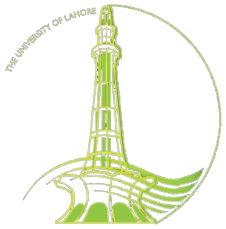

THE UNIVERSITY OF LAHORE
ISLAMABAD CAMPUS

Data Structures and Algorithms
(CS09203)

Lab Report

Name: Jassim Bashir
Registration #: SEU-F16-101

Lab Report #: 05
Dated: 30-04-2018
Submitted To: Sir. Usman Ahmed

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

Experiment # 5
Link list-Basic Deletion at desired position

Objective
The objective of this session is to insertion, traversal and deletion at desired position in link list using C++.

Software Tool
1. I use Code Blocks with GCC compiler.

Theory

This section discusses how to insert an item into, and delete an item from, a linked list. Consider the following definition of a node. (For simplicity, we assume that the info type is int. struct nodeType int info nodeType* link; ; We will use the following variable nodeType *head, *p, *q, *newNode; INSERTION:- Algorithms which insert nodes into the linked list come up in various situations. We discuss three of them here. The first one inserts a node at the beginning of the list, the second one inserts a node after a node with a given location, and the third one inserts a node into the sorted list.

Task

Procedure: Task 5

Write a C++ code using functions for the following operations. 1.Creating a linked List. 2.Traversing a Linked List. 3.Inserting the node at the start of the list. 4.Inserting a node after a given node. 5.Inserting a node in a sorted list.

2.2

```
#include<iostream>
```

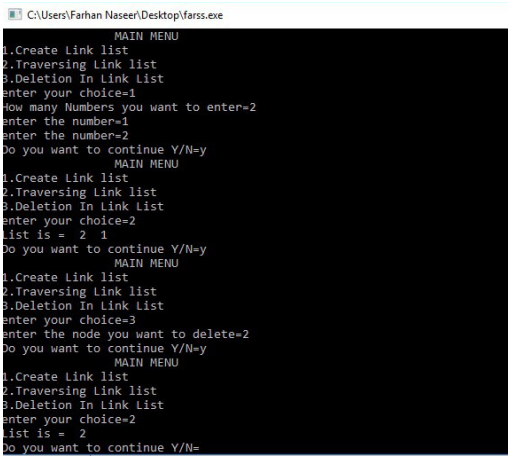

Figure 1: output

```
#include<stdlib .h>
#include<conio .h> using
namespace std ; struct
Node{ int data ;
        Node* next ;
};
struct Node* head ; void
Insert ( int x){
        Node* temp=(Node*) malloc ( sizeof (Node ));
        temp->data=x ; temp->next=head ; head=temp ;
}
void print (){
        Node* temp=head ; cout<<"List is =" ;
        while(temp != NULL){ cout<<"
            "<<temp->data ; temp=temp->next ;
        } cout<<endl ;
}
void Delete ( int n){
    struct Node* temp1=head ;
        if (n==1){ head=temp1->next ;
            free (temp1 );
            return ;
```

```
        }

        for ( int i =0;i<n-2; i++){ temp1=temp1->next ;}
                struct Node* temp2=temp1->next ;
                temp1->next=temp2->next ;
                free (temp2 );
}
int main(){ head=NULL; int
        size , j , k ;
        char ch , choice ; do{

        cout<<"\t\tMAIN MENU\t\t"<<endl ; cout<<"1.
        Create Link l i s t "<<endl ; cout<<"2. Traversing
        Link l i s t "<<endl ; cout<<"3. Deletion In Link
        List"<<endl ; cout<<"enter your choice=" ;
        cin>>choice ;
        switch( choice ){ case '1
                ' :
                        {
                                                cout<<"How many Numbers you want to enter=
                    cin>>size ; for ( j =0;j<size ; j++){ cout<<"enter the
                                number=" ; cin>>k ;
                                  Insert (k );
                        }}
                        break;
                case '2 ' :
                        {
                                                print ();
                        } break;
                case '3 ' :
                        {
        cin>>x ;
         Delete (x );
}
b
r
e
a
k;
```

```
d
ef
a
ul
t
:
        cout<<" invalid      choic    ! ! ! ! ! ! ! ! !
                             e        "<<endl ;
}
cout<<"Do        to                   continue
you want                              Y/N=" ;
                                                int x ; cout<<"enter the node you want to          delete=" ;
```

```
                cin>>ch ;
}
while (( ch=='Y' ) | | ( ch=='y ' ));
getch ();
return 0;
}
```

Conclusion

In today lab we have discussed how we can create a link list and alose learn to delete a node and display it on a screen by having a code.