Faculty of Arts & Sciences
Department of Computer Science
CMPS 200—Introduction to Programming
Assignment 6 – Due Monday Nov 5, 2012

Notes and Announcements

- Reading Material: Sections 7.1-7.4 (pp 449-496)

Exercises

1. ArrayOps

Write the following methods that operate on 1D arrays.

```
// generate and return an array of doubles with n elements filled with random numbers from 0 to 1
public static double[] generate1D(int n)

// generate and return a 1D  array of doubles filled with random numbers between min and max
public static double[] generate1D(int n, int min, int max)

// returns the sum of all the entries of a 1D array
public static double sum(double[] a)

// returns the sum of the entries that are less than the average of the array elements
public static double sumLessThanAverage(double[] a)
```

Write a `main()` method that calls the methods above to test them.

2. MinMax

Write a program that reads N integers from the command line and prints their minimum and maximum. The program should first read the input into an appropriate array and then operate on the array.

```
C:> java MinMax 3 7 -1 90 71
Minimum: -1.
Maximum:  90.
```

3. AlternatingSum

Write a static method `aSum()` with the signature below that computes and returns the alternating sum of the elements in an array:

```
public static int aSum(int[] x)
```

Write a method getInput() that takes an array of strings and returns an array of integers (assuming all the strings can be properly parsed as integers).

```
public static int[] getInput(String[] x)
```

Use the two methods above to write a program `ArraySum.java` that prints out the alternating sum of its arguments. The main method should read a sequence of strings from the command line.

4. Reverse

In this program you are to write two versions of a method that takes an array of integers and reverses it (i.e., puts the last element first, the next to last element second, etc.). The first version of the method reverses the array in place, i.e. it modifies the contents of the array that is passed in. The second version creates a new array and returns it without modifying the contents of the array that is passed in. The signatures of the methods are:

```
public static void reverseInPlace(int[] a)
public static int[] reverseCopy(int[] a)
```

Write a `main()` method that tests these two versions of reverse. The main method should read a sequence of integers from the command line, call the methods above, and print out the appropriate arrays.

5. LongestSortedSequence

Write a program that reads a sequence of integers from the command line and prints the length of the longest sorted (nondecreasing) sequence of integers in the array. For example, in the array [3, 8, 10, 1, 9, -14, -3, 0, 14, 207, 56, 98, 12] the longest sorted sequence had four values -3, 0, 14, 207, so your program should print 4 if passed this array. Sorted means nondecreasing so the sequence could contain duplicates. The program should print 0 if the command line argument is empty. Make sure you decompose the program into appropriate methods.

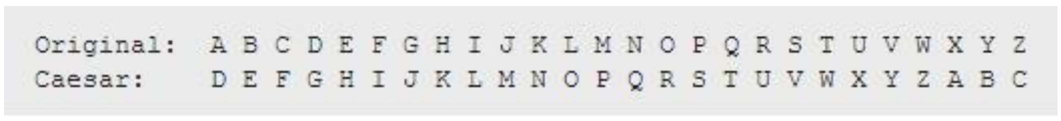
6. WordLengths.

Write a program that counts the number of letters in each word of its command line argument and prints a histogram of word lengths (how many times a word of a given number of letters appears). For example,

```
> java WordLengths how are you feeling today
1: 0
2: 0
3: 3 ***
4: 0
5: 1 *
6: 0
7: 1 *
```

7. Caesar

One of the oldest encryption schemes is attributed to Julius Caesar. Caesar used to send secret messages to his friends using a scheme that is now known as a *Caesar cipher*. Each letter is replaced by the letter k positions ahead of it in the alphabet (and you wrap around if needed). The table below gives the Caesar cipher when k = 3.



```
Original:  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Caesar:    D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

For example the message "VENI, VIDI, VICI" is converted to "YHQL, YLGL, YLFL". Write a program `Caesar.java` that takes a command line parameter k and a string to be encrypted, and applies a Caesar cipher with shift = k to the sequence of letters in the string. You program should handle upper case letters, and leave other characters unchanged. For example,

```
> java Caesar 3 "VENI, VIDI, VICI"
YHQL, YLGL, YLFL
```

There are many ways to solve this problem. But in order to practice arrays, you will to efine an array of 26 characters containing 'A' to 'Z' and write a method that takes this array and returns the cipher array. The program should, for every letter in the input, find the corresponding letter in the cipher array.