



## MET Institute of Computer Science

Roll No	1345
Name	Shubham Sarang
Topic	DLL
Title of Program	Doubly linked list

CODE:

```
/*
```

Name: Shubham Sarang

Roll no: 1345

Program: Doubly Linked List

```
*/
```

```
#include<iostream>
```

```
#include<conio.h>
```

```
using namespace std;
```

```
//Node template
```

```
class DNode
```

```
{
```

```
    public:
```

```
        int data;
```

```
        DNode *left;
```

```
        DNode *right;
```

```
};
```

```
class DList
```

```
{
```



## MET Institute of Computer Science

```
DNode *head;
```

```
DNode *tail;
```

```
public:
```

```
    DList()
```

```
    {
```

```
        head = NULL;
```

```
        tail = NULL;
```

```
    }
```

```
    void Insert(int x);
```

```
    void Display();
```

```
    void Length();
```

```
    void Search(int x);
```

```
    void Remove(int x);
```

```
};
```

```
//List template
```

```
void DList :: Insert(int x)
```

```
{
```

```
    DNode *t = new DNode();
```

```
    t->data = x;
```

```
    t->left = NULL;
```

```
    t->right = NULL;
```

```
    if(head == NULL)
```



## MET Institute of Computer Science

```
{  
    head = t;  
    tail = t;  
}  
else  
{  
    tail->right = t;  
    t->left = tail;  
    tail = t;  
}  
}  
  
void DList :: Display()  
{  
    if(head == NULL)  
    {  
        cout<<"Empty list";  
        return;  
    }  
  
    DNode *tmp = head;  
    cout<<"Forward Direction: ";
```



## MET Institute of Computer Science

```
while( tmp!= NULL)
{
    cout<<tmp->data<<"<->";
    tmp = tmp->right;
}
cout<<"End of list"<<endl;
tmp = tail;
cout<<"Reverse Direction: ";
while( tmp!= NULL)
{
    cout<<tmp->data<<"<->";
    tmp = tmp->left;
}
cout<<"Start of list"<<endl;
}
```

```
void DList::Length()
{
    int count=0;
    if(head == NULL)
    {
        cout<<"Empty list";
    }
}
```



## MET Institute of Computer Science

```
return;
```

```
}
```

```
DNode *tmp = head;
```

```
while( tmp!= NULL)
```

```
{
```

```
    count++;
```

```
    tmp = tmp->right;
```

```
}
```

```
cout<<"Length of list: "<<count;
```

```
}
```

```
void DList::Search(int x)
```

```
{
```

```
    if(head == NULL)
```

```
    {
```

```
        cout<<"Empty list";
```

```
        return;
```

```
    }
```

```
    int flag = 0;
```

```
    DNode *tmp = head;
```

```
    while(tmp!=NULL)
```



## MET Institute of Computer Science

```
{  
    if(tmp->data==x)  
    {  
        flag=1;  
        break;  
    }  
    tmp=tmp->right;  
}  
  
if(flag==1)  
{  
    cout<<"Element "<<x<<" found";  
}  
else  
{  
    cout<<"Element "<<x<<" not found";  
}  
}  
  
void DList::Remove(int x)  
{  
    if(head == NULL)
```



## MET Institute of Computer Science

```
{  
  
    cout<<"Empty list";  
  
    return;  
  
}  
  
DNode *tmp = head;  
DNode *prev = NULL;  
int flag = 0;  
while(tmp!=NULL)  
{  
    if(tmp->data == x)  
    {  
        flag = 1;  
        break;  
    }  
    prev = tmp;  
    tmp = tmp->right;  
}  
  
//step 3: unsuccessful search return control  
if(flag==0)  
{  
    cout<<"Element "<<x<<" not found ";  
  
    return;  
}
```



## MET Institute of Computer Science

}

//Step 4: Successful search, a single node deletion

if(head == tail)

{

head = NULL;

tail = NULL;

}

else if(tmp==head) //step 4 b: head node deletion

{

head = tmp->right;

head->left = NULL;

}

else if(tmp == tail) //step4 c: tail node deletion

{

tail = prev;

tail->right = NULL;

}

else //Step 4 d: any node in the middle

{

prev->right = tmp->right;

(tmp->right)->left = prev;





## MET Institute of Computer Science

```
}
```

```
//Step 5: delete the node containing x
```

```
delete tmp;
```

```
cout<<"Element deleted ";
```

```
}
```

```
//Functions
```

```
//Menu
```

```
int main()
```

```
{
```

```
    int ch, x, y,z, num;
```

```
    DList d;
```

```
    while(1)
```

```
    {
```

```
        system("cls");
```

```
        cout<<"**Doubly linked list**"<<endl;
```

```
        cout<<"1. Insert in DLL\n";
```

```
        cout<<"2. Display List\n";
```

```
        cout<<"3. Length of DLL\n";
```

```
        cout<<"4. Search for the node in DLL\n";
```

```
        cout<<"5. Remove a node\n";
```



## MET Institute of Computer Science

```
cout<<"6. Exit\n\n";
```

```
cout<<"Enter your choice: ";
```

```
cin>>ch;
```

```
switch(ch)
```

```
{
```

```
    case 1:
```

```
        cout<<"Enter the data: ";
```

```
        cin>>num;
```

```
        d.Insert(num);
```

```
        getch();
```

```
        break;
```

```
    case 2:
```

```
        cout<<"The list is: ";
```

```
        d.Display();
```

```
        getch();
```

```
        break;
```

```
    case 3:
```

```
        cout<<"Length of the list is: ";
```

```
        d.Length();
```

```
        getch();
```



## MET Institute of Computer Science

break;

case 4:

cout<<"Enter the element to be searched: ";

cin>>y;

d.Search(y);

getch();

break;

case 5:

cout<<"Enter the element to be removed: ";

cin>>z;

d.Remove(z);

d.Display();

getch();

break;

case 6:

exit(1);

default:

cout<<"Incorrect option";

getch();

break;

}//end of switch

}//end while

## MET Institute of Computer Science

}//end main

Enter element in list:

```
C:\Geralt\DSL\Linked list\DLL\DLL.exe
**Doubly linked list**
1. Insert in DLL
2. Display List
3. Length of DLL
4. Search for the node in DLL
5. Remove a node
6. Exit

Enter your choice: 1
Enter the data: 24_
```

Display list:

```
C:\Geralt\DSL\Linked list\DLL\DLL.exe
2. Display List
3. Length of DLL
4. Search for the node in DLL
5. Remove a node
6. Exit

Enter your choice: 2
The list is: Forward Direction: 20<->22<->24<->26<->28<->30<->16<->End of list
Reverse Direction: 16<->30<->28<->26<->24<->22<->20<->Start of list
```

```
C:\Geralt\DSL\Linked list\DLL\DLL.exe
**Doubly linked list**
1. Insert in DLL
2. Display List
3. Length of DLL
4. Search for the node in DLL
5. Remove a node
6. Exit

Enter your choice: 3
Length of the list is: Length of list: 7_
```

Length of list:



## MET Institute of Computer Science

Search element:

Element found:

```
C:\Geralt\DSL\Linked list\DLL\DLL.exe  -  □  ×

**Doubly linked list**
1. Insert in DLL
2. Display List
3. Length of DLL
4. Search for the node in DLL
5. Remove a node
6. Exit

Enter your choice: 4
Enter the element to be searched: 22
Element 22 found
```

Element not found:

```
C:\Geralt\DSL\Linked list\DLL\DLL.exe  -  □  ×

**Doubly linked list**
1. Insert in DLL
2. Display List
3. Length of DLL
4. Search for the node in DLL
5. Remove a node
6. Exit

Enter your choice: 4
Enter the element to be searched: 25
Element 25 not found_
```

Delete element:

Head node deletion:

## MET Institute of Computer Science

```
C:\Geralt\DSL\Linked list\DLL\DLL.exe
**Doubly linked list**
1. Insert in DLL
2. Display List
3. Length of DLL
4. Search for the node in DLL
5. Remove a node
6. Exit

Enter your choice: 5
Enter the element to be removed: 22
Element deleted Forward Direction: 24<->26<->28<->30<->End of list
Reverse Direction: 30<->28<->26<->24<->Start of list
```

Tail node deletion:

```
C:\Geralt\DSL\Linked list\DLL\DLL.exe
**Doubly linked list**
1. Insert in DLL
2. Display List
3. Length of DLL
4. Search for the node in DLL
5. Remove a node
6. Exit

Enter your choice: 5
Enter the element to be removed: 28
Element deleted Forward Direction: 20<->22<->24<->26<->End of list
Reverse Direction: 26<->24<->22<->20<->Start of list
```

Mid node deletion:



# MUMBAI EDUCATIONAL TRUST

THE MET LEAGUE OF COLLEGES  
**MET**  
AS SHARP AS YOU CAN GET  
Bhujbal Knowledge City

## MET Institute of Computer Science

```
C:\Geralt\DSL\Linked list\DLL\DLL.exe
**Doubly linked list**
1. Insert in DLL
2. Display List
3. Length of DLL
4. Search for the node in DLL
5. Remove a node
6. Exit

Enter your choice: 5
Enter the element to be removed: 24
Element deleted Forward Direction: 20<->22<->26<->End of list
Reverse Direction: 26<->22<->20<->Start of list
```