INTRODUCTION TO COMPUTER SCIENCE I
LAB 6
Working with arrays

1 Setup

Make a new directory for this lab, change into that directory, and copy a java file:

```
[sfkaplan@remus ~]$ mkdir lab-6
[sfkaplan@remus ~]$ cd lab-6
[sfkaplan@remus ~/lab-6]$ wget -nv -i https://goo.gl/62zJUn
```

The file `Arrays.java` currently contains two methods. Each method takes an `int[]` (a pointer to an array of `int` values) as its parameter and sets all of the values in the given array. The `fillRandom()` method does this by choosing a random number between 0 and 19 (inclusive) for each array element; the `fillKeyboard()` method does this by repeatedly prompting the user to enter the next element. These methods are there to help you test the additional methods that you will write in this lab.

2 Practice Using Arrays

Your job is to write six new methods that manipulate arrays in different ways. After writing each method, you will probably want to add some code to the main method that tests to make sure your methods are working properly.

1. Write a method called `print` that, given an `int[]` parameter, prints the contents of that array. The elements should all print on the same line, with spaces in between them, and with a line break after the last element prints. For example:

   ```
   3 6 9 13 2 6 3 5
   ```

2. Write a method called `fillFibonacci` that, given an `int[]` parameter called `myArray`, fills that array with the first `myArray.length` Fibonacci numbers.[1]

3. Write a method called `addAllElements` that, given an `int[]` parameter, adds up the values of all elements stored in the array and returns the total.

4. Write a method called `average` that, given a `int[]` parameter, computes the average of all elements stored in the array and returns the result. (Hint: use your `addAllElements` method!)

[1] Recall that the Fibonacci sequence is 1, 1, 2, 3, 5, 8, ...: the i^{th} Fibonacci number is obtained by adding the $i-1^{st}$ and $i-2^{nd}$ Fibonacci numbers.

5. Write a method called `countElements` that, given a `int[]` parameter, counts the number of times each number from 0 to 19 appears in the array and prints the results. (Hint: use your `print` method!) For example, if the input array is:

   ```
   3 6 9 13 2 6 3 5
   ```

   then the method should print:

   ```
   0 0 1 2 0 1 2 0 0 1 0 0 0 1 0 0 0 0 0 0
   ```

6. Write a method called `reverse` that, given a `int[]` parameter, reverses the order of the elements in the array. For example, if the input array is:

   ```
   3 6 9 13 2 6 3 5
   ```

   then after running the method the array should contain:

   ```
   5 3 6 2 13 9 6 3
   ```

   Your method should *not* print the reversed array.

3 Submit your work

Submit your modified `Arrays.java` file using either the CS submission web site or the `cssubmit` command.

This assignment is due on Thursday, October 19, 11:59 pm.