//Singly Linked List Implementation

#include<iostream>

#include<conio.h>

#include<stdlib.h>

using namespace std;

class node

{

public:

int data;

node\* next;

};

class List:public node

{

node \*head,\*tail;

public:

List()

{

head=NULL;

tail=NULL;

}

void create();

void insert();

void delet();

void display();

void search();

};

int main()

{

List l;

int ch;

while(1)

{

cout<<"\n\*\*\*\* MENU \*\*\*\*";

cout<<"\n1:CREATE\n2:INSERT\n3:DELETE\n4:SEARCH\n5:DISPLAY\n6:EXIT\n";

cout<<"\nEnter Your Choice:";

cin>>ch;

switch(ch)

{

case 1:

l.create();

break;

case 2:

l.insert();

break;

case 3:

l.delet();

break;

case 4:

l.search();

break;

case 5:

l.display();

break;

case 6:

return 0;

}

}

return 0;

}

void List::create()

{

node \*temp;

temp=new node;

int value;

cout<<"\nEnter an Element:";

cin>>value;

temp->data=value;

temp->next=NULL;

if(head==NULL)

{

head=temp;

tail=head;

}

else

{

tail->next=temp;

tail=temp;

}

}

void List::insert()

{

node \*prev,\*cur;

prev=NULL;

cur=head;

int i=1,pos,ch,value;

node \*temp=new node;

cout<<"\nEnter an Element:";

cin>>value;

temp->data=value;

temp->next=NULL;

cout<<"\nINSERT AS\n1:FIRSTNODE\n2:LASTNODE\n3:IN BETWEEN head & tail NODES";

cout<<"\nEnter Your Choice:";

cin>>ch;

switch(ch)

{

case 1:

temp->next=head;

head=temp;

break;

case 2:

tail->next=temp;

tail=temp;

break;

case 3:

cout<<"\nEnter the Position to Insert:";

cin>>pos;

while(i!=pos)

{

prev=cur;

cur=cur->next;

i++;

}

if(i==pos)

{

prev->next=temp;

temp->next=cur;

}

else

cout<<"\nNot Able to Insert";

break;

}

}

void List::delet()

{

node \*prev=NULL,\*cur=head;

int i=1,pos,ch;

cout<<"\nDELETE\n1:FIRSTNODE\n2:LASTNODE\n3:IN BETWEEN head&tail NODES";

cout<<"\nEnter Your Choice:";

cin>>ch;

switch(ch)

{

case 1:

if(head!=NULL)

{

cout<<"\nDeleted Element is "<<head->data;

head=head->next;

}

else

cout<<"\nNot Able to Delete";

break;

case 2:

while(cur!=tail)

{

prev=cur;

cur=cur->next;

}

if(cur==tail)

{

cout<<"\nDeleted Element is: "<<cur->data;

prev->next=NULL;

tail=prev;

}

else

cout<<"\nNot Able to Delete";

break;

case 3:

cout<<"\nEnter the Position of Deletion:";

cin>>pos;

while(i!=pos)

{

prev=cur;

cur=cur->next;

i++;

}

if(i==pos)

{

cout<<"\nDeleted Element is: "<<cur->data;

prev->next=cur->next;

}

else

cout<<"\nNot Able to Delete";

break;

}

}

void List::display()

{

node \*temp=head;

if(temp==NULL)

{

cout<<"\nList is Empty";

}

while(temp!=NULL)

{

cout<<temp->data;

cout<<"-->";

temp=temp->next;

}

cout<<"NULL";

}

void List::search()

{

int value,pos=0;

bool flag=false;

if(head==NULL)

{

cout<<"List is Empty";

return;

}

cout<<"Enter the data to be Searched:";

cin>>value;

node \*temp;

temp=head;

while(temp!=NULL)

{

pos++;

if(temp->data==value)

{

flag=true;

cout<<"Element"<<value<<"is Found at "<<pos<<" Position";

return;

}

temp=temp->next;

}

if(!flag)

{

cout<<"Element "<<value<<" not Found in the List";

}

}