Homework 5- CECS 174   Due date: Mar 25th, 2015

1. Write a program that reads a word and prints all substrings, sorted by length. For example, if the user provides the input "`rum`", the program prints

```
r
u
m
ru
um
rum
```

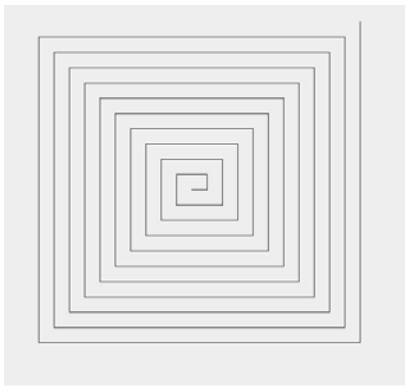
2. *Prime numbers*. Write a program that prompts the user for an integer and then prints out all prime numbers up to that integer. For example, when the user enters `20`, the program should print

```
2
3
5
7
11
13
17
19
```

Recall that a number is a prime number if it is not divisible by any number except 1 and itself. Use a class `PrimeGenerator` with methods `nextPrime` and `isPrime`. Supply a class `PrimePrinter` whose `main` method reads a user input, constructs a `PrimeGenerator` object, and prints the primes.

3. The game of Nim. This is a well-known game with a number of variants. The following variants has an interesting winning strategy. Two players alternately take marbles from a pile. In each move, a player chooses how many marbles to take. The player must take at least one but at most half of the marbles. Then the other player takes a turn. The player who takes the last marble loses.
Write a program in which the computer plays against a human opponent. Generate a random integer between 10 and 100 to denote the initial size of the pile. Generate a random integer between 0 and 1 to decide whether the computer or the human takes the first turn. Generate a random integer between 0 and 1 to decide whether the computer plays smart or stupid. In stupid mode the computer simply takes a random legal value (between 1 and n/2) from the pile whenever it has a turn. In smart mode the computer takes off enough marbles to make the size of the pile a power of two minus 1 – that is, 3, 7, 15, 31, or 63. That is always a legal move, except when the size of the pile is currently one less than a power of two. In that case, the computer makes a random legal move.
You will note that the computer cannot be beaten in smart mode when it has the first move, unless the pile size happens to be 15, 31, or 63. Of course a human player who has the first turn and knows the winning strategy can win against the computer.

4. Write a graphical application that draws a spiral, such as this one:



5. It is easy and fun to draw graphs of curves with the Java graphics library. Simply draw 100 line segments joining the points (x,f(x)) and (x+d, f(x+d)), where x ranges from $x_{min}\ to\ x_{max}$ and $d = (x_{min} - x_{max}) / 100$. Draw the curve `f(x) = 0.00005x³ - 0.03x² + 4x + 200`, where x ranges from 0 to 400 in this fashion.

6. Write array methods that carry out the following tasks for an array of integers by completing the `ArrayMethods` class below. For each method, provide a test program.

```
public class ArrayMethods
{
      private int[] values;
      public ArrayMethods(int[] initialValues) { values = initialValues; }
      public void swapFirstAndLast(){…}
      public void shiftRight(){…}
      …
}
```

a. Swap the first and last elements in the array.
b. Shift all elements by one to the right and move the last element into the first position. For example, 1 4 9 16 25 would be transformed into 25 1 4 9 16.
c. Replace all even elements with 0.
d. Replace each element except the first and last by the larger of its two neighbors.
e. Remove the middle element if the array length is odd, or the middle two elements if the length is even.
f. Move all even elements to the front, otherwise preserving the order of the elements.
g. Return the second-largest element in the array.
h. Return true if the array is currently sorted in increasing order.
i. Return true if the array contains two adjacent duplicate elements.
j. Return true if the array contains duplicate elements (which need not be adjacent).