

**Name: Syed Muhammad Fahad Fiaz**

**Roll No: BSSEM-S24-036**

**Section: SE 3A**

**Subject: Data Structure and Algorithms**

**Submitted To: Sir Rasikh Ali**

**Submission Date:**

**Assignment 7**

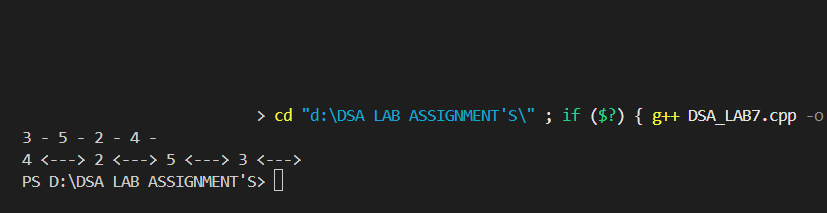
**DSA LAB TASK’S**

**LAB 7: Doubly Linked List (Insert & Display Nodes)**

**Explanation:**

* **Node**:
* Think of a Node as a person in the list. Each person (node) has three things:
  + A **name** (the data).
  + A link to the person in front of them (the prev pointer).
  + A link to the person behind them (the next pointer).
* **LinkedList**:
* This is like the whole list of people. At the start, the list is empty.
* The first person in the list is called the **head**.
* **Functions**:
* **insert\_first(int val)**:
  + This adds a new person at the front of the list.
  + If there are other people already, the new person (node) stands at the front and holds hands with the previous first person, who now knows the new person as the one in front of them.
* **insert\_last(int val)**:
  + This adds a person at the very end of the list.
  + If no one is in the list, this person becomes the first person (head). Otherwise, the last person holds hands with the new person, and the new person knows the last person as the one behind them.
* **specific\_location(int val, int pos)**:
  + This puts a new person at a certain spot in the list.
  + If the spot is 1 (the front), it does the same as insert\_first.
  + Otherwise, it finds the right place in the list and places the new person there, linking them to the others.
* **display()**:
  + This shows the list of people starting from the first to the last, one after the other.
* **display\_reverse()**:
  + This shows the list, but starting from the last person and going backward.

**OUTPUT**

****