OOPI Final Exam

This exam is to be done in personal, please do not discuss your answer with others. Plagiarism is strongly prohibited!

Due: 06/25/2021 23:00

You will get 0 points on this exam for late submission.

Description

In this exam, we will exercise your acquired coding skills by using Java language.

Hand in format

The file which contain main function should be named as studentID_Question.java

```
A1073314_Q1.java

A1073314_Q2.java

……
```

And you need to create a folder name as studentID_ OOPI_Final_Exam, put all the java file into it, and compressed the folder as a zip file, then upload the zip file on google classroom.

```
A1073314_OOPI_Final_Exam.zip
```

Way of reviewing

The command of execute and compile:

```
javac studentID_Question.java

java studentID_Question
```

We will only execute and compile the file named as studentID_Question.java in the linux environment.
If the file doesn't work properly you will get zero for this question, so please make sure your code runnable on the VM we provided, and the output format same as we expected.

Questions

1. (5pts) There are n people in a room, where n is an integer greater than or equal to 2. Each person shakes hands once with every other person. What is the total number of handshakes in the room? Write a recursive method to solve this problem with the following header:

```
public static int handshake(int n)
```

where handshake(n) returns the total number of handshakes for n people in the room. To get you started, if there are only one or two people in the room, then:

```
handshake(1) = 0

handshake(2) = 1
```

Expected output:

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java Handshake
If there are 0 people in the room then there is a total of 0 handshakes.
If there are 1 people in the room then there is a total of 0 handshakes.
If there are 2 people in the room then there is a total of 1 handshakes.
If there are 3 people in the room then there is a total of 3 handshakes.
If there are 4 people in the room then there is a total of 6 handshakes.
If there are 5 people in the room then there is a total of 10 handshakes.
If there are 6 people in the room then there is a total of 15 handshakes.
If there are 7 people in the room then there is a total of 21 handshakes.
If there are 8 people in the room then there is a total of 28 handshakes.
If there are 9 people in the room then there is a total of 36 handshakes.
```

2. (6pts) Add a method bubbleSort to the class ArraySorter, as given in Listing 7.10, that performs a bubble sort of an array. The bubble sort algorithm examines all adjacent pairs of elements in the array from the beginning to the end and interchanges any two elements that are out of order. Each interchange makes the array more sorted than it was, until it is entirely sorted. The algorithm in pseudocode follows:

```
Bubble sort algorithm to sort an array a

Repeat the following until the array a is sorted:

for (index = 0; index < a.length − 1; index++)

    if (a[index] > a[index + 1])

        Interchange the values of a[index] and a[index + 1].
```

The bubble sort algorithm usually requires more time than other sorting methods.

Expected output:

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution/2$ java BubbleSortDemo
Array values before sorting:
7 5 11 2 16 4 18 14 12 30
Array values after sorting:
2 4 5 7 11 12 14 16 18 30
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution/2$
```

3. (9pts) Write a class TelephoneNumber that will hold a telephone number. An object of this class will have the attributes

   - areaCode—a three-digit integer
   - exchangeCode—a three-digit integer
   - number—a four-digit integer

   and the methods

   - TelephoneNumber(aString)—a constructor that creates and returns a new instance of its class, given a string in the form xxx-xxx-xxxx or, if the area code is missing, xxx-xxxx. Throw an exception if the format is not valid. Hint: To simplify the constructor, you can replace each hyphen in the telephone number with a blank. To accept a telephone number containing hyphens, you could process the string one character at a time or learn how to use Scanner to read words separated by a character—such as a hyphen—other than whitespace.
   - toString—returns a string in either of the two formats shown previously for the constructor.

   Using a text editor, create a text file of several telephone numbers, using the two formats described previously. Write a program that reads this file, displays the data on the screen, and creates an array whose base type is TelephoneNumber. Allow the user to either add or delete one telephone number. Write the modified data on the text file, replacing its original contents. Then read and display the numbers in the modified file.

   Example format of text file:

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution/3$ cat numbers.txt
555-2345
203-555-0230
330-012-0001
123-4567
```

   Expected output:

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution/3$ java TelephoneProgram
Read the number 555-2345
Read the number 203-555-0230
Read the number 330-012-0001
Number 0 is 555-2345
Number 1 is 203-555-0230
Number 2 is 330-012-0001
Change a number (c) or add a number (a)
a
Enter the phone number to add
111-222-3333
The file has been written back.
```

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution/3$ java TelephoneProgram
Read the number 555-2345
Read the number 203-555-0230
Read the number 330-012-0001
Read the number 111-222-3333
Number 0 is 555-2345
Number 1 is 203-555-0230
Number 2 is 330-012-0001
Number 3 is 111-222-3333
Change a number (c) or add a number (a)
c
Which number do you want to change (enter the index)
3
Enter the phone number
111-222-444
Sorry, bad format for the numberNumber is not 4 digits
The file has been written back.
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution/3$ java TelephoneProgram
Read the number 555-2345
Read the number 203-555-0230
Read the number 330-012-0001
Read the number 111-222-3333
Number 0 is 555-2345
Number 1 is 203-555-0230
Number 2 is 330-012-0001
Number 3 is 111-222-3333
Change a number (c) or add a number (a)
c
Which number do you want to change (enter the index)
3
Enter the phone number
123-4567
The file has been written back.
```

4. (6pts) Write a program that determines the change to be dispensed from a vending machine. An item in the machine can cost between 25 cents and a dollar, in 5-cent increments (25, 30, 35, . . ., 90, 95, or 100), and the machine accepts only a single dollar bill to pay for the item. For example, a possible dialogue with the user might be

```
Enter price of item

(from 25 cents to a dollar, in 5-cent increments):

45

You bought an item for 45 cents and gave me a dollar,
```

so your change is

2 quarters,

0 dimes, and

1 nickel.

Expected output:

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java VendingChange
Enter price of item
(from 25 cents to a dollar, in 5-cent increments):
30
You bought an item for 30 and gave me a dollar.
so your change is
2 quarters,
2 dimes, and
0 nickel.
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java VendingChange
Enter price of item
(from 25 cents to a dollar, in 5-cent increments):
55
You bought an item for 55 and gave me a dollar.
so your change is
1 quarters,
2 dimes, and
0 nickel.
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java VendingChange
Enter price of item
(from 25 cents to a dollar, in 5-cent increments):
95
You bought an item for 95 and gave me a dollar.
so your change is
0 quarters,
0 dimes, and
1 nickel.
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$
```

5. (7pts) A palindrome is a string that reads the same forward and backward, such as "radar". Write a static recursive method that has one parameter of type String and returns true if the argument is a palindrome and false otherwise. Disregard spaces and punctuation marks in the string, and consider upper- and lowercase versions of the same letter to be equal. For example, the following strings should be considered palindromes by your method:

"Straw? No, too stupid a fad, I put soot on warts."

"xyzcZYx?"

Your method need not check that the string is a correct English phrase or word. Embed the method in a program, and test it.

Expected output:

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java PalindromeTestDemo
Enter a string
hieveryone

It is NOT a palindrome

Enter another sentence? (y/n)
y
Enter a string
eye

The sentence is a palindrome

Enter another sentence? (y/n)
n
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$
```

6. (7pts) Write a program that reads a string from the keyboard and tests whether it contains a valid date. Display the date and a message that indicates whether it is valid. If it is not valid, also display a message explaining why it is not valid. The input date will have the format mm/dd/yyyy. A valid month value mm must be from 1 to 12 (January is 1). The day value dd must be from 1 to a value that is appropriate for the given month. September, April, June, and November each have 30 days. February has 28 days except for leap years when it has 29. The remaining months all have 31 days each. A leap year is any year that is divisible by 4 but not divisible by 100 unless it is also divisible by 400.

   Expected output:

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java CheckDate
Please enter a date to be checked
01/30/2020
date is 1:30:2020
Your date was 01/30/2020
It is a valid date.
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java CheckDate
Please enter a date to be checked
02/29/2021
date is 2:29:2021
Your date was 02/29/2021
It is not a valid date.
The reason it is invalid: The day value is greater than 28 in February in a non leap year.
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java CheckDate
Please enter a date to be checked
02/29/2020
date is 2:29:2020
Your date was 02/29/2020
It is a valid date.
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java CheckDate
Please enter a date to be checked
06/31/2021
date is 6:31:2021
Your date was 06/31/2021
It is not a valid date.
The reason it is invalid: The day value is greater than 30 in a month with just 30 days.
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$
```

7. (9pts) Write a class ContactInfo to store contact information for a person. It should have attributes for a person's name, business phone, home phone, cell phone, email address, and home address. It should have a toString method that returns this data as a string, making appropriate replacements for any attributes that do not have values. It should

have a constructor ContactInfo(aString) that creates and returns a new instance of the class, using data in the string aString. The constructor should use a format consistent with what the toString method produces. Using a text editor, create a text file of contact information, as described in the previous paragraph, for several people. Write a program that reads this file, displays the data on the screen, and creates an array whose base type is ContactInfo. Allow the user to do one of the following: change some data in one contact, add a contact, or delete a contact. Finally, write over the file with the modified contacts.

Expected format of text file and output:

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution/7$ java ContactProgram
contact 0 is NAME:bob PHONE(BUSINESS): 555-1212 PHONE(HOME):  PHONE(CELL):  EMAIL: bob@funky.com ADDRESS:
contact 1 is NAME:fred PHONE(BUSINESS):  PHONE(HOME): 333-3456 PHONE(CELL):  EMAIL:  ADDRESS: Lover's Lane
contact 2 is NAME:susan PHONE(BUSINESS):  PHONE(HOME):  PHONE(CELL): 555-2135 EMAIL:  ADDRESS: easy street
Change a contact (c)  add a contact (a) or delete a contact (d)
a
Enter the contact info to add
jone# # # 111-2345# # Unknown
The file has been written back.
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution/7$ cat contacts.txt
bob# 555-1212# # # bob@funky.com#
fred# # 333-3456# # # Lover's Lane
susan# # # 555-2135# # easy street
jone# # # 111-2345# # Unknown
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution/7$ java ContactProgram
contact 0 is NAME:bob PHONE(BUSINESS): 555-1212 PHONE(HOME):  PHONE(CELL):  EMAIL: bob@funky.com ADDRESS:
contact 1 is NAME:fred PHONE(BUSINESS):  PHONE(HOME): 333-3456 PHONE(CELL):  EMAIL:  ADDRESS: Lover's Lane
contact 2 is NAME:susan PHONE(BUSINESS):  PHONE(HOME):  PHONE(CELL): 555-2135 EMAIL:  ADDRESS: easy street
contact 3 is NAME:jone PHONE(BUSINESS):  PHONE(HOME):  PHONE(CELL): 111-2345 EMAIL:  ADDRESS: Unknown
Change a contact (c)  add a contact (a) or delete a contact (d)
c
Which contact do you want to change (enter the index)
3
The contact is NAME:jone PHONE(BUSINESS):  PHONE(HOME):  PHONE(CELL): 111-2345 EMAIL:  ADDRESS: Unknown
Which field to you want to change?
PHONE(CELL)
Please enter the new cell phone number
223-4567
The file has been written back.
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution/7$ cat contacts.txt
bob# 555-1212# # # bob@funky.com#
fred# # 333-3456# # # Lover's Lane
susan# # # 555-2135# # easy street
jone# # # 223-4567# # Unknown
```

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution/7$ java ContactProgram
contact 0 is NAME:bob PHONE(BUSINESS): 555-1212 PHONE(HOME):  PHONE(CELL):  EMAIL: bob@funky.com ADDRESS:
contact 1 is NAME:fred PHONE(BUSINESS):  PHONE(HOME): 333-3456 PHONE(CELL):  EMAIL:  ADDRESS: Lover's Lane
contact 2 is NAME:susan PHONE(BUSINESS):  PHONE(HOME):  PHONE(CELL): 555-2135 EMAIL:  ADDRESS: easy street
contact 3 is NAME:jone PHONE(BUSINESS):  PHONE(HOME):  PHONE(CELL): 223-4567 EMAIL:  ADDRESS: Unknown
Change a contact (c)  add a contact (a) or delete a contact (d)
d
Which contact do you want to delete (enter the index)
3
The file has been written back.
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution/7$ cat contacts.txt
bob# 555-1212# # # bob@funky.com#
fred# # 333-3456# # # Lover's Lane
susan# # # 555-2135# # easy street
```

8. (7pts) Imagine a candy bar that has k places where it can be cut. You would like to know how many different sequences of cuts are possible to divide the bar into pieces. For example, if k is 3, you could cut the bar at location 1, then location 2, and finally at location 3. We indicate this sequence of cuts by 123. So if k is 3, we have six ways to divide the bar: 123, 132, 213, 231, 312, or 321. Notice that we have k possibilities for making the first cut. Once we make the first cut we have k – 1 places where a cut must be made. Recursively, this can be expressed as

$$C(k) = kC(k - 1)$$

Let's make this a bit more interesting by adding a restriction. You must always cut the leftmost pieces that can be cut. Now if k is 3, we can cut the bar at locations 123, 132, 213, 312, or 321. A cutting sequence of 231 would not be allowed, because after the cut at 2 we would have to make the cut at location 1, since it is the leftmost piece. We still have k possibilities for making the first cut, but now we have to count the number of ways to cut two pieces and multiply. Recursively, this can be expressed as

$$D(k) = \sum_{i=1}^{k} D(i-1)D(k-i)$$

When k is 3, we would compute

$$D(3) = \sum_{i=1}^{3} D(i-1)D(3-i) = D(0)D(2) + D(1)D(1) + D(2)D(0)$$

$$D(2) = \sum_{i=1}^{2} D(i-1)D(2-i) = D(0)D(1) + D(1)D(0)$$

$$D(1) = \sum_{i=1}^{1} D(i-1)D(1-i) = D(0)D(0)$$

For both recursive formulas, if k = 0, there is exactly one way to divide the bar.

Develop a program that will read a value of k from the keyboard and then display C(k) and D(k). (D(k) is interesting because it turns out to be the number of ways that we can parenthesize an arithmetic expression that has k binary operators.)

Expected output:

```
a1073316@2021_DBMS_practice:~/OOP17/Final_Exam/solution$ java Cuts
Computing the number of ways you can cut up a candy bar.
Enter an integer value k (Number of places you can cut the bar):
3
The number of ways to cut the bar with cuts in any order: 6
The number of ways to cut the bar with cuts only in the leftmost piece: 5
```

9. (7pts) Traditional password entry schemes are susceptible to "shoulder surfing" in which an attacker watches an unsuspecting user enter their password or PIN number and uses it later to gain access to the account. One way to combat this problem is with a randomized challenge-response system. In these systems the user enters different information every time based on a secret in response to a randomly generated challenge. Consider the following scheme in which the password consists of a five-digit PIN number (00000 to 99999). Each digit is assigned a random number that is 1, 2, or 3. The user enters the random numbers that correspond to their PIN instead of their actual PIN numbers.

   For example, consider an actual PIN number of 12345. To authenticate the user would be presented with a screen such as:

PIN: 0 1 2 3 4 5 6 7 8 9

NUM: 3 2 3 1 1 3 2 2 1 3

The user would enter 23113 instead of 12345. This doesn't divulge the password even if an attacker intercepts the entry because 23113 could correspond to other PIN numbers, such as 69440 or 70439. The next time the user logs in, a different sequence of random numbers would be generated, such as:

PIN: 0 1 2 3 4 5 6 7 8 9

NUM: 1 1 2 3 1 2 2 3 3 3

Write a program to simulate the authentication process. Store an actual PIN number (Please use "99508") in your program. The program should use an array to assign random numbers to the digits from 0 to 9. Output the random digits to the screen, input the response from the user, and output whether or not the user's response correctly matches the PIN number.

Expected output:

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java Authenticate
Welcome!  To log in, enter the random digits from 1-3 that
correspond to your PIN number.

PIN digit: 0 1 2 3 4 5 6 7 8 9
Random #:  3 2 2 1 1 3 1 3 3 2

Enter code.
22333
Correct!  You may now proceed.
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java Authenticate
Welcome!  To log in, enter the random digits from 1-3 that
correspond to your PIN number.

PIN digit: 0 1 2 3 4 5 6 7 8 9
Random #:  2 1 2 3 2 3 1 3 3 1

Enter code.
12123
Error, invalid password entered.
```

10. (7pts) Write a program that will make a copy of a text file, line by line. Read the name of the existing file and the name of the new file—the copy—from the keyboard. Use the methods of the class File to test whether the original file exists and can be read. If not, display an error message and abort the program. Similarly, see whether the name of the new file already exists. If so, display a warning message and allow the user to either abort the program, overwrite the existing file, or enter a new name for the file.

Expected format of text file:

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ cat test.txt
origin file
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ cat test1.txt
copy1
```

Expected output:

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java FileCopier
What is the file to read from?
test.txt
What is the file to write to?
test1.txt
That file already exists, do you want to over write it? (y)
y
File processing completed.
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java FileCopier
What is the file to read from?
test.txt
What is the file to write to?
test2.txt
File processing completed.
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ cat test.txt
origin file
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ cat test1.txt
origin file
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ cat test2.txt
origin file
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$
```

11. (12pts) Consider a class that could be used to play a game of hangman. The class has the following attributes:

    - The secret word.
    - The disguised word, in which each unknown letter in the secret word is replaced with a question mark (?). For example, if the secret word is abracadabra and the letters a, b, and e have been guessed, the disguised word would be ab?a?a?ab?a.
    - The number of guesses made.
    - The number of incorrect guesses.

    It will have the following methods:

    - makeGuess(c)—guesses that character c is in the word.
    - getDisguisedWord—returns a string containing correctly guessed letters in their correct positions and unknown letters replaced with ?.
    - getSecretWord—returns the secret word.
    - getGuessCount—returns the number of guesses made.
    - isFound—returns true if the hidden word has been discovered.

a. Write a method heading for each method.

b. Write preconditions and postconditions for each method.

c. Write some Java statements that test the class.

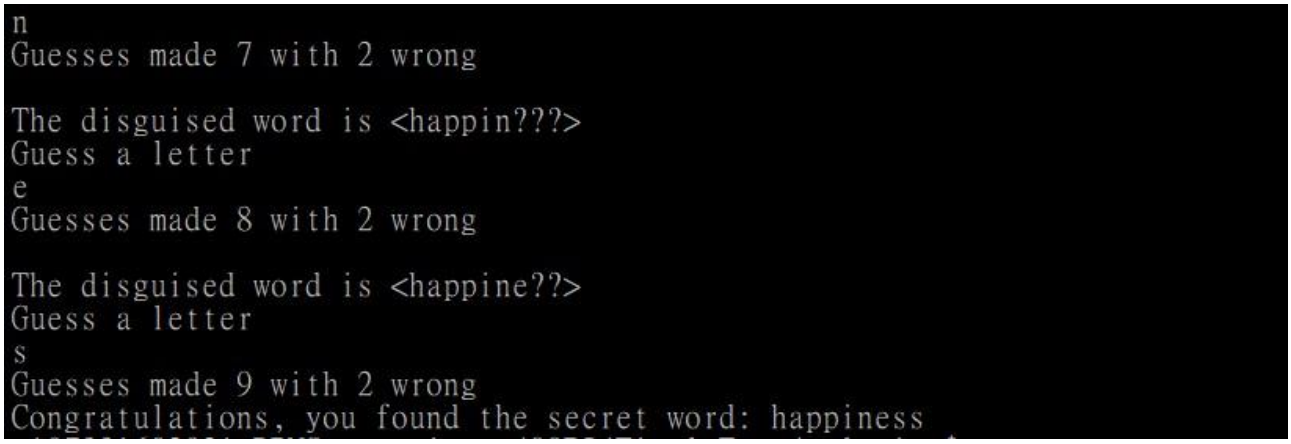
d. Implement the class.

e. List any additional methods and attributes needed in the implementation that were not listed in the original design. List any other changes made to the original design.

f. Write a program that implements the game of hangman using the class you wrote for Part d.

Expected output:

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java Hangman
Lets play a round of hangman.
We are playing hangman

The disguised word is <?????????>
Guess a letter
a
Guesses made 1 with 0 wrong

The disguised word is <?a???????>
Guess a letter
b
Guesses made 2 with 1 wrong

The disguised word is <?a???????>
Guess a letter
c
Guesses made 3 with 2 wrong

The disguised word is <?a???????>
Guess a letter
h
Guesses made 4 with 2 wrong

The disguised word is <ha???????>
Guess a letter
p
Guesses made 5 with 2 wrong

The disguised word is <happ?????>
Guess a letter
i
Guesses made 6 with 2 wrong

The disguised word is <happi????>
Guess a letter
```



12. (12pts) Suppose that you are in charge of customer service for a certain business. As phone calls come in, the name of the caller is recorded and eventually a service representative return the call and handles the request.

    Write a class ServiceRequests that keeps track of the names of callers. The class should have the following methods:

    - addName(name)—adds a name to the list of names.
      Throws a ServiceBackUpException if there is no free space in the list.
    - removeName(name)—removes a name from the list.
      Throws a NoServiceRequestException if the name is not in the list.
    - getName(i)—returns the ith name in the list.
    - getNumber—returns the current number of service requests.

    Write a program that uses an object of type ServiceRequests to keep track of customers that have called. It should have a loop that, in each iteration, attempts to add a name, remove a name, or print all names. Use an array of size 10 as the list of names.

    Expected output:

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution/12$ java ServiceProgram
Welcome to the Service Request program
Add a request (A), Remove a request (R), Show all requests (S) or quit (Q)
s
Current requests are:

Add a request (A), Remove a request (R), Show all requests (S) or quit (Q)
a
What is the name to add?
request1
Add a request (A), Remove a request (R), Show all requests (S) or quit (Q)
s
Current requests are:
Request 0 is request1

Add a request (A), Remove a request (R), Show all requests (S) or quit (Q)
r
What is the name to remove?
0
Sorry, that name was not in the list
Add a request (A), Remove a request (R), Show all requests (S) or quit (Q)
r
What is the name to remove?
request1
Add a request (A), Remove a request (R), Show all requests (S) or quit (Q)
s
Current requests are:

Add a request (A), Remove a request (R), Show all requests (S) or quit (Q)
a
What is the name to add?
request2
Add a request (A), Remove a request (R), Show all requests (S) or quit (Q)
q
Ok.  We are done.
```

13. (12pts) Write a program that will record the votes for one of two candidates by using the class VoteRecorder, which you will design and create. VoteRecorder will have static variables to keep track of the total votes for candidates and instance variables to keep track of the votes made by a single person. It will have the following attributes:

    - nameCandidatePresident1—a static string that holds the name of the first candidate for president.
    - nameCandidatePresident2—a static string that holds the name of the second candidate for president.
    - nameCandidateVicePresident1—a static string that holds the name of the first candidate for vice president.
    - nameCandidateVicePresident2—a static string that holds the name of the second candidate for vice president.
    - votesCandidatePresident1—a static integer that holds the number of votes for the first candidate for president.
    - votesCandidatePresident2—a static integer that holds the number of votes for the second candidate for president.
    - votesCandidateVicePresident1—a static integer that holds the number of votes for the first candidate for vice president.

- votesCandidateVicePresident2—a static integer that holds the number of votes for the second candidate for vice president.
- myVoteForPresident—an integer that holds the vote of a single individual for president (0 for no choice, 1 for the first candidate, and 2 for the second candidate).
- myVoteForVicePresident—an integer that holds the vote of a single individual for vice president (0 for no choice, 1 for the first candidate, and 2 for the second candidate)

In addition to appropriate constructors, VoteRecorder has the following methods:

- setCandidatesPresident(String name1, String name2)—a static method that sets the names of the two candidates for president.
- setCandidatesVicePresident(String name1, String name2)—a static method that sets the names of the two candidates for vice president.
- resetVotes—a static method that resets the vote counts to zero.
- getCurrentVotePresident—a static method that returns a string with the current total number of votes for both presidential candidates.
- getCurrentVoteVicePresident—a static method that returns a string with the current total number of votes for both vice presidential candidates.
- getAndConfirmVotes—a nonstatic method that gets an individual's votes, confirms them, and then records them.
- getAVote(String name1, String name2)—a private method that returns a vote choice for a single race from an individual (0 for no choice, 1 for the first candidate, and 2 for the second candidate).
- getVotes—a private method that returns a vote choice for president and vice president from an individual.
- confirmVotes—a private method that displays a person's vote for president and vice president, asks whether the voter is happy with these choices, and returns true or false according to a yes or no response.
- recordVotes—a private method that will add an individual's votes to the appropriate static variables.

Create a program that will conduct an election. The candidates for president are Annie and Bob. The candidates for vice president are John and Susan. Use a loop to record the votes of many voters. Create a new VoteRecorder object for each voter. After all the voters are done, present the results.

Expected output:

```
a1073316@2021_DBMS_practice:~/OOPI/Final_Exam/solution$ java VoteRecorder
YOU ARE VOTING FOR PRESIDENT
Please choose a candidate:
      0 - No one
      1 - Annie
      2 - Bob
0
YOU ARE VOTING FOR VICE PRESIDENT
Please choose a candidate:
      0 - No one
      1 - John
      2 - Susan
1
Your vote for president is no one
Your vote for president is John
Type yes if you are happy with your vote
yes
Type yes if there is another voter
yes
```

```
YOU ARE VOTING FOR PRESIDENT
Please choose a candidate:
      0 - No one
      1 - Annie
      2 - Bob
1
YOU ARE VOTING FOR VICE PRESIDENT
Please choose a candidate:
      0 - No one
      1 - John
      2 - Susan
2
Your vote for president is Annie
Your vote for president is Susan
Type yes if you are happy with your vote
y
YOU ARE VOTING FOR PRESIDENT
Please choose a candidate:
      0 - No one
      1 - Annie
      2 - Bob
2
YOU ARE VOTING FOR VICE PRESIDENT
Please choose a candidate:
      0 - No one
      1 - John
      2 - Susan
1
Your vote for president is Bob
Your vote for president is John
Type yes if you are happy with your vote
yes
Type yes if there is another voter
ynn
Annie has 0 votes; Bob has 1 votes;
John has 2 votes; Susan has 0 votes;
```