**NAME : MALIHA SHAHID**

**ROLL NO:SU92-BSSEM-S24-071**

**LAB 5**

#include <iostream>

using namespace std;

class ListNode {

public:

int data;

ListNode\* nextNode;

ListNode(int val) {

data = val;

nextNode = NULL;

}

};

class LinkedList {

public:

ListNode\* first;

LinkedList() {

first = NULL;

}

void addAtBeginning(int val) {

ListNode\* newNode = new ListNode(val);

newNode->nextNode = first;

first = newNode;

}

void addAtEnd(int val) {

ListNode\* newNode = new ListNode(val);

if (first == NULL) {

first = newNode;

return;

}

ListNode\* temp = first;

while (temp->nextNode != NULL) {

temp = temp->nextNode;

}

temp->nextNode = newNode;

}

void printList() {

if (first == NULL) {

cout << "The list is currently empty!" << endl;

return;

}

ListNode\* temp = first;

while (temp != NULL) {

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

temp = temp->nextNode;

}

cout << "NULL" << endl;

}

void printFirstNode() {

if (first == NULL) {

cout << "The list is empty!" << endl;

} else {

cout << "First Element: " << first->data << endl;

}

}

void printLastNode() {

if (first == NULL) {

cout << "The list is empty!" << endl;

return;

}

ListNode\* temp = first;

while (temp->nextNode != NULL) {

temp = temp->nextNode;

}

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

}

void printNthNode(int n) {

if (first == NULL) {

cout << "The list is empty!" << endl;

return;

}

ListNode\* temp = first;

int index = 1;

while (temp != NULL) {

if (index == n) {

cout << "Element at Position " << n << ": " << temp->data << endl;

return;

}

temp = temp->nextNode;

index++;

}

cout << "Position " << n << " is out of bounds!" << endl;

}

void printMiddleNode() {

if (first == NULL) {

cout << "The list is empty!" << endl;

return;

}

ListNode\* slow = first;

ListNode\* fast = first;

while (fast != NULL && fast->nextNode != NULL) {

slow = slow->nextNode;

fast = fast->nextNode->nextNode;

}

cout << "Middle Element: " << slow->data << endl;

}

};

int main() {

LinkedList myList;

myList.addAtEnd(1);

myList.addAtEnd(2);

myList.addAtEnd(3);

myList.addAtEnd(4);

myList.addAtEnd(5);

myList.printList();

myList.printFirstNode();

myList.printLastNode();

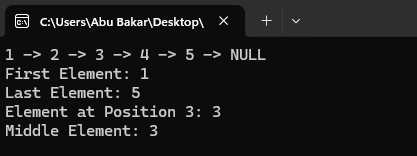
myList.printNthNode(3);

myList.printMiddleNode();

return 0;

}

**OUTPUT**

****

**EXPLANATION**

* ListNode Class – Defines a node with data and nextNode**.**
* LinkedList Class – Manages the list with first as the head.
* addAtBeginning / addAtEnd – Adds a node at the start or end.
* printList – Prints all elements in the list.
* printFirstNode / printLastNode – Displays the first or last node.
* printNthNode(n) – Prints the element at position n.
* printMiddleNode – Finds and prints the middle node using two-pointer technique.
* main Function – Demonstrates all functions with test values.