

A well-known scientist (some say it was Bertrand Russell) once gave a public lecture on astronomy. He described how the Earth orbits around the sun and how the sun, in turn, orbits around the centre of a vast collection of stars called our galaxy.

At the end of the lecture, a little old lady at the back of the room got up and said: “What you have told us is rubbish. The world is really a flat plate supported on the back of a giant tortoise.”

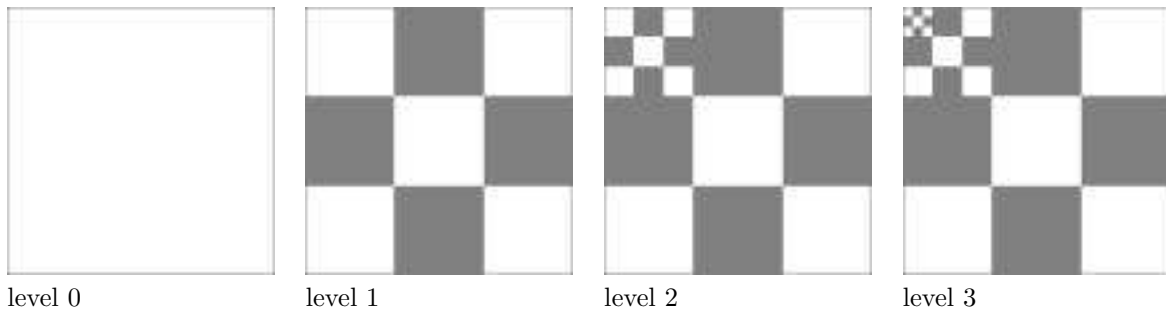
The scientist gave a superior smile before replying, “What is the tortoise standing on?”

“You’re very clever, young man, very clever,” said the old lady. “But it’s turtles all the way down.”

— reported by Stephen Hawking (b. 1942), *A Brief History of Time*

This is a pair-programming assignment. **All work must be done with both partners present!** Your task in this assignment is to get some practice with writing recursive methods.

1. First, we’ll start with a visual example of recursion. A *fractal* is an image that is composed of smaller copies of itself. There are a number of well-known examples that you can find by searching on the web, but we’ll deal with a particularly simple example in this question. When drawing fractals, you typically specify a *level*, which tells you how “deep” to go in drawing smaller versions of the image. Here is an example fractal:



This particular fractal is produced as follows: the level-0 fractal consists simply of an empty white square. To produce a level- i fractal, do the following: divide up the square into 9 pieces, solidly filling in the non-corner, non-middle pieces, and then fill in the upper-left corner piece with a level- $(i - 1)$ fractal. Your task will be to write a class that produces these fractals. *Your code must be recursive.*

Write a class `FractalLab` that contains a private 3-dimensional array `pixels` representing the pixels of the image. Your class should also include the following methods:

- `public FractalLab(int size)`, a constructor, which should initialize all pixels to white.
- `public EzImage createSquareFractal(int level, int size, Color color)`. This method should do the following: if `level` is zero, then the method should create and return a new `EzImage` using the current contents of the `pixels` array. If `level` is greater than zero, it should create a level-`level` fractal of size `size` in the upper-left corner of `pixels`. So your method will need to fill in the four solid squares with the color `color`, and then make an appropriate recursive call to fill in the upper-left square with a level-`(level - 1)` fractal.

Also write a tester class to show your code working.

Optional: if you are feeling somewhat ambitious, also write a method to do the same thing as above, except that the *middle* square (instead of the upper-left square) is a level- $(i - 1)$ fractal.

Optional #2: if you are feeling even more ambitious, also write a method to do the same thing as above, except that instead of filling in the four squares with solid colors, instead fill in those squares with an appropriately scaled image specified by the user.

2. Write a class called `BackString` with the following three methods:

- `public static void iterBackwards(String text)`
This method displays—that is, prints out to the console using `System.out.print()`—the String backwards, using iteration (that is, using a `for` loop or a `while` loop).
- `public static void recurseBackwards(String text)`
This prints the String backwards, using recursion instead of iteration. (This method should produce the same output as `iterBackwards()`.)
- `public static void main(String[] args)`
This is a main method to test both of your methods by displaying a String of your choice backwards, once using each method.

You may find the `charAt()`, `length()`, and `substring()` methods for Strings useful.

The long-awaited story on the Snowplow lab:

The Snowplow assignment will be accepted through 5:00p on Friday, 10 March 2006. It will be scored as up to one lab's worth of extra credit on problem sets and labs. You are *not* required to do it, but it would be yet another good way to practice for the second exam.

If you started it with a partner, you may continue to work with that partner. If you started it solo, you may join forces with another person who also started it solo.