**Lab 4**

**Doubly Linked List**

**Introduction:**

A doubly linked list is a linked list in which every node has a next pointer and a back pointer. In other words, every node contains the address of the next node (last node points to null as next node), and every node contains the address of the previous node (first node points to null as the previous node).



**Objective:**

The objective of this lab is to implement Doubly Linked List.

**Applications:**

**Doubly linked list** can be used in navigation systems where both front and back navigation is required. It is used by browsers to implement backward and forward navigation of visited web pages i.e. back and forward button. It is also used by various **application** to implement Undo and Redo functionality.

**Issue:**

I faced an issue in inserting at the last node of the list.

**Conclusion:**

To conclude, the **doubly linked list** can be traversed in forward as well as backward directions, unlike singly **linked list** which can be traversed in the forward direction only. Delete operation in a **doubly**-**linked list** is more efficient when compared to singly **list** when a given node is given.