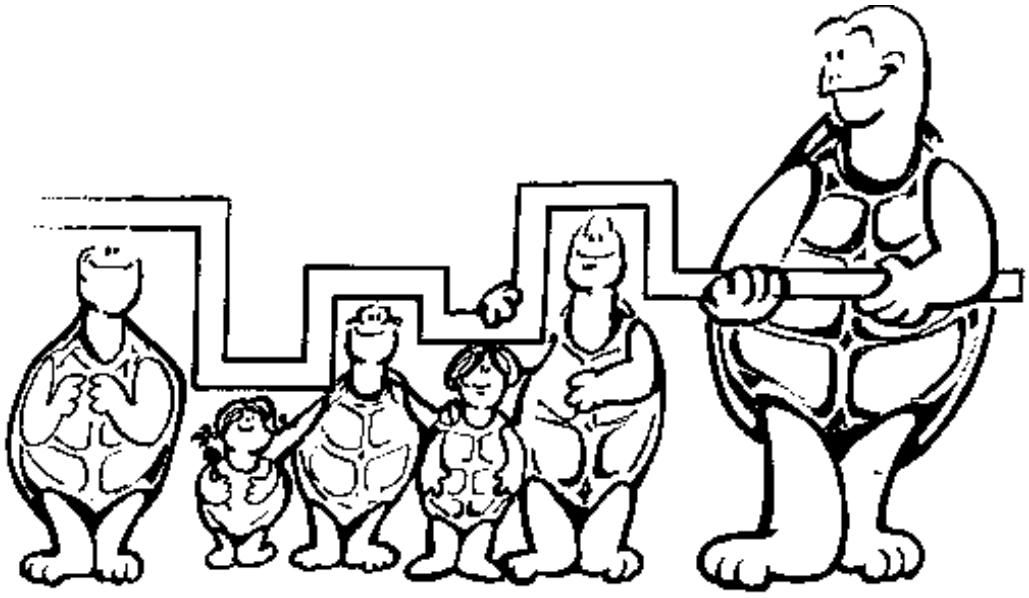
Chapter 5. Varying Variables

How many people are in your family? In your class? Are they all the same? Or are they Variable?



Got one in on you, didn't I? There's that word, "Variable."

How many of your friends have the same color hair? The same color eyes? How many are the same age? How many were born in the same month as you? Anyone born on the same day?

How many things about you and your group are the same? How many are "variable."

Now let's talk about Logo.

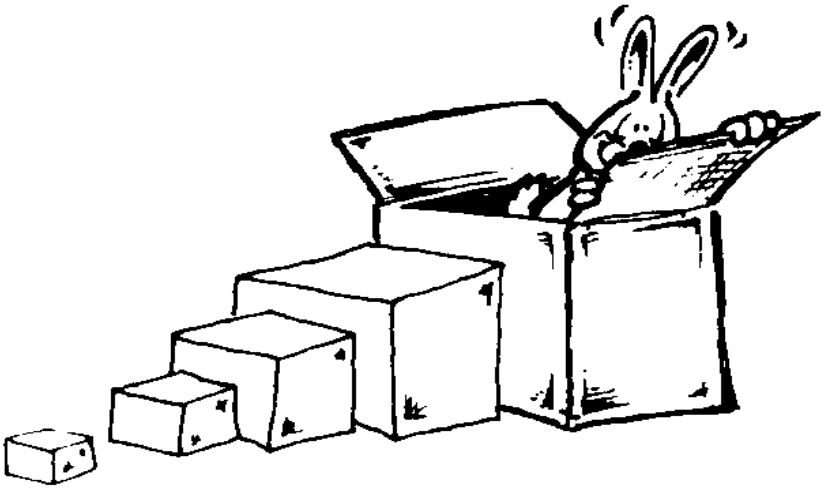
By now, you should know what this procedure will look like after it's been run. What do you think?

```
TO BOXES
REPEAT 4 [FD 100 RT 90]
RT 90 PU FD 120 PD LT 90
REPEAT 4 [FD 100 RT 90]
END
```

Sure…that's a procedure to draw two boxes side by side.

But what if you wanted to draw 20 boxes?

What if you want each box to be bigger than the last?



What if you want them smaller?

In other words, what if you want to vary the size or the number of boxes?

No problem...this is where those things called "variables" come in. A variable is something you put into a procedure so you can change it every time you run it.

Yes, that does sound confusing, doesn't it?

To help explain it, let's take another look at an experiment you did in chapter 2, page 66. Find a big sheet of paper and draw a picture using your favorite shape. Use triangles, squares, rectangles, or even circles.

But you can only use one type of shape. And you can only use one color. You can vary the size of the shape all you want. There's that word again, "vary."

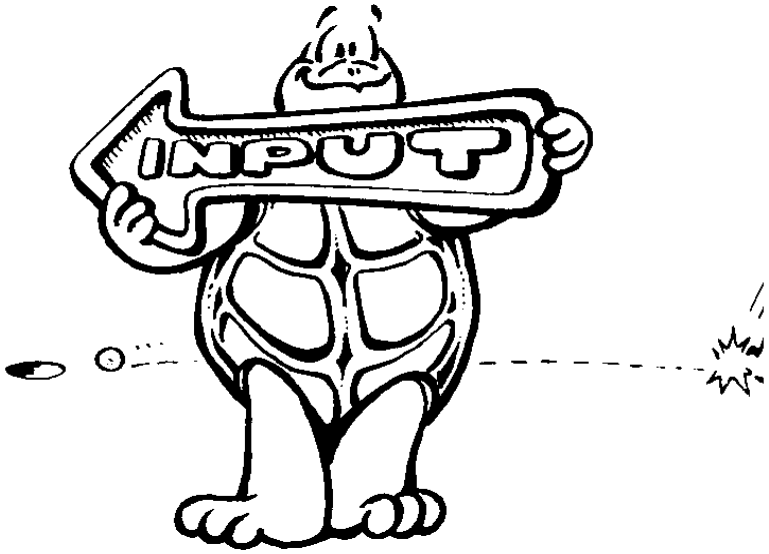
Remember the caterpillar example? That's a picture drawn using squares (and a little bit of a line).

OK…got your drawing done? Before you try to put your picture on the computer, let's take another look at the BOXES procedure.

```
TO BOXES :SIZE
REPEAT 4 [FD :SIZE RT 90]
RT 90 PU FD :SIZE + 20 PD LT 90
REPEAT 4 [FD :SIZE RT 90]
END
```

Bet you already know what the variable is, don't you? It's the :SIZE. That's right.

Now when you type BOXES to run the procedure, you have to provide an input.



Try it out. Type...

BOXES 20

BOXES 40

BOXES 60

BOXES 100

When you type BOXES 20, you tell the :SIZE variable to use the :SIZE of 20. What about BOXES 60. What will :SIZE be then?

Variables must always have an input, or value. They must also have the two dots in front.

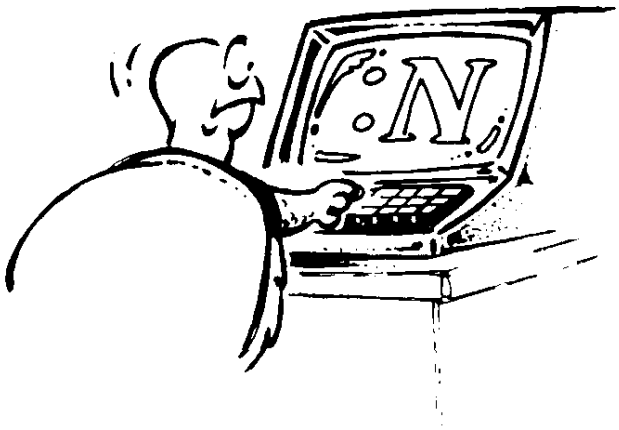
Yes, that's a colon. But in Logo, we call them "dots." You'll find they can save you a lot of time and typing.

Take a look...

Remember the TRI procedure? Let's add a variable.

```
TO TRI :N
REPEAT 3 [FD :N RT 120]
END
```

See! You can name variables just about anything you want. Rather than call this one :SIZE, we call it :N. The :N can stand for number. You could call it :X, :Z, or :WHATEVER.



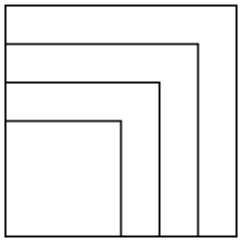
But we still use the dots...we have to do that.

Here are some examples that a 7-year-old enjoyed…using the SQUARE procedure.

```
TO SQUARE :N
REPEAT 4 [FD :N RT 90]
END
```

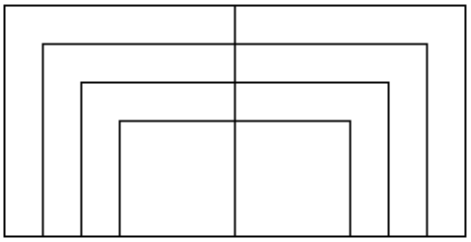
It started as a simple exercise to see what different squares would look like.

```
TO SQUARES
SQUARE 60
SQUARE 80
SQUARE 100
SQUARE 120
END
```



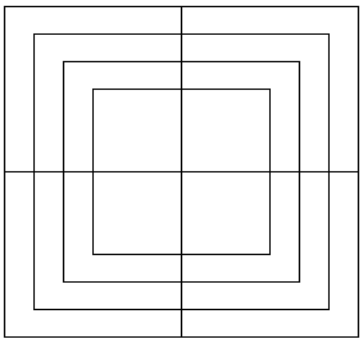
Then she added a left turn…and that reminded her of her mom's stacking tables.

```
TO TABLES
SQUARES
LT 90
SQUARES
END
```



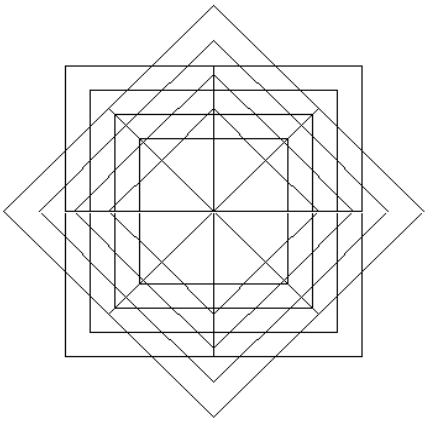
The more she looked at the tables, the more it looked like half of a decorative mirror.

```
TO MIRROR
TABLES
LT 90
TABLES
END
```



And what would happen if you stacked mirrors on top of one another?

```
TO MIRRORS
MIRROR
LT 45
MIRROR
END
```

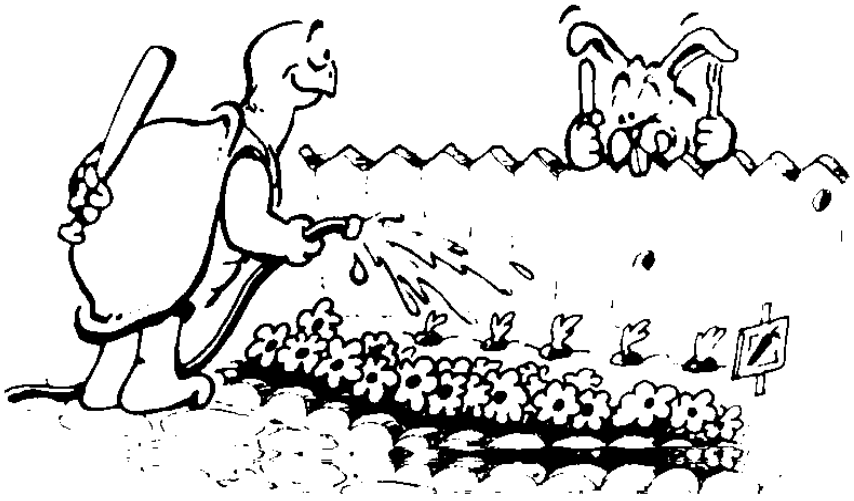


This is a lot to think about. So why not stop for awhile and experiment using one shape in a design.

After you've had fun with one shape, try doing something with two shapes.

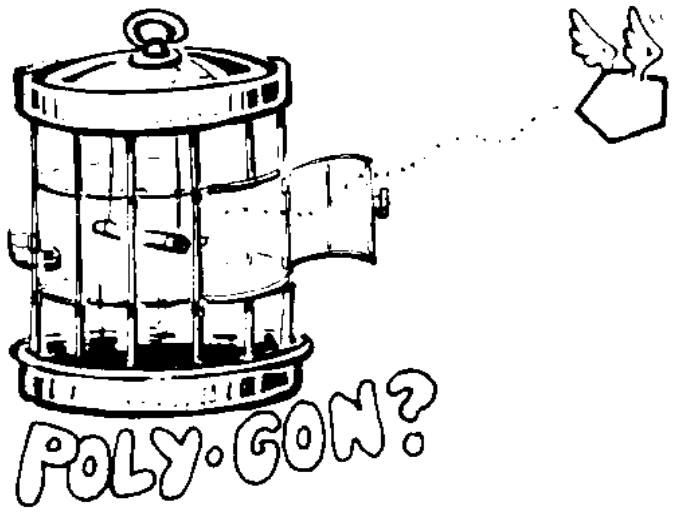
You've already seen what you can do with a square and a triangle. These were combined to make a house. Then they were used to make a wheel.

You've also made some flowers. Maybe you can "plant another garden."



Polygons and Things

Polygon? Now there's a new word for you. Know what it means? No, it doesn't mean that Poly flew away.



Think about this for a moment.

Squares, triangles, and rectangles are polygons. So are pentagons, hexagons, and octagons.

All of these shapes have one thing in common. They all enclose an area that has at least three sides. (You can't enclose anything with two sides, can you?)

Triangles have three sides, squares and rectangles have four, pentagons have five, and octagons have eight.

A polygon is a closed shape with at least three sides.

Remember the review you did at the end of Chapter 2 page 67? You added up the angles used to make squares, triangles, and rectangles. What was the answer?

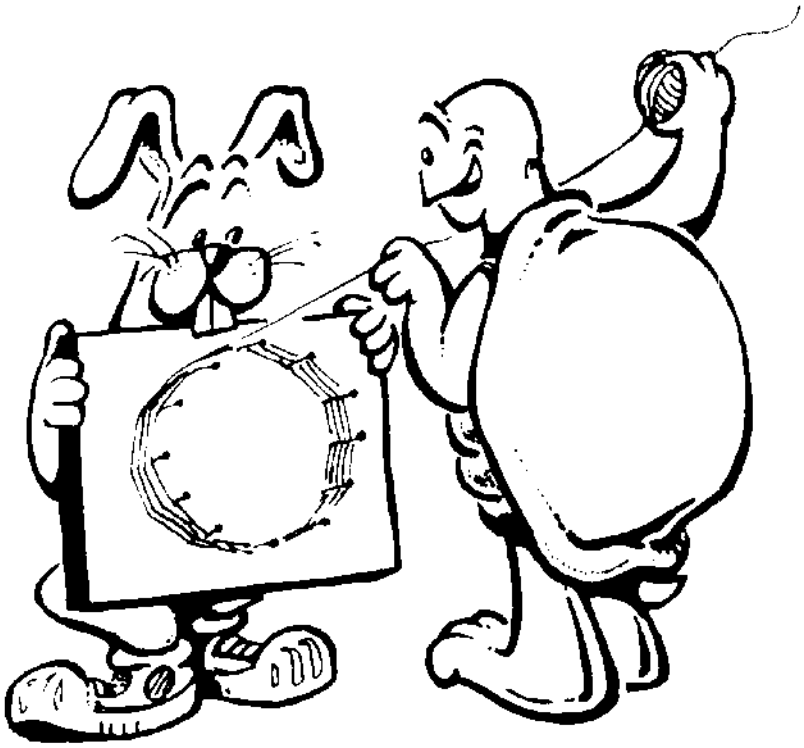
They all added up to 360, right?

Remember Morf's Rabbit Trail about the clock? How many degrees are in the clock face? 360, right? Well, remember that number as we talk about polygons.

Rabbit Trail 13. Learning With a Ball of String

You're going to need a piece of wood from which you can cut a square that is about 12 inches on each side. Use a piece of shelf board or a piece of plywood.

You're also going to need at least thirteen small nails about 1-1/2 inches long, a hammer, a pencil, and about 12 feet of yarn. String will do.



1. **Hammer a nail into the center of the board...just part way so that you have an inch or more sticking up from the board.**
2. **Tie the string to the nail in the center. Now you're going to make a circle around the center nail.**

3. Stretch the string out to the edge of the board.

4. Put your pencil out near the edge of the board. (Hold it up straight.) Then wrap the string around the pencil.

5. Hold the pencil up straight and stretch the string out from the nail. Make sure you hold the string so that it doesn't come off the pencil. Then draw a circle around the center nail.

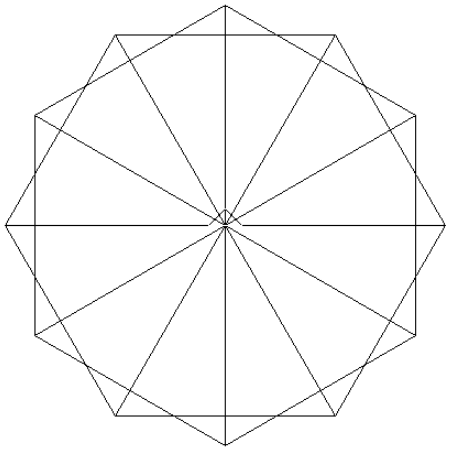
Now we're ready to hammer the other twelve nails into the board. But first, do you have a printer that can print pictures? If so, the turtle can make you a pattern for your string board.

Do you remember the triangle procedure, the one for a triangle with equal sides?

```
TO TRI
REPEAT 3 [FORWARD 100 RIGHT 120]
END
```

Now lets use that procedure to make a pattern.

```
REPEAT 12 [TRI RIGHT 30]
```



Wow! There's a pattern with twelve points, just like the numbers on a clock. Print the screen.

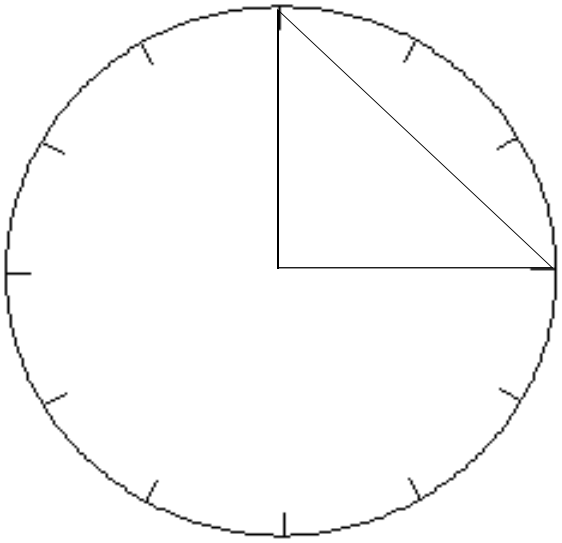
Now carefully push the pattern over the nail in the center of the board. Hold or tape the pattern in place. Then draw a line along each of the pattern lines to the circle you drew before. Where each line crosses the circle, hammer in a nail.

Looks like a clock, doesn't it?

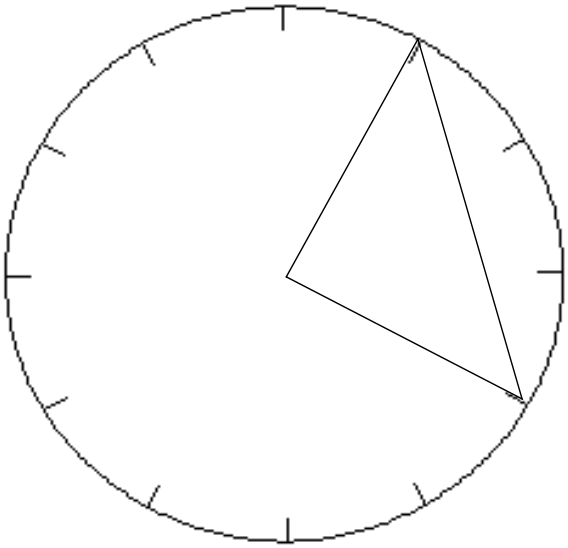
Well, now you and the string can play turtle graphics.

Take the string from the center nail...that's HOME on the screen...and stretch it up to the nail at 12 o'clock.

Go around the 12:00 o'clock nail and take the string around the 3:00 o'clock nail. Then take the string home.



Now do the same thing, only go around the 1:00 o'clock nail and the 4:00 o'clock nail.

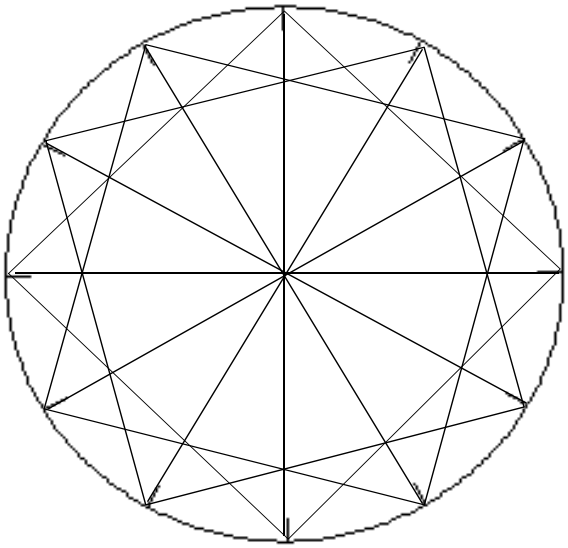


What's really going on here?

You're moving a triangle through space…turning it on a vertical axis that passes through Home. Can you think of a better way to demonstrate how things move through space?

Now, go around the 2:00 o'clock nail and the 5:00 o'clock nail, the 3:00 o'clock nail and the 6:00 o'clock nail, the 4:00 o'clock nail and the 7:00 o'clock nail. Continue on all the way around.

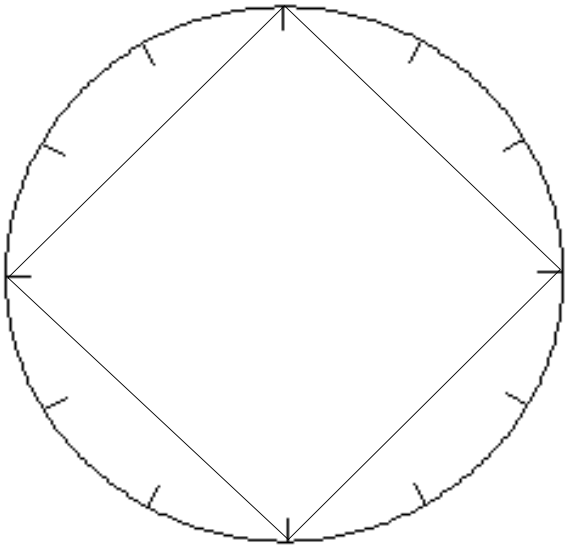
What pattern has the string made? Looks just like the one you printed, doesn't it. Only this one's inside a circle.



Your hand acted like the turtle, didn't it…as it moved around the string board. But instead of drawing lines on the screen, you made a line of string.

Take a close look at this drawing. We've been talking about and using triangles to create this. But do you see some other shapes here?

Do you see the squares?



How many squares can you find?

Look carefully at the procedure that made the pattern...

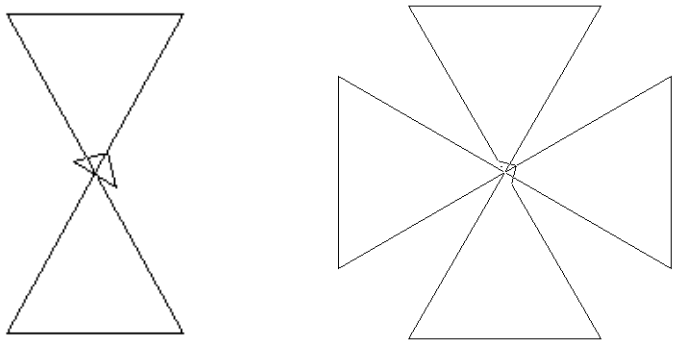
REPEAT 12 [TRI RIGHT 30]

Your first TRI went from HOME, around the 12:00 o'clock nail, around the 3:00 o'clock nail, and then HOME. Then did you turn RIGHT 30 turtle turns?

Guess so, right?

If you turn back to Chapter 4, you'll find that you can do all the clock activities from Rabbit Trail 9 on your string board. Remember, you did those using a circle drawn on the floor.

How about doing your string activities on the floor, too!



Rabbit Trail 14. String Toss Game

This can be a great game with a group of friends. It gives you the chance to act out Logo commands by drawing with a ball of string…better yet, a ball of colored yarn.

Have one person stand in the middle of your chalk circle. Then put other friends at each of the twelve points at the edge of the circle. You can also have one friend write down commands. Another can do the commands on the computer.

The person in the middle is like the turtle. The turtle always starts facing 12:00 o'clock.

LEFT 30

Toss the string to 11:00 o'clock.

Toss the string from 11:00 o'clock to 1:00 o'clock.

Toss the string Home.

What shape is this?

Another game you can play with string is called FD :N. We're sneaking the variables back in here. The idea is to create a design using string. The :N variable can equal one step or as many as you want.

Let's say you want to create a square of string. That's real easy. One person plays Turtle starting at Home. The turtle gives the string to the first person and says FD :N times 5. The first person takes 5 steps.

The first person then turns RT 90 and gives the string to the second person. That person goes FD :N * 5 and RT 90. A third person takes the string and goes FD :N * 5 RT 90. And finally a fourth person takes the string and brings it HOME.

See how this works?

It's more fun when you make crazy shapes. Try it.

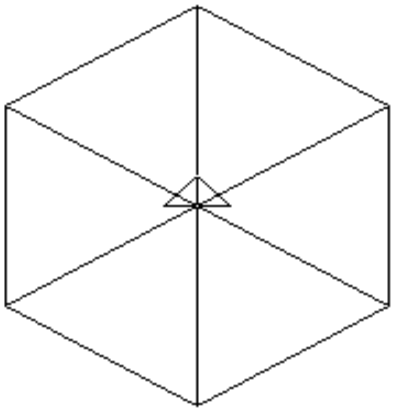
Hexagons and Spiderwebs

To make that String Toss Game design on the computer, you can use the TRI :N procedure you wrote earlier in this chapter.

```
TO TRI :N
REPEAT 3 [FD :N RT 120]
END
```

What would happen if you repeated the TRI :N procedure, turning a bit after each triangle?

```
REPEAT 6 [TRI :N
        RT 60]
```



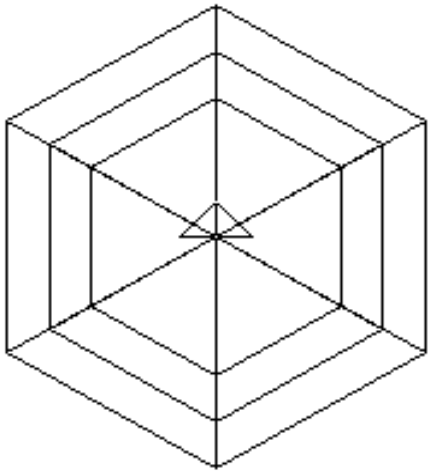
What do you call a shape that has six sides like this?

It's a hexagon, right?

```
TO HEXAGON :N
REPEAT 6 [TRI :N RT 60]
END
```

Be sure to tell the turtle how big to make the hexagon.

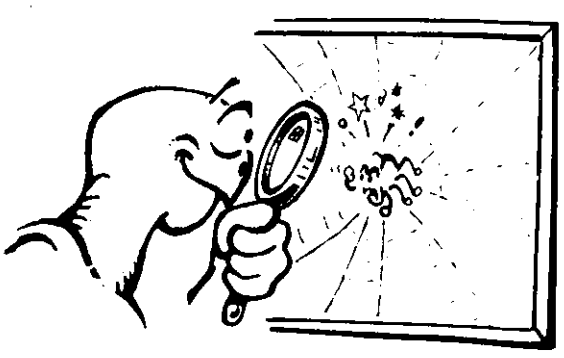
Try this…



```
HEXAGON 60
HEXAGON 80
HEXAGON 100
```

What's this look like?

Of course, it's a Spiderweb!



Can you think of another way to write this procedure so that the turtle will do the same thing? How about this...

```
TO SPIDERWEB :N
HEXAGON :N
HEXAGON :N + 20
HEXAGON :N + 40
END
```

Go ahead. Type this procedure and then enter...

```
SPIDERWEB 40
```

Play around with this idea to see what it can do. Make up some other shapes using variables to which you add numbers.

Adding More Variables

Can you think of a way to use more variables in the SPIDERWEB procedure? What about substituting a variable for 10? For 20? For both?

```
TO SPIDERWEB :N :X :Y
HEXAGON :N
HEXAGON :N + :X
HEXAGON :N + :X * :Y
END
```

This is getting complicated. Now you have three variables.

:N gives you the size of each side.

:X tells you how much to add to :N

:Y tells you to multiply :X by this number

After you've typed in this procedure, try this and see what happens.

```
SPIDERWEB 30 10 2
```

Does this look like the first spiderweb the turtle drew? It should. Take a look...

```
TO SPIDERWEB 30 10 2
HEXAGON 30
HEXAGON 30 + 10
HEXAGON 30 + 10 * 2
END
```

Changing a Variable

This is fine when you want to make three hexagons that have sides of 30, 40, and 50. But what if you want to do five hexagons? Seven hexagons? Seventy hexagons?

Let's try something!

When you write a procedure, it becomes another command you can use, right?

OK…then let's make the most of it. Tell Spiderweb to draw a hexagon using the variable :N. Then tell Spiderweb to add 10 to itself and do the same thing again.

```
TO SPIDERWEB :N
HEXAGON :N
SPIDERWEB :N + 10
END
```

Try it! What happens?

This is one way to change a variable. Maybe there are others. Maybe there's a command that will let you MAKE the variable whatever you want it to be.

Does that give you a clue? It should…take a look.

```
TO SPIDERWEB :N
HEXAGON :N
MAKE "N :N + 10
SPIDERWEB :N
END
```

Let's trace this procedure.

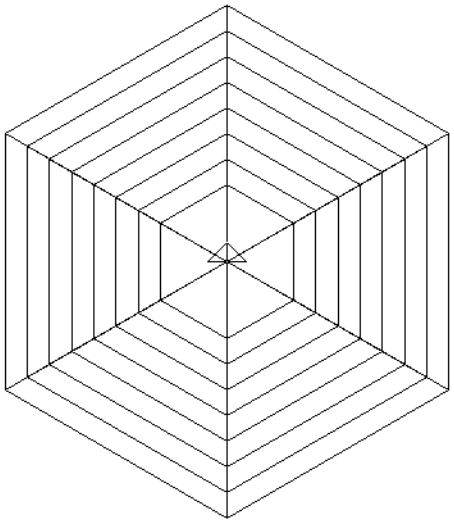
When you type SPIDERWEB 30, the first thing the turtle will do is draw a hexagon with sides of 30. Then the turtle reads the next line…

MAKE "N :N + 10

This says to MAKE the variable :N have the value of :N + 10...or MAKE "N 30 + 10, or 40.

The turtle then draws a hexagon with sides of 40...and then reads the next line. Then what happens to :N?

You can see what happens by watching the turtle draw lots of hexagons. Soon it will fill the screen with them.



Of course...this raises another question. How can you get the turtle to stop?

One way to stop the turtle in MSW Logo is to press the HALT button. This stops the turtle wherever she is. But wouldn't it be better if you could get the turtle to stop after she's finished doing the last hexagon?

Conditional Things

Well...there is a way. You just tell the turtle that IF the last hexagon that it drew was as big as you want the spiderweb to be, THEN stop drawing.

```
TO SPIDERWEB :N
IF :N > 200 [STOP]
HEXAGON :N
SPIDERWEB :N + 10
END
```

Look at that first line in this new procedure. When the turtle reads this line, it learns that IF :N is greater than 200, then stop drawing.

In MSW Logo, you don't have to type the word, THEN. The turtle knows what you mean.

```
IF :N > 200 [STOP]
```

That thing that looks like an arrowhead after the :N > is the symbol for "greater than." It means that if the value of :N is greater than 100, then STOP.

If > means "greater than," what does that other arrow symbol… < …mean?

You guessed it. It means "less than." An easy way to remember which symbol is which is that the arrow always points to the smaller value.

IF :N > 100 means that the value of :N must be larger than 100…at least 101.

IF :N < 100 means that the value of :N must be less than 100…no more than 99.

For our example, we picked 200 as a place to stop. You can select your own stopping point. Or you can make the stopping point another variable. How would you do that?

Go ahead. Give it a try. But remember, if you're going to use a variable like this, you have to add it to the procedure name.

```
TO SPIDERWEB :N  ____
IF :N > ____ [STOP]
HEXAGON :N
MAKE “N :N + 10
SPIDERWEB :N ____
END
```

When you've finished with your spiderwebs, why not try adding variables to your procedures for drawing other shapes? See what you can do with squares, rectangles and things.

There are other things you can do with variables and conditional statements. You’ll discover these as you move through the rest of this book.

In the meantime, see what you can do on your own.

Remember, this whole book is about Discovery!

More Ways to Vary Variables

You learned about MAKE and IF. Well, Logo gives you lots of other ways to vary your variables.

```
MAKE <name> <object>
or
MAKE “JOE 2
MAKE “TOM 4
MAKE “SAM :JOE + :TOM
```

So what does :SAM equal? If you said six, you get a Gold Star.

You can also NAME :JOE + :TOM “SAM

This does the same thing. You NAME <object> <name>.

If you want to see what :SAM equals, you can tell the computer to…

PRINT :SAM
or
SHOW :SAM

You can also tell Logo to…

SHOW THING “SAM
or
PRINT THING “SAM

THING is like the dots. It outputs the value of the variable named in the word that follows THING.

Local and Global Variables

You can write two kinds of variables…global and local. Global variables are used by any procedure. Take a look...

TO SHAPES :N
TRI :N
SQUARE :N
RECTANGLE :N
END

You've already got a procedure called TRI :N. Now write procedures for a square and a rectangle using :N to represent the distance forward.

```
TO SQUARE :N
REPEAT 4 [FORWARD :N RIGHT 90]
END

TO RECTANGLE :N
REPEAT 2 [FD :N RIGHT 90 FD :N * 2 RT 90]
END
```

If you type SHAPES 100, each of the procedures will use 100 wherever there is an :N.

Local variables are "local" to the one procedure where it is used. You write them like this...

```
TO TRI
LOCAL "X
MAKE "X 100
REPEAT 3 [FD :X RT 120]
END
```

Go ahead. Change your TRI procedure and then run the SHAPES procedure using SHAPES 100 again. Now what does the picture look like? Why?

_____

Musical Variables

In the last chapter, we talked about making music. Now that you've read about variables, how about some musical variables?

Want to turn your keyboard into musical keys?

```
TO MUSIC
MAKE “KEY RC
IF :KEY = “C [SOUND 262 100]
IF :KEY = “D [SOUND 294 100]
IF :KEY = “E [SOUND 330 100]
IF :KEY = “F [SOUND 349 100]
IF :KEY = “G [SOUND 392 100]
IF :KEY = “A [SOUND 440 100]
IF :KEY = “B [SOUND 494 100]
IF :KEY = “C [SOUND 523 100]
MUSIC
END
```

There’s another new command…RC. That’s short for READCHAR.

When Logo sees the READCHAR or RC command, it stops and waits for you to type a character. In this case, the letter you type becomes the variable :KEY.

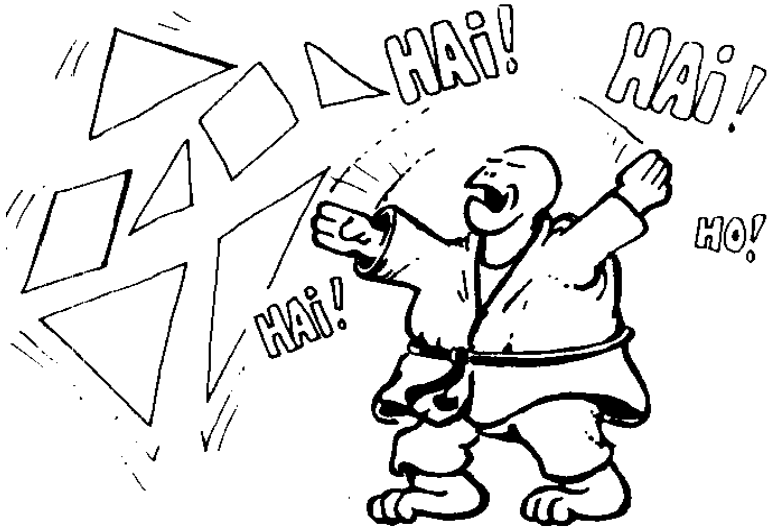
If you type one of the keys…A, B, C, D, E, F, G…you hear a note. Otherwise Logo just runs the MUSIC procedure again and again until you hit one of the sound keys and press Enter.

You’ll have more fun with RC later…also READLIST and READWORD.

Rabbit Trail 15. Tangrams

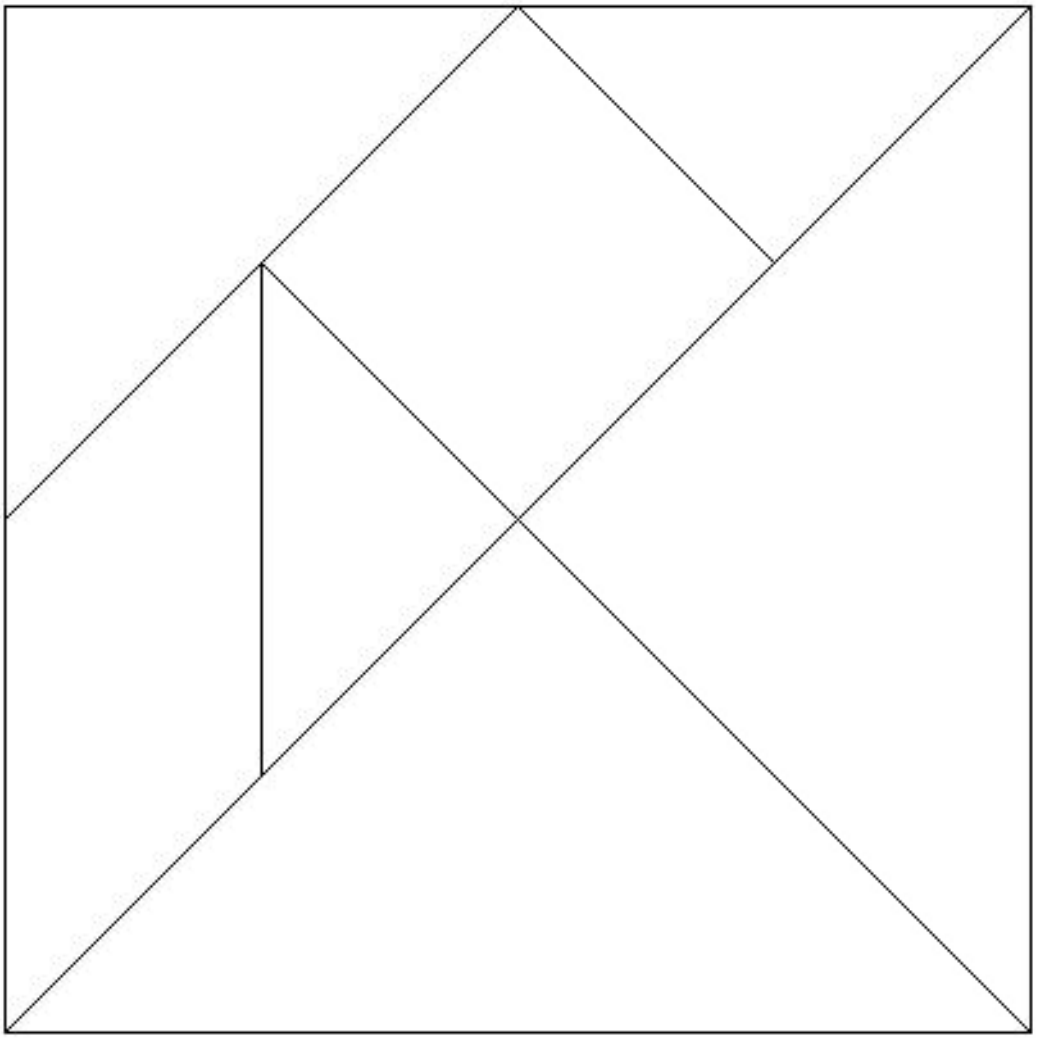
The Tangram is an Oriental puzzle with seven shapes of different sizes. The puzzle is to use these shapes to make lots of different things.

You'll find procedures to draw these shapes on the diskette that came with this book. They're all in the file called TANGRAM.LGO.



On the next page you will find a large copy of a square made with the tangram pieces. Here's how to make the pieces of the tangram puzzle.

1. **Copy the page and paste it to a piece of cardboard.**
2. **Carefully cut out the pieces.**
3. **Now you can play with the pieces to create interesting shapes…birds, ships, dragons, and other interesting designs.**
4. **Then draw them on the computer.**

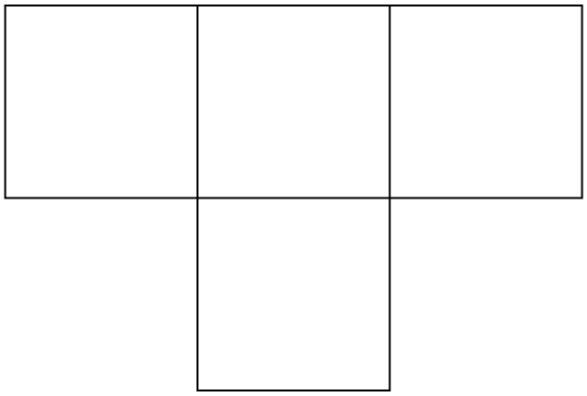


Why not visit your local library or bookstore? You'll find there are a number of books on tangrams that will give you lots of ideas of what to do with your new puzzle pieces.

Rabbit Trail 16. More on Logo Puzzles

Remember the Logo Puzzles back in Chapter 2?

Get some straws or some sticks and make this puzzle on a table top. Now…take away just one straw or stick and make a picture that has only three squares.



You can solve this puzzle by picking up any of the sticks to see if three squares are left on the table. But we told you there was a procedure that would solve the puzzle for you. Let's see what it is.

First of all, let's write procedures to create the puzzle. Obviously, you'll need a SQUARE procedure.

```
TO SQUARE
REPEAT 4 [FD 100 RT 90]
END
```

Now you can write a PUZZLE procedure.

```
TO PUZZLE
CS HT
REPEAT 2 [SQUARE MOVE]
REPEAT 2 [SQUARE RT 180]
```

```
HOME
END

TO MOVE
RT 90 FD 100 LT 90
END
```

The next step is to solve the puzzle. But how could the computer do that? It doesn't think? It simply does what you tell it to do.

Since the computer does things much faster than you can, one way to have the computer help you solve the problem is to have it erase each line in the puzzle and then draw it again.

Sound confusing? Try this…

```
TO SOLVE
REPEAT 2 [SQ MOVE]
REPEAT 2 [SQ RT 180]
HOME
END

TO SQ
REPEAT 4 [SIDE RT 90]
END

TO SIDE
PE FD 100 WAIT 100 PENPAINT
PD BK 100 FD 100
END
```

Waiting

There are times that you want to slow the computer down so you can see what's going on…or when you just want it to wait a few seconds. That's where the WAIT command comes in.

You'll have to experiment with numbers to find out how long you want the wait to be. Some computers are faster than others. WAIT 100 may be too slow…or too fast.

The Tacit Assumption

Here's another puzzle for you, the Tacit Assumption. Draw this pattern on paper or create it on the screen and print it.

Your challenge is to draw four straight lines that pass through all nine of the stars. You can start anywhere you want. The catch is that once you put your pencil down, it cannot leave the paper until you are done.

Here's a hint. When most people try to solve this puzzle, they *assume* that they must stay within the limits of the square made by the nine stars. That's what we mean by the *Tacit* or unspoken *Assumption*.

This procedure draws the nine stars.

```
TO TACIT
CS HT PU
REPEAT 3 [LINE MOVE]
HOME
END

TO LINE
REPEAT 3 [STAR FD 100]
BK 300
END


TO MOVE
RT 90 FD 100 LT 90
END

TO STAR
PD REPEAT 5 [FD 10 RT 144] PU
END
```

Here's a procedure that will solve the puzzle. However, you have to figure out what number to use for the variable. All you have to do is type SOLVE and add a guess. If your guess doesn't quite do it, type TACIT to draw the puzzle. Then guess again!

```
TO SOLVE :SIDE
FD :SIDE/SQRT 2 RT 135
FD :SIDE RT 135
FD :SIDE/SQRT 2 RT 90
RT 45 FD :SIDE/SQRT 2
END
```

That :SIDE / SQRT 2 may seem like something strange. But don't worry about it now. It's just part of a math formula for drawing the long side of a right triangle. You'll learn more about *SQ*uare *RooT*s in the Great Math Adventure chapter.**

The Non-Stop Puzzles

Do you remember the puzzles about the triangle patterns? Draw the patterns without retracing any line and without lifting the pen from the paper?

Here are the procedures. They give you some new things to think about.

```
TO TRIPUZZLE
CS CT
PR [Here's a puzzle for you!]
PR "
NSTRIANGLE 100 WAIT 100 CT
PR [Draw this pattern of triangles without lifting]
PR [your pencil from the paper and without retracing]
PR [any of the lines.] WAIT 200 CT
PR [Or press any key to have Logo do it for you!]
IGNORE RC
TRIANGLE 100
END
```

Here's another example of the PRINT…PR, for short…command. This time it's used to add instructions on what to do.

But what about that line…PR " ?

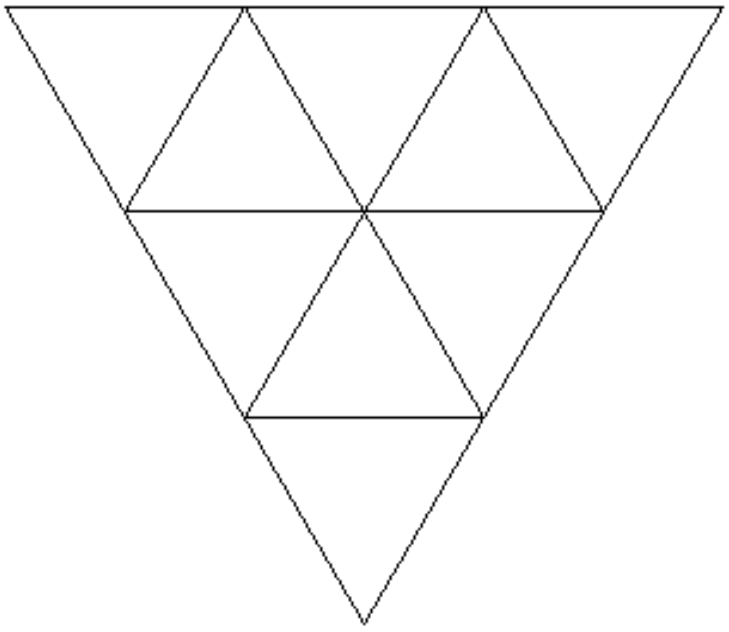
You can write it as PR " or PR [ ]. What it says is, print nothing. And that's exactly what it does, it prints nothing…leaving you with a blank line.

And how about that line IGNORE RC?

IGNORE is an MSW Logo command. Some other versions of Logo don't have it, but it's real easy to create.

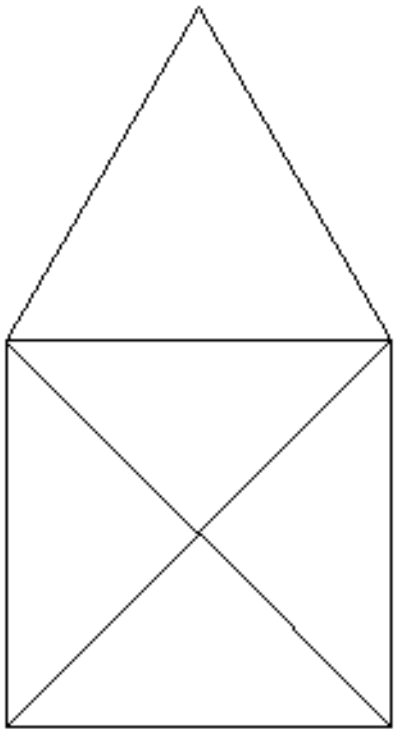
```
TO IGNORE :X
END
```

This procedure shows what the command does… absolutely nothing! When you replace the :X variable with RC…or READCHAR…the procedure just stops and waits for you to press a key.



```
TO NSTRIANGLE :D
CS HT LT 150 FD :D RT 120 FD :D * 3
RT 120 FD :D * 3 RT 120 FD :D * 2
RT 120 FD :D * 2 LT 120 FD :D
LT 120 FD :D * 2 LT 120 FD :D
LT 120 FD :D * 2 LT 120 FD :D
END
```

```
TO TRIANGLE :D
CS ST LT 150
FD :D RT 120 FD :D * 3 WAIT 50
RT 120 FD :D * 3 RT 120 FD :D * 2 WAIT 50
RT 120 FD :D RT 120 WAIT 50
REPEAT 3 [FD :D LT 120] LT 180 WAIT 50
REPEAT 3 [FD :D RT 120] WAIT 50
LT 60 FD :D LT 120 FD :D
END
```



Remember this puzzle?

Try this one, only draw it without crossing any line…and without retracing any line or lifting your pencil from the paper.

```
TO SOLVE
CS CT
PR [OK!  How about trying the same thing with this house? Only don't cross any lines either!]
HOUSE 100 WAIT 200 CT
```

```
PR [Or press any key to have Logo do it for you!]
IGNORE RC
NSHOUSE 100
END

TO HOUSE :D
CS FD :D LT 30
REPEAT 3 [FD :D LT 120]
LT 105 FD :D * .71 LT 90
FD :D * .71 RT 135 FD :D
RT 90 FD :D RT 135
FD :D * .71 RT 90
FD :D * .71
END

TO NSHOUSE :D
CS FD :D LT 30
REPEAT 3 [FD :D LT 120] WAIT 50
LT 105 FD :D * .71 LT 90 WAIT 50
FD :D * .71 RT 135 FD :D WAIT 50
RT 90 FD :D RT 135 WAIT 50
FD :D * .71 RT 90 WAIT 50
FD :D * .71
END
```

Enough of this house business. Now it's time to start getting serious about turtle geometry…starting with the next chapter.
