DATA STRUCTURE
(CS13217)

Lab Report

Name:	Zain ul abideen
Registration #:	SEU-F16-133
Lab Report #:	04
Dated:	25-04-2018
Submitted To:	Mr. Usman Ahmed

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

Experiment # 1
Implementing the link list..

Objective
To understand and implement the link list

Software Tool
1.
dev c++

1 Theory

a linked list is a linear collection of data elements, whose order is not given by their physical placement in memory. Instead, each element points to the next. It is a data structure consisting of a collection of nodes which together represent a sequence. In its most basic form, each node contains: data, and a reference (in other words, a link) to the next node in the sequence. This structure allows for efficient insertion or removal of elements from any position in the sequence during iteration. More complex variants add additional links, allowing more efficient insertion or removal of nodes at arbitrary positions. A drawback of linked lists is that access time is linear (and difficult to pipeline). Faster access, such as random access, is not feasible. Arrays have better cache locality compared to linked lists. Linked lists are among the simplest and most common data structures. They can be used to implement several other common abstract data types, including lists, stacks, queues, associative arrays, and S-expressions, though it is not uncommon to implement those data structures directly without using a linked list as the basis. .

2 Task

2.1 procedure: Task 1

```
 #include<iostream>
#include<stdio.h>
#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  ";
        while(temp!=NULL)
        {
                cout<<"  "<<temp->data;
                temp=temp->next;
        }
        cout<<" \n  ";

}


int main()
{
        insert(1);
        insert(2);
        insert(3);
        print();
;
        print();
}
```

3 Conclusion

In this lab we perform the application of llink list and also how its work and about the link list practical in dev c++.