#include<iostream>

#include<conio.h>

using namespace std;

struct node //structure type node

{

node \*prev; //prevoius part

int data; // data part

node \*next; //next part

};

class DLL

{

public:

node \*start, \*temp1, \*temp2, \*temp3;

public :

DLL() //constructure

{

start = NULL;

}

///////////////////////////////////////

void add (int value)

{

node \*s, \*temp;

temp = new node;

temp->data = value;

temp->next = NULL;

if (start == NULL)

{

temp->prev = NULL;

start = temp;

}

else

{

s = start;

while (s->next != NULL)

s = s->next;

s->next = temp;

temp->prev = s;

}

}

/////////////////////////////////////////////////

void display() //display function simple one

{

temp3=start;

if(start==NULL)

cout<<"no node to display"<<endl;

else

{

while(temp3->next!=NULL)

{

cout<<"Data stored is "<<temp3->data<<" at "<<temp3<<endl;

temp3=temp3->next;

}

cout<<"Data stored is "<<temp3->data<<" at "<<temp3<<endl;

}

}

////////////////////////////////////////////////

void search() //searching the elements that found first in this link list.

{

int p;

cout<<"enter no to search"<<endl;

cin>>p;

temp1=start;

while(temp1->next!=NULL)

{

if(temp1->data==p)

{

cout<<temp1->data<<" is stored in "<< temp1->next<<endl;

}

temp1=temp1->next;

}

}

////////////////////////////////////////////////

void delnode() //deleting the node at various locations

{

int num;

char d;

cout<<"press 's' to delete from start,'m' for midd , 'e' for end"<<endl;

cin>>d;

switch (d)

{

case's': //delete at strat

if(start==NULL)

{

cout<<"no node to delete"<<endl;

}

else

{

temp1=start;

start=start->next;

start->prev=NULL;

delete temp1;

}

break;

case'e': //delete end

if(start==NULL)

{

cout<<"no node to delete"<<endl;

}

else

{

temp1=start;

while(temp1->next!=NULL)

{

temp2=temp1;

temp1=temp1->next;

}

delete temp1;

temp2->next=NULL;

}

break;

case'm': //delete midint num;

cout<<"enter node you want to delete"<<endl;

cin>>num;

temp1=start;

for(int i=1;i<num;i++)

{

if(start==NULL)

cout<<"given node does not exist"<<endl;

else

{

temp2=temp1;

temp1=temp1->next;

}

}

temp3=temp1->next;

temp2->next=temp3;

temp3->prev=temp2;

delete temp1;

break;

}

}

void show() //backward display

{

cout<<"backward display"<<endl;

temp3=start;

if(start==NULL)

cout<<"no node to display"<<endl;

else

{

while(temp3->next!=NULL)

{

temp3=temp3->next;

}

while(temp3->prev!=NULL)

{

cout<<"Data stored is "<<temp3->data<<" at "<<temp3<<endl;

temp3=temp3->prev;

}

cout<<"Data stored is "<<temp3->data<<" at "<<temp3<<endl;

}

}

};

////////////////////////////////////////////////////

int main()

{

cout<<"Hi"<<endl;

int a;

DLL s;

cout<<"Enter Number to add in Link List"<<endl;

cin>>a;

s.add(a);

cout<<"Enter Number to add in Link List"<<endl;

cin>>a;

s.add(a);

cout<<"Enter Number to add in Link List"<<endl;

cin>>a;

s.add(a);

cout<<"Enter Number to add in Link List"<<endl;

cin>>a;

s.add(a);

cout<<"Enter Number to add in Link List"<<endl;

cin>>a;

s.add(a);

s.display();

//create menu in main.....

// This is one approach for making DLL have a closed look.

getchar();

getchar();

}