CODE:

#include<iostream>

using namespace std;

class node

{ public:

int data;

node \*next;

};

class singly\_linked

{

node \*ptr;

node \*head;

public:

singly\_linked()

{

head=NULL;

}

void create\_node(int element)

{

ptr=new node;

ptr->data=element;

ptr->next= NULL;

}

void insert\_beg()

{

if(head==NULL)

{

head=ptr;

}

else

{

ptr->next=head;

head=ptr;

}

}

void insert\_tail()

{

if(head==NULL)

{

head=ptr;

}

node\* temp;

temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=ptr;

}

void insert\_intermediary(int n)

{

node\* temp;

temp=head;

int i=n,length=1;

/\*while(temp->next!=NULL)

{

temp=temp->next;

length++;

cout<<length<<"\n";

}

if(n<1)

{

cout<<"No such position\n";

}

if(i=length)

{

insert\_tail();

}

if(i>length)

{

cout<<"No insertion\n";

}

else{\*/

temp=head;

while(i<n-1)

{

temp=temp->next;

i++;

}

ptr->next=temp->next;

temp->next=ptr;

}

void remove\_beg()

{

if(head==NULL)

{

cout<<"Underflow";

}

node\* temp;

temp=head;

if(temp->next==NULL)

{

delete temp;

}

else

head=temp->next;

delete temp;

}

void remove\_tail()

{

node\*temp1;

node\*temp2;

temp1=head;

while(temp1->next!=NULL)

{

temp2=temp1;

temp1=temp1->next;

}

temp2->next=NULL;

delete temp1;

}

void remove\_intermediary(int n)

{

node\*temp1;

node\*temp2;

temp1=head;

int i=1;

while(i<n)

{

temp2=temp1;

temp1=temp1->next;

i++;

}

temp2->next=temp1->next;

delete temp1;

}

void no\_of\_element()

{

node\*temp;

int length=0;

while(temp->next!=NULL)

{

temp=temp->next;

length++;

}

cout<<"Number of elements are "<<length<<endl;

}

void display()

{

node\* temp;

temp=head;

while(temp!=NULL)

{

cout<<temp->data;

cout<<"-->";

temp=temp->next;

}

}

void disp\_nth()

{

int a;

cout<<"Enter the position of node"<<endl;

cin>>a;

int i = 1;

node \*temp;

temp = head;

while (i < a)

{

temp = temp->next;

i++;

}

cout<<"The requested data is "<<temp->data<<endl;

}

void search\_ele(int d)

{

node \*tmp;

tmp = head;

int i = 1, flag = 0;

while(tmp != NULL)

{

if (tmp->data == d)

{

flag = 1;

break;

}

else

{

tmp = tmp->next;

i++;

}

}

if(flag == 1)

cout<<d<<" found at "<<i<<" position"<<endl;

else

cout<<"Not found"<<endl;

}

};

int main()

{

singly\_linked l;

int a,choice,n;

char ch='y';

while((ch=='y')||(ch=='Y'))

{

cout<<"Enter the element to be inserted\n";

cin>>a;

l.create\_node(a);

cout<<"1.Insert at begining\n2.Insert at tail\n3.Insert at intermediary\n";

cin>>choice;

if(choice==1)

{

l.insert\_beg();

}

if(choice==2)

{

l.insert\_tail();

}

if(choice==3)

{

cout<<"Enter the position: ";

cin>>n;

l.insert\_intermediary(n);

}

cout<<"Do you want to enter more?\n";

cin>>ch;

}

l.display();

cout<<endl;

cout<<"DELETION\n";

char ch1='y';

while((ch1=='y')||(ch1=='Y'))

{

cout<<"Do you want to delete:\n";

cin>>ch1;

cout<<"1.Delete at begining\n2.Delete at tail\n3.Delete at intermediary\n";

cin>>choice;

if(choice==1)

{

l.remove\_beg();

}

if(choice==2)

{

l.remove\_tail();

}

if(choice==3)

{

cout<<"Enter the position: ";

cin>>n;

l.remove\_intermediary(n);

}

cout<<"Do you want to delete more?\n";

cin>>ch1;

}

l.display();

char ch2='y';

int choice1;

while((ch2=='y')||(ch2=='Y'))

{

cout<<"1.Number of elements\n2.Display nth element\n3.Search an element\n";

cin>>choice1;

if(choice1==1)

{

l.no\_of\_element();

}

if(choice1==1)

{

l.disp\_nth();

}

if(choice1==3)

{

int d;

cout<<"Enter the element to be found:\n";

cin>>d;

l.search\_ele(d);

}

}

return (0);

}

OUTPUT:

