JAVA ASSIGNMENT

Problem 1 -> Take the department names as input from the user and then sorting the departments using the Comparator interface and using the lambda expressions.

Problem 2 -> To connect the program to a database and to store the department names in the database using JDBC, making use of POM.xml to do dependency injection.

Problem 3 -> Given a list of integers, sort all the values present in it in descending order using Stream functions?

10,15,8,49,25,98,98,32,15

Problem 4 -> Given a list of integers, find the maximum value element present in it using Stream functions?

0,15,8,49,25,98,98,32,15

Problem 5 ->

Description:

Develop an application to generate Two Contact Reports, Old Friends and New Friends Reports.

The application starts by reading Friend information from a file “AllFriendsFile” (Name – year ofmeeting), then asking the user if he/she wants to add each friend to the list.

- If the user wants to keep a specific friend, then the program will ask for the friend’s city.

- If the user does not want to keep the friend with his/her list, that friend will be ignored.
  Now, for each friend, we have Name, year of meeting read from the file, and city read from the keyboard.
  The program will create a Friend object for each friend.

If the friend’s year of meeting is before 2000, then the friend information will be printed to “OldFriendReport”. Otherwise, if the year of meeting is after 2000, the friend information will be printed to “NewFriendReport”.
You should create and use the following classes: o MainApp

Example:

▪ Implement the main logic for the problem using the developed classes and methods.

o Contact Class

- ▪ ·Should have fields like name, yearOfMeeting and city.
- ▪ ·Accessor and mutator methods for the fields
- ▪ ·Constructor to set default values to fields.
- ▪ ·Print function to return a string containing the contact information.

AllFriendsFile contains this list:

Mary 2019 Sharon 1988 Jone 2015 Lee 2000 Smith

1990

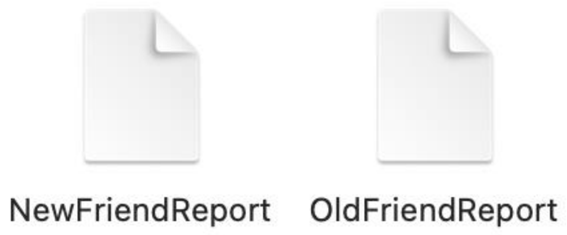
Sample implementation:

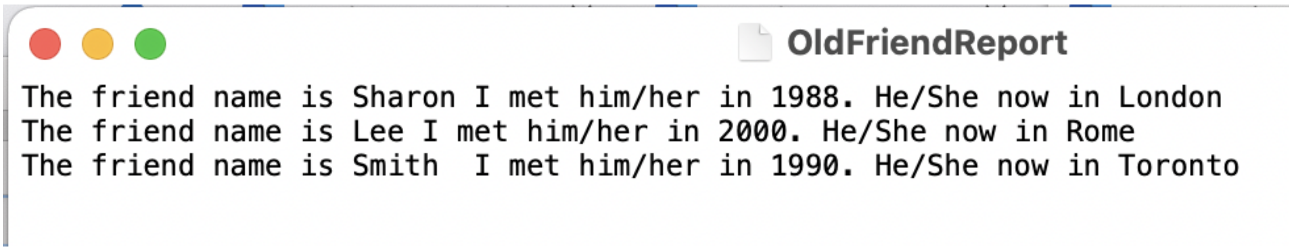
What is the file name that has friends' info
Friends
Do you want to add Mary To your new Contact List? y

Enter the address of Mary
Toronto
Do you want to add Sharon to your new Contact List? y
Enter the address of Sharon
London
Do you want to add Jone to your new Contact List?
n

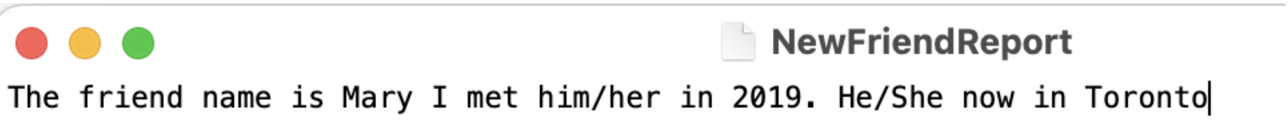
Do you want to add Lee To your new Contact List?
y
Enter the address of Lee
Rome
Do you want to add Smith To your new Contact List? y
Enter the address of Smith
Toronto

After finishing this implementation, we should have this output in the two reports: OldFriendReport





```
The friend name is Sharon I met him/her in 1988. He/She now in London
The friend name is Lee I met him/her in 2000. He/She now in Rome
The friend name is Smith  I met him/her in 1990. He/She now in Toronto
```



```
The friend name is Mary I met him/her in 2019. He/She now in Toronto
```