**NAME : MALIHA SHAHID**

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

**LAB 7**

#include <iostream>

using namespace std;

class DLLNode {

public:

int value;

DLLNode\* nextNode;

DLLNode\* prevNode;

DLLNode(int v) {

value = v;

nextNode = NULL;

prevNode = NULL;

}

};

class DoublyLL {

public:

DLLNode\* firstNode;

DoublyLL() {

firstNode = NULL;

}

void addAtBeginning(int v) {

DLLNode\* newNode = new DLLNode(v);

if (firstNode == NULL) {

firstNode = newNode;

return;

}

newNode->nextNode = firstNode;

firstNode->prevNode = newNode;

firstNode = newNode;

}

void addAtEnd(int v) {

DLLNode\* newNode = new DLLNode(v);

if (firstNode == NULL) {

firstNode = newNode;

return;

}

DLLNode\* temp = firstNode;

while (temp->nextNode != NULL) {

temp = temp->nextNode;

}

temp->nextNode = newNode;

newNode->prevNode = temp;

}

void addAtPosition(int pos, int v) {

if (pos < 1) {

cout << "INVALID POSITION!" << endl;

return;

}

if (pos == 1) {

addAtBeginning(v);

return;

}

DLLNode\* newNode = new DLLNode(v);

DLLNode\* temp = firstNode;

int count = 1;

while (temp != NULL && count < pos - 1) {

temp = temp->nextNode;

count++;

}

if (temp == NULL) {

cout << "INVALID POSITION!" << endl;

return;

}

newNode->nextNode = temp->nextNode;

if (temp->nextNode != NULL) {

temp->nextNode->prevNode = newNode;

}

temp->nextNode = newNode;

newNode->prevNode = temp;

}

void addAtMiddle(int v) {

if (firstNode == NULL) {

addAtBeginning(v);

return;

}

DLLNode\* slow = firstNode;

DLLNode\* fast = firstNode;

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

slow = slow->nextNode;

fast = fast->nextNode->nextNode;

}

DLLNode\* newNode = new DLLNode(v);

newNode->nextNode = slow->nextNode;

newNode->prevNode = slow;

if (slow->nextNode != NULL) {

slow->nextNode->prevNode = newNode;

}

slow->nextNode = newNode;

}

void printList() {

DLLNode\* temp = firstNode;

while (temp != NULL) {

cout << temp->value << " <-> ";

temp = temp->nextNode;

}

cout << "NULL" << endl;

}

void printReverse() {

if (firstNode == NULL) return;

DLLNode\* temp = firstNode;

while (temp->nextNode != NULL) {

temp = temp->nextNode;

}

while (temp != NULL) {

cout << temp->value << " <-> ";

temp = temp->prevNode;

}

cout << "NULL" << endl;

}

};

int main() {

DoublyLL obj;

obj.addAtBeginning(10);

obj.addAtBeginning(20);

obj.addAtEnd(30);

obj.addAtEnd(40);

obj.addAtPosition(3, 25);

obj.addAtMiddle(15);

cout << "Doubly Linked List in Order: " << endl;

obj.printList();

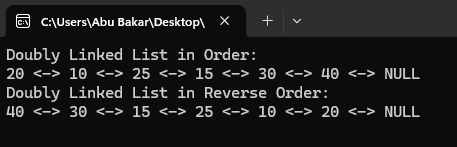
cout << "Doubly Linked List in Reverse Order: " << endl;

obj.printReverse();

return 0;

}

**OUTPUT**

****

**EXPLANATION**

This program implements a Doubly Linked List (DLL) in C++. Each node has a value, a pointer to the next node, and a pointer to the previous node. The DoublyLL class manages the list, allowing insertion at the beginning, end, a specific position, and the middle. The printList() function displays elements in order, while printReverse() prints them in reverse. The main() function demonstrates these operations by adding and printing elements.