National University

Of Computer & Emerging Sciences Peshawar Campus

CL-2001
Data Structures
Lab # 4 and Lab # 5 Task (5 wtg)

Objectives:

- Stack
- Queue

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. Submit all .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 **19F1234_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.

Lab 4 – Task 2.5 wtg

Problem: 1 | Queue using stack [7.5 marks]

Implement a stack using queue. You need to implement following functions

a) Enqueue()
b) Dequeue()
c) IsEmpty()
a) IsFull()
b) Print()

Problem: 2 | Queue using Doubly LinkedList [7.5 marks]

Create a function that will duplicate the nodes of queue depending on number of nodes. For example, if the nodes are 3 it is to be duplicated 3 more time

Input Queue: 3->4->5

Output Queue: 3->3->3->4->4->4->4->5->5->5->5->5

(Only doubly linked List codes will be get credit)

Lab 5 – Task 2.5 wtg

Problem: 3 | Array Based Queue [10 marks]

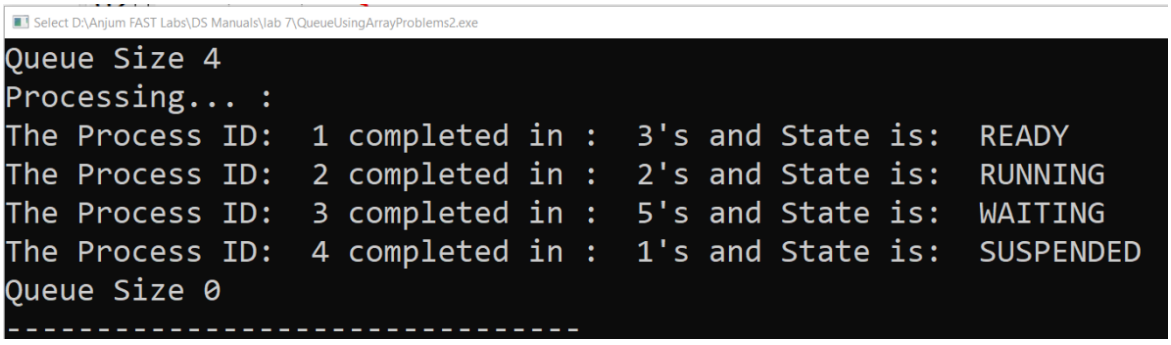
You need to implement an algorithm for processor that is based on FIFO (First In First out) Principle you need to take input the **process id**, **process state and time** that will be taken by the process in seconds keep taking input until the user enters the sentinel value or **array base queue size** is not going to be full. 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.

Note: Perform this task using Array Based Queue

Input:

1 30 READY
2 25 RUNNING
3 60 WAITING
3 3 SUSPENDED
-1

Output Sample:



Hints: Every index of Queue Array will contain a (**process id**, **process state and time)**