

<b>Ex. No. 5</b>	<b>IMPLEMENTATION OF SINGLY LINKED LIST</b>
16-08-2017	

**Question:**

Implement the following operations in a singly linked list: Insertion operation at the front of the list, Deletion of an element from the list, displaying all elements in the list.

**Algorithm:**

1. Create structure node and declare info and the pointer node\*link.
2. Assign pointer head for structure.
3. For the empty function: Create a new node and assign temp ->info=data.
4. Also temp->link=null and assign head to temp.
5. For adding at the beginning: Create new node temp and assign temp-> info = data.
6. Assign temp->link=head and assign head=temp.
7. For the delete function: create two pointers \*p and \*temp of type node.
8. Assign p=head.
9. If head ==null, then display" List is empty".
- 10.Check the condition: If head->info=data and assign temp=p.
- 11.Make head=head->link and delete temp. The element is removed.
- 12.Use while loop and check if p->link! =null.
- 13.Also, if p->link->info=data, temp=p->link and assign p->link=temp->link.

- 14.Delete the element at temp->info.
- 15.For display function: If head==null, display” List is empty”.
- 16.If else, display the elements.
- 17.Use switch case for the operations.
- 18.End.

### **Program:**

/\*To Implement the following operations in a singly linked list: Insertion operation at front of the list, Deletion of an element from the list, displaying all elements in the list. \*/

```
#include <iostream>
```

```
#include<stdlib.h>
```

```
using namespace std;
```

```
struct node
```

```
{
```

```
    int info;
```

```
    node*link;
```

```
} *head;
```

```
void addAtEmpty(int data)
```

```
{
```

```
    node *temp;
```

```
temp =new node;
temp->info=data;
cout<<"Node inserted: "<<data<<endl;
temp->link=NULL;
head=temp;
}

void addAtBeginng(int data)
{
    node* temp;
    temp=new node;
    temp->info=data;
    cout<<"Node inserted: "<<data<<endl;
    temp->link=head;
    head=temp;
}

void display()
{
    node *p;
    p=head;
    if(head==NULL)
        cout<<"\n Linked List is Empty\n";
    cout<<"The elements in the linked list are: ";
```

```
while(p!=NULL)
{
    cout<<p->info<<" -> ";
    p=p->link;
}
cout<<"NULL\n\n";
}

int del(int data)
{
    node*temp,*p;
    p=head;
    if(head==NULL)
    {
        cout<<"\n Linked List is empty\n";
        return 0;
    }
    if(head->info==data)
    {
        temp=p;
        head=head->link;
        delete temp;
        return 0;}
}
```

```
while(p->link!=NULL)
{
if(p->link->info==data)
{
temp=p->link;
p->link=temp->link;
cout<<"Deleted node: "<<temp->info<<endl;
delete(temp);
return 0;
}
p=p->link;
}
cout<<"Element not found";
}
int main()
{
int opt,data,pos;
while(1)
{
cout<<"\n Choices to do ";
cout<<"\n -----";

cout<<"\n1.Add to Empty\n2.Add at Beginning\n3.delete \n 4.Display \n
5.Exit\n" ;
```

```
cout<<"\nEnter your Option: ";
cin>>opt;
switch(opt)
{
case 1:
    cout<<"Enter the Data: ";
    cin>>data;
    addAtEmpty(data);
    break;
case 2:
    cout<<"Enter the Data: ";
    cin>>data;
    addAtBeginng(data);
    break;
case 3:
    cout<<"Enter the Data to be deleted: ";
    cin>>data;
    del(data);

    break;
case 4:
    display();
```

```
        break;

    case 5:

        exit(0);

    default:

        cout<<"Sorry wrong option";

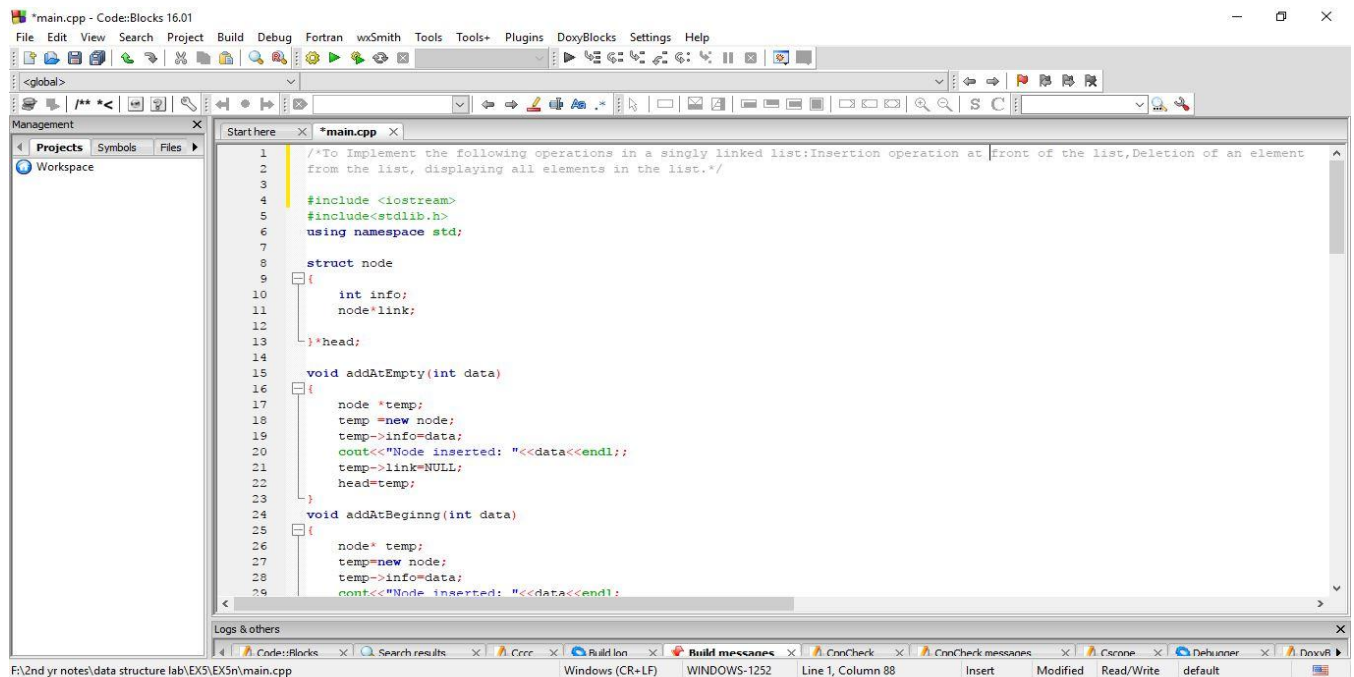
        break;

    }}

    return 0;

}
```

## Output:



# DATA STRUCTURES LAB

```
*main.cpp - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Management
  Projects Symbols Files
  Workspace
Start here *main.cpp
24 void addAtBeginning(int data)
25 {
26     node* temp;
27     temp=new node;
28     temp->info=data;
29     cout<<"Node inserted: "<<data<<endl;
30     temp->link=head;
31     head=temp;
32 }
33 void display()
34 {
35     node *p;
36     p=head;
37     if(head==NULL)
38         cout<<"\nLinked List is Empty\n";
39     cout<<"The elements in the linked list are: ";
40     while(p!=NULL)
41     {
42         cout<<p->info<<" -> ";
43         p=p->link;
44     }
45     cout<<"NULL\n\n";
46 }
47 int del(int data)
48 {
49     node*temp,*p;
50     p=head;
51     if(head==NULL)
52     {
53         cout<<"\nLinked List is empty\n";
54         return 0;
55     }
56     if(head->info==data)
57     {
58         temp=p;
59         head=head->link;
60         delete temp;
61         return 0;
62     }
63     while(p->link!=NULL)
64     {
65         if(p->link->info==data)
66         {
67             temp=p->link;
68             p->link=temp->link;
69             delete(temp);
70             return 0;
71         }
72         p=p->link;
73     }
74     return 0;
75 }
76
77 int main()
78 {
79     int opt,data,pos;
80     while(1)
81     {
82         cout<<"\n Choices to do : ";
83         cout<<"\n -----";
84         cout<<"\n1.Add to Empty\n2.Add at Beginning\n3.delete \n4.Display\n5.Exit\n";
85         cout<<"\nEnter your Option: ";
86         cin>>opt;
87         switch(opt)
88         {
89             case 1:
90                 cout<<"Enter the Data: ";
91                 cin>>data;
92                 addAtEmpty(data);
93                 break;
94             case 2:
95                 cout<<"Enter the Data: ";
96                 cin>>data;
97                 addAtBeginning(data);
98                 break;
99             case 3:
100                 del(data);
101                 break;
102             case 4:
103                 display();
104                 break;
105             case 5:
106                 exit(0);
107         }
108     }
109 }
```

```
*main.cpp - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Management
  Projects Symbols Files
  Workspace
Start here *main.cpp
48 int del(int data)
49 {
50     node*temp,*p;
51     p=head;
52     if(head==NULL)
53     {
54         cout<<"\nLinked List is empty\n";
55         return 0;
56     }
57     if(head->info==data)
58     {
59         temp=p;
60         head=head->link;
61         delete temp;
62         return 0;
63     }
64     while(p->link!=NULL)
65     {
66         if(p->link->info==data)
67         {
68             temp=p->link;
69             p->link=temp->link;
70             delete(temp);
71             return 0;
72         }
73         p=p->link;
74     }
75     return 0;
76 }
```

```
*main.cpp - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Management
  Projects Symbols Files
  Workspace
Start here *main.cpp
78 p=p->link;
79 }
80 }
81 }
82 cout<<"Element not found";
83 }
84 }
85 int main()
86 {
87     int opt,data,pos;
88     while(1)
89     {
90         cout<<"\n Choices to do : ";
91         cout<<"\n -----";
92         cout<<"\n1.Add to Empty\n2.Add at Beginning\n3.delete \n4.Display\n5.Exit\n";
93         cout<<"\nEnter your Option: ";
94         cin>>opt;
95         switch(opt)
96         {
97             case 1:
98                 cout<<"Enter the Data: ";
99                 cin>>data;
100                 addAtEmpty(data);
101                 break;
102             case 2:
103                 cout<<"Enter the Data: ";
104                 cin>>data;
105                 addAtBeginning(data);
106                 break;
107             case 3:
108                 del(data);
109                 break;
110             case 4:
111                 display();
112                 break;
113             case 5:
114                 exit(0);
115         }
116     }
117 }
```



# DATA STRUCTURES LAB

```
96 {
97     case 1:
98         cout<<"Enter the Data: ";
99         cin>>data;
100         addAtEmpty(data);
101         break;
102     case 2:
103         cout<<"Enter the Data: ";
104         cin>>data;
105         addAtBeginng(data);
106         break;
107     case 3:
108         cout<<"Enter the Data to be deleted: ";
109         cin>>data;
110         del(data);
111         break;
112     case 4:
113         display();
114         break;
115     case 5:
116         exit(0);
117     default:
118         cout<<"Sorry wrong option";
119         break;
120 }
121 return 0;
122 }
123 }
```

"F:\2nd yr notes\data structure lab\EX5\EX5n\main.exe"

```
Choices to do
-----
1.Add to Empty
2.Add at Beginning
3.delete
4.Display
5.Exit
Enter your Option: 1
Enter the Data: 111
Node inserted: 111
```

```
Choices to do
-----
1.Add to Empty
2.Add at Beginning
3.delete
4.Display
5.Exit
Enter your Option: 2
Enter the Data: 231
Node inserted: 231
```

```
Choices to do
-----
1.Add to Empty
2.Add at Beginning
3.delete
4.Display
5.Exit
Enter your Option: 2
Enter the Data: 213
Node inserted: 213
```

"F:\2nd yr notes\data structure lab\EX5\EX5n\main.exe"

```
Choices to do
-----
1.Add to Empty
2.Add at Beginning
3.delete
4.Display
5.Exit
Enter your Option: 4
The elements in the linked list are: 213 -> 231 -> 111 -> NULL
```

```
Choices to do
-----
1.Add to Empty
2.Add at Beginning
3.delete
4.Display
5.Exit
Enter your Option: 3
Enter the Data to be deleted: 231
Deleted node: 231
```

```
Choices to do
-----
1.Add to Empty
2.Add at Beginning
3.delete
4.Display
5.Exit
Enter your Option: 4
The elements in the linked list are: 213 -> 111 -> NULL
```

### VIDEO URL:

**<https://youtu.be/TyYIbDbNnFs>**

### RESULT:

The program to implement the following operations in a singly linked list: Insertion operation at the front of the list, Deletion of an element from the list, displaying all elements in the list is implemented successfully and the output is verified.