EECS2030 Fall 2017
Additional Exercises for Preparing Lab Test 1
Utility Methods

Chen-Wei Wang

– Create a new Java class `MyUtilities` with the following methods:

1. `public static int rps(char p1, char p2)` simulates a rock-paper-scissors game. The precondition is that both `p1` and `p2` are either `r`, `p`, or `s`. Your program returns 1 if player `p1` wins, or 2 if player `p2` wins.
2. `public static int max2(int a, int b, int c)` returns the second maximum of the given three numbers.
3. `public static int max2(int[] a)` returns the second maximum of the given array of numbers. The precondition is that `a` not null and `a` contains at least two numbers.
4. `public static boolean allPositive(int[] a)` returns if all numbers in array `a` are positive. The precondition is that `a` not null.
5. `public static boolean somePositive(int[] a)` returns if there is at least one number in array `a` that is positive. The precondition is that `a` not null.
6. `public static void swap2(int[] a)` which swaps the first and second elements of array `a`. The precondition is that `a` is not null and it contains exactly two elements.
7. `public static int[] fibSeq(int n)` returns the first n numbers in a Fibonacci sequence (i.e., $\langle 1, 1, 2, 3, 5, 8, 13, \ldots \rangle$). The precondition is that `n` is at least one. For example, calling `fibSeq(5)` returns an array $\{1, 1, 2, 3, 5\}$.
8. `public static boolean isFibSeq(int[] seq)` returns if *seq* is a Fibonacci sequence. The precondition is that `a` is not null and it contains at least two elements.
9. `public static int[] arithSeq(int start, int diff, int n)` returns an arithmetic sequence of size n, which starts with number `start` and terms have a common difference `diff`. For example, calling `arithSeq(5, 2, 0)` returns an empty array. Calling `arithSeq(5, 2, 4)` returns an array `{5, 7, 9, 11}`.
10. `public static boolean isArithSeq(int[] seq)` determines if array `seq` is an arithmetic sequence (meaning that you can figure out a common difference between terms). The precondition is that `seq` is not null and it contains at least 2 elements.
11. `public static boolean isSorted(int[] seq)` determines if array `seq` is sorted in a non-deceasing order. The precondition is that `seq` is not null. For example, calling `isSorted({1, 1, 2, 3, 3, 4, 5, 5})` returns true, and calling `isSorted({1, 1, 2, 3, 2, 4, 5, 5})` returns false.
12. `public static int[] reverseOf(int[] seq)` returns the reverse of array *seq*. The precondition is that `a` is not null. For example, calling `reverseOf({1, 2, 3, 4, 5})` returns an array `{5, 4, 3, 2, 1}`.