

Contents

Introduction	ix
About the projects. About this series. How to read this book. Versions of Logo.	

Acknowledgments	xv
---------------------------	----

I Cryptography

1 Cryptographer's Helper	1
------------------------------------	---

This project is a tool to assist the user in solving cryptograms (simple substitution ciphers). It makes some guesses automatically and records other guesses made by the user.

Program structure and modularity. Flag variables. Guided tour of global variables. Stacks. ASCII character codes. Computed variable names. Long instruction lines. Further explorations. Program listing.

2 Playfair Cipher	35
-----------------------------	----

This project demonstrates the use of a more complex cipher, in which a given letter in the original text is not always represented by the same letter in the coded version.

Data redundancy. Composition of functions. LOCAL vs. subprocedures. Multiple end tests. Stacks. Tail recursion. Conversational front end. Further explorations. Program listing.

II Games

3 Tic-Tac-Toe	61
-------------------------	----

This program plays the game of tic-tac-toe against the user. It follows a perfect strategy, so it always wins or ties.

Program structure and modularity. Predicates with side effects. Data representation. Variables in the workspace. Flag variables. Computing an instruction. Indirect variable assignment. Computing two things at once. Implementing the strategy rules. Further explorations. Program listing.

4 Solitaire 97

This project deals a simulated deck of cards and allows the user to play a game of solitaire. The program checks for legal moves and displays the board layout.

Program structure and modularity. Machine dependencies and portability. Data abstraction. Stacks. Program as data. Shuffling the deck. Exclusive or. Multiple branching. Further explorations. Program listing.

III Mathematics

5 Fourier Series Plotter 137

This program displays graphs of the Fourier (harmonics of sine waves) approximation to a square wave. It illustrates how harmonics are combined to control the timbres of notes in computer-generated music.

Keyword inputs. Indirect assignment. Evaluation environments. Numeric precision. Dynamic scope. Further explorations. Program listing.

6 Pitcher Problem Solver 157

This program solves the category of problems in which you are given pitchers of certain sizes and asked to use them to measure a specified amount of water from a river.

Tree search. Depth-first and breadth-first searching. Data representation. SENTENCE as a combiner. Finding the children of a node. Computing a new state. Efficiency: what really matters?. Stopping the program early. Avoiding meaningless pourings. Eliminating duplicate states. Further explorations. Program listing.

IV Programming Utilities

7 Pretty Print 185

This program prints a complex list structure in a readable way, using indentation to indicate the hierarchy of sublists.

Iteration over a list. Mutual recursion. Tree traversal. Global and local variables. Output and effect. Implementing the formatting rules. Further explorations. Program listing.

8 Iteration Compiler 201

This project modifies a Logo program that uses mapping and iteration procedures, replacing those general-purpose tools with individual recursive procedures that can run faster. It allows the Logo programmer to think in a more powerful way while programming, without giving up efficiency.

Procedure patterns. Parsing a logo instruction. Generated symbols. `TEXT` and `DEFINE`. Compiling the instructions. Defining the auxiliary procedures. Combining `SENTENCE` and `LIST`. Generic `COMBINE`. Avoiding long instructions: `ADDLINE`. Avoiding long instructions: table driven compilation. Further explorations. Program listing.

V Pattern Matching

9 Pattern Matcher 237

This program looks for patterns in a list. It can be used, for example, to recognize particular types of English sentences.

Reinventing `EQUALP` for lists. A simple pattern matcher. Efficiency and elegance. Logo's evaluation of inputs. Indirect assignment. Defaults. Program as data. Parsing rules. Predicates with side effects. Further explorations. Program listing.

10 DOCTOR 269

This project uses the pattern matcher to carry out a conversation with the user, pretending to be a psychiatrist. It is a Logo version of a program originally written by Joseph Weizenbaum.

ELIZA and artificial intelligence. ELIZA's linguistic strategy. Stimulus-response psychology. Property lists. Generated symbols. Modification of list structure. Linguistic structure. Data in workspace files. Further explorations. Program listing: SETUP. Program listing: DOCTOR.

Appendix A: Versions of Logo 307

Index 313