**OUTPUT:**

Welcome to the linked list

The operation available in the list are:

0.Insert at the beginning

1.Insert at the end

2.Insert after the specific node

3.Insert before specific node

4.Insert at specific position

5.Delete from the beginning

6.Delte from the end

7.Delete after specific node

8.Delete at specific position

9.Display

Enter your choice as per above menu and 'e' or 'E' to end the program:0

Enter the number you want to insert at beginning of the list:1

Item 1 inserted at the beginning of list

Enter your choice as per above menu and 'e' or 'E' to end the program:1

Enter the number you want to insert at end of the list:2

Item 2 inserted at the end of list

Enter your choice as per above menu and 'e' or 'E' to end the program:3

Enter the element before which you want to perform insertion:2

Enter the number you want to insert before the element 2 :4

Item 4 added before 2

Enter your choice as per above menu and 'e' or 'E' to end the program:2

Enter the element after which you want to perform insertion:1

Enter the number you want to insert after the element 1 :3

Item 3 added after 1

Enter your choice as per above menu and 'e' or 'E' to end the program:4

Enter the position at which you want to perform insertion:4

Enter the number you want to insert at the position 4 :5

Enter your choice as per above menu and 'e' or 'E' to end the program:9

The elements in the list are:

Element 1 =1

Element 2 =3

Element 3 =4

Element 4 =5

Element 5 =2

Enter your choice as per above menu and 'e' or 'E' to end the program:5

The item is deleted from beginning is.

The element deleted is:

1

Enter your choice as per above menu and 'e' or 'E' to end the program:6

The item is deleted from end is.

The element deleted is:

2

Enter your choice as per above menu and 'e' or 'E' to end the program:7

Enter the element after which you want to perform deletion:4

The item is deleted after element 4

The element deleted is:5

Enter your choice as per above menu and 'e' or 'E' to end the program:9

The elements in the list are:

Element 1 =3

Element 2 =4

Enter your choice as per above menu and 'e' or 'E' to end the program:8

Enter the position at which you want to perform deletion:2

The item at position 2 was deleted

The element deleted is:4

Enter your choice as per above menu and 'e' or 'E' to end the program:9

The elements in the list are:

Element 1 =3

Enter your choice as per above menu and 'e' or 'E' to end the program:e

SOURCE CODE:

//single linked list insertion and deletion

#include<iostream>

#include<cstring>

#include<cstdlib>

using namespace std;

template <class T>

struct Node{

T data;

Node \*next;

};

template <class T>

class LinkedList{

private:

Node<T> \*head;

public:

LinkedList()

{

head=NULL;

}

bool ListIsEmpty()

{

return(head==NULL);

}

Node<T> \*CreateNewNode(T value)

{

Node<T> \*newnode=new Node<T>;

if(newnode==NULL)

{

cout<<endl<<"New node cannot be created"<<endl;

exit(1);

}

else

{

newnode->data=value;

newnode->next=NULL;

}

return newnode;

}

void InsertAtBeginning(T item)

{

Node<T> \*NewNode=CreateNewNode(item);

NewNode->next=head;

head=NewNode;

cout<<endl<<"Item "<<item<<" inserted at the beginning of list"<<endl;

}

void InsertAtEnd(T item)

{

Node<T> \*ptr,\*NewNode;

NewNode=CreateNewNode(item);

ptr=head;

while(ptr->next!=NULL)

{

ptr=ptr->next;

}

ptr->next=NewNode;

cout<<endl<<"Item "<<item<<" inserted at the end of list"<<endl;

}

void InsertAfterSpecificNode(T item,T element)

{

Node<T> \*NewNode=CreateNewNode(item);

Node<T> \*ptr,\*preptr;

preptr=head;

ptr=head->next;

while(preptr->data!=element)

{

if(ptr->next==NULL)

{

cout<<endl<<"No such element found in the list"<<endl;

exit(1);

}

else

{

preptr=ptr;

ptr=ptr->next;

}

}

preptr->next=NewNode;

NewNode->next=ptr;

cout<<endl<<"Item "<<item<<" added after "<<element<<endl;

}

void InsertbeforeSpecificNode(T item,T element)

{

Node<T> \*NewNode=CreateNewNode(item);

Node<T> \*ptr,\*preptr;

preptr=head;

ptr=head->next;

while(ptr->data!=element)

{

if(ptr->next==NULL)

{

cout<<endl<<"No such element found in the list"<<endl;

exit(1);

}

else

{

preptr=ptr;

ptr=ptr->next;

}

}

preptr->next=NewNode;

NewNode->next=ptr;

cout<<endl<<"Item "<<item<<" added before "<<element<<endl;

}

void InsertAtSpecificPos(T item,int position)

{

Node<T> \*NewNode=CreateNewNode(item);

Node<T> \*ptr=head;

int count=1;

while(count<position-1)

{

ptr=ptr->next;

count++;

}

NewNode->next=ptr->next;

ptr->next=NewNode;

cout<<endl<< “Element”<<item<< “inserted at position”<<position<<endl;

}

T DeleteFromBeginning()

{

Node<T> \* temp;

T item;

if(ListIsEmpty())

{

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

exit(1);

}

else

{

item=head->data;

temp=head;

head=head->next;

delete temp;

}

cout<<endl<<"The item is deleted from beiginning"<<endl;

return item;

}

T DeleteFromEnd()

{

Node<T> \*preptr,\*ptr;

T item;

if(ListIsEmpty())

{

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

exit(1);

}

else

{

preptr=head;

ptr=head->next;

while(ptr->next!=NULL)

{

preptr=ptr;

ptr=ptr->next;

}

item=ptr->data;

preptr->next=NULL;

delete ptr;

}

cout<<endl<<"The item is deleted from end"<<endl;

return item;

}

T DeleteNodeAfterSpecificNode(T element)

{

T item;

Node<T> \*preptr,\*ptr;

if(ListIsEmpty())

{

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

exit(1);

}

else{

preptr=head;

ptr=head->next;

while(preptr->data!=element)

{

preptr=ptr;

ptr=ptr->next;

}

item=ptr->data;

preptr->next=ptr->next;

delete ptr;

}

cout<<endl<<"The item is deleted after element "<<element<<endl;

return item;

}

T DeleteAtSpecificPos(int position)

{

int count=1;

T item;

Node<T> \*ptr,\*preptr;

if(ListIsEmpty())

{

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

exit(1);

}

else{

preptr=head;

ptr=head->next;

while(count<position-1)

{

preptr=ptr;

ptr=ptr->next;

}

item=ptr->data;

preptr->next=ptr->next;

delete ptr;

}

cout<<endl<<"The item at position "<<position<<" was deleted"<<endl;

return item;

}

void displayList()

{

if(head==NULL)

{

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

exit(1);

}

else

{

Node<T> \*temp=head;

int i=1;

cout<<endl<<"The elements in the list are:";

cout<<endl<<"Element "<<i<<" ="<<temp->data;

while(temp->next!=NULL)

{

i++;

temp=temp->next;

cout<<endl<<"Element "<<i<<" ="<<temp->data;

}

}

}

};

int main()

{

char check='i';

LinkedList<int> obj;

int num;

cout<<endl<<"Welcome to the linked list"<<endl;

cout<<endl<<"The operation available in the list are:";

cout<<endl<<"0.Insert at the beginning";

cout<<endl<<"1.Insert at the end";

cout<<endl<<"2.Insert after the specific node";

cout<<endl<<"3.Insert before specific node";

cout<<endl<<"4.Insert at specific position";

cout<<endl<<"5.Delete from the beginning";

cout<<endl<<"6.Delte from the end";

cout<<endl<<"7.Delete after specific node";

cout<<endl<<"8.Delete at specific position";

cout<<endl<<"9.Display";

while(check!='e' && check!='E')

{

cout<<endl<<"Enter your choice as per above menu and 'e' or 'E' to end the program:";

cin>>check;

if(check=='0')

{

cout<<endl<<"Enter the number you want to insert at beginning of the list:";

cin>>num;

obj.InsertAtBeginning(num);

}

else if(check=='1')

{

cout<<endl<<"Enter the number you want to insert at end of the list:";

cin>>num;

obj.InsertAtEnd(num);

}

else if(check=='2')

{

int ele;

cout<<endl<<"Enter the element after which you want to perform insertion:";

cin>>ele;

cout<<endl<<"Enter the number you want to insert after the element "<<ele<<" :";

cin>>num;

obj.InsertAfterSpecificNode(num,ele);

}

else if(check=='3')

{

int ele;

cout<<endl<<"Enter the element before which you want to perform insertion:";

cin>>ele;

cout<<endl<<"Enter the number you want to insert before the element "<<ele<<" :";

cin>>num;

obj.InsertbeforeSpecificNode(num,ele);

}

else if(check=='4')

{

int ele;

cout<<endl<<"Enter the position at which you want to perform insertion:";

cin>>ele;

cout<<endl<<"Enter the number you want to insert at the position "<<ele<<" :";

cin>>num;

obj.InsertAtSpecificPos(num,ele);

}

else if(check=='5')

{

int answer=obj.DeleteFromBeginning();

cout<<endl<<"The element deleted is:"<<answer;

}

else if(check=='6')

{

int answer=obj.DeleteFromEnd();

cout<<endl<<"The element deleted is:"<<answer;

}

else if(check=='7')

{

int ele;

cout<<endl<<"Enter the element after which you want to perform deletion:";

cin>>ele;

int answer=obj.DeleteNodeAfterSpecificNode(ele);

cout<<endl<<"The element deleted is:"<<answer;

}

else if(check=='8')

{

int ele;

cout<<endl<<"Enter the position at which you want to perform deletion:";

cin>>ele;

int answer=obj.DeleteAtSpecificPos(ele);

cout<<endl<<"The element deleted is:"<<answer;

}

else if(check=='9')

obj.displayList();

else if(check=='e'||'E')

break;

else

cout<<endl<<"Please choose valid option";

} return 0;

}

**OUTPUT:**

Welcome to the double linked list

The operation available in the list are:

0.Insert at the beginning [0]

1.Insert at the end [1]

2.Insert after the specific node [2]

3.Insert before specific node [3]

4.Insert at specific position [4]

5.Delete from the beginning [5]

6.Delte from the end [6]

7.Delete after specific node [7]

8.Delete at specific position [8]

9.Delete before specific node [9]

10.Display [d or D]

Enter your choice as per above menu and 'e' or 'E' to end the program:0

Enter the number you want to insert at beginning of the list:1

Item 1 inserted at the beginning of list

Enter your choice as per above menu and 'e' or 'E' to end the program:0

Enter the number you want to insert at beginning of the list:3

Item 3 inserted at the beginning of list

Enter your choice as per above menu and 'e' or 'E' to end the program:1

Enter the number you want to insert at end of the list:2

Item 2 inserted at the end of the list

Enter your choice as per above menu and 'e' or 'E' to end the program:2

Enter the element after which you want to perform insertion:3

Enter the number you want to insert after the element 3 :5

Item 5 inserted after element 3

Enter your choice as per above menu and 'e' or 'E' to end the program:3

Enter the element before which you want to perform insertion:1

Enter the number you want to insert before the element 1 :6

Item 6 inserted before element 1

Enter your choice as per above menu and 'e' or 'E' to end the program:4

Enter the position at which you want to perform insertion:3

Enter the number you want to insert at the position 3 :7

Element 7 inserted at position 3

Enter your choice as per above menu and 'e' or 'E' to end the program:d

The elements in the list are:

Element 1 =3

Element 2 =5

Element 3 =7

Element 4 =6

Element 5 =1

Element 6 =2

Enter your choice as per above menu and 'e' or 'E' to end the program:5

The item is deleted from beiginning

The element deleted is:3

Enter your choice as per above menu and 'e' or 'E' to end the program:6

The item is deleted from the end

The element deleted is:2

Enter your choice as per above menu and 'e' or 'E' to end the program:7

Enter the element after which you want to perform deletion:7

Item deleted after element 7

The element deleted is:6

Enter your choice as per above menu and 'e' or 'E' to end the program:d

The elements in the list are:

Element 1 =5

Element 2 =7

Element 3 =1

Enter your choice as per above menu and 'e' or 'E' to end the program:8

Enter the position at which you want to perform deletion:2

The item at position 2 was deleted

The element deleted is:7

Enter your choice as per above menu and 'e' or 'E' to end the program:0

Enter the number you want to insert at beginning of the list:9

Item 9 inserted at the beginning of list

Enter your choice as per above menu and 'e' or 'E' to end the program:d

The elements in the list are:

Element 1 =9

Element 2 =5

Element 3 =1

Enter your choice as per above menu and 'e' or 'E' to end the program:9

Enter the element before which you want to perform deletion:5

Item deleted before element 5

the element deleted is:9

Enter your choice as per above menu and 'e' or 'E' to end the program:d

The elements in the list are:

Element 1 =5

Element 2 =1

Enter your choice as per above menu and 'e' or 'E' to end the program:e

SOURCE CODE:

//Double link insertion and deletion

#include<iostream>

#include<cstdlib>

#include<cmath>

using namespace std;

template <class T>

struct Node

{

T data;

Node \*prev,\*next;

};

template <class T>

class DoubleLinkedList

{

private:

Node<T> \*Head;

public:

DoubleLinkedList()

{

Head=NULL;

}

bool isEmpty()

{

return(Head==NULL);

}

Node<T> \*createNewNode(T value)

{

Node<T> \*newnode=new Node<T>;

if(newnode==NULL)

{

throw("New Node cnnot be created");

}

else

{

newnode->prev=NULL;

newnode->data=value;

newnode->next=NULL;

}

return newnode;

}

void insertAtBeginning(T value)

{

Node<T> \*NewNode=createNewNode(value);

if(isEmpty())

{

Head=NewNode;

}

else

{

NewNode->next=Head;

Head->prev=NewNode;

Head=NewNode;

}

cout<<endl<<"Item "<<value<<" inserted at the beginning of list"<<endl;

}

void insertAtEnd(T value)

{

Node<T> \*NewNode=createNewNode(value);

Node<T> \*ptr;

ptr=Head;

while(ptr->next!=NULL)

{

ptr=ptr->next;

}

ptr->next=NewNode;

NewNode->prev=ptr;

cout<<endl<<"Item "<<value<<" inserted at the end of the list"<<endl;

}

void insertAfterSpecificNode(T value,T element)

{

Node<T> \*NewNode,\*ptr,\*preptr;

NewNode=createNewNode(value);

preptr=Head;

ptr=Head->next;

while(preptr->data!=element)

{

if(ptr->next==NULL)

throw("No such element found in the list");

else

{

preptr=ptr;

ptr=ptr->next;

}

}

preptr->next=NewNode;

NewNode->prev=preptr;

NewNode->next=ptr;

ptr->prev=NewNode;

cout<<endl<<"Item "<<value<<" inserted after element "<<element<<endl;

}

void insertBeforeSpecificNode(T value,T element)

{

Node<T> \*NewNode,\*ptr,\*preptr;

NewNode=createNewNode(value);

preptr=Head;

ptr=Head->next;

while(ptr->data!=element)

{

if(ptr->next==NULL)

throw("No such element found in the list");

else

{

preptr=ptr;

ptr=ptr->next;

}

}

preptr->next=NewNode;

NewNode->prev=preptr;

NewNode->next=ptr;

ptr->prev=NewNode;

cout<<endl<<"Item "<<value<<" inserted before element "<<element<<endl;

}

void insertAtSpecificPos(T value,int position)

{

int count=1;

Node<T> \*NewNode,\*ptr,\*preptr;

NewNode=createNewNode(value);

preptr=Head;

ptr=Head->next;

while(count<position-1)

{

if(ptr->next==NULL)

throw("No such position exist in the list");

else

{

preptr=ptr;

ptr=ptr->next;

count++;

}

}

preptr->next=NewNode;

ptr->prev=NewNode;

NewNode->next=ptr;

NewNode->prev=preptr;

cout<<endl<<"Element "<<value<<" inserted at position "<<position<<endl;

}

T DeleteFromBeginning()

{

Node<T> \* temp;

T item;

if(isEmpty())

{

throw("List is empty");

}

else

{

item=Head->data;

temp=Head;

Head=Head->next;

Head->prev=NULL;

delete temp;

}

cout<<endl<<"The item is deleted from beiginning"<<endl;

return item;

}

T DeleteFromEnd()

{

Node<T>\* ptr,\*preptr;

T item;

if(isEmpty())

throw("List is empty");

else

{

preptr=Head;

ptr=Head->next;

while(ptr->next!=NULL)

{

preptr=ptr;

ptr=ptr->next;

}

item=ptr->data;

preptr->next=NULL;

delete ptr;

}

cout<<endl<<"The item is deleted from the end"<<endl;

return item;

}

T DeleteAfterSpecificNode(T element)

{

Node<T> \*ptr,\*preptr;

T item;

if(isEmpty())

throw("List is empty");

else

{

preptr=Head;

ptr=Head->next;

while(preptr->data!=element)

{

if(ptr->next==NULL)

{

throw("Element doesnot exist");

}

else

{

preptr=ptr;

ptr=ptr->next;

}

}

item=ptr->data;

preptr->next=ptr->next;

Node<T> \*postptr=ptr->next;

postptr->prev=preptr;

delete ptr;

}

cout<<endl<<"Item deleted after element "<<element<<endl;

return item;

}

T DeleteBeforeSpecificNode(T element)

{

Node<T> \*ptr,\*preptr,\*temp;

T item;

if(isEmpty())

throw("List is empty");

else

{

preptr=Head;

ptr=Head->next;

while(ptr->data!=element)

{

if(ptr->next==NULL)

{

throw("Element doesnot exist");

}

else

{

preptr=ptr;

ptr=ptr->next;

}

}

if(preptr==Head)

{

item=Head->data;

temp=Head;

Head=Head->next;

Head->prev=NULL;

delete temp;

}

else

{

item=preptr->data;

ptr->prev=preptr->prev;

Node<T> \*temp=preptr->prev;

temp->next=ptr;

delete preptr;

}

}

cout<<endl<<"Item deleted before element "<<element<<endl;

return item;

}

T DeleteAtSpecificPosition(int position)

{

Node<T> \*ptr,\*preptr,\*temp;

T item;

int count=1;

if(isEmpty())

throw("List is Empty");

else{

preptr=Head;

ptr=Head->next;

while(count<position-1)

{

if(ptr->next==NULL)

throw("No such position exist");

else

{

preptr=ptr;

ptr=ptr->next;

count++;

}

}

if(count==1)

{

item=Head->data;

temp=Head;

Head=Head->next;

Head->prev=NULL;

delete temp;

}

else{

item=ptr->data;

preptr->next=ptr->next;

Node<T> \*postptr=ptr->next;

postptr->prev=preptr;

delete ptr;

}

cout<<endl<<"The item at position "<<position<<" was deleted"<<endl;

return item;

}

}

void display()

{

if(Head==NULL)

throw("List is empty");

else

{

Node<T> \*temp=Head;

int i=1;

cout<<endl<<"The elements in the list are:";

cout<<endl<<"Element "<<i<<" ="<<temp->data;

while(temp->next!=NULL)

{

i++;

temp=temp->next;

cout<<endl<<"Element "<<i<<" ="<<temp->data;

}

}

}

};

int main()

{

char check='i';

DoubleLinkedList<int> obj;

int num;

try{

cout<<endl<<"Welcome to the double linked list"<<endl;

cout<<endl<<"The operation available in the list are:";

cout<<endl<<"0.Insert at the beginning [0]";

cout<<endl<<"1.Insert at the end [1]";

cout<<endl<<"2.Insert after the specific node [2]";

cout<<endl<<"3.Insert before specific node [3]";

cout<<endl<<"4.Insert at specific position [4]";

cout<<endl<<"5.Delete from the beginning [5]";

cout<<endl<<"6.Delte from the end [6]";

cout<<endl<<"7.Delete after specific node [7]";

cout<<endl<<"8.Delete at specific position [8]";

cout<<endl<<"9.Delete before specific node [9]";

cout<<endl<<"10.Display [d or D]";

while(check!='e' && check!='E')

{

cout<<endl<<"Enter your choice as per above menu and 'e' or 'E' to end the program:";

cin>>check;

if(check=='0')

{

cout<<endl<<"Enter the number you want to insert at beginning of the list:";

cin>>num;

obj.insertAtBeginning(num);

}

else if(check=='1')

{

cout<<endl<<"Enter the number you want to insert at end of the list:";

cin>>num;

obj.insertAtEnd(num);

}

else if(check=='2')

{

int ele;

cout<<endl<<"Enter the element after which you want to perform insertion:";

cin>>ele;

cout<<endl<<"Enter the number you want to insert after the element "<<ele<<" :";

cin>>num;

obj.insertAfterSpecificNode(num,ele);

}

else if(check=='3')

{

int ele;

cout<<endl<<"Enter the element before which you want to perform insertion:";

cin>>ele;

cout<<endl<<"Enter the number you want to insert before the element "<<ele<<" :";

cin>>num;

obj.insertBeforeSpecificNode(num,ele);

}

else if(check=='4')

{

int ele;

cout<<endl<<"Enter the position at which you want to perform insertion:";

cin>>ele;

cout<<endl<<"Enter the number you want to insert at the position "<<ele<<" :";

cin>>num;

obj.insertAtSpecificPos(num,ele);

}

else if(check=='5')

{

int answer=obj.DeleteFromBeginning();

cout<<endl<<"The element deleted is:"<<answer;

}

else if(check=='6')

{

int answer=obj.DeleteFromEnd();

cout<<endl<<"The element deleted is:"<<answer;

}

else if(check=='7')

{

int ele;

cout<<endl<<"Enter the element after which you want to perform deletion:";

cin>>ele;

int answer=obj.DeleteAfterSpecificNode(ele);

cout<<endl<<"The element deleted is:"<<answer;

}

else if(check=='8')

{

int ele;

cout<<endl<<"Enter the position at which you want to perform deletion:";

cin>>ele;

int answer=obj.DeleteAtSpecificPosition(ele);

cout<<endl<<"The element deleted is:"<<answer;

}

else if(check=='9')

{

int ele;

cout<<endl<<"Enter the element before which you want to perform deletion:";

cin>>ele;

int answer=obj.DeleteBeforeSpecificNode(ele);

cout<<endl<<"the element deleted is:"<<answer;

}

else if(check=='D'||check=='d')

obj.display();

else if(check=='e'||'E')

break;

else

cout<<endl<<"Please choose valid option";

}

}

catch(const char\* error)

{

cout<<error<<endl;

}

return 0;

}