— 7. GETB.LIST—8. GETB.YN—9. GETC.NUMBER— 10. GETC.LIST	

CHAPTER 3

UTILITIES FOR TEXT AND SCREEN MANIPULATION	55
Counting and Comparing Utilities—11. COUNT.W:WORD— 12. COUNT.C:LIST—13. COUNT.S:LIST—14. CHARP :CHAR:WORD—Text Formatting Utilities—15. CENTER :TEXT:YPOS—16. SP:X—17. SC:X:Y—18. TYPEF:FIELD :TEXT—19. SCROLL:X:Y:TEXT—20. PAUSE.ANY—21. PAUSE.RETURN—List Processing Utilities—22. RANPICK :LIST—23. REMOVE.O:OBJECT:LIST—24. INVERSE.O :OBJECT:LIST	

CHAPTER 4

MATH UTILITIES AND MATH PROGRAMS	75
Arithmetic Functions—25. SQR :X—26. POWER :BASE :EXP—	
27. FACTORIAL :N—Statistical Functions—28. AVERAGE	
:DATA—29. MAX :X :Y—30. MIN :X :Y—31. HI :DATA—	
32. LO :DATA—Trigonometric Functions—33. TAN :ANGLE—	
34. ARCSIN :X—35. PI—Miscellaneous Utilities—36.	
ROUND OFF :NUM :PLACES—37. SUMLIST :NUMLIST—	
38. ABS :X—Math Programs—39. STATISTICS—40.	
CURVE.FITTING—41. SERIES	

CHAPTER 5

EXERCISING THE TURTLE	109
Simple Graphics Programs—42. SPI.MENU—43. CIRCLE	
:RADIUS—44. ARCR :RADIUS—45. KITE :X :TAIL :NUM—	
46. OJO :SIZE—47. THREE.D—48. POLAR—49.	
MULTITANGLE :HEIGHT :LENGTH—50. LUNAR.ECLIPSE	
:RADIUS—51. SKETCH.N.SAVE—Graphics Using Complex	
Recursion—52. TREE :SIZE—53. NEST.TRI :SIZE—54.	
NEST.POLY :SIZE :NUMBER :DEPTH—55. ELLIPSE :SIZE—	
56. FRACTAL :SIZE :LEVEL—57. SNOW :SIZE :LEVEL—58.	
HILBERT :SIZE :LEVEL :INVERT—59. CARPET	

CHAPTER 6

SHORT GAMES	163
Logo Programming Techniques—60. ROLLEM—	
61. GUESS.NUMBER—62. REFLEX—63. BLACKJACK	

CHAPTER 7

A LONG GAME	181
How the Program Works—64. JOTTO	

CHAPTER 8

MATCHMAKER	199
How Matchmaker Works—65. MM.CREATE—	
66. MM.CHANGE—67. MM.PLAY	

CHAPTER 9

GRAPHMAKER	233
68. FUNCTION—69. POINTS—70. HISTOGRAM	

CHAPTER 10

BASEMAKER	259
71. DB.CREATE—72. DB.DISPLAY—73. DB.SEARCH—	
74. DB.CHANGE	

CHAPTER 11

MOTION.MAKER: THE TURTLE TAKES OFF	311
The Laws of Motion—Damped Harmonic Oscillator—	
75. SPRING—Flying The Heliturtle—76. FLY.H—Landing the	
Shuttle—77. FLY.S—Lunar Orbiter and Lander—78. FLY.O	

CHAPTER 12

MISCELLANEOUS	345
More Utilities—79. RING :N—80. BUZZ :N—	
81. STORE.DATA—Program Sampler—82. SORT—83. MODEL	
:RABGRO :FOXGRO—84. GROWTH—Business and Finance—	
85. INVEST—86. INTEREST—87. PAYMENTS—For	
Children—88. CHILDS.PLAY	

APPENDIX A

A SUMMARY OF LOGO COMMANDS	384
----------------------------------	-----

APPENDIX B

ASCII TABLE	405
Numerical Conversion—ASCII Definitions	

APPENDIX C

MANAGING YOUR WORKSPACE	412
-------------------------------	-----

APPENDIX D

THE “88 LOGO PROGRAMS” DISK	414
An Overview—Making Backup Copies—Running Programs From	
a Disk—The Workspace—Troubleshooting—Reporting Errors	

APPENDIX E

POSTSCRIPT—MAKING IMPROVEMENTS	420
--------------------------------------	-----

88 Apple Logo Programs

<i>No.</i>	<i>Program</i>	<i>Page</i>	<i>No.</i>	<i>Program</i>	<i>Page</i>
1.	GETA.WORD	40	29.	MAX :X :Y	81
2.	GETA.LIST	41	30.	MIN :X :Y	81
3.	GET.KEY	42	31.	HI :DATA	82
4.	GETB.NUMBER	43	32.	LO :DATA	83
5.	GETB.INTEGER	45	33.	TAN :ANGLE	83
6.	GETB.WORD	47	34.	ARCSIN :X	84
7.	GETB.LIST	48	35.	PI	85
8.	GETB.YN	49	36.	ROUND OFF :NUM	
9.	GETC.NUMBER	51		:PLACES	85
10.	GETC.LIST	52	37.	SUMLIST	
11.	COUNT.W :WORD	57		:NUMLIST	87
12.	COUNT.C :LIST	58	38.	ABS :X	87
13.	COUNT.S :LIST	59	39.	STATISTICS	88
14.	CHARP :CHAR		40.	CURVE.FITTING	92
	:WORD	61	41.	SERIES	101
15.	CENTER:TEXT		42.	SPI.MENU	111
	:YPOS	62	43.	CIRCLE :RADIUS	114
16.	SP :X	64	44.	ARCR :RADIUS	115
17.	SC :X :Y	65	45.	KITE :X :TAIL	
18.	TYPEF :FIELD			:NUM	116
	:TEXT	66	46.	OJO :SIZE	119
19.	SCROLL :X :Y		47.	THREE.D	122
	:TEXT	67	48.	POLAR	126
20.	PAUSE.ANY	68	49.	MULTITANGLE	
21.	PAUSE.RETURN	69		:HEIGHT :LENGTH	130
22.	RANPICK :LIST	70	50.	LUNAR.ECLIPSE	
23.	REMOVE.O			:RADIUS	132
	:OBJECT :LIST	71	51.	SKETCH.N.SAVE	134
24.	INVERSE.O		52.	TREE :SIZE	141
	:OBJECT :LIST	72	53.	NEST.TRI :SIZE	144
25.	SQR. :X	77	54.	NEST.POLY :SIZE	
26.	POWER :BASE			:NUMBER :DEPTH	146
	:EXP	78	55.	ELLIPSE :SIZE	149
27.	FACTORIAL :N	79	56.	FRACTAL :SIZE	
28.	AVERAGE :DATA	80		:LEVEL	151

<i>No.</i>	<i>Program</i>	<i>Page</i>	<i>No.</i>	<i>Program</i>	<i>Page</i>
57.	SNOW :SIZE		73.	DB.SEARCH	281
	:LEVEL	152	74.	DB.CHANGE	291
58.	HILBERT :SIZE		75.	SPRING	314
	:LEVEL :INVERT	154	76.	FLY.H	321
59.	CARPET	155	77.	FLY.S	328
60.	ROLLEM	164	78.	FLY.O	336
61.	GUESS.NUMBER	167	79.	RING :N	346
62.	REFLEX	170	80.	BUZZ :N	348
63.	BLACKJACK	172	81.	STORE.DATA	348
64.	JOTTO	183	82.	SORT	355
65.	MM.CREATE	203	83.	MODEL :RABGRO	
66.	MM.CHANGE	212		:FOXGRO	360
67.	MM.PLAY	224	84.	GROWTH	364
68.	FUNCTION	235	85.	INVEST	366
69.	POINTS	242	86.	INTEREST	368
70.	HISTOGRAM	251	87.	PAYMENTS	374
71.	DB.CREATE	263	88.	CHILDS.PLAY	377
72.	DB.DISPLAY	270			

Introduction to Logo

In this chapter, you will find:

- Why This Book?
- What Is Logo?
- How to Read a Logo Program
- The Secret to Good Program Writing
- Three Stages of Programmer Development
 - Exploration and Discovery
 - Use of Variables in Simple Procedures
 - Top-Down Design of Programs
- The Magic of Recursion
- How to Use This Book

1 INTRODUCTION TO LOGO

Why This Book?

Today, we are in the midst of an information explosion propelled by tremendous advances in computer hardware and software. Just as computer hardware, the “nuts and bolts” of a computer system, has been decreasing in cost while increasing in performance, so too has computer software become easier to use and more powerful.

Logo is one example of this evolving software. It is designed to give beginners control over sophisticated computing resources that can be used as tools for learning, playing and exploring. It is possible with Logo to write quite simple programs that do very complex things, especially using “turtle-graphics.”

In addition, Logo has most of the features of a good programming language, so learning Logo teaches good programming habits. In fact, a popular microcomputer journal recently recommended that Logo replace BASIC as the first programming language for beginning microcomputer users.

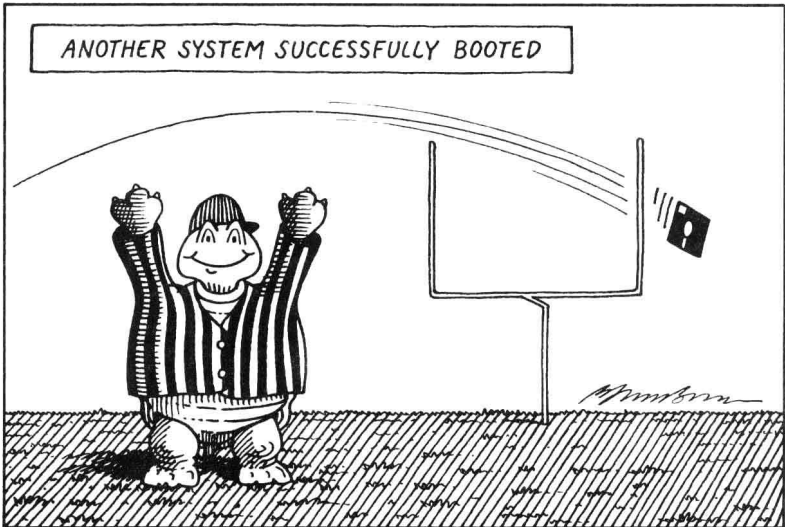
88 Apple Logo Programs has been written for three major reasons: to demonstrate, to stimulate and to educate. Our first goal is to demonstrate the wide range of programs that can be written using Logo. By doing this, we hope these programs will motivate and stimulate you into wanting to try them out and to explore making changes in them. Then, as you work through the programs, you will learn the language and learn good programming techniques.

Most of all, we hope this book will give you the tools for a lifetime of rewarding and enjoyable programming with Logo and the world of computers.

What Is Logo?

Logo has been described as being both a computer language and a philosophy of learning. The major theme of Logo’s philosophy is to learn by doing, to learn by trial and error, to learn by exploring and discovering. Errors do not reflect failure, but “bugs” that can be analyzed and fixed.

The language itself was designed to reflect this philosophy by being



easy to use and powerful enough for sophisticated applications. Logo achieves these two seemingly opposing goals by incorporating several major design characteristics.

The most important Logo feature is its ability to respond to new user-created words or programs in the same way Logo responds to its built-in words. Every computer language has certain “reserved” words that the computer knows and responds to. In Logo, these key words are called “primitives”, and include such commands as, MAKE, FIRST, AND, IF, REPEAT, FORWARD and RIGHT. For example, the command

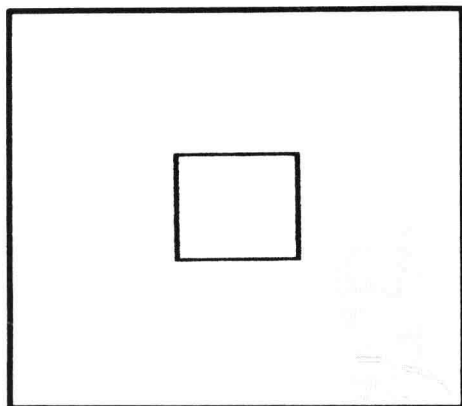
```
FORWARD 50
```

will cause the “turtle” to draw a line 50 steps long on the screen, while the command

```
RIGHT 90
```

will turn the turtle 90 degrees. We can create some rather interesting programs using these primitives. Here is one of the easiest programs to visualize.


```
TO BOX  
REPEAT 4 [FORWARD 50 RIGHT 90]  
END
```



The key point is that we can run this program, called a “procedure” in Logo, just by typing its name, the same as we run a Logo primitive. We can even include this newly defined procedure in other procedures. For example, here is a simple procedure that uses our BOX procedure.

```
TO TURNBOX  
REPEAT 20 [BOX RIGHT 18]  
END
```

