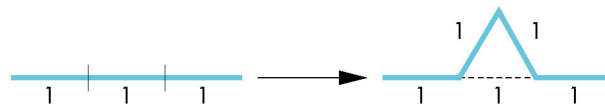


### Homework 3

This one will give you some practice with writing your own OpenGL program and compiling it, as well as additional practice thinking about iterative and and/or recursive algorithms:

Exercise 2.6 (page 111 in your text): Space-filling curves have interested mathematicians for centuries. In the limit, the curves have infinite length, but they are confined to a finite rectangle and never cross themselves. Many of these curves can be generated iteratively. Consider the “rule” pictured in Figure 2.48 that replaces a single line segment with four shorter segments. Write a program that starts with a triangle and **iteratively** applies the replacement rule to all line segments. The object that you generate is called the Koch snowflake. For other examples of space-filling curves, see [Hil07] and [Bar93].



For Graduate Students (20 points extra credit for undergrads) – can this be done using a recursive algorithm? If so, implement it – if not – tell me why??

NOTE: If you do not give me the **iterative algorithm** first, you will not receive full credit, let alone get extra credit. You don't need to do anything fancy with your shaders in this one, but if you implement your vertex shader to assign vertex colors based on input from your program, and do something interesting with those colors, I'll give you 10 bonus points.