Name: Tanzeela Asghar

Reg no: 2021 BSE 032

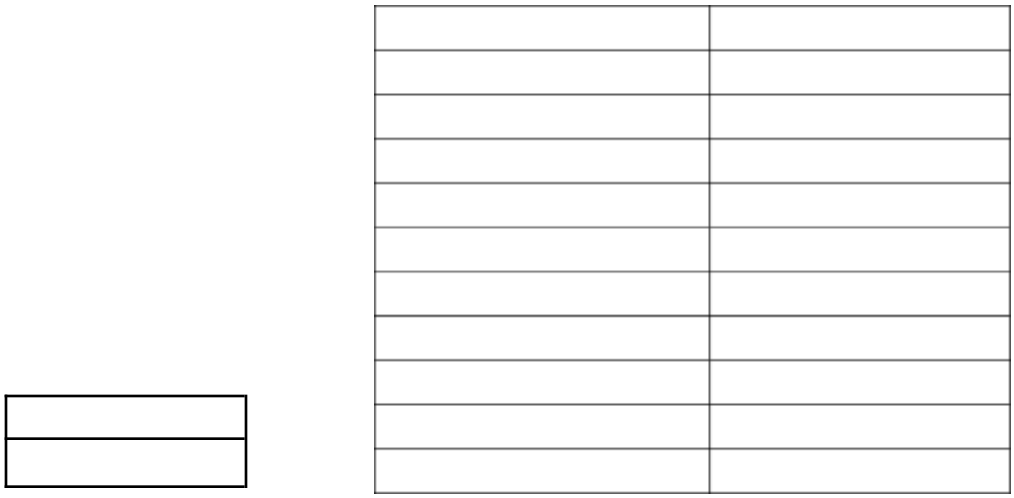
Section no: III-A

Course: Data Structure

Submitted to: Sir Rehan Ahmed Siddiqui

Lab#7

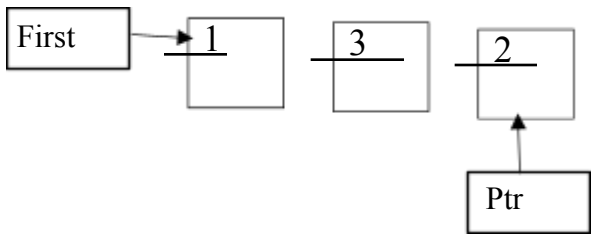
Task#1



1.	**START** Y **INFO** **LINK** (Q) Mary Q ->I (W) Helen W ->U (E) Barbara E ->T (R) Paula R-> O (T) Diana T ->P (Y) Audrey Y ->E (U) Karen U ->Q (I) Nancy I ->R (O) Ruth O -> NULL (P) Eileen P ->W
2.	first->data; 1 first->next->next->data; 3 ptr->next->data; 3 ptr->next->next; NULL

first->next->next->next;	NULL

3.



Task#2

Code:

```
#include "stdafx.h"
#include<iostream>
```

```
using namespace
std; class List
{
        struct node
        {
                                                        i
                                                        n
                                                        t
public:                                                 d
                                                        a
                                                        t
                                                        a
                                                        ;
                                                        n
                                                        o
                                                        d
                                                        e
                                                        *
                                                        n
                                                        e
                                                        x
                                                        t
                                                        ;
                                                }*head;

                                                List()
                                                {
                                                head=NULL;
        }
        ~List()
        {
        }

        void empty()
        {
                if(head==NULL)
                {
                        cout<<"List is empty"<<endl;
                }
        }
        void insert_begin(int val)
        {
                node *p;
                p=new node;
                p->data=val;
                p->next=head;
                head=p;
        }
        void insert_end(int val)
        {
                node *p,*q;
```

```
        p=head;
        while(p->next!=NULL)
        {
                p=p->next;
        }
        q=new
        node;
        q->data=val;
        p->next=q;
        q->next=NULL;

}
void insert_after(int oldV,int newV)
```

```
{
        int found=0;
        node *p;
        p=head;
        node *q;
        q=new
        node;
        q->data=newV;
        while(p->next!=NULL
        )
        {
                if(p->data==oldV)
                {
                        q->next=p->next
                        ; p->next=q;
                        found=1;
                        break;
                }
                p=p->next;
        }
        if(found==0)
        {
                q->next=NULL
                ; p->next=q;
        }
}
void deleteNode(int value)
{
        node *p,*q;
        p=head;
        if(p==NULL)
        {
                cout<<" Your list is empty you cant delete it :) ";
        }
        else if(p->data==value)
        {
                                                h
                                                e
        }                                       a
        else                                    d
        {                                       =
                                                h
                                                e
                                                a
                                                d
                                                -
                                                >
                                                n
                                                e
                                                x
                                                t
```

```
;                                                   e
d                                                   *
e                                                   p
l                                                   ,
e                                                   *
t                                                   q
e                                                   ;
p                                                   p
;                                                   =
                                                    h
                                                    e
                                                    a
n                                                   d
o                                                   ;
d                                                   while(p->data!=value)
                                                    {
                    q=p;
                    p=p->next;
                }
```

```
                                                        q
                                                        -
                                                        >
                        }                               n
                                                        e
                                                        x
        }                                               t
        void display()                                  =
        {                                               p
                                                        -
                                                        >
                                                        n
                                                        e
                                                        x
                                                        t
                                                        ;
                                                        d
                                                        e
                                                        l
                                                        e
                                                        t
                                                        e
                                                        p
                                                        ;
                                                        q=head;
                node *p;
                p=head;
                while(p!=NULL)
                {
                        cout<<p->data<<"
                        "; p=p->next;
                }
                cout<<endl;
        }
};

int _tmain(int argc, _TCHAR* argv[])
{
        List l1;
        cout<<"Check List is empty or not...\n";
        l1.empty();
        cout<<"Inserting values in list :\n";
        l1.insert_begin(8);
        l1.insert_begin(10);
        l1.insert_begin(7);
        l1.display();
        cout<<"Inserting values at the end of the list :\n";
        l1.insert_end(6);
        l1.insert_end(8)
        ; l1.display();
        cout<<"Inserting value after specific value:
```

```
        \n"; l1.insert_after(1,2);
        l1.display();
        cout<<"Deleting specific node:
        \n"; l1.deleteNode(7);
        l1.display();
        system("pause")
        ; return 0;
}
```

OUTPUT

```
Check List is empty or not...
List is empty
Inserting values in list :
7 10 8
Inserting values at the end of the list :
7 10 8 6 8
Inserting value after specific value:
7 10 8 6 8 2
Deleting specific node:
10 8 6 8 2
Press any key to continue . . .
```