C++ Assignments | Linkedlist - 1 | Week 15

Q.1 In a singly linked list, deletion of data requires modification of how many pointers?

Ans-: 2

Q.2 Predict the output for linked list = 1->2->3->4->5:

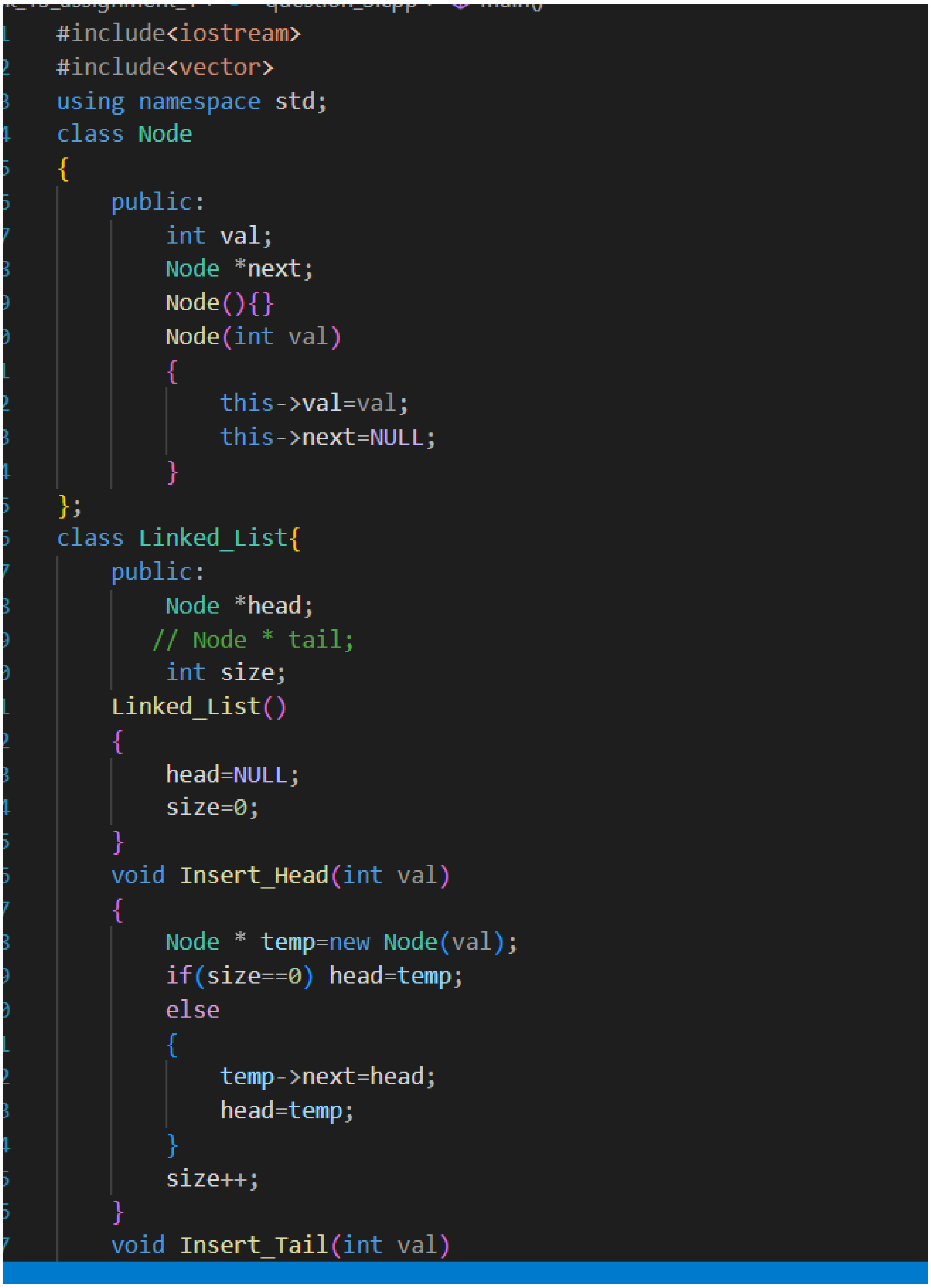
```
void traverse(Node* head) {
 while(head and head->next) {
 cout << head->data << ' ';
 head = head->next->next;
 }
}
```

Ans -: 1 3

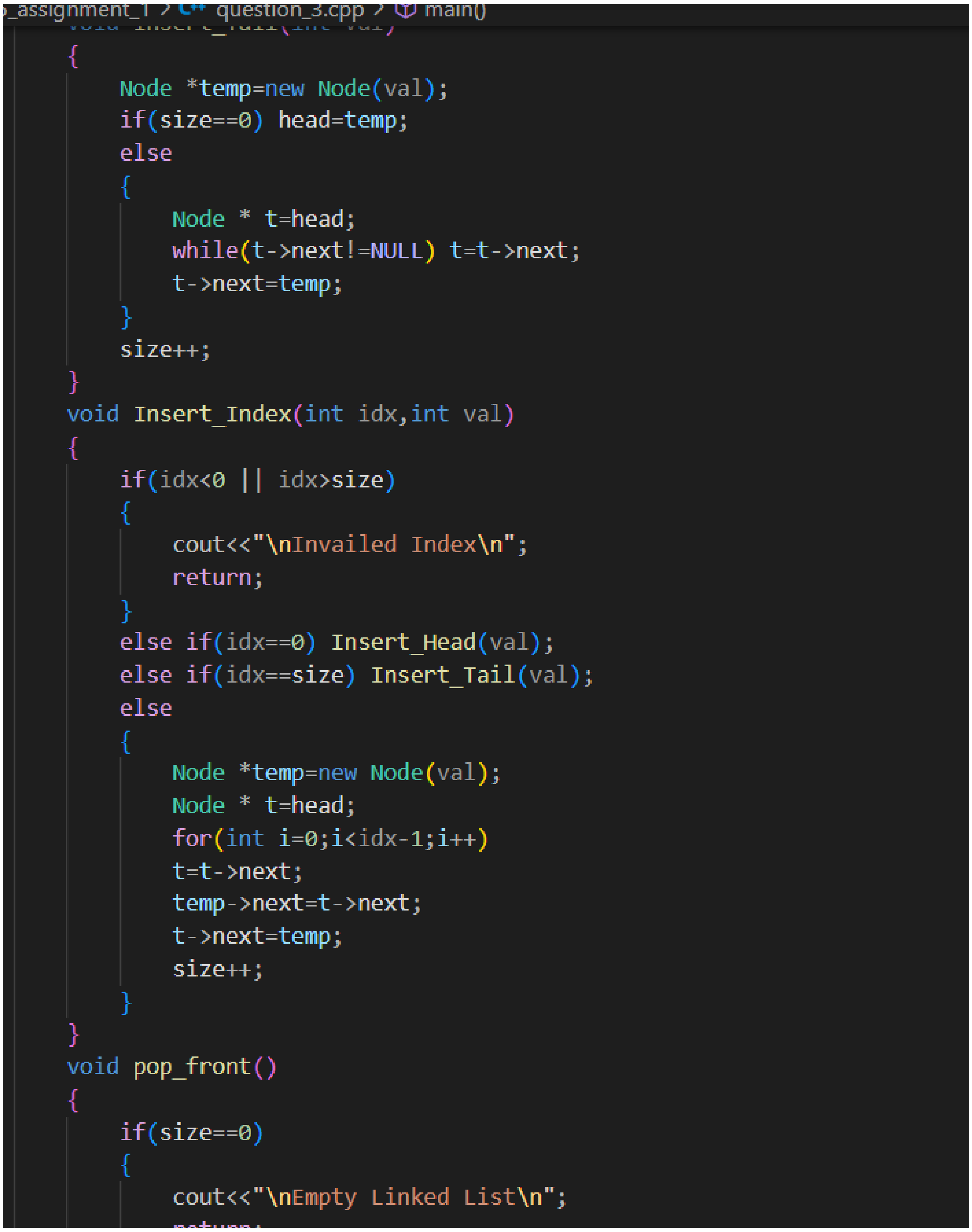
Q.3. Implement a Linked List class.

The user defined LL should have insert (head,tail,idx) , delete(head,tail,idx) , get(idx) and display functions.

Solution-:

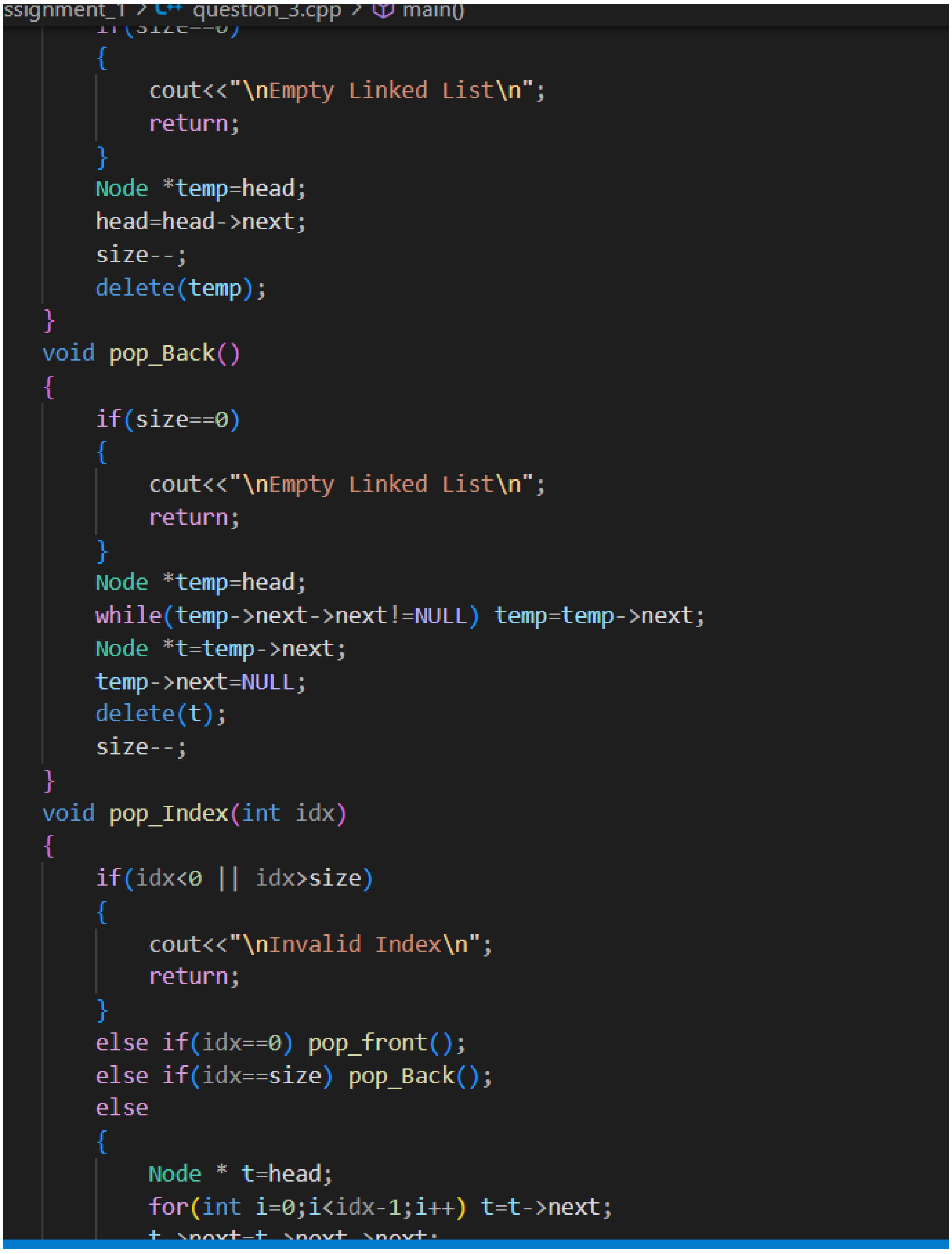


```cpp
#include<iostream>
#include<vector>
using namespace std;
class Node
{
    public:
        int val;
        Node *next;
        Node(){}
        Node(int val)
        {
            this->val=val;
            this->next=NULL;
        }
};
class Linked_List{
    public:
        Node *head;
       // Node * tail;
        int size;
    Linked_List()
    {
        head=NULL;
        size=0;
    }
    void Insert_Head(int val)
    {
        Node * temp=new Node(val);
        if(size==0) head=temp;
        else
        {
            temp->next=head;
            head=temp;
        }
        size++;
    }
    void Insert_Tail(int val)
```



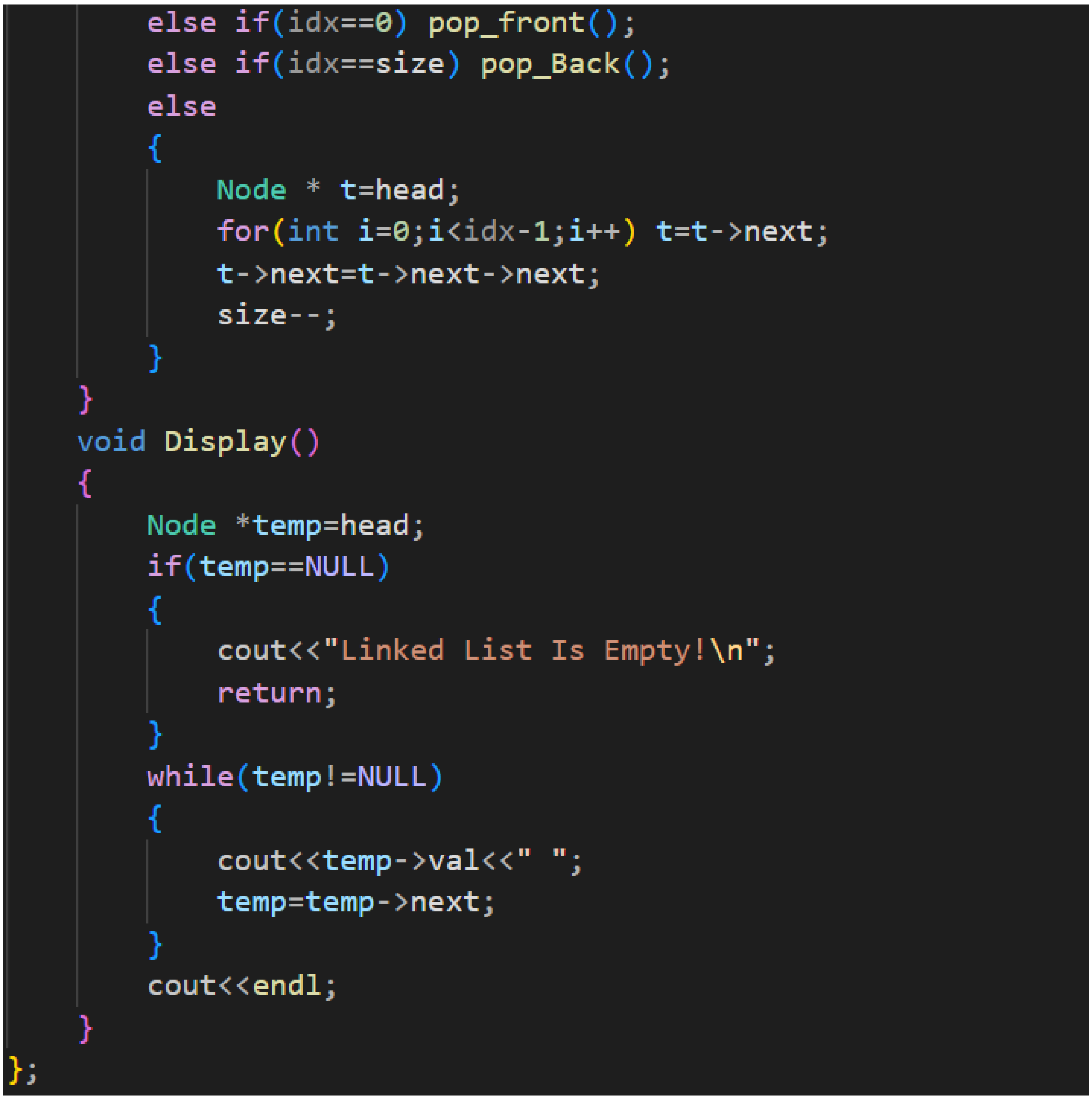
_assignment_1 > C++ question_3.cpp > main()

```
    {
        Node *temp=new Node(val);
        if(size==0) head=temp;
        else
        {
            Node * t=head;
            while(t->next!=NULL) t=t->next;
            t->next=temp;
        }
        size++;
    }
    void Insert_Index(int idx,int val)
    {
        if(idx<0 || idx>size)
        {
            cout<<"\nInvailed Index\n";
            return;
        }
        else if(idx==0) Insert_Head(val);
        else if(idx==size) Insert_Tail(val);
        else
        {
            Node *temp=new Node(val);
            Node * t=head;
            for(int i=0;i<idx-1;i++)
            t=t->next;
            temp->next=t->next;
            t->next=temp;
            size++;
        }
    }
    void pop_front()
    {
        if(size==0)
        {
            cout<<"\nEmpty Linked List\n";
```

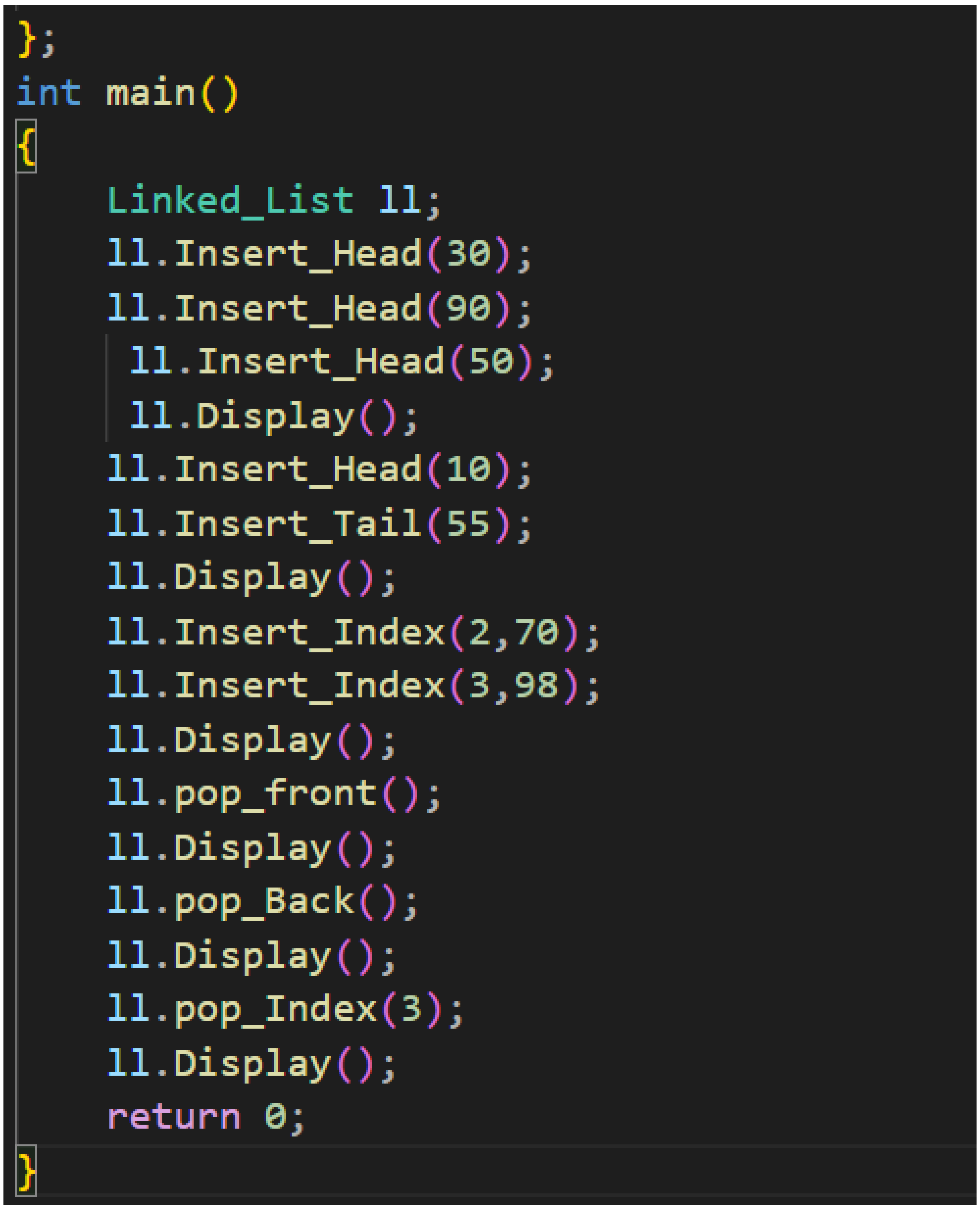


ssignment_1 > question_3.cpp > main()

```
        {
            cout<<"\nEmpty Linked List\n";
            return;
        }
        Node *temp=head;
        head=head->next;
        size--;
        delete(temp);
    }
    void pop_Back()
    {
        if(size==0)
        {
            cout<<"\nEmpty Linked List\n";
            return;
        }
        Node *temp=head;
        while(temp->next->next!=NULL) temp=temp->next;
        Node *t=temp->next;
        temp->next=NULL;
        delete(t);
        size--;
    }
    void pop_Index(int idx)
    {
        if(idx<0 || idx>size)
        {
            cout<<"\nInvalid Index\n";
            return;
        }
        else if(idx==0) pop_front();
        else if(idx==size) pop_Back();
        else
        {
            Node * t=head;
            for(int i=0;i<idx-1;i++) t=t->next;
```

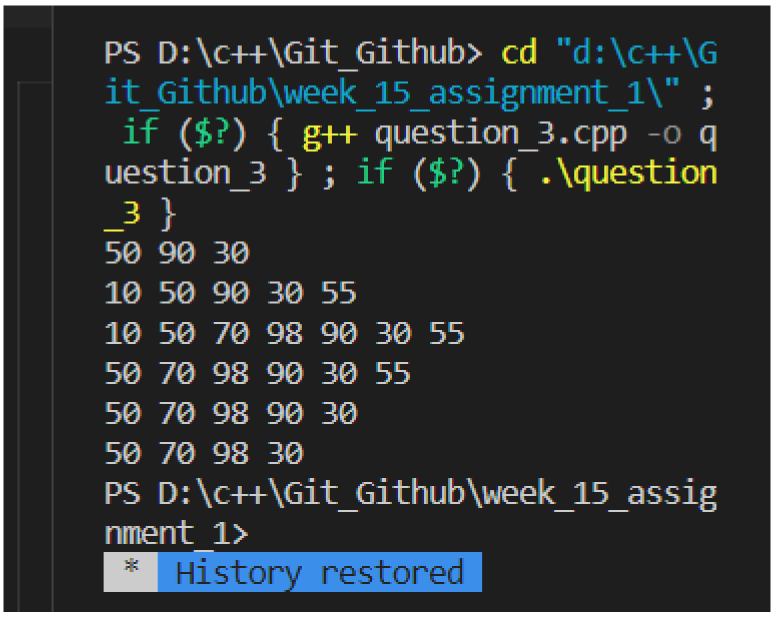


```
        else if(idx==0) pop_front();
        else if(idx==size) pop_Back();
        else
        {
            Node * t=head;
            for(int i=0;i<idx-1;i++) t=t->next;
            t->next=t->next->next;
            size--;
        }
    }
    void Display()
    {
        Node *temp=head;
        if(temp==NULL)
        {
            cout<<"Linked List Is Empty!\n";
            return;
        }
        while(temp!=NULL)
        {
            cout<<temp->val<<" ";
            temp=temp->next;
        }
        cout<<endl;
    }
};
```



```cpp
};
int main()
{
    Linked_List ll;
    ll.Insert_Head(30);
    ll.Insert_Head(90);
     ll.Insert_Head(50);
     ll.Display();
    ll.Insert_Head(10);
    ll.Insert_Tail(55);
    ll.Display();
    ll.Insert_Index(2,70);
    ll.Insert_Index(3,98);
    ll.Display();
    ll.pop_front();
    ll.Display();
    ll.pop_Back();
    ll.Display();
    ll.pop_Index(3);
    ll.Display();
    return 0;
}
```

OUTPUT ↓



```
PS D:\c++\Git_Github> cd "d:\c++\G
it_Github\week_15_assignment_1\" ;
 if ($?) { g++ question_3.cpp -o q
uestion_3 } ; if ($?) { .\question
_3 }
50 90 30
10 50 90 30 55
10 50 70 98 90 30 55
50 70 98 90 30 55
50 70 98 90 30
50 70 98 30
PS D:\c++\Git_Github\week_15_assig
nment_1>
 *  History restored
```