CL-2001
Data Structures
Lab # 7

Objectives:

- Queue ADT
- Enqueue, Deque
- Rear, Front
- Queue Linked List

Note: Carefully read the following instructions (*Each instruction contains a weightage*)

1. There must be a block of comments at start of every question's code by students; the block should contain brief description about functionality of code.
2. Comment on every function and about its functionality.
3. Mention comments where necessary such as comments with variables, loop, classes etc to increase code understandability.
4. Use understandable name of variables.
5. Proper indentation of code is essential.
6. Write a code in C++ language.
7. Make a Microsoft Word file and paste all of your C++ code with all possible screenshots of every task **outputs in Microsoft Word and submit word file. Do not submit .cpp file.**
8. First think about statement problems and then write/draw your logic on copy.
9. After copy pencil work, code the problem statement on MS Studio C++ compiler.
10. At the end when you done your tasks, attached C++ created files in MS word file and make your submission on Google Classroom. (Make sure your submission is completed).
11. Please submit your file in this format **20F1234_L4**.
12. Do not submit your assignment after deadline. Late and email submission is not accepted.
13. Do not copy code from any source otherwise you will be penalized with negative marks.

Problem: 1 | Queue ADT

Implement the QueueADT with size 10. Create the following driver functions:

Functions:

1. enqueue()
2. dequeue()
3. isempty()
4. isfull()
5. display()

Implement main() in such a way that working of all the functions mention above will be satisfied.

Problem: 2 | Circular Queue

Provide array base implementation of circular queue. Also create a driver function

Functions:

- Enqueue()
- Dequeue()
- Isempty()
- Isfull()

Problem: 3 | Queue Linked List

Provide Linked list base implementation of queue. Also create a driver function

Functions:

1. enqueue()
2. dequeue()
3. isempty()
4. display()

Implement main() in such a way that working of all the functions mention above will be satisfied.

Problem: 4 |

You need to implement an algorithm for processor that is based on FIFO(First In First out) Principle you need to take input the process id and time that will be taken by the process in seconds keep taking input until the user enters the sentinel value. As soon as input is complete the processing is started in the sequence as the processes were entered to show that the computer is processing you can use sleep function for specified amount of time specified with every process.

- **Input:**
- 1 3
- 2 4
- 3 5
- -1
- **Output:**
- 1 completed 3 seconds
- 2 completed in 4 seconds
- 3 completed in 5 seconds

Problem: 5 |

Round robin scheduling is the scheduling in which every process gets executed in a cyclic way, i.e. in this a particular time slice is allotted to each process which is known as time quantum. You have to design an algorithm which simulates the round robin scheduling using the quantum time of 2 seconds. Use problem 4.

☺ Best of luck