NAME :ZOHAIB KHAN

ROLL_ NO : 022

SUBJECT: DSA LAB

LAB NO 7

Task: Implement functions to insert node at first, last, Nth location, and centre of a doubly linked list. And display in order and display in reverse order.

```cpp
#include <iostream>
using namespace std;
struct Node {
    int data;
    Node* next;
};
class LinkedList {
private:
    Node* head;
public:
    LinkedList() : head(NULL) {}
    void insertAtEnd(int value);
    void deleteFirstNode();
    void deleteLastNode();
    void deleteNthNode(int n);
    void deleteCentreNode();
    void display();
};
void LinkedList::insertAtEnd(int value) {
    Node* newNode = new Node{value, NULL};
    if (!head) head = newNode;
    else {
        Node* temp = head;
        while (temp->next) temp = temp->next;
        temp->next = newNode;
    }
}
void LinkedList::deleteFirstNode() {
    if (!head) cout << "List is empty!" << endl;
    else {
        Node* temp = head;
        head = head->next;
        delete temp;
    }
}
void LinkedList::deleteLastNode() {
    if (!head) cout << "List is empty!" << endl;
    else if (!head->next) {
        delete head;
```

```cpp
    }
    Node* nodeToDelete = temp->next;
    temp->next = temp->next->next;
    delete nodeToDelete;
}
void LinkedList::deleteCentreNode() {
    if (!head) {
        cout << "List is empty!" << endl;
        return;
    }
    Node *slow = head, *fast = head, *prev = NULL;
    while (fast && fast->next) {
        prev = slow;
        slow = slow->next;
        fast = fast->next->next;
    }
    if (!prev) head = head->next;
    else prev->next = slow->next;
    delete slow;
}
void LinkedList::display() {
    Node* temp = head;
    while (temp) {
        cout << temp->data << " -> ";
        temp = temp->next;
    }
    cout << "nullptr" << endl;
}
int main() {
    LinkedList list;
    list.insertAtEnd(10); list.insertAtEnd(20); list.insertAtEnd(30);
    list.insertAtEnd(40); list.insertAtEnd(50); list.display();
    list.deleteFirstNode();   list.display();
    list.deleteLastNode();    list.display();
    list.deleteNthNode(1);    list.display();
    list.deleteCentreNode();  list.display();
    return 0;
}
```

```
10 -> 20 -> 30 -> 40 -> 50 -> nullptr
20 -> 30 -> 40 -> 50 -> nullptr
20 -> 30 -> 40 -> nullptr
20 -> 40 -> nullptr
20 -> nullptr

-------------------------------
Process exited after 0.746 seconds with return value 0
Press any key to continue . . .
```

1. nsertAtFirst(int value): Inserts a node at the beginning of the doubly linked list.
2. insertAtLast(int value): Inserts a node at the end of the doubly linked list.
3. insertAtNth(int value, int n): Inserts a node at the specified position n.
4. insertAtCentre(int value): Inserts a node at the centre of the doubly linked list using the two