singly\_07.cpp

**Compile:** g++ singly\_07.cpp -o singly\_07

**Run:** ./singly\_07

**Program:**

#include<iostream>

#include<stdlib.h>

using namespace std;

int ch,data1,data2 ;

struct node

{

int data;

struct node \*next;

};

struct node \*list=NULL,\*p,\*q,\*r,\*s;

classLinkedlist

{

public:

void menu()

{

do

{

cout<<"\nEnter your choice\n1. Insertion at the beginning\n2. Insert at the end\n3. Display\n4. Insert before a particular element\n5. Insert after a particular element\n6. Exit\n7. Delete from beginning\n8. Delete from end\n9. Delete a particular element\n10. Search an element\n11. Sort\n12. Reverse\n13. Count";

cin>>ch;

switch(ch)

{

case 1:

insertb();

break;

case 2:

inserte();

break;

case 3:

display();

break;

case 4:

insertbp();

break;

case 5:

insertap();

break;

case 6:

break;

case 7:

deleteb();

break;

case 8:

deletee();

break;

case 9:

deletep();

break;

case 10:

search();

break;

case 11:

sort();

break;

case 12:

reverse();

break;

case 13:

count();

break;

default:

cout<<"\n Enter proper option :)";

}

}while(ch!=6);

}

void count()

{

inti=0;

if(list==NULL)

{

cout<<endl<<"List is empty!!! ";

}

else

{

q=list;

while(q->next!=NULL)

{

q=q->next;

i++;

}

cout<<endl<<"number of element "<<i+1;

}

}

void reverse()

{

if(list==NULL)

{

cout<<endl<<"List is empty";

}

else

{

struct node \*temp;

temp=NULL;

q=s=list;

r=q->next;

while(r!=NULL)

{

temp=q;

q=r;

r=q->next;

q->next=temp;

}

list=q;

s->next=NULL;

}

}

void sort()

{

if(list==NULL)

{

cout<<endl<<"List is empty!!! ";

}

else

{

q=list;

//r=list;

while(q!=NULL)

{

r=q->next;

while(r!=NULL)

{

if(r->data < q->data)

{

int temp;

temp=r->data;

r->data=q->data;

q->data=temp;

}

r=r->next;

}

q=q->next;

}

}

}

void search()

{

if(list==NULL)

{

cout<<endl<<"List is empty!!! ";

}

else

{ inti=0;

cout<<endl<<"Enter the search element";

cin>>data2;

q=list;

while(q->data!=data2 && q->next!=NULL)

{

i=i+1;

q=q->next;

}

if(q->data==data2)

{

cout<<endl<<"Data found at location "<<i;

}

else

{

cout<<endl<<"Data not found";

}

}

}

voiddeletep()

{

if(list==NULL)

{

cout<<"List is empty"<<endl;

}

else

{

if(list->next==NULL)

{

cout<<"The element deleted is: "<<list->data<<endl;

//free(list);

list=NULL;

}

else

{

cout<<"Enter the element you want to delete"<<endl;

cin>>data2;

if(list->data==data2)

{

q=list;

list=list->next;

free(q);

}

else

{

q=list;

while(q->next!=NULL && q->data!=data2)

{

r=q;

q=q->next;

}

if(q->data==data2)

{

s=q->next;

r->next=s;

free(q);

}

else

cout<<"Data not found"<<endl;

}

}

}

}

voiddeleteb()

{

if(list==NULL)

{

cout<<"List is empty"<<endl;

}

else

{

if(list->next==NULL)

{

cout<<"The element deleted is: "<<list->data<<endl;

//free(list);

list=NULL;

}

else

{

q=list;

cout<<"The element deleted is: "<<q->data<<endl;

list=list->next;

free(q);

}

}

}

voiddeletee()

{

if(list==NULL)

{

cout<<"List is empty"<<endl;

}

else

{

if(list->next==NULL)

{

cout<<"The element deleted is: "<<list->data<<endl;

//free(list);

list=NULL;

}

else

{

q=list;

while(q->next!=NULL)

{

r=q;

q=q->next;

}

cout<<"The element deleted is: "<<q->data<<endl;

r->next=NULL;

free(q);

}

}

}

voidinsertap()

{

p=new node;

cout<<"Enter data\n";

cin>>data1;

p->data=data1;

if(list==NULL)

{

cout<<"\nList is empty!!! We cant insert an element :(:(";

}

else

{

cout<<"\n Enter the element after which you want to insert new element";

cin>>data2;

q=list;

r=list;

cout<< data2;

while(q->data != data2 && q->next!=NULL)

{

q=q->next;

}

if(q->data==data2)

{

r=q->next;

q->next=p;

p->next=r;

}

else

{

cout<<"\n Data not found :( :(";

}

}

}

voidinsertbp()

{

p=new node;

cout<<"\n Enter data";

cin>>data1;

p->data=data1;

if(list==NULL)

{

cout<<"\nList is empty!!! We cant insert an element :(:(:";

}

else

{

cout<<"\nEnter the element before which you want to insert new element";

cin>>data2;

q=r=list;

if(q->data == data2)

{

p->next=q;

list=p;

}

else

{

while(q->data != data2 && q->next!=NULL)

{

r=q;

q=q->next;

}

if(q->data == data2)

{

r->next=p;

p->next=q;

}

else

{

cout<<"\nData not found";

}

}

}

}

voidinserte()

{

p=new node;

cout<<"\n Enter data";

cin>>data1;

p->data=data1;

if(list==NULL)

{

p->next=NULL;

list=p;

}

else

{

p->next=NULL;

q=list;

while(q->next!=NULL)

{

q=q->next;

}

q->next=p;

}

}

voidinsertb()

{

p=new node;

cout<<"\n Enter data";

cin>>data1;

p->data=data1;

if(list==NULL)

{

p->next=NULL;

list=p;

cout<<"\n Data inserted";

}

else

{

p->next=list;

list=p;

cout<<"\n Data inserted";

}

}

void display()

{

if(list==NULL)

{

cout<<"\n List is empty :( :(";

}

else

{

q=list;

while(q!=NULL)

{

cout<< q->data<<"===>";

q=q->next;

}

}

}

};

int main()

{

Linkedlist l;

l.menu();

return 0;

}

**Output:**

