**Muhammad Kaleem**

**56614**

***Lab task no 7***

***Question no 1***

#include<iostream>

using namespace std;

class Node {

private:

float data;

Node\* next;

Node\* previous;

public:

Node(float data)

{

this->data = data;

this->next = NULL;

this->previous = NULL;

}

void setData(float data)

{

this->data = data;

}

void setNext(Node\* next)

{

this->next = next;

}

void setPrevious(Node\* previous)

{

this->previous = previous;

}

float getData()

{

return data;

}

Node\* getNext()

{

return next;

}

Node\* getPrevious()

{

return previous;

}

};

class link {

private:

Node\* head = NULL;

public:

link()

{

this->head = NULL;

}

void at\_head(float data)

{

Node\* n = new Node(data);

n->setNext(head);

if (head != NULL)

{

head->setPrevious(n);

}

head = n;

}

void at\_tail(float data)

{

Node\* n = new Node(data);

if (head == NULL)

{

at\_head(data);

return;

}

Node\* temp = head;

while (temp->getNext() != NULL)

{

temp = temp->getNext();

}

temp->setNext(n);

n->setPrevious(temp);

}

void remove\_num(float target)

{

if (head == NULL)

{

cout << "NO list to display \n";

return;

}

Node\* temp = head;

while (temp != NULL && temp->getData() != target)

{

temp = temp->getNext();

}

if (temp == NULL)

{

cout << "Number not found \n";

return;

}

if (temp == head)

{

head = temp->getNext();

if (head != NULL)

{

head->setPrevious(NULL);

}

delete temp;

return;

}

if (temp->getPrevious() != NULL)

{

temp->getPrevious()->setNext(temp->getNext());

}

else if (temp->getNext() != NULL)

{

temp->getNext()->setPrevious(temp->getPrevious());

}

delete temp;

}

void display()

{

Node\* temp = head;

while (temp != NULL)

{

cout << temp->getData() << "->";

temp = temp->getNext();

}

}

void search(int target)

{

Node\* temp = head;

bool found = false;

while (temp != NULL)

{

if (temp->getData() == target)

{

bool found = true;

break;

}

else

{

bool found = false;

break;

}

temp = temp->getNext();

}

if (found == true)

{

cout << "Number is present in the list \n";

}

else

{

cout << "Number is not present in the list \n";

}

}

};

int main()

{

link l1;

cout << "Inserting at tail \n";

l1.at\_tail(1.1);

l1.at\_tail(2.1);

l1.at\_tail(3.1);

l1.display();

cout << endl;

cout << "Inserting at head \n";

l1.at\_head(0.0);

l1.display();

cout << endl;

cout << "Deleting number \n";

l1.remove\_num(2.2);

l1.display();

cout << endl;

cout << "Searching number \n";

l1.search(2.2);

l1.display();

}

***Question no 2***

#include <iostream>

using namespace std;

class Node {

private:

float data;

Node\* next;

Node\* previous;

public:

Node(float data)

{

this->data = data;

this->next = NULL;

this->previous = NULL;

}

void setData(float data)

{

this->data = data;

}

void setNext(Node\* next)

{

this->next = next;

}

void setPrevious(Node\* previous)

{

this->previous = previous;

}

float getData()

{

return data;

}

Node\* getNext()

{

return next;

}

Node\* getPrevious()

{

return previous;

}

};

class Link {

private:

Node\* head;

public:

Link()

{

head = NULL;

}

void at\_head(float data)

{

Node\* n = new Node(data);

if (head == NULL)

{

n->setNext(n);

n->setPrevious(n);

head = n;

}

else

{

Node\* tail = head->getPrevious();

n->setNext(head);

n->setPrevious(tail);

head->setPrevious(n);

tail->setNext(n);

head = n;

}

}

void at\_tail(float data)

{

Node\* n = new Node(data);

if (head == NULL)

{

at\_head(data);

}

else

{

Node\* tail = head->getPrevious();

tail->setNext(n);

n->setPrevious(tail);

n->setNext(head);

head->setPrevious(n);

}

}

void display()

{

if (head == NULL) return;

Node\* temp = head;

do

{

cout << temp->getData() << " <-> ";

temp = temp->getNext();

}

while (temp != head);

cout << "(back to head)\n";

}

void search(int target)

{

Node\* temp = head;

bool found = false;

while (temp != NULL)

{

if (temp->getData() == target)

{

bool found = true;

break;

}

else

{

bool found = false;

break;

}

temp = temp->getNext();

}

if (found == true)

{

cout << "Number is present in the list \n";

}

else

{

cout << "Number is not present in the list \n";

}

}

};

int main() {

Link l1;

cout << "Inserting at tail \n";

l1.at\_tail(1.1);

l1.at\_tail(2.1);

l1.at\_tail(3.1);

l1.display();

cout << "Inserting at head \n";

l1.at\_head(0.0);

l1.display();

cout<<endl;

cout << "Searching number \n";

l1.search(2.1);

l1.display();

return 0;

}