INTRODUCTION TO COMPUTER SCIENCE I
PROJECT 3A
Conditionals and loops

Here is the first of a two-part project that will exercise your use of our new-found *conditional* and *iterative statements*. You will use them in various combinations to perform a few new types of calculations.

1 The Fibonacci sequence

Computer science teachers just love the Fibonacci sequence. It makes for good programming examples. I see no reason to restrain myself from it's use, so here we go:

$$F(n) = \begin{cases} n & \text{if } 0 \leq n < 2 \\ F(n-1) + F(n-2) & \text{if } n \geq 2 \end{cases}$$

That is, loosely speaking, each Fibonacci number is the sum of the two previous ones, with the exception of the first two values which are $F(0) = 0$ and $F(1) = 1$. Thus, the sequence begins: $0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 \ldots$

A program to write: You must write a new program, from scratch, composed of a few methods. Here is the progression that you should follow:

1. **Getting started:** Login to `remus/romulus`. At the command line, create a new directory and change into it, like so:

   ```
   $ mkdir project-3
   $ cd project-3
   ```

   Then, create and open a new, blank source code file with *Emacs*:

   ```
   $ emacs Fibonacci.java
   ```

   Finally, inside this file, put in the usual stuff that surrounds the methods that you write:

   ```
   import java.util.Scanner;

   public class Fibonacci {

     public static Scanner keyboard = new Scanner(System.in);

     // YOUR METHODS WILL GO HERE.

   }
   ```

2. **Write a Fibonacci calculating method:** Write a method named `fib` that accepts, as a parameter, an integer k. This method should then calculate the k^{th} Fibonacci number and return it. That is, your method should begin:

```
public static long fib (int k) {
```

Fill in the body of this method with appropriate code to perform the needed calculation. You should be using *iteration*—that is, a *while loop*—to perform this calculation. Notice the return type of `long` for this method. Since Fibonacci numbers grow to be quite large rather quickly, using a larger integer type is important and useful here.

As always, feel free to write a temporary `main` method to test your `fib` method. That is, have `main` pass some known value for k to your `fib` method, and the print the result that is returned. Verify that the value computed by your `fib` method is correct.

3. **Write a Fibonacci search method:** Even a `long` integer has a limited range. Any number larger than about 8 quintillion cannot be stored in such a variable. In fact, if you take the largest position integer that can be stored in a `long` integer and then add 1 to it, the value will *wrap around* into the negative numbers. Consequently, if we try increasing values of k on our `fib` method, eventually we will find a value (let's call it k_{max}) that yields the **largest** Fibonacci number that can be correctly contained in a `long` integer variable—let's call that one f_{max}. That is, if we pass $k_{max}+1$ to `fib`, the value returned will appear negative, which is, of course, incorrect.

We seek k_{max}. **Write a method** named `findMaxFib` that accepts no parameters and returns the value of k_{max}—that is, the largest k for which your `fib` method returns a correct result. The method should begin like this:

```
public static int findMaxFib () {
```

4. **Write your `main` method:** If you previously wrote a `main` method to test your previous work,[1], now is the time to delete that code and start the body of `main` anew. Specifically, you should write `main` so that it calls your `findMaxFib` method to obtain k_{max}. It should then call `fib` directly, passing it k_{max} to obtain that largest correct Fibonacci number, f_{max}. Finally, `main` should print both numbers, like so:

```
F(22) = 712371238124
```

[**Warning:** $k_{max} \neq 22$, and $f_{max} \neq 712371238124$. Your program should emit the correct answer in that *format*, but not using those exact values, which are incorrect.]

[1] Always a good idea

2 Finding part B

Once you have the correct f_{max} value, use it to fill in for `XYZ` in the following web address that you should provide to a web browser:

```
https://www.cs.amherst.edu/~sfkaplan/courses/spring-2010/cs11/XYZ.html
```

[**Warning:** Many people, on Project 2, encountered difficulties when copying-and-pasting this web address from this PDF document into their browser. In particular, the tilde character (`~`) seems not to be copied correctly on some types of computers, and thus mangles the web address. At the least, be sure that the tilde character appears correctly before my username (`sfkaplan`); to be truly cautious about this problem, type the entire web address by hand.]

This web page will redirect you to part B of this project.

3 How to submit your work

As usual, use the `cs11-submit` command:

```
cs11-submit project-3a Fibonacci.java
```

Part A is due on Thursday, February 18, at 11:59 pm