CSE101L Computer Programming Laboratory Quiz 2

Disclosure

You will submit your java file to an assignment that is given through MS Teams. Your filename **MUST BE** *Quiz2_{yourStudentNumber}.java*. Change the parentheses to your own student number. For example, your filename should look like this:

Quiz2_202051056016.java

Quiz will start at 14.45, you will have an hour to code then you will be given extra five minutes to finalize the upload process. Make sure that your file does not have any compile-time error. Any kind of code sharing, using internet, or previously written code snippets is forbidden. During the quiz, you can only open this pdf that you are currently reading and a code editor with only a single java file, that will be your quiz file opened, and nothing else.

You can use any editor you would like to use. DO NOT use any Turkish character, and if your editor crates a package automatically, remove that line (usually at the very top) BEFORE turning in your java file. Make sure that your uploaded file is a java file and not something else (for example, Quiz2_202051056016.java.iml, this is not a java file)

You do not have to have a main method, although you can use one to test your written methods. You can write and use any additional method you would like to, that is not mentioned in here. You will only be evaluated for the following methods that you will write. **All methods must be static.**

Questions

1. **Eliminate duplicates:** Write a method that returns a new array by eliminating the duplicate values in the array. Order of appearances for numbers must not change.

    - **Name:** eliminateDuplicates
    - **Args:** *int[]:* array containing duplicate elements
    - **Returns:** *int[]:* new array that does not contain duplicate elements, but keeps the order of appearances of numbers.

2. **All closest pairs of points:** Write a method that returns the array of indices of closest points to the given point. If *m* points share the same minimum distance to the given point, then your array will have all those points.

    - **Name:** getClosestPoints
    - **Args:**
        - *double[n][d]:* array of *n* points in *d* dimensional space
        - *int:* index of a point from *d* dimensional points array that we will search for closest *m* points
    - **Returns:** *int[m]:* array of indices of points that share the same minimum distance to given point

Hint: Calculate distances to one decimal place sensitivity.

3. **Weekly work hours:** Suppose the weekly hours for all employees are stored in a two-dimensional array. Each row records an employee's seven-day work hours.

   Write a method that sorts the employee's according to their sum of working hours for an entire week in decreasing order. Finally returns the sum of working hours for each employee. Sort both returned array and given employees array.

   - **Name:** employeeWorkingHours
   - **Args:** *int[8][7]:* the array of employees. Each row contains an employee's seven-day work hours.
   - **Returns:** int[8]: the array of sum working hours for each employee in decreasing order.

Hint: A single index will represent the same employee in both arrays.

4. **Random number chooser:** Write a method that returns a random number between given minimum and maximum, excluding the numbers given in variable arguments of the method. If a number generated is one of the excluded numbers, then generate and return another number.

   - **Name:** random
   - **Args:**
     - *int:* minimum number to generate
     - *int:* maximum number to generate
     - *int[optional]*: variable amount of excluded numbers
   - **Returns:** *int:* a randomly generated number between minimum and maximum (inclusive)

Hint: A user of this method should be able to use it with at least two, or more number of parameters.