National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Data Structures**	**Course Code:**	**CS 218**
Program:	**BS (SE)**	**Semester:**	**Fall 2024**
Due Date:		**Total Marks:**	**120**
Section:	**3A**		
Type:	**Assignment 1**		

Important Instructions:

Submit separate .cpp files for each question. The naming format of each file

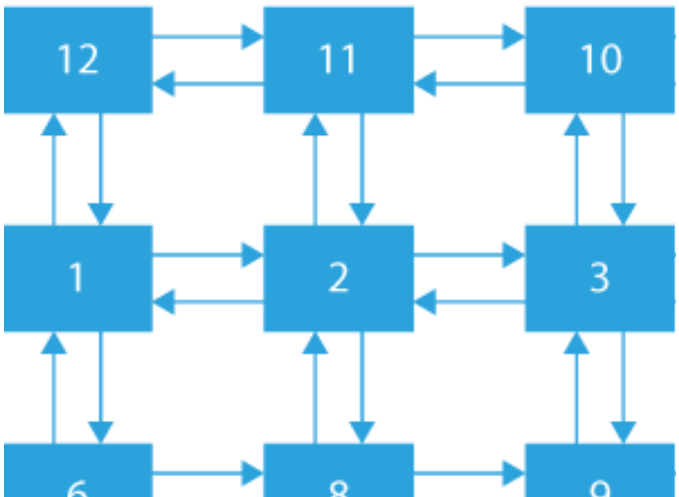
Should be: 22L-RollNo._Q_No.cpp i.e. 22L-XXX_Q_X.cpp. The files violating the

Format won't be considered for grading. DO NOT COMMENT THE CODE. Commented codes will be marked 0. Do not submit .zip files. Late Submissions won't be accepted.

Question 1 (100 marks)

All of us have used google maps at least once in our life and found a very helpful application. Have you ever wondered in which order they might have stored the whole earth. Now it's time to implement a kind of map using our newly learned data structure "Linked List".

We will have a node with four pointers (Left, Right, Up, Down) which will be connected to all nodes resides in its neighboring nodes.



Structure of Node class:

- Node* (Left, Right, Up, Down)
- string name
- Bool isRoad

Structure of Map class:

- Node* head
- Int rows
- Int columns

Functions in Map class:

- **Constructor (int rows, int columns)**
  It will initialize the rows * columns matrix as shown in above figure. Name attribute will be null and isRoad will be false initially.

- **ConstructRoad(String pointA, String pointB, int lengthOfRoad, String direction)**
  Here pointA and pointB are the names of starting and ending nodes. lengthOfNodes will be the number of nodes among pointA and pointB. Direction will specify in which direction you will construct road then in that direction, you will change the value of isRoad to true of lengthOfRoad nodes.

- **ConstructRoad(int row, int col, String pointA, String pointB, int lengthOfRoad, String direction)**
  It is the overloaded function, here we will construct the road after row and col index number. In start we will not have any road so we will specify row and column for road construction. If we do not specify row

and col then first we will find row and col index of pointA then we will construct road after that point.

- **RemoveRoad(String pointA, String pointB)**
  Now we will change the value of isRoad variable of nodes that are between pointA and pointB. First, we will locate the row and col indices of both points then we will change the value. If there is no direct path exist, print “No direct path found”

- **PrintMap()**
  This will just print the roads in the map, we can print white spaces where there is no road and * where road exist.

- **PrintPath(pointA, pointB)**
  This will print the path from pointA to pointB. Again, we can place * where road exists and white spaces where road does not exist.

Question 2 (20 marks)

You are given a file named **input.txt** with values Yes and No. Read the file and insert the strings in a 2D array of **strings**, where the elements are either "Yes" or "No". Do not use extra space. You must calculate number of rows and columns after reading the file. Now, your task is to compress the input array by counting the number of "Yes" values and creating a new array, which will contain the count of "Yes" values in the first column and the string "No" in the second column.

Yes	Yes	Yes No No No
Yes	No	No No Yes No

Yes	No	No No No No
Yes	Yes	Yes No No Yes

Output Values:
3 3
2 4
1 5
4 2

In this 2D array the first row will contain the location of “Yes”.
You need to solve this problem in $ON(\log_2 N)^2$