MICRO
COMPUTERS
SOFTWARE

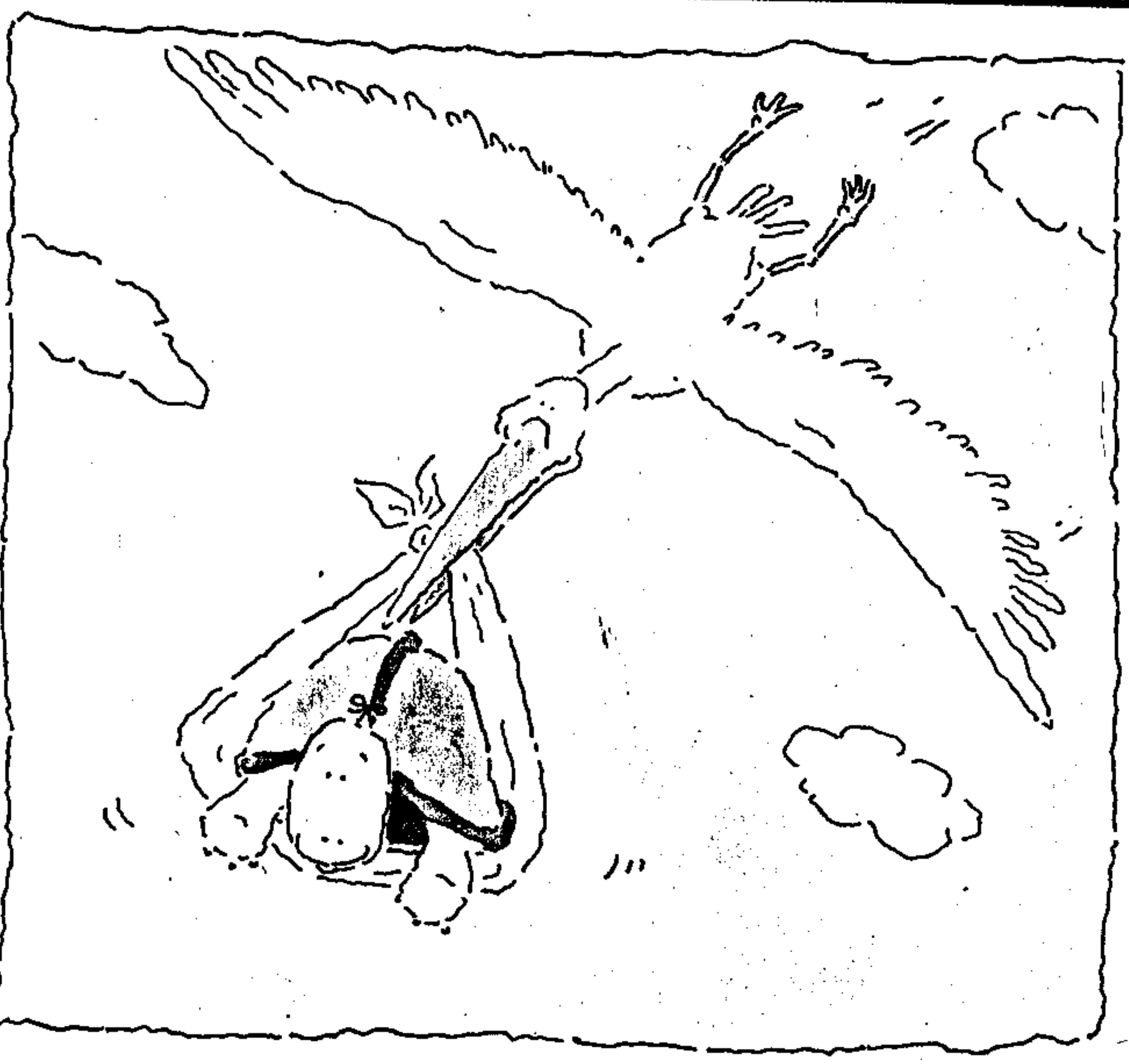
MCT0033

TINY LOGO

CASSETTE

FOR THE TEXAS INSTRUMENTS 99/4A HOME COMPUTER

Requires the use of a cassette tape recorder for loading program.



Contents

Welcome to Tiny Logo

Tiny Logo was invented by a teacher who wanted his students to have the benefits of the Logo language without having to buy expensive computer peripherals and software. It is an interpretation of the Logo languages incorporating many of the programming features of regular Logo, while only using the TI-99/4A console with its 16K memory. **Tiny Logo** requires nothing more to run than your TI-99/4A, a tv set or monitor, and a cassette recorder.

Like regular Logo, **Tiny Logo** is a computer language that uses the extensive graphic capabilities of your TI-99/4A to teach the fundamentals of programming. Although it was designed for school children to enjoy and learn with, anyone from preschooler to adult can enjoy drawing in color with the **Tiny Logo** turtle and learn about programming while having fun.

Like other computer languages, **Tiny Logo** can be used in either an immediate mode, where the turtle will execute your commands one at a time as you type them in, or in the programming or procedure mode.

This booklet has all the information you need to draw and program with the **Tiny Logo** turtle. Follow the instructions in *Getting Started* and you will soon be on your way.

Important

A box (⬓) is used to represent the turtle (≮) key throughout this booklet. Whenever it appears in a procedure, press the turtle (≮) key.

SECTION 1

Getting Started

Loading the Tiny Logo Program

There are no special loading instructions for the **Tiny Logo** cassette. Type in OLD CS1 and follow the instructions on your screen. The blinking cursor will appear after the DATA OK message. When you see the cursor type RUN and press ENTER. **Tiny Logo** should now be ready to use. If you have any problems loading your **Tiny Logo** cassette consult your TI-99/4A *User's Reference Guide*.

Diskette Saving and Loading Instructions

If you have a disk drive, you can save your **Tiny Logo** cassette program on diskette. Follow the standard instructions in your TI-99/4A User's Reference Guide for saving programs on diskette.

To make sure **Tiny Logo** runs properly from diskette it is necessary to type the following sequence of loading instructions:

CALL FILES (1) *press* ENTER
NEW *press* ENTER

Now type in:
OLD DSK1.
TINYLOGO *press* ENTER

Talking to the Turtle

After loading the **Tiny Logo** program, you will see the title screen and then a green board with the **Tiny Logo** turtle in the center. Just outside the board's lower left corner, there will be a small white box and the word "HELLO", along with a dash. The box shows the color that the turtle will draw in. The dash under the greeting "HELLO" is where you will type instructions to the turtle. You can talk to the turtle with the keyboard.

When you type in an instruction and press ENTER the turtle reads the instruction. If everything is in order, the turtle will do what you command. If the turtle finds an error, it will report it. When the turtle has completed your instruction, it will report "DONE" at the lower left corner of the screen. After each report the turtle will wait for your next instruction. If you want the turtle to stop performing an instruction press the period "." until the turtle reports: "STOPPED".

It is important to remember never to use a space between the turtle sign ■ and your instruction. Also, never use more than one space between any two words or parts of your instruction. The turtle will not accept your instructions if you have not followed the above rules and will report an error.

Special Keys

Before we begin to draw with the turtle, there are some special keys to become familiar with.

< , **The comma** is used to type the turtle ■ sign.

SHIFT & 7 **The ampersand** is used instead of the plus sign "+" for addition. (The "+" in ± will not work).

FCTN ! 1 While holding down the function key, press 1 to erase any errors you might make when typing your instructions.

> **The period** is used to stop the turtle while performing any of your instructions, and can also be used to indicate the end of a turtle procedure.

Drawing with the Turtle

There are fifteen commands you can use to direct the turtle to draw, move, make sounds, or change colors. We will now use some of them to get you started using **Tiny Logo**. Type in ■MOVE 3 (use the comma "<" key to type the turtle ■ sign). Now press enter and watch what happens. The turtle has moved 3 spaces up the green board and has drawn a short white line. Note that the square at the bottom left of the screen is the same color as the line.

Let's instruct the turtle to go into its shell. Type ■IN and press ENTER. Now type ■MOVE 3 again (don't forget to press ENTER after every turtle instruction). What happened? The turtle pulled its legs into its shell and advanced 3 steps without drawing. Use ■IN whenever you want to position the turtle on the board without drawing a line.

Now type in ■HOME. This instruction moves the turtle back to the center of the board with the turtle in its shell. To draw, we must bring the turtle back out of the shell. Simply type ■OUT. This command makes the turtle's legs reappear and when instructed to move, it will once again leave a trail.

We will now draw three more lines on the screen and make a square. Each side of the square will be a different color. Type in the following commands, and remember to press ENTER after each one.

■TURN RIGHT
■TAKE RED
■MOVE 3
■TURN LEFT
■TAKE YELLOW
■MOVE 3
■TURN LEFT
■TAKE BLUE
■MOVE 2
■CORNER

If you accurately typed in the above instructions and remembered to press ENTER after each line, you should now see a square with red, white, yellow and blue sides. The turtle will now be in its shell in the lower left corner of the board. You may notice that the turtle, when given the command ■MOVE, always goes in the direction that it is pointing. The dimensions of the turtle's board are twenty-two spaces from top to bottom, and twenty-nine spaces from left to right.

After having finished this example, you now know how to do the following things:

1. Make the turtle move (■MOVE).
2. Make the turtle turn (■TURN).
3. Put the turtle into its shell so it will not draw (■IN).
4. Make the turtle come out of its shell, ready to draw (■OUT).
5. Make the turtle choose a color to draw in (■TAKE).
6. Send the turtle home to the center of the board in its shell (■HOME).
7. Send the turtle to the lower left corner of the screen, in its shell (■CORNER).

The turtle can draw in seven colors. They are white, black, red, blue, green, yellow and violet. Use the command ■TAKE followed by a space and the name of the color you choose.

What happens when the turtle draws in green? To find out, type in the following:

■MOVE 6
■TURN EAST
■TAKE GREEN
■OUT
■MOVE 10

Don't forget to press ENTER after each command. The turtle moved, but did he really draw in green? Let's see. Type in the command ■BOARD BLACK. As you can see, the turtle actually did draw in green, but the line is invisible on the green board. You can use this method to erase lines that you want to take out of your drawing. Simply go over them with the turtle drawing in the same color as the board. There is one more turtle instruction you should know before trying the many examples given in this booklet. First type in ■HOME and press ENTER. Now type in the new command ■CLEAR and once again press ENTER. You should now see a black board without any drawing on it. The turtle is in its shell in the HOME position.

You can change the color of the board to be any of the same seven colors that the turtle draws in. Choose the color that you like best and type in ■BOARD followed by a space and the color. Before going on to the sections on turtle procedures and programming, try experimenting with the turtle using the commands you have just learned.

SECTION 2

Turtle Procedures

A **Tiny Logo** turtle procedure is a list of instructions for the turtle to perform. Turtle procedures are very much like small computer programs, and are written in the LEARNing mode. (See the definition of the LEARN command for step-by-step instructions.)

In the LEARNing mode, the turtle will not perform instructions as they are typed in. Instead, your entire set of instructions will be given a name. Only when that name is typed in will the turtle perform the complete sequence of instructions.

Writing procedures for the turtle to LEARN can be extremely challenging and fascinating. The following list of ten procedures has been carefully created as an introduction. Explanatory notes accompany some of them to help you understand the procedures as well as expand your general knowledge of programming in **Tiny Logo**.

The procedures presented become progressively more complex. By the time you have finished typing in and studying them, you will have been exposed to all of the most important facets of programming in **Tiny Logo**.

Additional turtle procedure examples are given in the back of the booklet for you to learn from and enjoy. All the information you need to program your turtle is in this booklet. Before using any commands to write procedures, read about them in the command section. Have patience, use your imagination, and you will soon be creating unique turtle procedures of your own.

Tiny Logo Procedure Examples

1. SQUARE *Simple Linear Procedure*
2. BOX :SIZE *Linear Procedure with One Variable*
3. EASYSQUARE *Calling a Procedure Using REPEAT*
4. SIMPLESQUARE :SIZE *Passing a Variable to a Called Procedure*
5. FLOWER :L :S :PCOL :CCOL *Many Procedures With Multiple Variables*
6. JOURNEY *Simple Recursion Involving One Decision*
7. SPIRAL :LENGTH *Simple Recursion With an Incremented Variable*
8. BOXGROW :SIZE *Complex Recursive Procedure With an Incremented Variable*
9. DUPLICATE *Recursion, Multiple Procedures, Decisions and Incremented Variables*
10. COIL :N *Complex Recursive Procedure*

Important

A box (■) is used to represent the turtle (≺) key throughout this booklet. Whenever it appears in a procedure, press the turtle (≺) key.

SQUARE
Simple Linear Procedure

```
SQUARE
CLEAR
HOME
OUT
MOVE 6
TURN
MOVE 6
TURN
MOVE 6
TURN
MOVE 6
CORNER
```

The first three commands clear the screen and put the turtle in postion to draw. The last command removes the turtle from what has been drawn.

To call this and all turtle procedures, type the procedure's name and press ENTER. Do not use the turtle sign (), when calling procedures.

BOX :SIZE
Linear Procedure With One Variable

```
BOX :SIZE
CLEAR
HOME
OUT
MOVE :SIZE
TURN
MOVE :SIZE
TURN
MOVE :SIZE
TURN
MOVE :SIZE
CORNER
```

Choose a number to determine the size of the BOX. Turtle procedures with variables will not run without being given a value for the variable. (For example, to call this procedure, type in BOX 6, BOX 4, or BOX 13).

EASYSQUARE
Calling a Procedure Using REPEAT

```
EASYSQUARE
CLEAR
HOME
OUT
REPEAT 4 SIDE
CORNER

SIDE
MOVE
TURN
```

Note that the procedure EASYSQUARE calls a second procedure named SIDE.

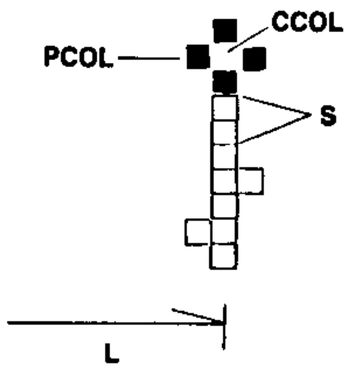
SIMPLESQUARE :SIZE
Passing a Variable to a Called Procedure

```
SIMPLESQUARE :SIZE
CLEAR
HOME
OUT
REPEAT 4 (EDGE :SIZE)
CORNER

EDGE :SIZE
MOVE :SIZE
TURN
```


To call this procedure, type in SIMPLESQUARE and a number. Note that this number will be passed on to the procedure called EDGE.

FLOWER :L :S :PCOL :CCOL
Many Procedures With Multiple Variables



L = Distance across screen
S = Stem size
PCOL = Petal Color
CCOL = Center Color

```
FLOWER :L :S :PCOL :CCOL
CORNER
TURN
MOVE :L
TURN LEFT
TAKE GREEN
STALK
STEM :S
MOVE 2
CENTER :CCOL
MOVE
PETALS :PCOL
CORNER
```

```
STEM :S
MOVE :S
IN
```

```
CENTER :COLOR
TAKE :COLOR
OUT
IN
```

```
PETALS :COLOR
TAKE :COLOR
TURN
REPEAT 4 PETAL
```

```
PETAL
OUT
IN
MOVE
TURN
MOVE
```

```
STALK
OUT
MOVE
TURN LEFT
MOVE
TURN BACK
MOVE
TURN LEFT
MOVE 2
TURN
MOVE
TURN BACK
MOVE
TURN
MOVE
```

The FLOWER procedure is a complex set of procedures. Although it may look confusing at first, it does not use anything you have not already learned. It

just uses more of it. FLOWER is an excellent example of building a complex procedure.

Here is an example of how to call this procedure. You change any of the colors or numbers.

FLOWER 9 5 YELLOW RED

If you use the colors we selected, type in BOARD BLACK.

You can draw more than one flower on the screen. Simply call FLOWER again and assign your own color and number values.

JOURNEY
Simple Recursion Involving One Decision

```
JOURNEY
MOVE
IF ONEDGE (TURN BACK)
JOURNEY
```

Before calling this procedure, enter the following command:

HOME

A procedure which calls itself is recursive. In the JOURNEY procedure, the turtle will move one step, then decide whether or not it is on the edge of the board. If it is, it will reverse its direction. If it is not, it will not change direction. In either case, the turtle will follow the recursive instruction on the third line of this procedure and begin its JOURNEY all over again. This type of procedure results in a continuous repetition of the instructions and will not stop until the Tiny Logo program has reached its memory limit. The message "I CAN'T CONTINUE. SORRY" will appear at the bottom of the screen when this occurs.

SPIRAL :LENGTH
Simple Recursion With an Incremented Variable

```
SPIRAL :LENGTH
MOVE :LENGTH
TURN
SPIRAL :LENGTH+1
```

Before calling this procedure, enter the following commands:

```
HOME
CLEAR
OUT
```

Note: to type "+", use the ampersand key (SHIFT 6).

BOXGROW :SIZE
Complex Recursive Procedure With an Incremented Variable

```
BOXGROW :SIZE
BOX :SIZE
IN
TURN BACK
MOVE 2
TURN
MOVE 2
TURN
OUT
BOXGROW :SIZE+4

BOX :SIZE
REPEAT 4 (PERIM :SIZE)

PERIM :LENGTH
MOVE :LENGTH
TURN
```


Before calling this procedure, enter the following commands :

```
CORNER
OUT
```

DUPLICATE
Recursion, Multiple Procedures, Decisions and Incremented Variables

```
DUPLICATE
MOVE
TURN
CHECK 1
TURN
IF INSIDE DUPLICATE
IF ONEDGE (TURN BACK)
MOVE

CHECK :N
MOVE
IF COLORED (COPY :N)
IF INSIDE (CHECK :N+1)
IF ONEDGE (TURN BACK)
MOVE

COPY :N
TURN BACK
REPEAT 2 (MOVE :N)
OUT
TURN BACK
IN
REPEAT 2 (MOVE :N)
```

Before calling and executing this procedure, carry out the following instructions:

```
CLEAR
```

Draw a line or simple shape on the bottom right quadrant of the BOARD.

```
CORNER
TURN
MOVE 14
TURN LEFT
```

COIL :N
Complex Recursive Procedure

```
COIL :N
MOVE
IF INSIDE COIL :N+1
TURN
MOVE :N
```

Before calling this procedure, enter the following commands:

```
CLEAR
HOME
OUT
```

SECTION 3

Commands

General Commands

■MOVE [number of steps]
■TURN [direction]
■TAKE [color]
■IN
■OUT
■HOME
■CORNER
■BOARD [color]
■CLEAR
■SOUND [sound code number]
■REPEAT [number <instruction>]
■IF [condition <instruction>]

Procedure Commands

■LEARN
■EDIT
■LIST

■MOVE [numeric expression]

The turtle always moves in the direction it is pointing. The numeric expression follows the command and tells the turtle the number of steps to take. If no numeric expression is given, the turtle will move one step.

Examples

■MOVE (*the turtle will move one step*)
■MOVE 4

This command can also be used with a variable or a variable plus a numeric expression as shown below.

■MOVE :LENGTH
■MOVE :LENGTH+3

Errors

If you instruct the turtle to move beyond the edge of the board by using too large a numeric expression with the ■MOVE command, the turtle will stop when it reaches the end of the board, and the message "STOPPED" will be displayed.

■TURN [direction]

The turtle can MOVE in four directions. If no direction follows the command ■TURN, the turtle will turn right. There are eight expressions that can be used to turn the turtle. They are:

■TURN *(turtle turns right)*
■TURN RIGHT
■TURN LEFT
■TURN BACK
■TURN NORTH
■TURN SOUTH
■TURN EAST
■TURN WEST

The ■TURN command can also be used with a variable:

■TURN :DIRECTION

■ = Turtle (⊰) Key

■TAKE [color]

The turtle can draw in any one of the seven colors listed below. To use a specific color, type it in after the command ■TAKE. If a color is not specified, the turtle will draw in a color selected at random.

The square at the lower left corner of the screen always shows the turtle's current color selection.

■TAKE *(turtle takes random color)*
■TAKE WHITE
■TAKE BLACK
■TAKE RED
■TAKE GREEN
■TAKE YELLOW
■TAKE BLUE
■TAKE VIOLET

The command ■TAKE can also be used with a variable:

■TAKE :COLOR

■IN

When the turtle is given the command ■IN, it pulls its legs into its shell and will not draw when it MOVEs.

■OUT

When the turtle is given the command ■OUT, it will come out of its shell and draw in place and whenever it MOVEs.

■HOME

The command ■HOME instructs the turtle to go to the center of the BOARD, pointing north and IN its shell. You must use the command ■OUT to draw from the HOME position.

■CORNER

The command ■CORNER instructs the turtle to go to the lower left corner of the board, pointing north and IN its shell. You must use the command ■OUT to draw from the CORNER position.

■BOARD [color]*

Use the command ■BOARD to change the turtle's drawing board to any of the seven colors listed below.

■BOARD WHITE
■BOARD BLACK
■BOARD RED
■BOARD GREEN
■BOARD YELLOW
■BOARD BLUE
■BOARD VIOLET

The command ■BOARD can also be used with a variable:

■BOARD :COLOR

*The ■BOARD command must always be used with a color.

■CLEAR

The command ■CLEAR clears the board of anything previously drawn. When the command is given the turtle will go in its shell and remain in place.

SOUND [numeric expression]

Use the command SOUND to make the turtle sing a note. A numeric expression following this command determines the pitch of the tone. When given the command SOUND or SOUND 0 the turtle will sing low C (131 Hz). Each step of one in the numeric expression following the SOUND command results in a sound one half tone higher, up to SOUND 99 which is the highest tone and cannot be heard. The duration of all notes is fixed at 0.5 seconds.

Examples

SOUND	*(low C)*
SOUND 0	*(low C)*
SOUND 12	*(middle C)*
SOUND 24	*(high C)*
SOUND 19	*(middle G)*
SOUND 99	*(highest tone—cannot be heard)*

The command SOUND can also be used with a variable or a variable plus a numeric expression:

SOUND :TONE
SOUND :OCTAVE+7 *(G in variable octave)*

REPEAT [numeric expression <instruction>]*

The command REPEAT is used when you want the turtle to perform the same instruction more than once. It is always used with a numeric expression telling the turtle how many times to repeat the instruction, followed by an expression that tells the turtle which instruction to perform.

Examples

REPEAT 9 <SOUND 12>
REPEAT 6 <MOVE 4>
REPEAT 3 <TURN LEFT>

A variable may be used with this command in both the numeric and instructing expressions:

REPEAT 9 SOUND
REPEAT 2 :INSTRUCTION
REPEAT :TIMES <SOUND 12>
REPEAT 4 <SIDE :SIZE>

Errors

If either the numeric expression or the instructing expression is missing, the turtle will report an error.

Example

```
REPEAT 5
```
Report: REPEAT WHAT? 'REPEAT 5'

If an incorrect value follows an instruction, the turtle will report an error.

```
REPEAT 4 <SOUND RED>
```
Report: BAD VALUE: 'SOUND RED'

*Brackets (<>) must be used whenever an instruction has more than one part.

IF [condition <instructing expression>]*

The IF command is always followed by a condition and an instruction to be performed if that condition is true. This command can be used with any one of the four conditions listed below:

IF INSIDE MOVE
IF ONEDGE <TURN BACK>
IF COLORED <SOUND 12>
IF EMPTY <FLOWER RED>

Additional examples of condition statements.

IF INSIDE <IF EMPTY FLOWER>
IF INSIDE <IF ONEDGE DRAW>
(*thus DRAW is never performed!*)

Conditions

INSIDE	The turtle is INSIDE whenever it is anywhere on the board and not touching any of the four edges.
ONEDGE	The turtle is ONEDGE whenever it rests on any of the outermost squares that frame the board.
COLORED	Any area that the turtle has visited while OUT of his shell is colored. Although the turtle's trail is not visible when it draws in the same color as the board, the area covered is still considered to be colored.
EMPTY	An area is EMPTY if the turtle has either not visited it or visited it only when IN its shell.

Errors

If you are either missing a condition or an instructing expression, the turtle will report an error:

```
IF OUTSIDE SOUND
```
Report: BAD CONDITION: 'IF OUTSIDE SOUND'

If an incorrect name or number follows a command or instruction, the turtle will report an error:

```
IF INSIDE <TAKE 5>
```
Report: BAD COLOR: 'TAKE 5'

*Brackets (<>) must be used whenever an instruction has more than one part.

LEARN

The command LEARN is used whenever you want to teach the turtle a procedure. After typing in the command LEARN and pressing ENTER, the turtle asks for the name you want to give the procedure and a variable list (if there is one) with the message: "WHAT". If you have already used the name for some other procedure, the turtle will respond with an error report: "YOU TAUGHT ME THAT."

The turtle can LEARN no more than fifteen procedures. If you begin typing in a sixteenth procedure, the turtle will respond with the error report: "I CAN LEARN NO MORE." If everything is in order, the name of your procedure and variable list will be printed on the screen. When using variables, always double check that you have included all of them and that they appear in the right order. Continue entering your procedure by typing each command or instruction and then pressing ENTER.

The turtle will accept no more than fourteen instructions for any one procedure and will automatically print "<END>." Use the period "." to end the procedure.

If you are still working on a line you want to change and have not yet pressed ENTER, you can use the FCTN/1 key to erase and then retype. If you want to make any changes after the complete procedure or any of its lines have been ENTERed, follow the instructions given for the EDIT command.

EDIT

The EDIT command is used to display a procedure on the screen in order to make changes or corrections. After typ-

ing ■EDIT and pressing ENTER, the message "WHAT?" will appear, asking you to type in the procedure's NAME. If you make a mistake and type in the name incorrectly or if the name you enter has not been used for a procedure, the turtle will report "I DIDN'T LEARN [procedure name]."

If a procedure has already been compiled (learned by the turtle) the EDIT mode will list the procedure and variable list, followed by each of the procedure's instructions. You will be able to review the entire procedure before making any changes. If the procedure has not yet been compiled, the EDIT mode will begin with the message "NEW NAME:". If you want to change the procedure's name or any of the variables, type in the new information and then press ENTER. If no name change is required, just press ENTER and the EDIT mode will display the first line of your procedure for you to either change as just described or to move on to the next line.

Summary of EDITing procedures

If you want to:

Edit or change a line, type in your changes and press ENTER.

Leave a line as it is, press ENTER.

Delete a line, type the minus sign "-", then your new line, and then press ENTER. The new line will take the number of the line last listed. For example, if you type "/" when line 4 is listed, your line will become the new number 4, and the old line 4 becomes line 5.

Exit the editing mode without changing the remaining instructions, type the equal sign " = " and then press ENTER.

End the procedure, type period "." and then press ENTER.

As is the case when writing a new procedure, the EDIT mode will allow no procedure to be more than fourteen lines long, excluding the period or end statement. After the fourteenth instruction is typed in, editing will be automatically terminated, and the fourteen instructions displayed on the screen will comprise the entire procedure.

■LIST

This command is used when you want a list of the names of procedures already learned. The turtle reports each procedure name on a separate line, followed by a list of all variables that might appear in the procedure.

If no procedures have been learned by the turtle, the turtle will report "NOTHING TO LIST!".

SECTION 4

Definitions

Command

Refer to the *Commands* section of this booklet for definitions of each of the **Command names** listed below.
■MOVE, ■TURN, ■TAKE, ■IN, ■OUT
■HOME, ■CORNER, ■BOARD, ■CLEAR
■SOUND
■REPEAT, ■IF
■LEARN, ■EDIT, ■LIST.

Command Errors:

If an invalid Command name is used, an error is reported.

Example
Spelling mistake:

Error: ■HAME
Report: BAD COMMAND: '■HAME'

Color

Any of the following seven colors may be used:

WHITE
BLACK
RED
GREEN
YELLOW
BLUE
VIOLET

or a variable whose value is one of the above.

Color Errors:

If the turtle is looking for a color and finds an error, he reports it.

Examples:
Nonexistent color:

Error: ■TAKE PINK
Report: BAD COLOR: '■TAKE PINK'

Missing color in ■BOARD:

Error: ■BOARD
Report: BAD COLOR: '■BOARD'

Spelling mistake:

Error: ■TAKE BLU
Report: BAD COLOR: '■TAKE BLU'

■ = **Turtle (≺) Key**

Direction

Any of the following is a **direction**:
RIGHT
LEFT
BACK
NORTH
SOUTH
EAST
WEST
or a variable whose value is one of the above.

Direction Errors:

If the turtle is looking for a direction and finds an error, he reports it.

Example
Spelling mistake

Error: TURN SUOTH
Report: BAD DIRECTION: 'TURN SUOTH'

Condition

Any of the following is a **condition**:
COLORED
EMPTY
INSIDE
ONEDGE
or a variable whose value is one of the above.

Condition Errors:

If the turtle is looking for a condition and finds an error, he reports it.

Example
Nonexistent condition:

Error: IF CLEAR SOUND
Report: BAD CONDITION: 'IF CLEAR SOUND'

Number

Number refers to any one of the positive numbers from 0 to 99 inclusive.

Numeric Expression

A **numeric expression** can be any one of the following:

- number
- variable *(The value of a variable must be a number.)*
- *variable + number (The value of a sum* "variable + number" *must be a number.)*

Examples:

17
66
:LENGTH
:TIMES
:LENGTH + 5
:TIMES + 10

Errors in Numeric Expressions:

If the turtle finds an error while looking for a numeric expression, it reports the error.

Examples
A number outside the range of 0 to 99:

Error: MOVE 100
Report: BAD VALUE: 'MOVE 100'

Letter O instead of zero:

Error: MOVE 2O
Report: BAD VALUE: 'MOVE 2O'

A sum outside the range of 0 to 99 (in this example the value of :LENGTH has been set at 60):

Error: MOVE :LENGTH + 43
Report: BAD VALUE 'MOVE :LENGTH + 43'

Procedure

Procedure is a sequence of instructions LEARNed by the turtle. A procedure must be given a name.

Procedure Name

A **procedure name** is any combination of letters and/or numbers used to call a procedure. Hyphens may be used to separate words. Never include spaces in a procedure name.

Examples

EASYSQUARE
EASY-SQUARE
EASYSQUARE2

Procedure Name Errors:

If the turtle is looking for a procedure name and finds a mistake, he reports it.

Example
Bad response to WHAT *in* LEARN:

Error: SQUARE
Report: BAD NAME: 'SQUARE'

Procedure Call

A **procedure call** is one of the following:
procedure name
procedure name and data list

The number of items in a data list must equal the number of variables in the procedure. The first variable in variable list is assigned the first value in data list, the second is assigned the second, etc.

Examples

Procedure name: SQUARE :SIZE :COLOR
Call: SQUARE 4 GREEN
Procedure name: FLOWER :IN-C :OUT-C :HEIGHT
Call: FLOWER RED YELLOW 11

Procedure name: SIDE :LENGTH
Call : REPEAT 4 (SIDE :SIZE)

Procedure Call Errors:

If a procedure is called that has not been LEARNed, an error is reported.

Example
Spelling error:

Error: SQURE 15 GREEN
Report: I DIDN'T LEARN 'SQURE'

If the number of data items in the call is greater or less than the number of variables in the called procedure, the turtle reports an error.

Examples

Procedure name: SQUARE :SIZE :COLOR
Error: SQUARE 4
Report: MISSING DATA: 'SQUARE 4'

Procedure name: SIDE :LENGTH
Error: SIDE 3 15 7
Report: TOO MUCH DATA: 'SIDE 3 15 7'

Variable

The colon ":" followed by a name or combination of letters is a **variable**. The combination cannot include spaces or the plus sign "+", but may include a hyphen.

Examples

:LENGTH
:TIMES
:TONE
:COLOR
:STALK-COLOR
:CONDITION
:INSTRUCTION
:SHAPE
:MELODY
:DIRECTION

Variable List

A **variable list** is a list of variables, each separated from the next by a single space.

Undeclared Variable Error:

If the turtle comes across an undeclared variable while performing an instruction, the error is reported.

Examples
Spelling mistake:

Procedure name: SQUARE :SIZE :COLOR
Error: ■TAKE :COLON
Report: I CAN'T FIND ':COLON'

Undeclared variable:

Procedure name: SQUARE :SIZE
Error: ■TAKE :COLOR
Report: I CAN'T FIND ':COLOR'

Bad Data Error

If a data-list includes an incorrect data item, the turtle will report it as an error.

Procedure name: SQUARE :COLOR
(*contains the instruction* ■TAKE :COLOR)
Error: SQUARE WHIT
Report: BAD COLOR: '■TAKE :COLOR'

Data List

A **data list** is made up of any of the following items:
numeric expression
color
direction
condition
instructing expression

A single space must separate each item in the data list.

Instruction

Any command or procedure call is an **instruction**.

Instructing Expression

The following are *instructing expressions*.
⟨instruction⟩
variable
instruction

An instruction that includes a data-list must be placed between brackets.

Examples

⟨■SOUND 12⟩
⟨SQUARE 4 GREEN⟩
⟨■REPEAT 4 ⟨SIDE :SIZE⟩⟩
⟨■IF INSIDE ⟨■SOUND 12⟩⟩
:INSTRUCTION
:PAINTING
■SOUND
SQUARE
⟨■REPEAT 4 SIDE⟩
⟨■IF INSIDE ■SOUND⟩

Errors in Instructing Expressions:

If the turtle finds an open bracket "⟨" without a closing bracket "⟩" an error is reported.

Example

Error: ■REPEAT 4 ⟨SIDE :SIZE
Report: '⟩' MISSING: '■REPEAT 4 ⟨SIDE :SIZE'

If the turtle is looking for an instructing expression and finds only a variable in brackets, he reports the error.

Example

Error: ■IF INSIDE ⟨:PAINTING⟩
Report: BAD NAME: ':PAINTING'

SECTION 5

Simple Turtle Procedures

SQUARE

```
SQUARE
■MOVE 5
■TURN
■MOVE 5
■TURN
■MOVE 5
■TURN
■MOVE 5
■CORNER
.
```

Before calling and executing this procedure, enter the following commands:

```
■CLEAR
■HOME
■OUT
```

BOX :SIZE

```
BOX :SIZE
■MOVE :SIZE
■TURN
■MOVE :SIZE
■TURN
■MOVE :SIZE
■TURN
■MOVE :SIZE
■TURN
.
```

Before calling and executing this procedure, enter the following commands:

```
■CLEAR
■HOME
■OUT
```

EASYSQUARE

```
EASYSQUARE
■REPEAT 4 SIDE
■CORNER
.
SIDE
■MOVE 4
■TURN
.
```

Before calling and executing this procedure, enter the following commands:

```
■CLEAR
■HOME
■OUT
```

SIMPLESQUARE :SIZE

```
SIMPLESQUARE :SIZE
■REPEAT 4 (SIDE :SIZE)
■CORNER
.
SIDE :SIZE
■MOVE :SIZE
■TURN
.
```

Before calling and executing this procedure, enter the following commands:

```
■CLEAR
■HOME
■OUT
```

■ = Turtle (≺) Key

SQUARE :SIZE :COLOR

```
SQUARE :SIZE :COLOR
■TAKE :COLOR
■REPEAT 4 (SIDE :SIZE)
■CORNER
.
SIDE :SIZE
■MOVE :SIZE
■TURN
.
```

Before calling and executing this procedure, enter the following commands:

```
■CLEAR
■HOME
■OUT
```

FRAME

```
FRAME
■CORNER
■IN
■REPEAT 2 SIDES
.
SIDES
■REPEAT 21 STEP
■TURN
■REPEAT 28 STEP
■TURN
.
STEP
■TAKE
■MOVE
.
```

LOLLIPOP

```
LOLLIPOP :S :IN-C :OUT-C
STICK :S
■MOVE 2
INSIDE :IN-C
■MOVE
■TURN LEFT
■MOVE
■TURN BACK
SHELL :OUT-C
.
STICK :S
■OUT
■MOVE :S
■IN
.
INSIDE :COLOR
■TAKE :COLOR
■TAKE :COLOR
■OUT
■IN
.
SHELL :COLOR
■OUT
■TAKE :COLOR
■REPEAT 4 CANDY
.
CANDY
■MOVE 2
■TURN RIGHT
.
```

Before calling and executing this procedure, type in the following commands:

```
■CLEAR
■CORNER
■TURN
■MOVE 14
■TURN LEFT
```

COLORED-SQUARES

```
COLORED-SQUARES
BEGINNING
■REPEAT 5 SQUARES
.
```

```
SQUARES
■REPEAT 6 SCAN
■TURN LEFT
■MOVE 4
■TURN LEFT
■MOVE 24
■TURN BACK
.

BEGINNING
■CORNER
■MOVE 20
■TURN RIGHT
■MOVE 26
■TURN BACK
.

SCAN
■SQUARE
■IN
■MOVE 4
.

SQUARE
■TAKE
■OUT
■REPEAT 4 SIDE
■MOVE
■TURN LEFT
MIDDLE
PASSAGE
.

MIDDLE
■IN
■TAKE
■MOVE
■OUT
■IN
.

SIDE
■MOVE 2
■TURN LEFT
.
```

```
PASSAGE
■TURN BACK
■MOVE
■TURN RIGHT
■MOVE
■TURN BACK
.
```

VANE :S

```
VANE :S
■REPEAT 8 (LINE :S)
PART :S
■CORNER
.

LINE :S
■OUT
■MOVE :S
■IN
■TURN LEFT
■MOVE
■IF COLORED PASSAGE
■TURN LEFT
.

PASSAGE
■TURN BACK
■MOVE 2
■TURN RIGHT
.

PART :S
■MOVE :S
■TURN LEFT
■MOVE
STRIP
■REPEAT 3 (PIECE :S)
.

STRIP
■OUT
■MOVE 3
■TURN LEFT
■MOVE
```



```
■TURN LEFT
■MOVE 3
■IN
.

PIECE :S
■TURN BACK
■MOVE
■MOVE :S
■TURN LEFT
■MOVE :S
STRIP
.
```

Before calling and executing this procedure, enter the following commands:

```
■CLEAR
■HOME
■OUT
```

FRAME :SIZE :D

```
FRAME :SIZE :D
■REPEAT 4 (SIDE :SIZE :D)
■CORNER
.

SIDE :SIZE :DIRECTION
■OUT
■REPEAT :SIZE STEP
```

```
■TURN :DIRECTION
.

STEP
■TAKE
■MOVE
.
```

Before calling and executing this procedure, enter the following commands:

```
■CLEAR
■CORNER
■OUT
```

SHOW :INSTRUCTION

```
SHOW :INSTRUCTION
■REPEAT 1 :INSTRUCTION
■CORNER
■REPEAT 3 (■BOARD VIOLET)
■REPEAT 3 (■BOARD WHITE)
■REPEAT 3 (■BOARD RED)
■REPEAT 3 (■BOARD BLUE)
■REPEAT 3 (■BOARD YELLOW)
■REPEAT 3 (■BOARD GREEN)
■REPEAT 3 (■BOARD BLACK)
.
```

Try these variations:

```
SHOW COLORED-SQUARES
SHOW FRAME
```

SECTION 6

Recursive Turtle Procedures

SEARCH

```
SEARCH
 MOVE
 IF INSIDE SEARCH
.
```

Before calling and executing this procedure, enter the following commands:

```
 CLEAR
 HOME
 OUT
```

BOOMERANG

```
BOOMERANG
 MOVE
 IF INSIDE BOOMERANG
 IF ONEDGE (TURN BACK)
 MOVE
.
```

Before calling and executing this procedure, enter the following commands:

```
 CLEAR
 HOME
 OUT
```

SCALE :TONE

```
SCALE :TONE
 MOVE
 SOUND :TONE
 IF INSIDE (SCALE :TONE + 1)
 SOUND :TONE
.
```

Before calling and executing this procedure, enter the following commands:

```
 CLEAR
 HOME
 OUT
```

SPIRAL :LENGTH

```
SPIRAL :LENGTH
 MOVE :LENGTH
 TURN
SPIRAL :LENGTH + 1
.
```

Before calling and executing this procedure, enter the following commands:

```
 CLEAR
 HOME
 OUT
```

COIL :N

```
COIL :N
 MOVE
 IF INSIDE (COIL :N + 1)
 TURN
 MOVE :N
.
```

■ = Turtle (≺) Key

Before calling and executing this procedure, enter the following commands:

```
 CLEAR
 HOME
 OUT
```

GROWBOX :SIZE

```
GROWBOX :SIZE
SQUARE :SIZE
GROWBOX :SIZE + 2
.

SQUARE :SIZE
 REPEAT 4 (SIDE :SIZE)
.

SIDE :LENGTH
 MOVE :LENGTH
 TURN
.
```

Before calling and executing this procedure, enter the following commands:

```
 CLEAR
 CORNER
 OUT
```

BOXOUT :SIZE

```
BOXOUT :SIZE
SQUARE :SIZES
 IN
 TURN BACK
 MOVE 2
 TURN
 MOVE 2
 TURN
 OUT
BOXOUT :SIZE + 4
.
```

```
SQUARE :SIZE
 REPEAT 4 (SIDE :SIZE)
.

SIDE :LENGTH
 MOVE :LENGTH
 TURN
.
```

Before calling and executing this procedure, enter the following commands:

```
 CLEAR
 CORNER
 OUT
```

DUPLICATE

```
DUPLICATE
 MOVE
 TURN
CHECK 1
 TURN
 IF INSIDE DUPLICATE
 IF ONEDGE ( TURN BACK)
 MOVE
.

CHECK :N
 MOVE
 IF COLORED (COPY :N)
 IF INSIDE (CHECK :N + 1)
 IF ONEDGE ( TURN BACK)
 MOVE
.

COPY :N
 TURN BACK
 MOVE :N
 MOVE :N
 OUT
 TURN BACK
 IN
```

```
■MOVE :N
■MOVE :N
.
```

Before calling and executing this procedure, carry out the following instructions:

```
■CLEAR
```
Draw a line or simple shape on the bottom right quadrant of the BOARD.
```
■CORNER
■TURN
■MOVE 14
■TURN LEFT
```

BOXIN :N

```
BOXIN :N
■MOVE
■IF COLORED (BOXIN :N + 4)
■IF EMPTY ■TURN
■TAKE
■OUT
SQUARE :N
■IN
■MOVE 2
■TURN
■MOVE 2
■TURN LEFT
■IF EMPTY ■OUT
.
SQUARE :SIZE
■REPEAT AT 4 (SIDE :SIZE)
.
SIDE :LENGTH
■MOVE :LENGTH
■TURN
.
```

Before calling and executing this procedure, enter the following commands:

```
■CLEAR
■CORNER
■OUT
```

PLAY

```
PLAY
SHOW 3
■CORNER
■MOVE 21
■TURN
■MOVE 14
■TURN
■TAKE BLUE
DUPLICATE
■REPEAT 1  CHNG-BRD-CLR
.
SHOW :SQ#
■BOARD BLACK
■CLEAR
■CORNER
■TAKE GREEN
■MOVE 14
■OUT
■MOVE :SQ#
■CORNER
■MOVE 15
BOXIN .
■CORNER
.
CHNG-BRD-CLR
■BOARD BLACK
■BOARD WHITE
.
```

CHAIN :S

```
CHAIN :S
■CLEAR
■CORNER
■OUT
LOOP :S
.
LOOP :S
■REPEAT 4 (SIDE :S)
■MOVE 2
■TURN RIGHT
■MOVE 2
■TURN LEFT
LOOP :S
.
SIDE :S
■MOVE :S
■TURN
.
```

FRAME

```
FRAME
BEGINNING
■REPEAT 4 LINE
.
LINE
COLOR
PASSAGE
.
COLOR
■TAKE
■MOVE
■IF INSIDE COLOR
.
BEGINNING
■CORNER
■MOVE
■TURN
■MOVE
■TURN LEFT
■OUT
.
```

```
PASSAGE
■IN
■TURN BACK
■MOVE
■TURN LEFT
■OUT
.
```

Before calling and executing this procedure, enter the following commands:

```
■CLEAR
■CORNER
■OUT
```

MAZETRIP :N

Before calling and executing this procedure, do the following:

1. Draw random line segments on the BOARD.
2. ■HOME
3. ■TAKE a color that is neither the color of the board nor the color of the line segments you have just drawn.

To call the procedure type in MAZE TRIP 1

```
MAZETRIP :N
■MOVE
■IF COLORED (AVOID :N)
■IF EMPTY DRAW
MAZETRIP :N
.
AVOID :N
■REPEAT :N ■TURN
TRIP :N + 1
.
DRAW
■OUT
■IN
.
```

Summary of Tiny Logo Terms

Commands

MOVE, TURN, IN, OUT
HOME, CORNER, BOARD
TAKE, CLEAR
SOUND
REPEAT, IF
LEARN, EDIT, LIST

Numbers

0 to 99 inclusive (positive numbers only)

Colors	Directions	Conditions
WHITE	RIGHT	INSIDE
BLACK	LEFT	ONEDGE
RED	BACK	COLORED
GREEN	NORTH	EMPTY
YELLOW	SOUTH	
BLUE	EAST	
VIOLET	WEST	

Index of Reports and Error Messages

Report	Meaning
BAD COLOR: 'command'	*see* COLOR.
BAD COMMAND: '. . .	*see* COMMAND.
BAD CONDITION: 'IF . . .'	*see* CONDITION
BAD DIRECTION: 'TURN datum'	*see* DIRECTION
BAD VALUE: 'command'	*see* numeric expression.
DATA MISSING: 'procedure call'	*see* procedure call.
DONE	*Turtle has completed performing the given instruction.*
I CAN LEARN NO MORE	*see* LEARN.
I CAN'T CONTINUE . . . SORRY	*Not enough memory to complete the given instruction.*
I CAN'T FIND 'variable'	*see* variable.
I DIDN'T LEARN 'procedure name'	*see* procedure call *or* EDIT.
NOTHING TO LIST!	*see* LIST.
REPEAT WHAT? 'REPEAT . . .'	*see* REPEAT.
STOPPED	Turtle stopped (*see* MOVE), *or* was stopped by ".".
TOO MUCH DATA: 'procedure call'	*see* procedure call.
YOU TAUGHT ME THAT	*see* LEARN.
'procedure name' USED	*see* EDIT.
'>' MISSING: 'instruction'	*see* instructing expression.

IMPORTANT NOTICE OF DISCLAIMER REGARDING SOFTWARE

Microdistributors International, Inc., does not warrant this software to be free from error or will meet the specific requirements of the end user. The end user or consumer assumes full responsibility for any decisions made or actions taken based on information obtained by using Microcomputers software.

MICRODISTRIBUTORS INTERNATIONAL, INC. EXCLUDES ALL WARRANTIES, BOTH EXPRESSED AND IMPLIED, REGARDING THE WARRANTY OF MERCHANTABILITY AND PERFORMANCE OF ANY SOFTWARE FOR A PARTICULAR PURPOSE. THE SOFTWARE IS MADE AVAILABLE ON AN "AS IS BASIS." THERE ARE NO WARRANTIES WHICH EXTEND BEYOND THE FACE HEREOF.

IN NO EVENT shall Microdistributors International, Inc., be liable to anyone for direct, incidental, or consequential damages in connection with the purchase or use of any Microcomputers software.

Some states have different limitations on implied warranties. The above exclusion may not apply to you in those states.

MICRODISTRIBUTORS INTERNATIONAL, INCORPORATED

34 Maple Avenue/Box 38
Armonk, New York 10504
914-273-6480