



To see how these procedures work, let's see what happens if we type HOUSE 30. LOGO reads the input line and assigns the value 30 to the variable "BIG in HOUSE. The first line of HOUSE is therefore now equivalent to SQUARE 30. The variable "SIZE in SQUARE is, in turn, assigned the value 30. SQUARE is now run, with FD :SIZE becoming FD 30. A similar procedure is followed when TRI is called.

Now try adapting the procedures for drawing the five-by-five board so that BOARD takes the size of the square as an input.

Here's a procedure that draws polygons, with the number of sides given as an input:

```
TO POLY :SIDES
  REPEAT :SIDES [FD 50 RT 360 / :SIDES]
END
```

Using this procedure with one input, POLY 3 will draw a triangle, POLY 4 a square, and so on. However, in all the polygons drawn by this procedure, the sides will be 50 units in length. A more general procedure that draws polygons of any size requires two inputs — one for the number of sides, and one for the required length. To do this, all that is needed is to adapt the POLY procedure to replace the 50 with a variable name and add that name to the title line:

```
TO POLY :SIDES :SIZE
  REPEAT :SIDES [FD :SIZE RT 360 / :SIDES]
END
```

Now POLY 5 30 will draw a pentagon with sides of 30 units in length. You might like to try adapting your new version of the board-drawing procedure so that it will draw any square board (not just five-by-five). There will now be two inputs — the number of squares in each direction, and their size.

GLOBAL VARIABLES

So far we have considered variables that are local to the procedures that use them. But variables may be defined that are available for use by all procedures. These are known as *global* variables and are useful for communicating information between different procedures. However, their use makes debugging more difficult and so they should be used sparingly.

The command MAKE is used to assign values to global variables. MAKE "SIDE 3 assigns 3 as the value of the variable "SIDE. MAKE "SIDE :SIDE + 1 increases the value of "SIDE by one. The exact meaning of the notation in this second example is: find the value of the variable "SIDE, add one, then assign the result back to the variable named "SIDE. In each case, MAKE requires two inputs — the name of the variable, and the value to be assigned to that variable.

To sum up the programming features we have covered in this instalment of the LOGO course, we've designed some procedures for drawing spirals. The main procedure is named EQSPI. This requires three inputs: the initial length of the line to be drawn, the angle that must be turned at each 'corner' of the spiral, and a scale factor by which

the initial length must be multiplied to produce the spiral effect. Different sets of inputs may be used to achieve different effects — we tried 70 283 0.95, 70 143 0.95, and 20 243 1.05. Try other sets of numbers and see what happens.

NOWRAP is a new command. This stops the turtle 'wrapping around' the screen — when the turtle reaches the screen boundary the procedure will stop with an 'out of bounds' error message. In many cases, the wrap-around effect can give interesting results. In this procedure it spoils the spiral effect, so NOWRAP is used to turn it off.

The main procedure EQSPI repeatedly draws a line (the length of which is determined by the scale factor), then turns through a fixed angle, and finally alters the scale factor. The length of the lines drawn either increases or decreases, depending on whether the scale factor is greater than or less than 1. The large number after REPEAT is simply to keep the procedure going for a long time. If you've seen enough, press Control-G (or Break) to stop the procedure running. Most of the variables are local, with the exception of "SCALE. This is global because "GROW changes its value, and this new value must be made available to S.FORWARD. Thus, "SCALE is used to communicate between the two procedures.

Spiral Procedure

```
TO EQSPI :SIZE :ANGLE :FACT
  SETUP
  REPEAT 1000 [S.FORWARD :SIZE RIGHT :ANGLE
    GROW :FACT]
END
TO SETUP
  DRAW NOWRAP MAKE "SCALE 1
END
TO GROW :NUMB
  MAKE "SCALE :SCALE 1 :NUMB
END
TO S.FORWARD :DIST
  FORWARD :SCALE 1 :DIST
END
```

Logo Flavours

LCSI versions use the command FENCE rather than NOWRAP to stop automatic wrapping.

The Atari version doesn't have FENCE, so use WINDOW instead. This stops the turtle from wrapping around, but, unlike FENCE, it doesn't halt the procedure when the turtle reaches the screen boundary.

In the Spiral procedure on the Spectrum, replace DRAW in the SETUP subprocedure with CS.

Procedure Problems 2

- 1) Write a procedure to draw a circle of radius 50. Modify the procedure so that the radius is given as an input to the procedure.
- 2) Write a procedure that draws a 'target' consisting of five concentric circles.