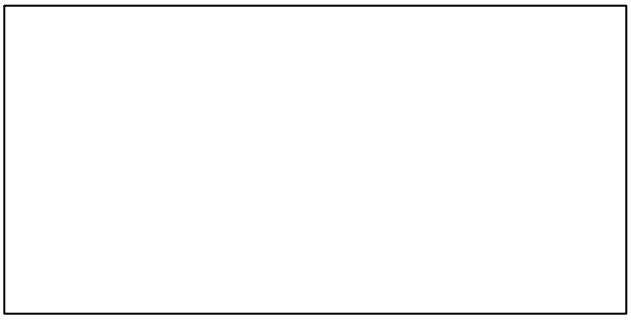
Names: _____ / _____ / _____ /

CHAPTER 10
Recursion in LOGO

In chapter 9 you were asked to draw a lot of pictures of regular polygons on the same screen. While the procedures for each were different, they required the same basic commands, forward so far and right so far, depending on the number of sides. A recursive procedure is one in which the same basic commands are used in succession, with some parameter changes depending on a previous output or on the position of the procedure. For example, with your pencil sketch the effect of the following command sequence:

FD 50 RT 90
FD 50+10 RT 90
FD 60+10 RT 90
FD 70+10 RT 90



If the sequence continued what would be the 100th command?

Let's write a procedure called **maze**, which does this all at once:

```
TO maze :L
FD :L RT 90
maze :L+10
END
```

Save and exit. Look carefully at this procedure. What should it do?

WARNING: To stop a program in progress click on **Halt** in the commander window**.**

Now run **maze 20**. :) Maybe you should try **Halt** now! What happened and why?

Notice **maze** calls itself and will continue forever unless you stop it. We can edit **maze** to include a "test" command designed to stop it when the parameter :L

reaches a certain size. Insert the command line

IF :L>100[STOP]

at the beginning of the definition of the procedure. This tells logo to check the parameter :L and stop the process if :L is bigger than 100.
Try the new edited version of **maze 30.**

Exercise 1. Edit **maze** one final time to allow the amount added each time to be a variable you get to choose also and the maximum length to be another variable you get to choose. So the procedure's command will be

maze :l :a :m

Write your program for maze :l :a :m in the table below.

Attach printouts of the applications of **maze 30 10 500**

Exercise 2. Write a recursive procedure called limpoly, which will construct all the regular polygons starting with whichever you wish, with all side lengths whatever you wish, and stopping at the number of sides you wish.
In other words complete the following procedure definition:

to limpoly :n :l :m

where :n is the initial number of sides, :l is the length of each side and :m is the final number of sides. Write your program in the table provided.

Attach a printout of the first 25 regular polygons of side length 30, beginning with the triangle.

Exercise 3. Repeat exercise 2 in its entirety, for all the regular "odd pointed" stars of a given side length beginning with a given number of points, and ending with a specified number of points. Write your program in the table provided.

Again attach a printout of the stars with side length 30, starting with the 5 pointedstar and ending with the 21 pointed star.

Aside, before recalling a procedure within a procedure the command **pause** causes the procedure to pause and await the command **continue** before proceeding. This allows you to see each consecutive construction. I wrote a procedure stored as c.log, which allows me simply enter the command **c** to continue. It's very handy, you might try it.

Exercise. Edit the procedures in Exercises 2 and 3 so as to pause before calling the procedure back. This allows you to see them drawn one at a time.

FINAL PROJECT

Write and save a program that draws a scene of a school house on a starry night with nice landscaping and a crescent moon. (A picket fence with a gate to the back yard playground would be nice, wouldn't it?) Don't forget to put the names of your group members somewhere on your picture. Include both the picture and your program for drawing it in your portfolio.