Programming Assignment 5

Due: March 18 at 11:59 pm

Exercise 1 (15 points)

Write a **class DataSet** that stores a number of values of type double. Use a regular array (not ArrayList) as an instance variable to store the values. You will also need to store the current size of your dataset (it is different from the maximum number of values). Provide a constructor

public DataSet(int maximumNumberOfValues)

and a method

public void add(double value)

that adds a value, provided there is still room.

Provide methods to compute the **sum**, **average**, **maximum**, and **minimum** value. Test all functionality in main method

Exercise 2 (15 points)

Create a **class Matrix** and write two methods

First method:

that sums all the numbers in the major diagonal in an n x n matrix of int values using the following header:

public int sumMajorDiagonal(int[ ][ ] m)

or if you want to store matrix as instance variable of Matrix class:

public int sumMajorDiagonal()

Second method:
that transpose a matrix so that columns become rows.
header:
public int[ ][ ] transposeMatrix(int[ ][ ] m)   or public int[ ][ ] transposeMatrix()

$$\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}$$

Input

$$\begin{bmatrix}1 & 4 & 7\\ 2 & 5 & 8\\ 3 & 6 & 9\end{bmatrix}$$

Output

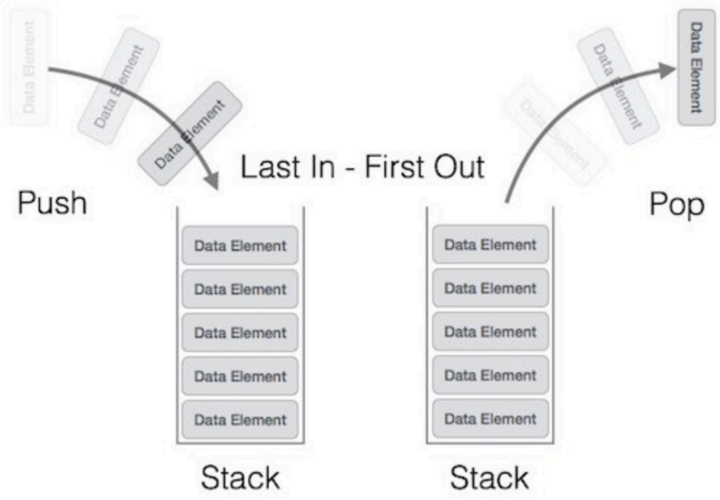
For testing generate a square matrix (4 by 4) and populate it with random values (using Random class and nextInt method), print your matrix and the result from sumMajorDiagonal and transposeMatrix methods

Exercise 3 (15 points)

For this exercise you will need to write your own data structure Stack, which add and remove values in the following order: last in - first out.

Implement **class MyStack** for storing and manipulating integers. Fill all the methods inside the initial code MyStack.java. Write test program to test the functionality. You do not need to provide any constructor (by default it will create an empty stack). Test your MyStack class in main method.

MyStack has an instance variable ArrayList<Integer> list to store elements.
public boolean isEmpty() method returns true if this stack is empty
public int getSize() method returns the number of elements in this stack
public int peek() method returns the top (the last added) element in this stack without removing it
public int pop() method returns and removes the top (the last added) element in this stack
p**ublic void push(int value)** method add a new element to the top (the end of the list) of this stack.



Submission

1. Push your project directory along with the source to GitHub by the due date.
2. Invite and share your project repository the Grader (yan.chen01@sjsu.edu) and Instructor (**mariia.surmenok@sjsu.edu**) or by usernames: **yanchen-01** and **msurmenok**
3. Submit a Readme.txt to Canvas including your name, repository access link, instructions to run your program (if any), **snapshot of your running program (screenshot of your output)** and citations (if any)
4. Your project directory will be graded according to the state your project directory was in at due time.