American University of Beirut
Department of Computer Science
CMPS 200 – Introduction to Programming
A05 – Deadline: Wednesday March 30 @11:00 PM

Exercise 1. Fill Array

- Write a method **fillArray**() that takes three integers: (***s, min, max***), and returns an array of size ***s*** having random integers with values between ***min and max***.
- Write a method fillTwoDArray() that takes four integers: rows, cols, min, and max. It returns a 2D array with the specified size, and random values between ***min and max***.
- You can use this method in any of the subsequent programs to generate arrays to be used to test your methods.

Exercise 2. Shuffle Array

Write a java method **shuffle(int[] array)** that shuffles the array randomly (hint: use Math.random). One way to do this is to swap every single element of the array with another element chosen randomly (at random index). Write a main method to test your work (Call the method shuffle different times and print out the array after each call).

Exercise 3. Longest Sorted Sequence

Write a java method **longestSorted(int[] array)** that takes an integer array and returns the length of the longest sequence of non-decreasing elements.

For example, an input of: array = {2, 5, 9, 9, 4, 11, -1, 10, 23, 23, 87, 11} should return 5 (longest sorted sequence: -1, 10, 23, 23, 87)

Exercise 4. Remove Duplicates

write a java method removeDuplicates() that takes a sorted array, and removes all duplicates integers in the array. For example, and input of [0, 0, 0, 1, 1, 3, 4, 4, 4, 4, 6, 6] should return [0, 1, 3, 4, 6]

Exercise 5. Array Sorting

- Write a function **getMinMax**() that, given an array of integers and two indexes, returns the minimum and the maximum values in the array bound by those indexes (built-in functions to manipulate any array may NOT be used at all).
  For example, if the array contains the values [5, 12, -3, 1, 0, - 11, 29, 8] and the indices were 2 and 5 then the function returns, somehow, -11 and 1 (since - 11 is the minimum and 1 is the maximum in the part of the array that starts at index 2 and ends at index 5).

Write a program that:

- Fills an array with ten random integers between 1 and 100 (using the function fillArray() implemented before).
- Using the function getMinMax(), your program should sort the array as follows:
  - It should use two indices, say i and j, and point i to the first element and j to the last one in the array.

- It should then call the function getMinMax() to find the minimum and maximum values in the range bound by i and j.
- Swap the first and last elements in the range [i, j] with the values returned by the function.
- Move i one location forward and j one location backward and repeat.
- Once i and j coincide or bypass each other, you should stop as the array is now sorted.

- Print the content of your sorted array using a new function that prints ANY array on integers (row by row).

Exercise 6. Dice Game

1. **Dice**. Write a program `Dice.java` that shows a pair of dice. The program prompts the user to enter a choice from the standard input:

   (1) **Play**: the dice should be rolled (that is, the dice should be assigned newly computed random values). Notice that there are 11 different combinations for the sum of the two dice. Your program should count the number of times of each sum.

   (2) **Exit:** before to exit you have to display the number of times of each sum.

```
Sum equals to:      2   3   4   5   6   7   8   9  10  11  12
Number of Times:    0   0   0   2   0   0   2   0   0   0   0
```

Exercise 7. Matrix Multiplication

Consider two matrices A and B of respective sizes (m × n) and (p × q). For matrix multiplication to be feasible, the inner dimensions must agree in the sense that n has to be equal to p. In this case, matrix multiplication can be performed and would result in a matrix, say R, that is of size (m × q). Denote by a(i,j), b(i,j) and r(i, j) to be the respective entries of matrices A, B and R, all located at row i and column j. The value of r(i,j) can be computed in such a way that:

$$r(i,j) = \sum_{k=1}^{n} a(i,k) \cdot b(k,j) \quad , \quad \forall\, i \in [1;m], \ \forall\, j \in [1,q]$$

Implement a JAVA method that takes two matrices A and B as input parameters and, if feasible, returns a matrix R resulting from the multiplication of A by B. In the case where the multiplication is not feasible, the method must return null. Test the correctness and validity of the above-implemented method by writing a main function. Print out your resulting operation to the screen. A sample output is provided next:

```
M1 = |4   5   2|
     |6   3   8|

M2 = |1   5|
     |1   2|
     |3   2|

M1 * M2 = |15    34|
          |33    52|
```