SinglyLinkedList

1. push_back(int Val)

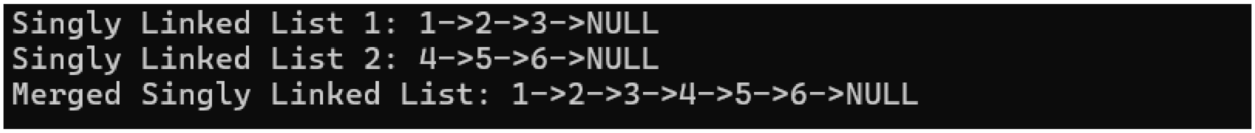
- Adds a new node to the **end** of the list.
- If the list is empty, the new node becomes both Head and Tail.
- Otherwise, links the new node to the current Tail and updates Tail.

2. display()

- Traverses from Head to NULL, printing each node’s data.

3. merge(SinglyLinkedList &OtherList)

- Appends another singly linked list (OtherList) to the current one.
- If current list is empty, takes over OtherList's head and tail.
- Otherwise, links current Tail to OtherList's Head and updates Tail.



```
Singly Linked List 1: 1->2->3->NULL
Singly Linked List 2: 4->5->6->NULL
Merged Singly Linked List: 1->2->3->4->5->6->NULL
```

DoublyLinkedList

1. push_back(int Val)

- Adds a new node to the **end**.
- Sets backward (Prev) and forward (Next) links appropriately.

2. displayForward()

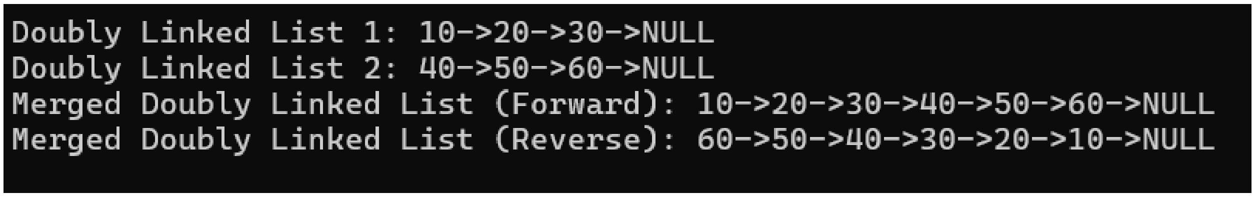
- Prints list from Head to Tail.

3. displayReverse()

- Prints list from Tail to Head.

4. merge(DoublyLinkedList &OtherList)

- Appends another doubly linked list.
- If empty, adopts OtherList's nodes.
- Else, links current Tail to OtherList's Head, updates Prev, and sets new Tail.



```
Doubly Linked List 1: 10->20->30->NULL
Doubly Linked List 2: 40->50->60->NULL
Merged Doubly Linked List (Forward): 10->20->30->40->50->60->NULL
Merged Doubly Linked List (Reverse): 60->50->40->30->20->10->NULL
```