Military Institute of Science and Technology
B.Sc. in Computer Science and Engineering
Online Examination-1 (Spring) : 20 May, 2021
Subject: CSE 204, Data Structures and Algorithms Sessional-I

Time: 80 minutes **Full Marks: 20**

INSTRUCTIONS:

a. **Question-1** in **Section-A** is **compulsory.**
b. Answer any of the **ONE** question from **Section-B**
c. Figures in the margin indicate **full marks**._____

SECTION-A

Question – 1

a. During World War II, the Germans used a machine named Enigma to send ciphered (coded) messages for its Military operations in secrecy. Which was then broken by a team of mathematicians in Great Britain that provided the groundwork for the invention of modern computers, the story made famous by the movie "*The Imitation Game*". **10**

Inspired by it, you decided to make a little ciphering machine of your own to send coded messages to your friend. The machine takes the message you want to send to your friend as an input and codes each character of your message to numeric digits as per the equation given below. Note that your machine can send only **8 digits** of code at a time.

Code = (ASCII value of the character + Key) % 20
Where, Key = Last two digits of your ID.

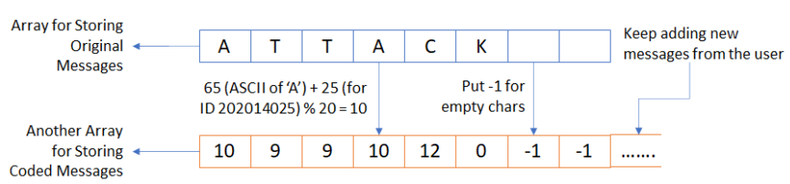
Now, write a code that runs your machine and does the following:

1. Takes input of the message (at most 8 characters) to be sent to your friend and stores it. Only storing the current message is enough.

2. Let the user delete all occurrences of a particular character from the input message.

3. Cipher (turn original input message to coded message) and store it. However, store **ALL** the coded digits in an array for future reference. Make the storage as efficient as possible, that is, **grow the coded message's array** as new ciphers are appended.

   Always add 8 values to the array. Put -1 in remaining spots if there's less than 8 characters.



Note:

1. Use one array to store the original message and one array to continuously store the coded messages. Use Static and Dynamic Array list appropriately.
2. You may use STL Vector to implement DAL.
3. **Code on top of this template. (Click here to download).**

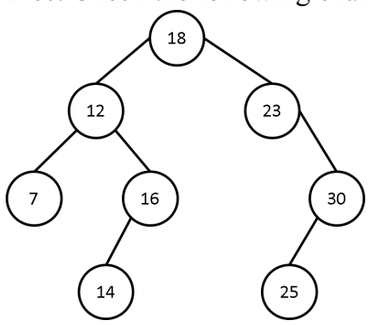
SECTION-B

Question – 2 **10**

This is a very common and straightforward problem of Binary Search Tree. A Binary Search Tree will be given to you and you will have to print all the pairs (a,b) from the Binary Search Tree where:

1. a+b = x; x is a positive integer and given as input
2. a < b

For simplicity you can assume that there are no repeated values in the Binary Search Tree. Check the following example for x = 30.



Here all possible pairs are: {(7, 23), (12, 18), (14,16)}. You can print the pairs in any sequence.

For your convenience a program is written as a template with a function inside the "BST" class named "*void insert(int p)*" that inserts **p** in a Binary Search Tree. Find the template-code *HERE*.

Now your task is to implement the necessary function/s in the "BST" class on the template-code that completes the described scenario and call it when choice=2 is submitted by the user.

Question – 3 **10**

A *double-ended queue* or *deque* (pronounced "deck") is a generalization of a stack and a queue that supports adding and removing items from either the front or the back of the data structure. In the last sessional class, you have implemented a normal queue [also can be found here]. You have to add functions so that this normal *queue* acts like a *deque*.

a. Add *push* function to insert an element at the front.

b. Add *pop* function to remove an element from the rear.

c. Add *complementarySequence* function to find the complement of a DNA sequence using the *deque* data structure. A DNA is a sequence of A, T, C, and G. This function will take DNA sequence and return the complement of that sequence.

d. In the main function, add an option which will take a DNA sequence from the user, call *complementarySequence* function and print the returned result.

Sample Input: ATAACGGA
Sample Output: AGGCAATA

Military Institute of Science and Technology
B.Sc. in Computer Science and Engineering
Online Examination-2 (Spring) : 01 July, 2021
Subject: CSE 204, Data Structures and Algorithms Sessional-I

Time: 1 hour 30 minutes **Full Marks: 30**

Question-1 **20**

A country consists of some interconnected cities. There are some bidirectional roads inside a country that connect 2 cities. Length of a bidirectional road is 100 km. Each country has a capital city. Shortest distance of a city is calculated from the capital city of that country. Now you will be given a graph consisting of some countries and you have to find out the terms described in the output section. For simplicity you will assume the following:

1. The name of a city consists of only lowercase letters except the capital city. The name of a capital city starts with an uppercase letter and the rest of the letters are lowercase.
2. The name of a country contains all uppercase letters.
3. String length of a city or country is at least 2.
4. The names of all the cities are different.
5. A country consists of at least 3 cities.
6. If two cities are unreachable from each other then it is guaranteed that they are from different countries.

Input

First line of input contains a single integer ***e*** denoting the total number of bidirectional roads. Next ***e*** lines of input contain 3 space separated strings each: ***X***, ***Y***, ***Z*** which denote that there is a bidirectional road between ***X*** and ***Y*** and the country name of both ***X*** and ***Y*** is ***Z***. If ***X*** or ***Y*** begins with an uppercase letter then that represents the capital city of ***Z***. Note that ***Z*** is a string that consists of all uppercase letters. Next line of input contains a single integer ***q*** that denotes the number of queries. Next ***q*** lines of input contain a single string ***P*** each.

Output

For each query print the following:

1. Name of the country of ***P*** [4 marks]
2. Shortest path of ***P*** from the capital city of the country of ***P*** [10 marks]
3. Shortest distance of ***P*** from the capital city of the country of ***P*** (in km) [6 marks]

If ***P*** is not found in the graph then just print "City not found". If there are multiple shortest paths then print any one of them. See the sample output for clarification.

Sample Input Output

Input	Output
9 chennai rajasthan INDIA Dhaka khulna BANGLADESH khulna rajshahi BANGLADESH barishal Dhaka BANGLADESH Delhi chennai INDIA Delhi rajasthan INDIA barishal rajshahi BANGLADESH rajshahi cumilla BANGLADESH cumilla khulna BANGLADESH 4 cumilla rajshahi coxbazar rajasthan	BANGLADESH Dhaka -> khulna -> cumilla 200 km BANGLADESH Dhaka -> barishal -> rajshahi 200 km City not found INDIA Delhi -> rajasthan 100 km

The following figure is the graph representation of the sample input output. This is attached for your convenience.

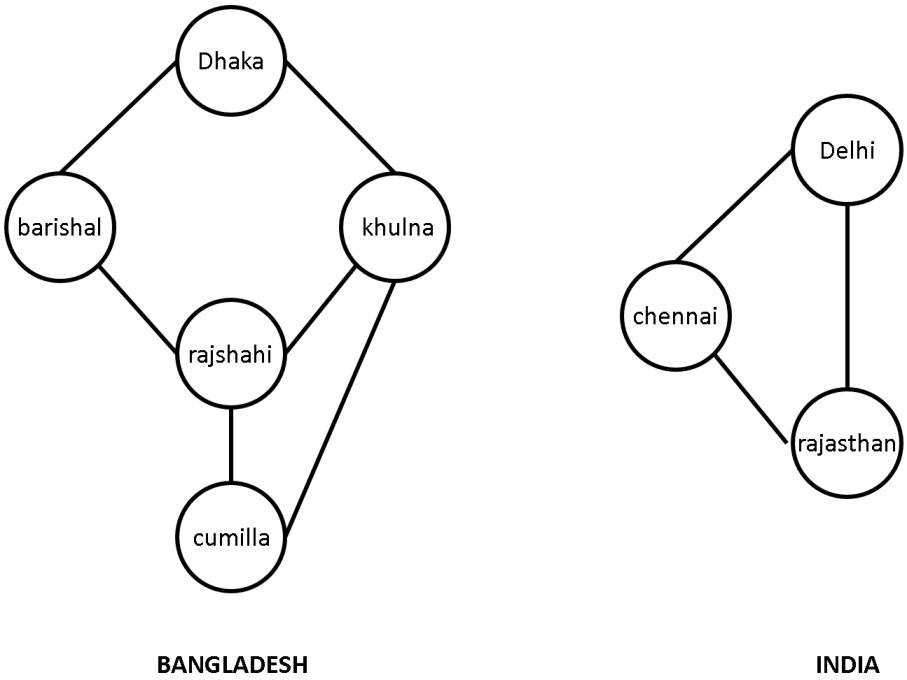


Figure 1

Question - 2 **10**

Template: Click Here (ideone) or here (drive, .cpp file)

Well, if you did last week's class, you now know how to implement quicksort to sort an integer array from smallest to largest number (ascending order).

But, we can do sorting in many different ways and even quicksort can be made more efficient for specific cases! Now, based on you knowledge from last class, implement the following -

1) Use quicksort to sort an integer array in **descending** order. [5 marks]
   (**Case A** of Figure 2.)

2) Modify quicksort so that instead of taking the rightmost element of a partition as pivot, it takes the **minimum number** of every partition (including the initial array) as the pivot. [5 marks]
   (**Case B** of Figure 2.)

3) Modify your entire code so that the quicksort algorithm now works for both integer numbers and float numbers. **You must not write another quicksort function that uses a float array!** [**Hint**: use templates] **[Bonus problem-3 marks]**

Note: implement on top of the given template.

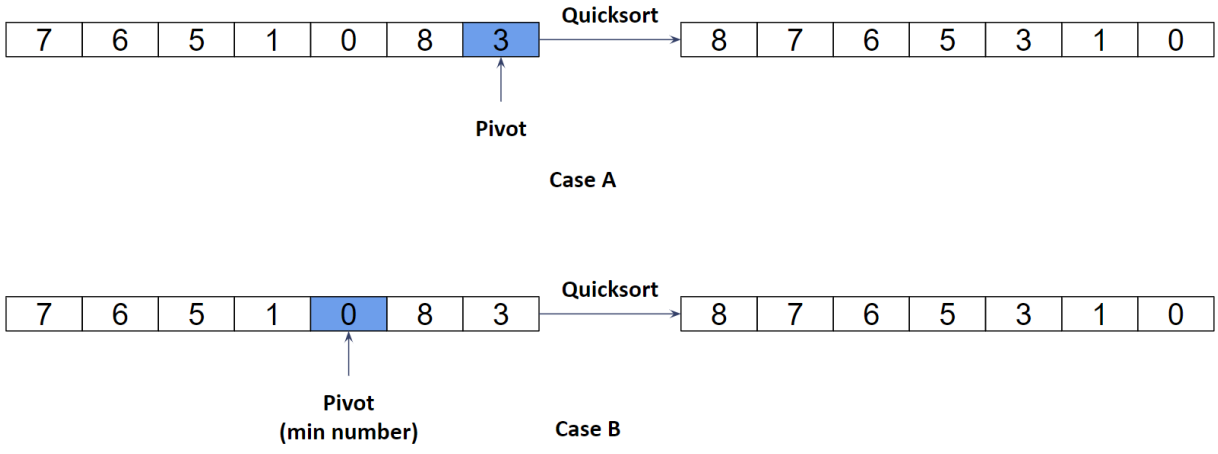


Figure 2