National University of Computer and Emerging Sciences, Lahore Campus

Course:	Data Structures	Course Code:	CS 218
Program:	BS (DS)	Semester:	Fall 2024
Due Date:		Total Marks:	
Section:	3A	Page(s):	5
Type:	Assignment 2		

Important Instructions:

Submit separate .cpp files for each question. The naming format of each file Should be: 2XX-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 No. 1

Consider that **'Hammad'** appeared in a national level examination. He went to hall and received his exam in accordance to the seating plan **based on roll numbers**. The examiner announced that those who have completed can place their exam at the top of the submission pile and leave. Hammad followed the instructions and submitted the exam after **69%** of the total people who appeared in exam. There was a small queue before submission desk, that consisted of **3%** of the total people appeared. Each paper will have name and roll number written on it, however there is only one person named Hammad with **roll number 420**.

After the submissions, the pile of the exams was taken to the marking center. The examiner started the marking of the exams by picking the top one first, and after checking it, placed it at the top of resultant pile.

On result day, Hammad was after **37%** of the total people, and at his turn, looked through the pile for his paper.

Write a program that manages the papers, and Hammad using Queue and Stack, and display the serving position at all instances.

Your program should be able to answer the following questions:

1. After how many students Hammad received is exam?
2. How long Hammad had to wait to submit his exam after completing it?
3. At what number was Hammad's exam placed in the submission stack?
4. At what number was Hammad's exam marked?
5. What place Hammad's exam got in the marked exams stack?
6. After how many students Hammad received his exam back?
7. What is the maximum number of exams Hammad would have been through while searching for his Paper?
8. What is the minimum number of exams Hammad would have been through while searching for his Paper?

You need to print all the stacks and queues formed while reaching the end.

You'll receive data of all students who appeared from a file named 'data'. **If some roll number isn't present in file, it means that particular student was absent.**

You are not allowed to use any other data structure other than one vector, and as many stacks and queues you want.

Question No. 2

Whenever we visit any place where we need to be served, the serving is often on first come first serve basis. Suppose you are to implement such a system of a Hospital's OPD, where patients are registered and enqueued for their medical examination. You need to take the following information from the patients:

1. Name
2. Age
3. Condition (Normal/Critical/Intense)
4. /Priority *(Derived)*

Your program should be able to perform the following functionalities:

- Enqueue Patients
- Specify Condition and Priority level
- Update Information
- View Current Queue (each patient detail)
- Prioritize Patients with severe condition

We, living in Pakistan, know that our system isn't rigid on rules. There are people with strong references that need to be served first, your program should be able to enqueue such people on high priority and make sure that the interface never witnesses it.

Senior Citizens are not to be made waiting in accordance to general ethics. Your program should be able to make such people wait for MAXIMUM 3 people to be served before them.
The program should be able to handle exceptions in a decent manner. Following exceptions may occur:

- Incomplete Information
- Improper derived priority
- Delayed serving

~~Use built in queue and stack~~.

Question No. 3

Make a Linked List where each node is a stack of queues.
Also add all the following Operations:

- **Insert**

    Provide an option for the following insertions:
    - ➤ Insert a Node
    - ➤ Insert in a Specific Stack
    - ➤ Insert in a Specific Queue
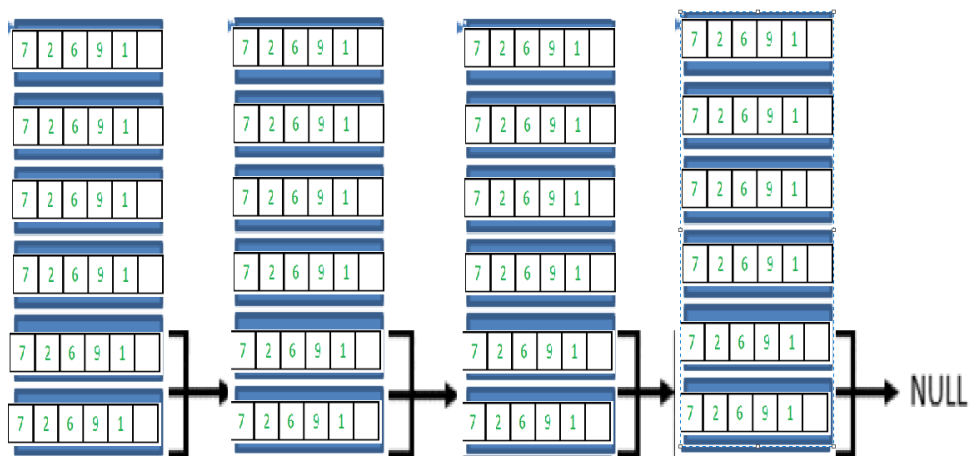
- **Delete**

    Provide an option for the following insertions:
    - ➤ Delete a Node
    - ➤ Delete from a Specific Stack
    - ➤ Delete from a Specific Queue

- **Sort**

    This function should sort
    - ➤ all queue in ascending
    - ➤ all stacks in ascending based on front element of queues
    - ➤ all nodes in ascending based on top element of stack

Question No. 4:

Let's consider that we have a group of people standing in a random order. Each person can be categorized as:

- Male/Female
- Facing side (left/right/front/back)

You need to arrange them all in such a way that no same person is next to other. The uniqueness can be defined by gender as well as body orientation. Let's say that you need to arrange them all in a form of nxm Matrix.

Make a dynamic user defined 2-D array, and read the data from a file named 'people' to arrange them all in the required order.
The data will be comma separated as:

M,F,F,F,F,F,M,M,M,M,M,M,F,F,F

Your program should try all possible combinations of people, and must use backtracking to undo incorrect positioning.

Display the Array at the end, each display each person's Gender and Facing position.
If no possible solution is found, the program should display a meaningful message.

NOTE:

FOUR PEOPLE AROUND A PERSON ARE CONSIDERED THE PEOPLE NEXT TO HIM/HER, NOT THE CORNER ONES.

Hint:

- Use stack to keep track of current order
- Use queue to keep track of person to be arranged next
- Use recursion to make combinations