

Doubly Linked List

Source Code :

```
#include<iostream>
#include<stdlib.h>
using namespace std;
struct node
{
    struct node *lptr;
    int data;
    struct node *rptr;
}*list=NULL,*p,*q,*r;

class doubly
{
public:
    int ch,data1,data2;
    void menu()
    {
        do
        {
            cout<<"\n\nEnter choice \n1. insert begining \n2. insert at end \n3. insert before
element\n4. insert after element \n5. display\n 6. exit";

            cin>>ch;
            switch(ch)
            {
                case 1:
                    insertbe();
                    break;
                case 2:
                    insertae();
                    break;
                case 3:
                    insertbee();
                    break;
                case 4:
                    insertatend();
                    break;
```

```

        case 5:
            display();
            break;
        default:
            cout<<"\nInvalid chice";
            break;
    }
}while(ch!=6);
}

```

```

void insertbe()
{
    p=(struct node*) malloc(sizeof(node));
    cout<<"\n\nEnter element : \n";
    cin>>data1;
    p->data=data1;

    if(list==NULL)
    {
        p->lptr=NULL;
        p->rptr=NULL;
        list=p;
    }
    else
    {
        p->rptr=list;
        list->lptr=p;
        p->lptr=NULL;
        list=p;
    }
}

```

```

void insertatend()
{
    if(list==NULL)
    {
        cout<<"\n\nList is empty";
    }
    else
    {
        p=(struct node*) malloc(sizeof(node));
        cout<<"\n\nEnter element : \n";
        cin>>data1;
        p->data=data1;
        cout<<"\n\nEnter the element after which you want to insert ";
        cin>>data2;

        q=list;

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

            q=q->rptr;
        }
        if(q->data==data2)
        {
            r=q->rptr;
            p->rptr=r;
            r->lptr=p;
            q->rptr=p;
            p->lptr=q;

        }
    }
    else
    {
        cout<<"\n\nData not found";
    }
}

void insertbee()
{

```

```

if(list==NULL)
{
    cout<<"\n\nList is empty";
}
else
{
    p=(struct node*) malloc(sizeof(node));
    cout<<"\n\nEnter element : \n";
    cin>>data1;
    p->data=data1;
    cout<<"\n\nEnter the element before which you want to insert ";
    cin>>data2;

    q=list;
    while(q->data!=data2 && q->rptr!=NULL)
    {
        r=q;
        q=q->rptr;
    }
    if(q->data==data2)
    {
        r->rptr=p;
        p->lptr=r;
        p->rptr=q;
        q->lptr=p;
    }
    else
    {
        cout<<"\n\nData not found";
    }
}
}

```

void insertae()

```

{
    p=(struct node*) malloc(sizeof(node));
    cout<<"\n\nEnter element : \n";
    cin>>data1;
    p->data=data1;

    if(list==NULL)
    {
        p->lptr=NULL;
        p->rptr=NULL;
        list=p;
    }
    else
    {
        q=list;
        while(q->rptr!=NULL)
        {
            q=q->rptr;
        }
        q->rptr=p;
        p->lptr=q;
        p->rptr=NULL;
    }
}

```

```

void display()
{
    if(list==NULL)
    {
        cout<<"\n\nList is empty";
    }
    else
    {
        q=list;
        while(q!=NULL)
        {
            cout<<q->data<<" ";
            q=q->rptr;
        }
    }
}

```

```
};
```

```
int main()
{
    doubly obj;
    obj.menu();

    return 0;
}
```

Output :

Enter choice

1. insert begining
2. insert at end
3. insert before element
- .4. insert after element
5. display
6. exit1

Enter element :

10

Enter choice

1. insert begining
2. insert at end
3. insert before element
- .4. insert after element
5. display
6. exit1020

Invalid chice

Enter choice

1. insert begining
2. insert at end
3. insert before element
- .4. insert after element
5. display
6. exit1

Enter element :

30

Enter choice

1. insert begining
2. insert at end
3. insert before element
- .4. insert after element
5. display
6. exit2

Enter element :

50

Enter choice

1. insert begining
 2. insert at end
 3. insert before element
 - .4. insert after element
 5. display
 6. exit5
- 30 10 50

Enter choice

1. insert begining
2. insert at end
3. insert before element
- .4. insert after element
5. display
6. exit3

Enter element :

10

Enter the element before which you want to insert 10

Enter choice

1. insert begining
 2. insert at end
 3. insert before element
 - .4. insert after element
 5. display
 6. exit5
- 30 10 10 50

Enter choice

1. insert begining
2. insert at end
3. insert before element
- .4. insert after element
5. display
6. exit