



THE UNIVERSITY  
OF LAHORE  
**ISLAMABAD  
CAMPUS**

## **Data Structures and algorithm (CS09203)**

### **Lab Report**

Name: Muhammad Talha Khalid  
Registration #: CSU-S16-135  
Lab Report #: 5  
Dated: 16-04-2018  
Submitted To: Mr. Usman Ahmed

The University of Lahore, Islamabad Campus  
Department of Computer Science & Information Technology

## Experiment # 3

### Inserti and deletion in Link list

#### Objective

How to insert and delete data from nodes in link list.

#### Software Tool

1. Windows 8.1
2. dev c++
3. c++

## 1 Theory

In this experiment i insert 6 calues in link list and then delete the value from it.First of all it will check the root of the node if data entered is there it will delete it otherwise it will move to next node until given value is there or next node will be NULL

link list has 3 rules:

1. it has unlimited space.
2. you can insert and delete nde easily.
3. for deletion it will check 1st node then other till the nod is !=NULL

## 2 Task

### 2.1 Procedure: Task 1

We can remove any number from that link list

### 2.2 Procedure: Task 2

```
#include<stdio.h>
#include<iostream>
```

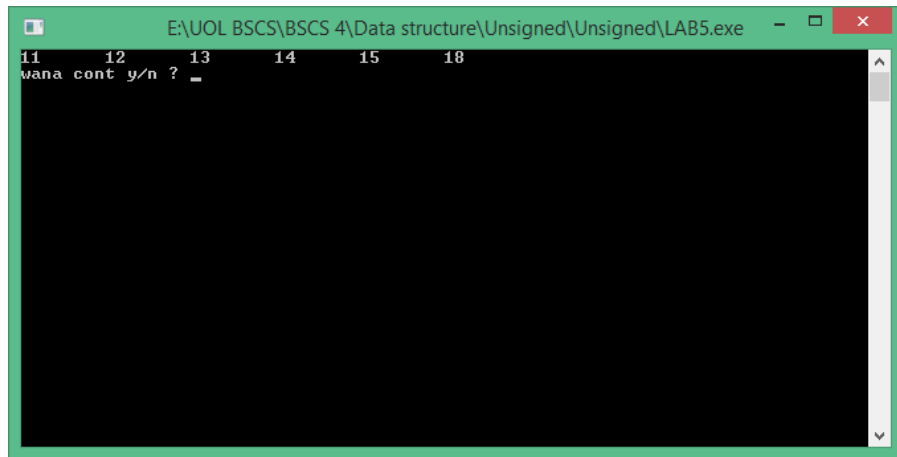


Figure 1: 6 elements in link list

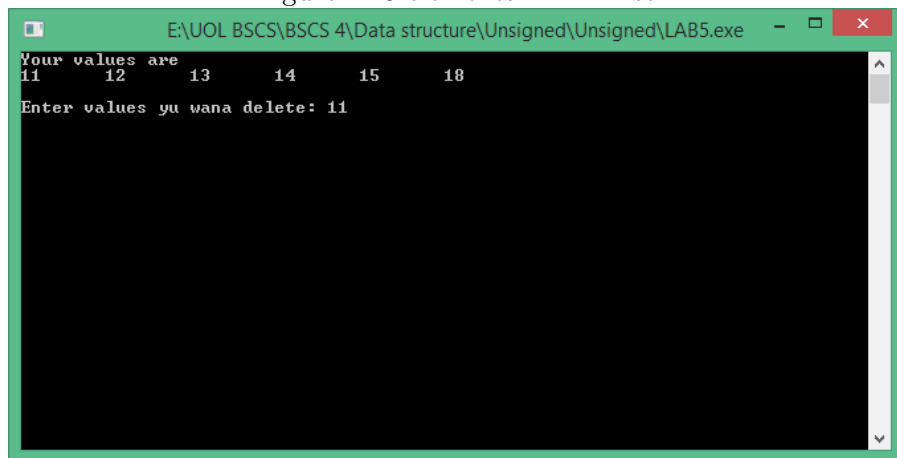


Figure 2: delete 11 from link list

```

#include<dos.h>
#include<unistd.h>
#include<string>
#include<stdlib.h>
#include<menu.h>
#include <list>
using namespace std;
class Node {
public:
int val;
Node *next;
Node() {
next=NULL;
val=0;
}
};
class ll {
public:
Node *start,*temp,*newnode;
ll() {
start=NULL;
}
void addnode(int a) {
if(start==NULL) {
temp=new Node;
start=temp;
temp->val=a;
} else {
temp=start;
while(temp->next!=NULL) {
temp=temp->next;
}
newnode=new Node;
newnode->val=a;
temp->next=newnode;
}
}
void del(int n) {
Node *prev;
temp=start;

```

```

        if (start==NULL) {

            cout << "Nothing to delete , the list is empty!\n";
        } else if(temp->val==n) {
            start=temp->next;
            delete temp;
            cout <<" value deleted\n";
            return ;
        } else {

            while (temp!=NULL) {
                if(temp->val==n) {
                    cout <<" value deleted\n";
                    prev->next=temp->next;
                    delete temp;
                    break;
                } else {

                    prev=temp;
                    temp=temp->next;
                }
            }

            //      cout << "value not found!"<<endl;
        }

    void ShowTime() {
        Node *imp;
        imp=start;
        while(imp!=NULL) {
            cout<<imp->val<<"\t";
            imp=imp->next;
        }
        cout<<endl;
    }
};

int choice;
string subchoice="n";
int num;
void menu() {
    system("cls");

```

```

cout<<" press_1_to_enter_data\n";
cout<<" press_2_to_display_data\n";
cout<<" press_3_to_remove_data\n";
cout<<" press_4_to_exit\n\n";
}
int main()
{
    ll lo;
    do {
        menu();
        cout<<" choose_from_above:_";
        cin>>choice;
        switch (choice) {
            case 1:
                do {
                    system(" cls");
                    cout<<" Enter_values:_";
                    cin>>num;
                    lo.addnode(num);
                    cout<<" wana_cont_y/n?_";
                    cin>>subchoice;
                }while(subchoice!="y");
                break;
            case 2:
                system(" cls");
                do {
                    lo.ShowTime();
                    cout<<" wana_cont_y/n?_";
                    cin>>subchoice;
                }while(subchoice!="y");
                break;
            case 3:
                system(" cls");
                do {
                    cout<<" Your_values_are\n";
                    lo.ShowTime();
                    cout<<" \nEnter_values_yu_wana_delete:_";
                    cin>>num;
                    lo.del(num);
                    cout<<" wana_exit_y/n?_";

```

```
cin>>subchoice;  
}while(subchoice!="y");  
break;  
}  
}while(choice!=4);  
return 0;  
}
```

### 3 Output:

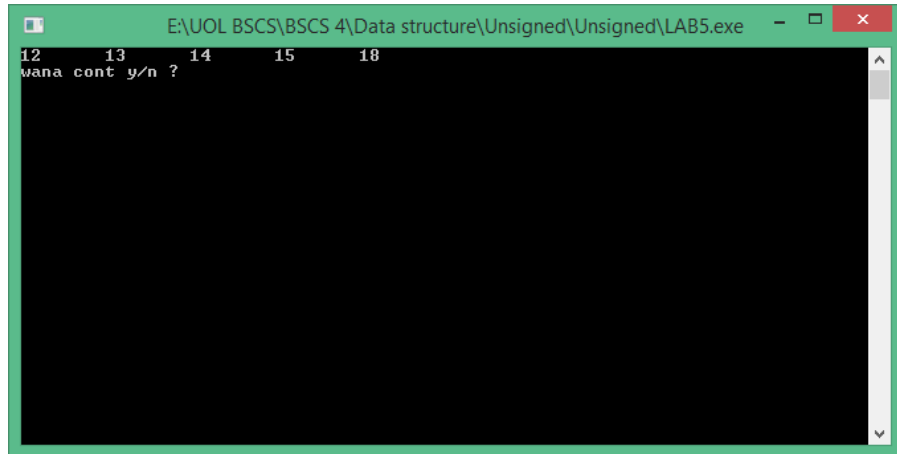


Figure 3: Output of the program after deletion

### 4 Conclusion

this is basic concept of how we add and remove nodes from link list using the address of the nodes linklist is basic concept about how to manage our data