***Superior university***

***Be superior***

***Be kind***

***Assignment no 5***

***Name . Aqeel Abbas***

***Submitted to***

***Sir Rasikh Sahab***

***Roll no . 056***

***Subject . Lab Da***

***ta structure***

Lab .5

Questions no 1 Implement functions to display the first node, last node, Nth node, and centre node of a singly linked list.

#include <iostream>

using namespace std;

class Node {

public:

int data;

Node\* next;

Node(int value) {

data = value;

next = nullptr;

}

};

class SinglyLinkedList {

public:

Node\* head;

SinglyLinkedList() {

head = nullptr;

}

void insertAtEnd(int data) {

Node\* newNode = new Node(data);

if (head == nullptr) {

head = newNode;

return;

}

Node\* last = head;

while (last->next != nullptr) {

last = last->next;

}

last->next = newNode;

}

void displayFirstNode() {

if (head != nullptr) {

cout << "First Node: " << head->data << endl;

} else {

cout << "List is empty." << endl;

}

}

void displayLastNode() {

if (head == nullptr) {

cout << "List is empty." << endl;

return;

}

Node\* temp = head;

while (temp->next != nullptr) {

temp = temp->next;

}

cout << "Last Node: " << temp->data << endl;

}

void displayNthNode(int n) {

if (head == nullptr) {

cout << "List is empty." << endl;

return;

}

Node\* temp = head;

int count = 1;

while (temp != nullptr && count < n) {

temp = temp->next;

count++;

}

if (temp != nullptr) {

cout << "Nth Node: " << temp->data << endl;

} else {

cout << "Position exceeds list length." << endl;

}

}

void displayCenterNode() {

if (head == nullptr) {

cout << "List is empty." << endl;

return;

}

Node\* slow = head;

Node\* fast = head;

while (fast != nullptr && fast->next != nullptr) {

slow = slow->next;

fast = fast->next->next;

}

cout << "Center Node: " << slow->data << endl;

}

void display() {

if (head == nullptr) {

cout << "List is empty." << endl;

return;

}

Node\* temp = head;

while (temp != nullptr) {

cout << temp->data << " -> ";

temp = temp->next;

}

cout << "None" << endl;

}

};

int main() {

SinglyLinkedList list;

list.insertAtEnd(10);

list.insertAtEnd(20);

list.insertAtEnd(30);

list.insertAtEnd(40);

list.insertAtEnd(50);

list.display();

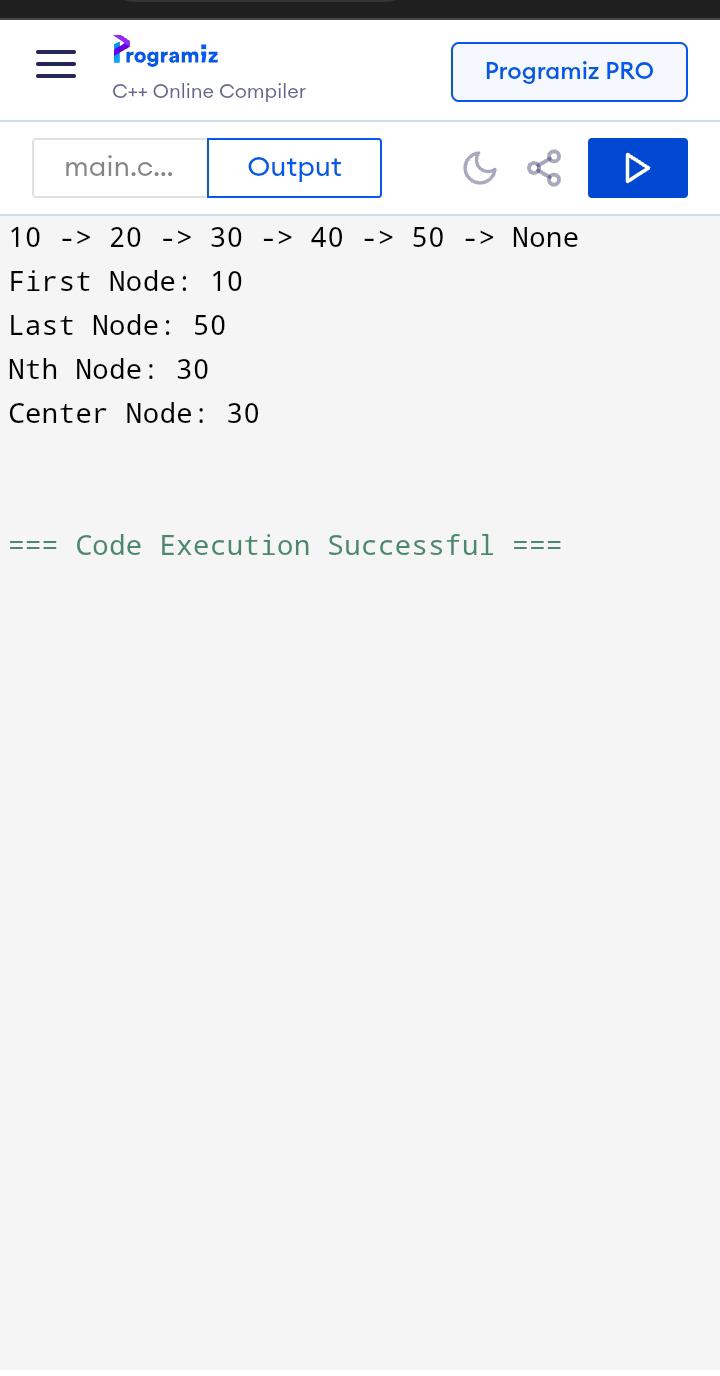
list.displayFirstNode();

list.displayLastNode();

list.displayNthNode(3);

list.displayCenterNode();

return 0;

}