Lab 2

1. Write a Java program that reads a positive integer **N** and prints all **prime numbers** between 1 and **N**. [You must validate the input using do while statement until a positive integer is entered].

   Example1
   Enter N: -50
   You should enter a positive integer number: -100
   You should enter a positive integer number: 14
   The prime numbers between 1 and 14: 2, 3, 5, 7, 11, 13
   Example2
   Enter N: 1
   There are no prime numbers between 1 and 1.
   Example3
   Enter N: 0
   You should enter a positive integer number: 20
   The prime numbers between 1 and 20: 2, 3, 5, 7, 11, 13, 17, 19

2. Create a Java class named **CustomSet** to represent a set of integers. A set stores any number of distinct integers. The class will contain two private members, **data** and **size**, and two member methods**, contains** and **union**.
   The class will also have a constructor that takes a list of the initial set members as a parameter. **Deduplication should be handled properly**.
   The method (**contains)** takes an integer parameter and returns a boolean indicating whether the number is a member of the set.
   The method (**union**) takes a parameter of type **CustomSet** and returns an object of type CustomSet.

   It is important that the class includes getter and setter methods for its member variables.

Dr. Layla Abou-Hadeed

Eng. Ahmed ElSayed
Eng. Ali Hassan
Eng. Ahmed Ashraf
Eng. Seif Eldin Mahmoud
Eng. Miar Mamdouh
Eng. Ahmed AboEleid
Eng. Ahmed Essam
Eng. Menna Tullah Ihab
Eng. AbdElaziz Mohamed
Eng. Mahmoud Ramzy
Eng. AbdElrahman ElSayed

Create another class named **Lab2_p2** with a main function where you will create two objects of the **CustomSet** class. The data for both will be provided by the user. You should also create a third **CustomSet** which is the union of the first two, also print whether the union contains a prime number.

Example

Please enter the members of the first set: 2, 5, 9, 7, 9
Please enter the members of the second set: -8, 3, 3, 4, 7, 1, 640
The union of the two sets is [-8,1,2,3,4,5,7,9,640]
The prime numbers that are members of the union: 2, 3, 5, 7

3. Create a class named **Invoice** that contains three private member variables:

a. an integer variable named **invoiceNumber.**
b. a string variable named **customerName.**
c. an array of double variables named **prices.**

The class will have only one constructor that takes 3 arguments, one for each member variable. The class also contains two member methods named **report** and **calculateInvoice**. The (**report)** method does not accept any parameters and does not return a value. The (**calculateInvoice)** method takes no parameters and returns a value of type double. The (**report)** method will display the data of the calling object. The (**calculateInvoice)** method will compute and return the total cost after applying the discount based on the following table:

Discount rate %	Total cost
1	=> 1000
4	=> 5000
6	=> 8000
8	=> 10000

The class should contain getters and setters methods for each member variable. Create another class named **Lab2_p3** with a main function where you will create two objects of the class **Invoice** and read their data from the user. Then, their data is displayed as shown in the example:

Example

Enter account 1 data:
Invoice number: 1
Customer name: Ahmed
Number of elements: 3
Prices: 150.2, 250.36, 845
Enter account 2 data:
Invoice number: 2
Customer name: Ali
Number of elements: 4
Prices: 188.9, 365.36, 410, 145.78

The data of the first account:
Account number: 1
Name: Ahmed
Total cost: 1245.56
The total cost after discount = 1233.10
The data of the second account:
Account number: 2
Name: Ali
Total cost: 1110.04
The total cost after discount = 1098.93

4. Using the **Invoice** class you have defined in the previous problem, write a Java class named **Lab2_p4** that contains two functions, the main function and a function called **readInvoiceDetailsFromFile**. The header of readInvoiceDetailsFromFile function should be:
public static Invoice[] readInvoiceDetailsFromFile(String fileName)

   The function should read the invoice data from the file specified by the fileName variable, store the data in an array of type **Invoice (previously defined in the prior problem)**, and return this array.
   In the main function, first read the name of the file that contains the invoice data. Then, use the readInvoiceDetailsFromFile function to load this data into an array. After that, sort the array by total cost (after applying discounts) in descending order and print the results. Finally, collect new invoice data from the user and add it to the same file from which the original data was read.
   The file will have the following structure:
   - First line will be an integer number representing the number of invoices stored in the file.
   - Each line of the next lines represents the data of one invoice comma separated.
     (Ex: the line 1,Ahmed,3,150.2,250.36,845 represent data of a banking account whose number is 1, owner name is Ahmed, and the (3 products) prices (150.2,250.36,845) which mean total cost before discount is 1245.56).

Required:

1. You are required to solve **all** the above questions and deliver them online through a Google form that will be available for you in the next few days.
2. The deadline for the delivery is **Friday, October 11, 2024, at 11 PM**.
3. A discussion is made with you at your lab time next week on what you have delivered.

What to be delivered:

- On the Google form, you should deliver a zipped file that contains the .java files.
- Your zip file should be named as id_groupNumber. For example, 4678_G2.

Policies:

- You should work individually.
- Delivering a copy will be severely penalized for both parties, so delivering nothing is so much better than delivering a copy.
- No late submission is allowed

Dr. Layla Abou-Hadeed

Eng. Ahmed ElSayed
Eng. Ahmed Ashraf
Eng. Miar Mamdouh
Eng. Ahmed Essam
Eng. AbdElaziz Mohamed
Eng. AbdElrahman ElSayed
Eng. Ali Hassan
Eng. Seif Eldin Mahmoud
Eng. Ahmed AboEleid
Eng. Menna Tullah Ihab
Eng. Mahmoud Ramzy