SUPERIOR UNIVERSITY

NAME :ZOHAIB KHAN

ROLL_ NO : 022

SUBJECT: DSA LAB

LAB NO 5

Task: Implement functions to display the first node, last node, Nth node, and centre node of a singly linked list.

```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 displayFirstNode();
    void displayLastNode();
    void displayNthNode(int n);
    void displayCentreNode();
    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::displayFirstNode() {
```

```
void LinkedList::displayLastNode() {
    if (!head) cout << "List is empty!" << endl;
    else {
        Node* temp = head;
        while (temp->next) temp = temp->next;
        cout << "Last node: " << temp->data << endl;
    }
}

void LinkedList::displayNthNode(int n) {
    if (n < 0 || !head) {
        cout << "Invalid position or list is empty!" << endl;
        return;
    }
    Node* temp = head;
    for (int i = 0; i < n && temp; i++) temp = temp->next;
    if (!temp) cout << "Position out of range!" << endl;
    else cout << "Node at position " << n << ": " << temp->data << endl;
}

void LinkedList::displayCentreNode() {
    if (!head) {
        cout << "List is empty!" << endl;
        return;
    }
    Node *slow = head, *fast = head;
    while (fast && fast->next) {
        slow = slow->next;
        fast = fast->next->next;
    }
    cout << "Centre node: " << slow->data << endl;
}
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.displayFirstNode();
    list.displayLastNode();
    list.displayNthNode(2);
    list.displayCentreNode();
    return 0;
}
```

```
10 -> 20 -> 30 -> 40 -> 50 -> nullptr
First node: 10
Last node: 50
Node at position 2: 30
Centre node: 30

--------------------------------
Process exited after 0.4702 seconds with return value 0
Press any key to continue . . .
```

1. insertAtEnd(int value): Inserts a node at the end of the linked list.
2. displayFirstNode(): Displays the value of the first node.
3. displayLastNode(): Displays the value of the last node.