CS2001 – Data Structures

Assignment # 02

Instructor	**Mr. Muhammad Yousaf**
Session	**Fall 2022**
Section	**BCS 3D, MCS1A**

General Guidelines

1. Write neat and clean code. Avoid any memory leaks and dangling pointers while implementing the scenarios required in this assignment.
2. You can lose the marks if conventions are not strictly followed.
3. Peer plagiarism and the late submissions are strictly not allowed. In case, zero marks will be awarded for whole assignment. You're not allowed to use any built-in libraries.
4. Total Marks: 250

Submission Guidelines

1. You will upload the assignment on CLASSROOM in given timeline.
2. Don't email your solution to instructor or TA for submission. Submit your assignment in given deadline said LMS.
3. You have already given one extra day for submission. No submission will be accepted later than said deadline.
4. Set file name as ***Roll-no_Section_Assignment#***

Deadline: October 20, 2022, 4:00 PM

Stack and Queue Implementation

Task # 1: Calculator using Stack ADT [Marks 30]

Implement the following class.

```
class Stack
{
Node* Top;

public:
Stack();
bool puch(char);
char pop();
bool isEmpty();
void display();
int calculator(string);//You are supposed to implement a basic calculator that will perform addition, Subtraction, multiplication, Division on numbers from 0 to 9. It should follow the BODMAS rule. Like  (7+(6*3+2)-(6/3)) = 25, Use STACK ADT to implement this.
```

Task # 2: Sequence Mutation [Marks 10]

Suppose you are writing a messaging App that will change the sequences of words in the message, the changing mechanism is simple though, it removes the punctuation (if any) from the message for example, a message like below: "Hurray! Live for assignment." would get changed to "assignment for live Hurray". *Use the ADT you have implemented in task 1 with implementation of following function.*

string sequenceMutation(string){}

Note: You're not allowed to use any built-in libraries.

Task # 3: Organizing plates using Stack ADT [Marks 40]

Imagine a scenario, you are on vacations and your manager asked you to help her with organizing crockery. You have three (03) empty cupboards and you placed all plates of five(05) different sizes in first cupboard with a sequence of largest set of plates at very bottom and the smallest plates at top, you asked your mother if it was fine and she said that you were supposed to place the plates in third cupboard not the first one so, now you are to move plates to third cupboard. You planned to do this by playing a game. The rules of the game are:

- You can move only one set of plates at a time.
- Set of larger plates cannot placed on a set of smaller plates.
- You can move only upper most set of plates.

You are to implement a console-based game for above scenario. The main purpose of game is to move the plates in minimum moves by following the rules of the game. For better understanding you can play a similar game on the link https://www.mathsisfun.com/games/towerofhanoi.html

Note: you can use numbers to represent set of plates. The state of each cupboard should be displayed after each iteration.

Task # 4: Traffic management using DEQUE [Marks 30]

Suppose you have joined the city traffic police (CTF) as a software engineer, a few days after your hiring a sudden traffic jam happened on Sargodha Road. The situation of the traffic jam is bad the vehicles are all stuck and more vehicles keep on coming that is causing the situation to get even worse. Some cars are leaving and joining this traffic jam from somewhere middle. Traffic police have arrived at the spot, and they have started to move traffic from both ends to get rid of it ASAP. You will be designing a program to deal with this situation while keeping a record of incoming and outgoing cars from traffic jams. You will be using an array-based double-ended queue (DEQUE) to implement this program that will show an appropriate message when all the cars are gone, and the situation is clear.

Note: your DEQUE class should include insertion (front, middle, end) and Deletion(front and rear end).

Task # 4: Senate win prediction using Queue ADT [Marks 40]

Suppose we have two parties: The Government and the Opposition. The senate consists of senators coming from these two parties. Now the senate wants to decide about an amendment in the law. The voting for this change is a round-based procedure. In each round, each senator can exercise one of the two rights:

1. Ban one senator's right:
   A senator can make another senator lose **all his rights** in this and all the following rounds.
2. Announce the victory:
   If this senator found the senators who still have rights to vote are all from **the same party**, he can announce the victory and make the decision about amendment in the law.

You have n senators, and you must deal with them using Queue ADT. The character 'G' and 'O' will represent Government, and the Opposition respectively the belonging party of senator.
The round-based procedure starts from the first senator to the last senator in the given order. This procedure will last until the end of voting. All the senators who have lost their rights will be skipped during the procedure. Suppose every senator is smart enough and will play the best strategy for his own party, you need to predict which party will finally announce the victory and make the amendment. The output should be Government or Opposition.

```
/*
Input: "OG"
Output: "Opposition"
Explanation: The first senator comes from Opposition and he can just ban the next senator's
right in the round 1 and the second senator can't exercise any rights any more since his right
has been banned and in the round 2, the first senator can just announce the victory since he is
the only guy in the senate who can vote.
*/
```

```
/*
Input: "OGG"
Output: "Government"
Explanation:
The first senator comes from Opposition and he can just ban the next senator's right in the round 1.
And the second senator can't exercise any rights anymore since his right has been banned.
And the third senator comes from Government and he can ban the first senator's right in the round 1.
And in the round 2, the third senator can just announce the victory since he is the only guy in the senate.
*/
```

Task # 5: Student Facilitation center management system using Queue ADT [Marks 50]

FAST-NUCES Chiniot Faisalabad campus has initiated the Student Facilitation Center to facilitate students. All the departments that a student has concern with have been assigned a counter there. All the building and infrastructure is done what left is to implement a modern management system to start the working of facilitation center. You are hired by the admin of the university as a developer to design a CLI based menu driven management system.

Whenever a person comes to the facilitation center the system will show two options:

1) New entry
2) Display

On choosing new entry system will ask his name and roll number for record after that it will show him a list of Departments available as shown below:

- Accounts
- Academics
- Admin
- Examination
- Lost and Found
- Sports
- Student Affairs

System should assign a code as per the choice of the user and will add him to the Queue.

On choosing Display the system will show the state of queues on the Command prompt in a structured form a sample is attached bellow:

Accouts	Academics	Admin	Examination	Lost&Found	Sports	Student Affair
Acc01	Acd01	Adm01	Exm01	L&F01	SP01	SA01
Acc02	Acd02	Adm02	Exm02	-	SP02	SA02
Acc03	Acd03	Adm03	-	-	SP03	SA03
Acc04	Acd04	Adm04	-	-	SP04	SA04
Acc05	-	Adm05	-	-	SP05	SA05

Assume that each person is staying for 2 seconds on the counter and update the queues. You should also keep the record of all the visitors and save it into file. When a person is processed its record should be added into file can be 1 or multiple as per your choice.

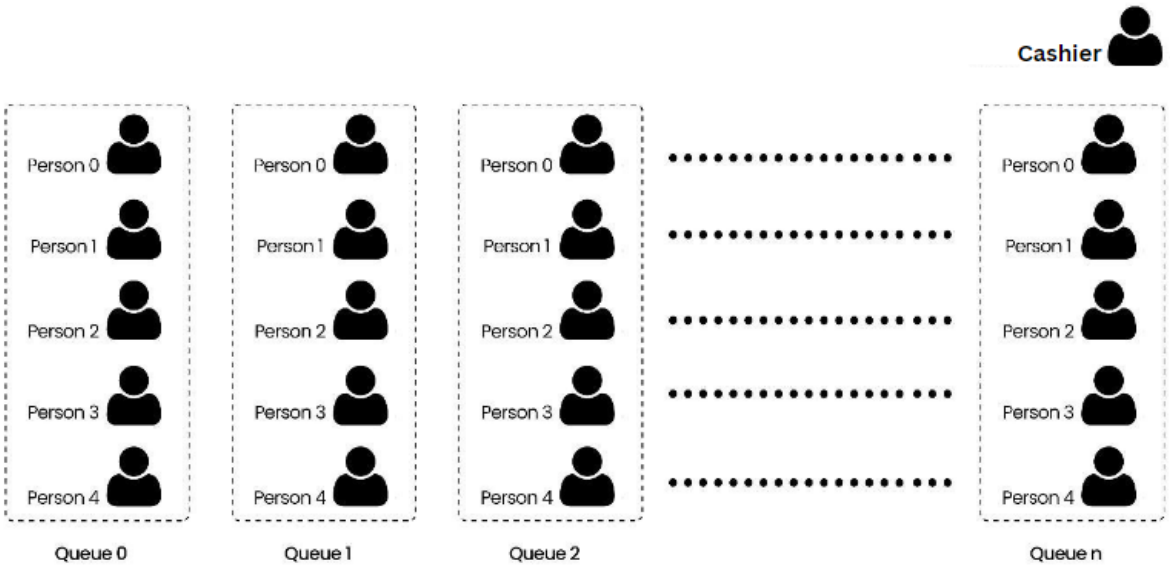
Note: There is a limit of at most 15 persons in each queue, if the limit reaches and more visitors are coming you should display an appropriate message to ask them to wait.

You will be using **Queue ADT** to implement above functionalities. If you think that some other Data structure in addition to the queue can be fruitful you can use that too but use of queue is compulsory.

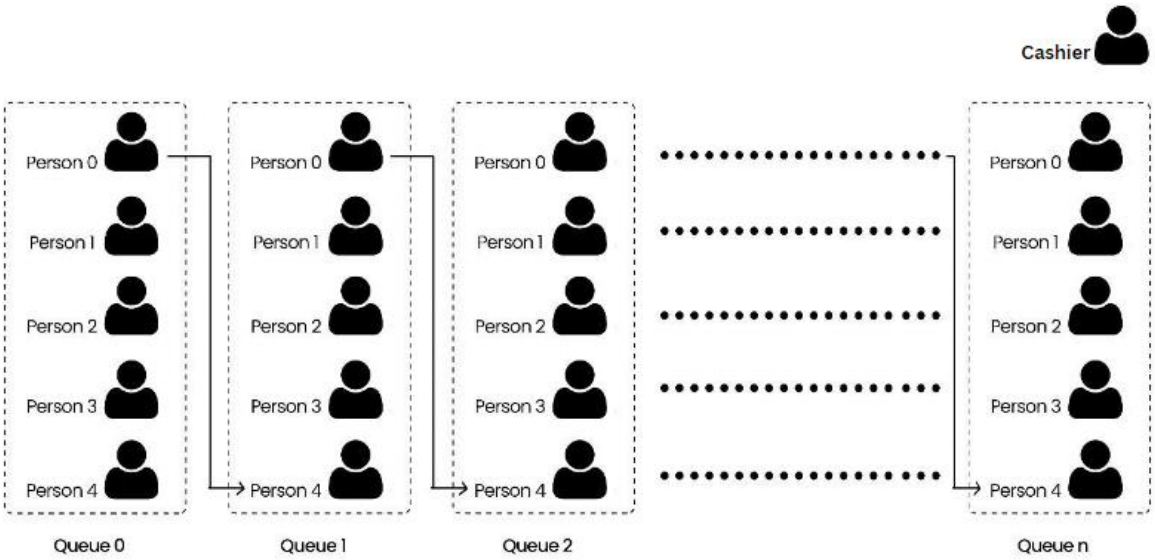
Note: You're not allowed to use any built-in libraries like vectors or queues.

Task # 6: Bank queue management using Queue ADT [Marks 50]

The digital technology is still new in Pakistan and Elder generations do not prefer it. It is the beginning of the month of October and all the pensions have been transferred to the accounts of pensioners. Due to some National level issues only one bank is allowed to work in each city and even that one bank will have only one single cashier to deal with all customers. So there is a large number of pensioners who wants to withdraw their money. The rush is going to be insane. To avoid longer queues and to keep track of queues inside the bank, they have decided to divide the queue into sub-queues. The Cashier takes 2 seconds to process a person. The queues look like this:



The queues work in the following way:

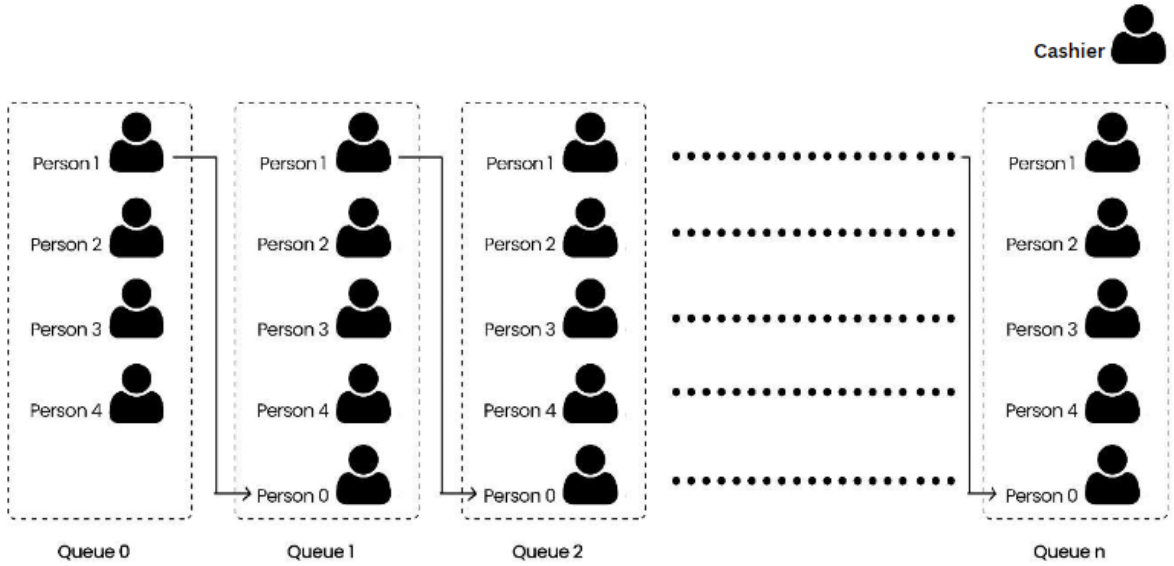


i.e: After processing "Person 0" of "Queue n", the queue moves forward like this:

1. "Person 0" of "Queue n-1" leaves his/her queue and enters in "Queue n".
2. "Person 0" of "Queue n-2" leaves his/her queue and enters in "Queue n-1".
3. "Person 0" of "Queue n-3" leaves his/her queue and enters in "Queue n-2".

4. This keeps happening all the way to queue 0 and at the end, "Person 0" of "Queue 0" leaves his/her queue and enters in "Queue 1".

After the queue is moved one-step forward, here's how it looks:



By the end, the cashier processes all the people in the queues so that everyone can get their money and manage their expenses.

You are going to simulate the above explained process using Queues ADT in C++. To implement this program, you will also be creating a Template Queue Class all by yourself to keep it generic and to create queues of any-types. The flow of the program will be like this:

User inputs an integer and N queues (<int>) are created. After creating N queues, you will enqueue some number of persons. For your ease, you can use same number of persons in every queue (minimum number of people in a queue must be 10). The ticket collector starts to process persons in "Queue n" one by one until all the persons are processed OR all the queues are empty.

Note: You're not allowed to use any built-in libraries like vectors or queues.