Project 11: Examination Extra Credit

Due on Monday, November 15 by 11:59 p.m.

Description: Since the last examination was a bit too long for many people to finish, I'm giving you an opportunity to earn some point back. The number of points you can earn back depends on your examination grade. The details of how many points you can receive were sent out in an email on 11/7/21. Point will be awarded based on what percentage of the 100 points of this project you complete. So if you could earn 10 points back and you complete 50 points of the project, you would receive 5 points.

Objectives: Your program will be graded according to the rubric below.

1. Write the final program from the CS 1323 Midterm 2 and submit it to the Project 11a Zylab.
    a. Write the countInRange method. (10 points)
    b. Write the readFile method. (10 points)
    c. Write the summarizeDataByHour method. (15 points)
    d. Write the main program. (15 points)
2. Write the final program from the CS 1324 Midterm 2 and submit it to the Project 11b Zylab.
    a. Write the contains method. (10 points)
    b. Write the findMutualFriend method. (15 points)
    c. Write the readArrayFromFile method. (10 points)
    d. Write the main program. (15 points)

Note: the readFile and readArrayFromFile methods are somewhat similar, but not identical. The format in which the data is stored is different. Pay attention to the details to avoid frustration.

CS 1323 Midterm 2 Program

VisageBook has hired you to write a program to analyze when individuals tend to post. They have files that contain the times that each individual user posts, represented as doubles on a 24-hour clock. This means that 7.25 is actually 7:15 a.m., which 18.50 would be 6:30 p.m. The program should accept the file name, count the number of times that this particular user posted in one-hour increments. If there were no postings during this time, the count should not be shown.

A sample run of the program is below. User input is given in bold, italics and underline. Notice that the program is using 0 for 12.

```
Enter the name of the file that contains the usage data
Person.txt

Usage Summary
4 to 5 there were 1 entries
7 to 8 there were 1 entries
8 to 9 there were 1 entries
9 to 10 there were 1 entries
10 to 11 there were 1 entries
11 to 12 there were 1 entries
15 to 16 there were 3 entries
19 to 20 there were 1 entries
23 to 24 there were 2 entries
```

This program must be implemented using the methods below.

The file Person.txt contained:

```
12
8.3
9.1
11.5
4.6
15.2
7.9
10.4
15.8
19.2
15.0
23.4
23.6
```

public static int countInRange(double[] data, int lower, int upper): This method takes a perfect size array of doubles and returns the number of times that a value between lower (inclusive) and upper (exclusive) occurs. For example, if data contained {1.1, 2.2, 3.3, 4.4} and lower was 2 and upper was 6, the method would return 3 (because 2.2, 3,3 and 4.4 are in that range).

public static double[] readFile(String fileName): This method reads a file of the given format into a perfect size array of doubles. For example, if the file contained four lines: 3, 2.1, 2.2, 2.3, the returned array would contain {2.1, 2.2, 2.3} and be of length 3.

public static int[] summarizeDataByHour(double[] data): This method takes a perfect size array of doubles and returns a perfect size array of integers. The returned array will have indices from 0 to 23. In index 0 will contain the number of times that the data array had a value between 0 (inclusive) and 1 (exclusive) in the data array. Index 1 will contain the number of times that the data array had a value between 1 (inclusive) and 2 (exclusive) in the data array. And so on. This method can be implemented with or without calling countInRange. It does not require a nested loop, although you may use one if you wish to. For example, if data contained {1.1, 2.2, 3.3, 3.5, 3.6, 4.0} then the returned array would contain: {0, 1, 1, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

public static void main(String args). This method must call readFile and summarizeDataByHour.

The class name must be Project11a.

CS 1324 Midterm 2 Program

You are employed by a social media company called VisageBook. Visage book keeps track of who your social media friends are, among other things. The company keeps track of which users are friends by having a file with the user's name contain a list of their friends. We will foolishly assume that no two people have the same name and that all names are entered perfectly at the keyboard. For example: the file named "Puss And Boots.txt" contains:

```
Shrek The Ogre
Fiona The Princess
Donkey The Donkey
```

Visagebook has asked you to write a program that determines whether two users have any mutual friends.

A sample run of the program is below. User input is in bold and underlined.

```
Enter your name
Puss And Boots
Enter your friend's name
Shrek The Ogre
Fiona The Princess is a mutual friend
Enter your friend's name
Lord Farquad
You two do not have a mutual friend
Enter your friend's name
Quit
Bye
```

```
"Shrek The Ogre.txt" contains:
Fiona The Princess
The GingerBread Man
Cinderella The Princess
Robin Hood
```

```
"Lord Farquad.txt" contains:
Prince Charming
The Dragon
The Fairy Godmother
The Talking Mirror
```

The program uses four methods:

public static boolean contains(String fileName, String element): This method returns true if element is on a line by itself in the file called fileName, and false otherwise. For example, if the fileName was "sample.txt" and the element was "Third Line" and sample.txt contained:

First Line
Second Line
Third Line

The method would return true. If element was "Fourth Line", however, the method would return false.

public static String findMutualFriend(String[] mine, String fileName) This method returns a String containing the name of a person that is found both in the array called mine and as a line in the file called fileName. If no mutual friend is found, then the method returns null. For example, if the filename was "sample.txt" and the file contained:

First Line
Second Line
Third Line

And the array mine contained {"Fourth Line", "First Line", "Fifth Line"} then "First Line" would be returned. If the array mine contained {"Fourth Line", "Fifth Line", "Sixth Line"} then null would be returned.

This method should call the contains method above, although it may also be written other ways.

public static String[] readArrayFromFile(String fileName) This method returns a perfect size array that contains the lines of the file named fileName, one line to each array element.

public static void main(String[] args). The main program must call the readArrayFromFile method and the findMutualFriend method, written above.

The class name must be Project11b.