**Lab Task 07**



**Superior University Gold Campus**

|  |  |
| --- | --- |
| **Submitted to** | ****Mr. Rasikh Ali**** |
| **Submitted by** | **Javaid Ali** |
| **Roll No** | **SU92-BSSEM-S24-029 (Section – 3A)** |
| **Subject** | **Data Structures and Algorithms (Lab)** |
| **Class** | **BS – Software Engineering** |

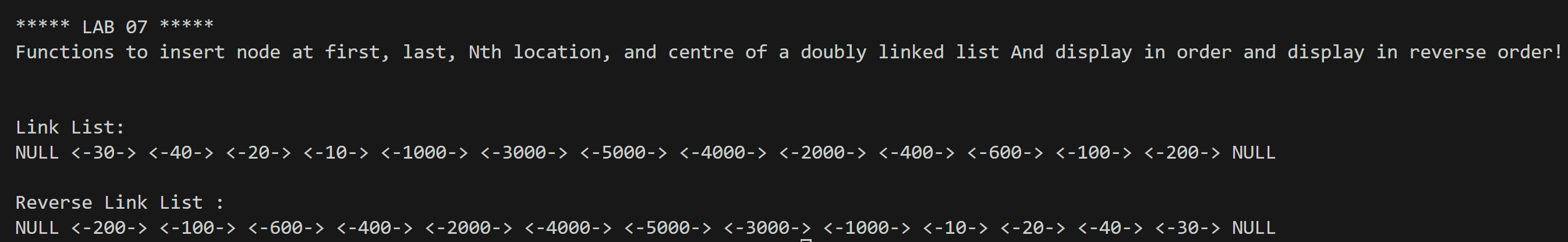
# **Lab 07: Doubly Linked List (Insert & Display Nodes)**

This program implements a **doubly linked list** with functions to insert nodes at the **start**, **end**, **specific position**, and **middle** of the list. The **Node** class stores data, a pointer to the next node (next), and a pointer to the previous node (prev).

1. The **insert\_at\_start** function adds a node at the beginning, updating both next and prev pointers.
2. The **insert\_at\_end** function appends a node at the end, linking it to the previous last node.
3. The **insert\_at\_pos** function inserts a node at a specified position, handling invalid positions.
4. The **insert\_at\_mid** function calculates the middle position and inserts a node there.
5. The **display** function prints the list in order, while **display\_reverse** traverses backward using the **prev** pointers to print the list in reverse.

The program demonstrates these operations, showcasing the flexibility of a doubly linked list for bidirectional traversal and insertion.

**Outputs:**

****