

# Linked List

```
// Aneesh Panchal
// 2K20/MC/21

#include<bits/stdc++.h>
using namespace std;

class Node {
public:
    int data;
    Node* next=NULL;
};

class Linkedlist {
    Node* head;

public:
    Linkedlist(){head = NULL;}

    void insert(int data_){
        Node* newNode = new Node();
        newNode->data = data_;

        Node* temp=head;

        if(head == NULL){
            head = newNode;
            return;
        }

        else if(head->data > data_){
            newNode->next = head;
            head = newNode;
            return;
        }

        while(temp->next != NULL && temp->next->data<data_){
            temp = temp->next;
        }
        newNode->next = temp->next;
        temp->next = newNode;
    }

    void print(){
        Node* temp=head;

        while(temp!=NULL){
            cout<< temp->data <<" -> ";
            temp = temp->next;
        }
        cout<<"NULL"<<endl<<endl;
    }
}
```

```

void deleted(int data_){
    Node* temp=head;

    if(head==NULL){
        cout<<"List empty"<<endl<<endl;
        return;
    }

    else if(head->data == data_){
        head = temp->next;
        return;
    }

    while(temp->next->data!=data_){
        if(temp->next->next==NULL){
            cout<<data_<<" " <<"not present in the list"<<endl<<endl;
            return;
        }
        temp=temp->next;
    }
    Node* del = temp->next;
    temp->next = temp->next->next;
    delete del;
}

};

int main()
{
    Linkedlist linkedL;
    cout<<endl;
    linkedL.deleted(4);
    linkedL.print();

    linkedL.insert(9999);
    linkedL.print();

    linkedL.insert(445);
    linkedL.print();

    linkedL.insert(3214);
    linkedL.print();

    linkedL.insert(4);
    linkedL.print();

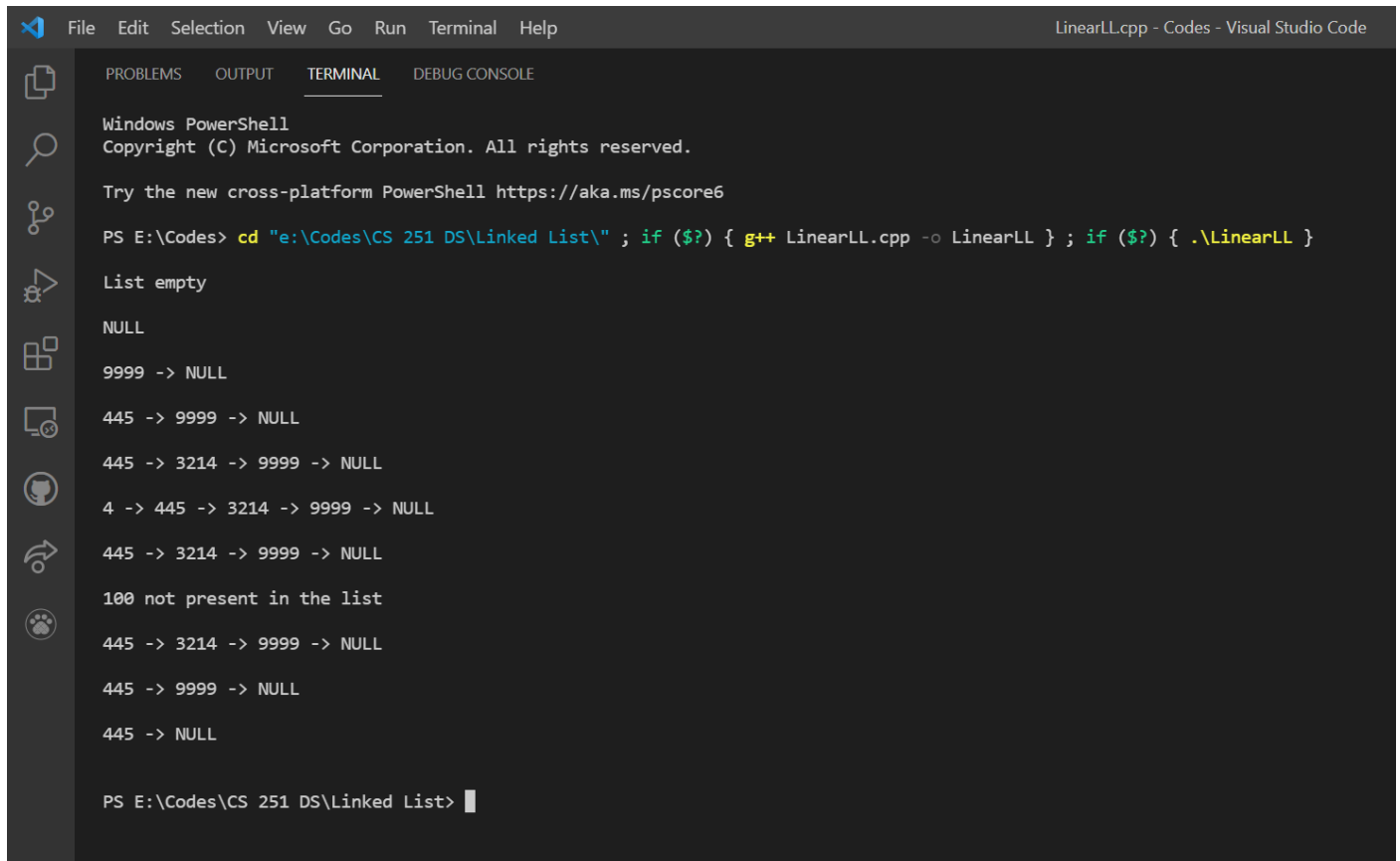
    linkedL.deleted(4);
    linkedL.print();

    linkedL.deleted(100);
    linkedL.print();

    linkedL.deleted(3214);
    linkedL.print();
}

```

```
linkedL.deleted(9999);  
linkedL.print();  
cout<<endl;  
  
return 0;  
}
```



The image shows a Visual Studio Code window with the title bar "LinearLL.cpp - Codes - Visual Studio Code". The interface includes a sidebar on the left with icons for Explorer, Search, Source Control, Run and Debug, Extensions, Testing, and Remote Explorer. The main area displays the "TERMINAL" tab, which contains the following text:

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
  
PS E:\Codes> cd "e:\Codes\CS 251 DS\Linked List\" ; if ($?) { g++ LinearLL.cpp -o LinearLL } ; if ($?) { .\LinearLL }  
  
List empty  
  
NULL  
  
9999 -> NULL  
  
445 -> 9999 -> NULL  
  
445 -> 3214 -> 9999 -> NULL  
  
4 -> 445 -> 3214 -> 9999 -> NULL  
  
445 -> 3214 -> 9999 -> NULL  
  
100 not present in the list  
  
445 -> 3214 -> 9999 -> NULL  
  
445 -> 9999 -> NULL  
  
445 -> NULL  
  
PS E:\Codes\CS 251 DS\Linked List> 
```

# Doubly Ended Linked List

```
// Aneesh Panchal
// 2K20/MC/21

#include<bits/stdc++.h>
using namespace std;

class Node {
public:
    int data;
    Node* next=NULL;
    Node* prev=NULL;
};

class DoublyEndedLinkedList {
    Node* head;

public:
    DoublyEndedLinkedList(){head = NULL;}

    void insert(int data_){
        Node* newNode = new Node();
        newNode->data = data_;

        Node* temp=head;

        if(head == NULL){
            head = newNode;
            return;
        }

        else if(head->data > data_){
            newNode->prev=NULL;
            newNode->next = head;
            head->prev=newNode;
            head=newNode;
            return;
        }

        while(temp->next != NULL && temp->next->data<data_){
            temp = temp->next;
        }
        newNode->next = temp->next;
        newNode->prev = temp;
        temp->next = newNode;
        temp->next->prev = newNode;
    }

    void print(){
        Node* temp=head;

        while(temp!=NULL){
```

```

        cout<< temp->data <<" -> ";
        temp = temp->next;
    }
    cout<<"NULL"<<endl<<endl;
}

void deleted(int data_){
    Node* temp=head;

    if(head==NULL){
        cout<<"List empty"<<endl<<endl;
        return;
    }

    else if(head->data == data_){
        head = head->next;
        head->prev = NULL;
        return;
    }

    while(temp->next->data!=data_){
        if(temp->next->next==NULL){
            cout<<data_<<" " <<"not present in the list"<<endl<<endl;
            return;
        }
        temp=temp->next;
    }

    if(temp->next->next == NULL)
    {
        Node* del = temp->next->next;
        temp->next = NULL;
        delete del;
    }
    else
    {
        Node* del = temp->next;
        temp->next = del->next;
        del->next->prev = temp;
        delete del;
    }
}

};

int main()
{
    DoublyEndedLinkedList DElinkedL;
    cout<<endl;
    DElinkedL.deleted(12);
    DElinkedL.print();

    DElinkedL.insert(2365);
    DElinkedL.print();
}

```

```

DElinkedL.insert(440);
DElinkedL.print();

DElinkedL.insert(1021);
DElinkedL.print();

DElinkedL.insert(144);
DElinkedL.print();

DElinkedL.deleted(144);
DElinkedL.print();

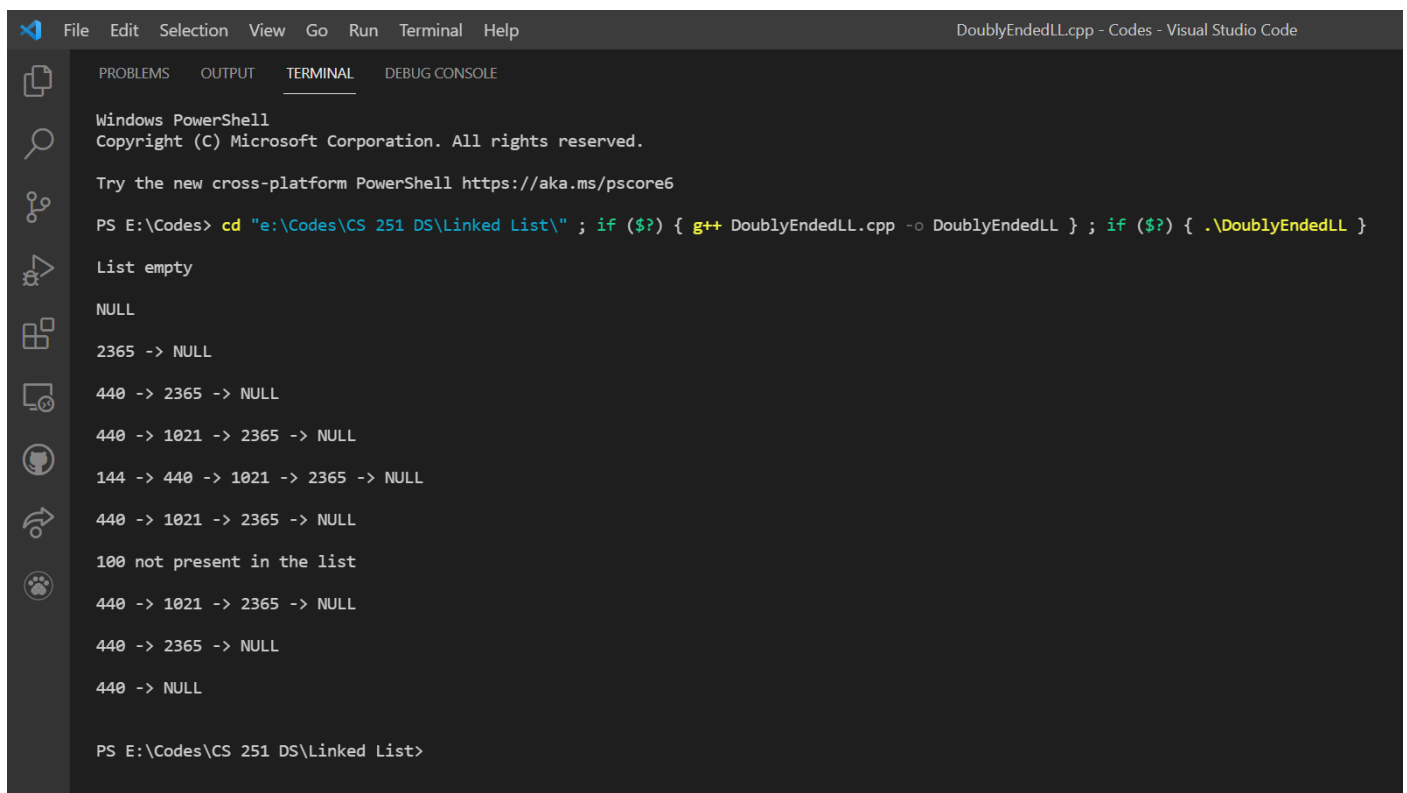
DElinkedL.deleted(100);
DElinkedL.print();

DElinkedL.deleted(1021);
DElinkedL.print();

DElinkedL.deleted(2365);
DElinkedL.print();
cout<<endl;

return 0;
}

```



Visual Studio Code interface showing the execution of a C++ program. The terminal window displays the output of the program, which includes the insertion and deletion of nodes in a doubly linked list. The output shows the list state after each operation, confirming that the nodes are correctly added and removed.

```

DoublyEndedLL.cpp - Codes - Visual Studio Code
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\Codes> cd "e:\Codes\CS 251 DS\Linked List\" ; if ($?) { g++ DoublyEndedLL.cpp -o DoublyEndedLL } ; if ($?) { .\DoublyEndedLL }

List empty

NULL

2365 -> NULL

440 -> 2365 -> NULL

440 -> 1021 -> 2365 -> NULL

144 -> 440 -> 1021 -> 2365 -> NULL

440 -> 1021 -> 2365 -> NULL

100 not present in the list

440 -> 1021 -> 2365 -> NULL

440 -> 2365 -> NULL

440 -> NULL

PS E:\Codes\CS 251 DS\Linked List>

```