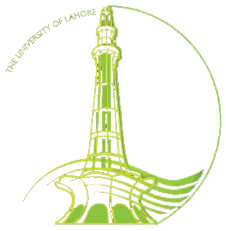

THE UNIVERSITY OF LAHORE
ISLAMABAD CAMPUS

Data Structures and Algorithms
(CS09203)

Lab Report

Name:	M.Usman Ali
Registration #:	SEU-F16-135
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.

1 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.

2 Task

2.1 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

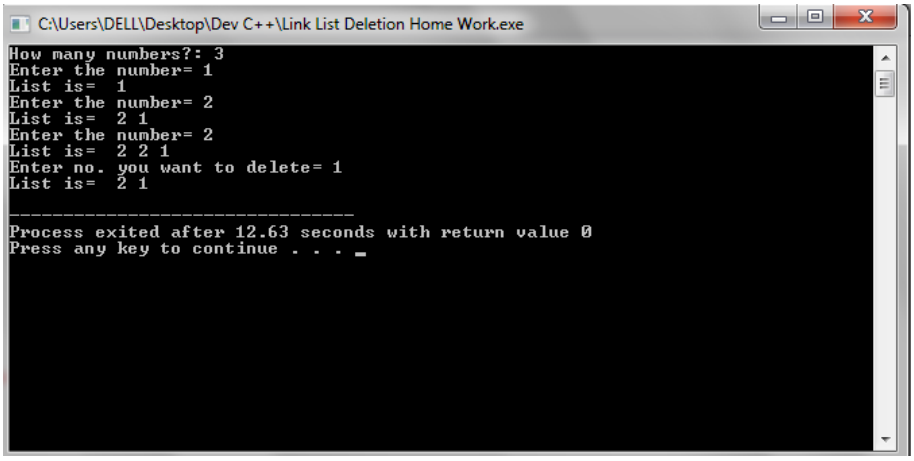

Figure 1: output

```
#include<iostream>
#include<stdlib.h>
using namespace std;
struct Node{
        int data;
        struct Node*next;

};
struct Node*head;
void insert(int x){
        struct Node*temp=(Node*)malloc(sizeof(struct Node));
        temp->data=x;
        temp->next=head;
        head=temp;
}
void print()
{
        struct Node*temp=head;
        cout<<"list is"<<endl;
        while(temp!=NULL)
        {
                cout<<temp->data;
                temp=temp->next;
```

```
        }
        cout<<endl;
}
void Delete(int n){
        int temp;
        struct Node* temp1= head;
        if(n==1){
                head=temp1->next;
                free(temp1);
                return;
        }
        int i;
        for(i=0;i<n-1;i++){
                temp1=temp1->next;
                struct Node* temp2=temp1->next;

                temp1->next=temp2->next;
                free(temp2);
        }
}
int main(){
        head=NULL;
        cout<<"how many numbers"<<endl;
        int n,i,x,y;
        cin>>n;
        for(i=0;i<n;i++){
                cout<<"enter the number"<<endl;
                cin>>x;
                insert(x);
                print();          }
                cout<<"enter no u want to delete"<<endl;
                cin>>y;
                Delete(y);
                print();

}
```

3 Conclusion

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