**SUBMITTED BY : ROSHNEE RAZA**

**SUBMITTED TO : MAM REHANA**

**SAP :**

**TASK 1:**

#include <iostream>

using namespace std;

class Node {

public:

int data;

Node\* next;

};

class LinkedList {

public:

Node\* head;

LinkedList() {

head = NULL;

}

void insertAtEnd(int value) {

Node\* newNode = new Node();

newNode->data = value;

newNode->next = NULL;

if (head == NULL) {

head = newNode;

} else {

Node\* temp = head;

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = newNode;

}

}

void display() {

Node\* temp = head;

while (temp != NULL) {

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

temp = temp->next;

}

cout << endl;

}

};

int main() {

LinkedList list;

list.insertAtEnd(10);

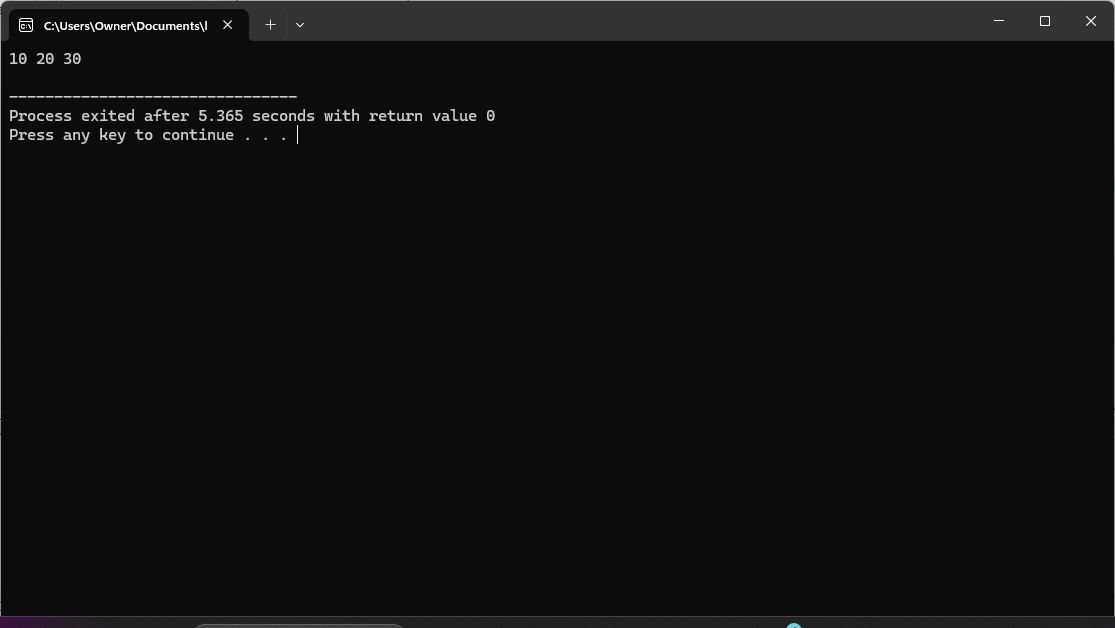
list.insertAtEnd(20);

list.insertAtEnd(30);

list.display(); // Output: 10 20 30

return 0;

}



**TASK 2:**

#include <iostream>

using namespace std;

class Node {

public:

int data;

Node\* next;

};

class LinkedList {

public:

Node\* head;

LinkedList() {

head = NULL;

}

void insertAtEnd(int value) {

Node\* newNode = new Node();

newNode->data = value;

newNode->next = NULL;

if (head == NULL) {

head = newNode;

} else {

Node\* temp = head;

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = newNode;

}

}

void deleteAtStart() {

if (head == NULL) {

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

} else {

Node\* temp = head;

head = head->next;

delete temp;

}

}

void deleteAtPosition(int pos) {

if (head == NULL) {

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

} else if (pos == 1) {

deleteAtStart();

} else {

Node\* temp = head;

for (int i = 1; i < pos - 1; i++) {

temp = temp->next;

}

Node\* nodeToDelete = temp->next;

temp->next = nodeToDelete->next;

delete nodeToDelete;

}

}

void deleteAtEnd() {

if (head == NULL) {

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

} else if (head->next == NULL) {

delete head;

head = NULL;

} else {

Node\* temp = head;

while (temp->next->next != NULL) {

temp = temp->next;

}

delete temp->next;

temp->next = NULL;

}

}

void display() {

Node\* temp = head;

while (temp != NULL) {

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

temp = temp->next;

}

cout << endl;

}

};

int main() {

LinkedList list;

list.insertAtEnd(10);

list.insertAtEnd(20);

list.insertAtEnd(30);

list.insertAtEnd(40);

list.display(); // Output: 10 20 30 40

list.deleteAtStart();

list.display(); // Output: 20 30 40

list.deleteAtPosition(2);

list.display(); // Output: 20 40

list.deleteAtEnd();

list.display(); // Output: 20

return 0;

}

