University of Central Punjab

**Faculty of Information Technology**

# Data Structures and Algorithms

# Summer 2024

|  |  |  |
| --- | --- | --- |
| **Lab 08** | |  |
| **Topic** | * LinkedList |
| **Objective** | * The basic purpose of this lab is to implement ADT of Linked List   and test its applications. |
|  | | |

**Instructions:**

* Indent your code.
* Comment your code.
* Use meaningful variable names.
* Plan your code carefully on a piece of paper before you implement it.
* Name of the program should be same as the task name. i.e. the first program should be Task\_1.cpp

**Students are required to complete the following tasks in lab timings.**

## Task 1

Modify the code done in class and implement the linked list using **Head** and **Tail** pointers.

**Functions of Linked List:**

**front()** – Returns the value of the first element in the linked list.

**back()** – Returns the value of the last element in the linked list.

**insetAtFront(int)** – Adds a new element at the beginning of the linked list.

**insetAtEnd(int)** – Adds a new element at the end of the linked list**.**

**removeFromFront()** – Removes the first element of the linked list, and reduces size of the linked list by 1.

**RemoveFromEnd()** – Removes the last element of the linked list, and reduces size of the linked list by 1.

**empty()** – Returns whether the list is empty(1) or not(0).

**size()** – Returns the number of elements in the list.

## Task 2

Implement **Stack** (LIFO) **and Queue** (FIFO) using **Linked Lists** only.

It should be a menu driven program which should first ask whether you want to use stack or queue. Then, operations related to stack or queue may be called. The cost of push, pop, enqueue and dequeue should be O(1).

## Task 3

Modify the code done in class and implement the linked list using **Head** pointer only (you are not allowed to use **Tail pointer).**

**Functions of Linked List:**

**front()** – Returns the value of the first element in the linked list.

**back()** – Returns the value of the last element in the linked list.

**insetAtFront(int)** – Adds a new element at the beginning of the linked list.

**insetAtEnd(int)** – Adds a new element at the end of the linked list**.**

**removeFromFront()** – Removes the first element of the linked list, and reduces size of the linked list by 1.

**RemoveFromEnd()** – Removes the last element of the linked list, and reduces size of the linked list by 1.

**empty()** – Returns whether the list is empty(1) or not(0).

**size()** – Returns the number of elements in the list.

## Task 4

Implement **Stack** (LIFO) **and Queue** (FIFO) using **Linked Lists only**.

It should be a menu driven program which should first ask whether you want to use stack or queue. Then, operations related to stack or queue may be called.

Comment on the Big O of push, pop, enqueue and dequeue in this implementation.

## Task 5

Add a function in linked list class called **void destroyLL()**. This function should destroy (that is delete all nodes of the linked list) when called.

## Task 6

Right now, there is NO destructor in Linked Lists. This results in non-deletion of dynamically created memories. Your task to create appropriate destructors to ensure that there is NO memory leakage.