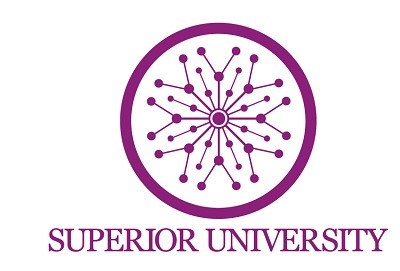
# **Lab # 03**

**Lab Tasks**



Submitted to

:

Sir Rasikh Ali

rR

Submitted by

:

Arooj Fatima

Roll No

:

SU92-BSSEM-S24-091

Subject:

DSA (Lab)

Class:

BS – Software Engineering

# **- Singly Linked List (Insert at End, Insert at Start)**

This lab shows the insertion of a node in the start and end of the singly linked list.

The insert\_at\_start (int val) function creates a new node with the given value. If the list is empty, it assigns this node as the head. Otherwise, it links the new node to the current head and updates the head pointer to the new node, ensuring the new node becomes the first element.

The insert\_At\_Last (int val) function inserts a new node at the end of the list. If the list is empty, it sets the new node as the head. Otherwise, it traverses the list until it reaches the last node and then appends the new node by updating the last node's next pointer.

Both functions efficiently handle inserting elements at different positions in a **singly linked list**.

# **Lab # 04**

# **- Singly Linked List (Insert at Specific Location)**

This lab shows the insertion of a node at the specific location in a singly linked list. This function, insert\_at\_pos (int pos, int val), inserts a new node at a specified position in a **singly linked list**.

It first creates a new node with the given value. If the position is less than 1, it prints an error message indicating an invalid position. If the position is **1**, it inserts the new node at the beginning by linking it to the current head and updating the head pointer.

For other positions, the function traverses the list to find the node just before the desired position. If the list is empty during traversal, it prints an error message. Once the correct position is reached, the new node is inserted by updating the pointers. This ensures that the node is correctly placed in the list while maintaining the linked structure

# **Lab # 05**

# **- Singly Linked List (Display Nodes)**

This lab shows the functions to display the first node, last node, Nth node, and Centre node of a singly linked list. The display() function prints all the elements of a **singly linked list** in a structured format. It starts at the head node and iterates through the list using a temporary pointer temp. During each iteration, it prints the data stored in the current node, followed by " -> " to indicate the link between nodes. The pointer then moves to the next node. Once the traversal reaches the end of the list (NULL), it prints "NULL" to signify the termination of the linked list. This function effectively visualizes the structure of the linked list.

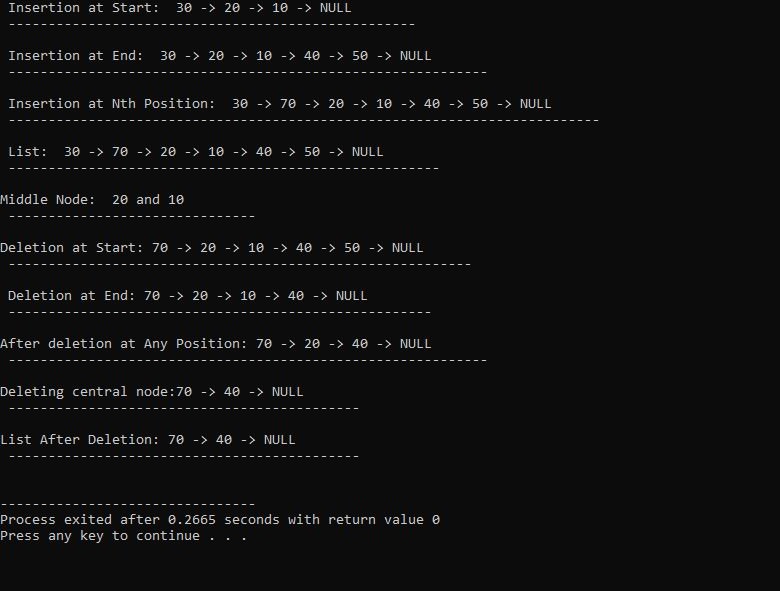
# **Lab # 06**

# **- Singly Linked List (Delete Nodes)**

This lab provides functions to delete the first node, last node, Nth node and Centre node of a singly linked list.

* The delete\_at\_start () function removes the first node by updating the head pointer to the second node, disconnecting the first node, and freeing its memory.
* The delete\_at\_end () function traverses the list to find the second-last node, deletes the last node, and sets the second-last node’s next pointer to NULL.
* The delete\_at\_pos (int pos) function deletes a node at a specified position by iterating through the list to locate the preceding node, adjusting its pointer to skip the target node, and freeing its memory. If the position is invalid, it displays an error message.
* The delete\_at\_center () function finds the middle node by first counting the total number of nodes, then traversing to the node before the middle one, updates its pointer to bypass the middle node, and deletes the middle node.

Each function ensures proper memory management and avoids memory leaks while maintaining the structure of the linked list.

****