Practical No.6

DOUBLLY LINKLIST.

```
#include<stdio.h>
#include<stdlib.h>
typedef struct node
{
    struct node*next;
    struct node*pre;
    int data;
}Node;
void display(Node *,Node *);
void insertAfter(Node **,Node**,int );
void insertBefore(Node **,Node**,int );
void insertEnd(Node**,int );
void DeletionAfter(Node**,Node**);
void DeletionBefore(Node**,Node**);
void Deletionnum(Node**,Node**);
int count;
int main()
{
     Node *node,*head=NULL,*tail=NULL,*ptr,*p;
    int co,count=0;
    printf("Enter Number of nodes to be created : ");
    scanf("%d",&co);
    while(co>0)
    {
        node = (Node*)malloc(sizeof(Node));
        printf("\nEnter the element : ");
        scanf("%d",&node->data);
        node->next=NULL;
        node->pre=NULL;
        if(head==NULL && tail==NULL)
        {
            head=node;
            tail=node;
            co--;
        }
        else
        {
            ptr=head,p=tail;
            while(ptr->next!=NULL && p->pre!=NULL)
            {
```

```
                ptr=ptr->next;
                p=p->pre;
            }
            ptr->next=node;
            node->pre=ptr;
            tail=node;
            co--;
        }
    }

    display(head,tail);
    int d,a;
    //Insertion After a particular location
    printf("\n\t\tInsertion After  a particular
number\n");
    printf("\nEnter after which number a new number to
be added : ");
    scanf("%d",&d);
    insertAfter(&head,&tail,d);
    printf("\n\tList after insertion of number After a
particular number \n");
    display(head,tail);
    //Insertion before a particular number
     printf("\n\t\tInsertion Before  a particular
number\n");
    printf("\nEnter Before which number a new number
to be added : ");
    scanf("%d",&a);
    insertBefore(&head,&tail,a);
    printf("\n\tList after insertion of number After a
particular number \n");
    display(head,tail);
    //Deletion After A particular number
    printf("\n\t\tDeletion of Element After a
Particular Number\n");
    DeletionAfter(&head,&tail);
    printf("\n\tList after Deletion\n");
    display(head,tail);
    //Deletion Before A particular number
```

```
    printf("\n\t\tDeletion of Element Before a
Particular Number\n");
    DeletionBefore(&head,&tail);
    printf("\n\tList after Deletion\n");
    display(head,tail);
    //Deletion of a Particular Number
     printf("\n\t\tDeletion of Element of Particular
Number\n");
    Deletionnum(&head,&tail);
    printf("\n\tList after Deletion of a Number \n");
    display(head,tail);

}
void insertAfter(Node **temp,Node**t,int b)
{
    Node *node=(Node *)malloc(sizeof(Node));
    printf("\n\t\tEnter a number to be inserted : ");
    scanf("%d",&node->data);
    node->next=NULL;
    Node*ptr;
    ptr=*temp;
    int flag=0;
    while(ptr->next!=NULL  &&  ptr->data!=b)
    {
        ptr=ptr->next;
    }
     if(ptr->data==b)
        flag=1;
    if(flag==0)
        printf("\n\t Number Not Found !");
     if(ptr->next==NULL)
     {
         ptr->next=node;
         node->pre=ptr;
         *t=node;
     }
    else
    {
        node->next=ptr->next;
        (ptr->next)->pre=node;
```

```
        node->pre=ptr;
        ptr->next=node;
    }
}
void insertBefore(Node **temp,Node**t,int b)
{
    Node *node=(Node *)malloc(sizeof(Node));
    printf("\n\t\tEnter a number to be inserted : ");
    scanf("%d",&node->data);
    node->next=NULL;
    Node*ptr;
    ptr=*t;
    int flag=0;
    while(ptr->pre!=NULL  &&  ptr->data!=b)
    {

        ptr=ptr->pre;
    }
    if(ptr->data==b)
        flag=1;
    if(flag==0)
        printf("\n\t Number Not Found !");
         if(ptr->pre==NULL)
        {
            node->next=ptr;
            ptr->pre=node;
            *temp=node;
        }
    else
    {

        node->next=ptr;
        (ptr->pre)->next=node;
        node->pre=ptr->pre;
        ptr->pre=node;
    }
}
void DeletionAfter(Node **temp,Node**t)
{
    int b;
```

```
    printf("\n\t\tEnter a number After which a Number
to be Deleted : ");
    scanf("%d",&b);
    Node*ptr,*p;
    ptr=*temp;
    int flag=0;
    while(ptr->next!=NULL && ptr->data!=b)
    {
        ptr=ptr->next;
    }
     if(ptr->data==b)
          {
               flag=1;
               p=ptr->next;
           }

    if(flag==0)
        printf("\n\t Number Not Found !");
    if(p->next==NULL)
        {
            *t=ptr;
            ptr->next=NULL;
            free(p);
        }
    else
    {
       ptr->next=p->next;
       (p->next)->pre=ptr;
       free(p);
    }
}
void DeletionBefore(Node **temp,Node**t)
{
    int b;
    printf("\n\t\tEnter a number Before which a number
to be Deleted  : ");
    scanf("%d",&b);
    Node*ptr,*p;
    ptr=*t;
    int flag=0;
```

```
    while(ptr->pre!=NULL && ptr->data!=b)
    {
        ptr=ptr->pre;
    }
     if(ptr->data==b)
          {
               flag=1;
               p=ptr->pre;
           }

    if(flag==0)
        printf("\n\t Number Not Found !");
   if(p->pre==NULL)
         {
             *t=ptr;
             ptr->pre=NULL;
             free(p);
         }
    else
    {
      ptr->pre=p->pre;
      (p->pre)->next=ptr;
      free(p);
    }
}
void Deletionnum(Node**head,Node**tail)
{
    int b,flag=0;
    printf("\n\t\tEnter a number to be Deleted  : ");
    scanf("%d",&b);
    Node *ptr=*head,*p=*tail;
    while(ptr->next!=NULL && ptr->data!=b)
        ptr=ptr->next;
    if(ptr->data==b)
        flag=1;
    if(ptr->pre==NULL)
    {
        *head=ptr->next;
        ptr->next=NULL;
        free(ptr);
```

```
    }
    if(ptr->next==NULL)
    {
        *tail=ptr->pre;
        ptr->pre=NULL;
        free(ptr);
    }
    else
       {
            (ptr->pre)->next=ptr->next;
            (ptr->next)->pre=ptr->pre;
       }

}
void display(Node *ptr,Node *p)
{
    count = 0;
    printf("\nLIST : ");
    while(ptr!=NULL)
    {
        printf("%d  ",ptr->data);
        count++;
        ptr=ptr->next;
    }
    printf("\nReversed List : ");
    while(p!=NULL)
    {
         printf("%d  ",p->data);
         p=p->pre;
    }
    printf("\nNumber of Nodes : %d\n",count);
}
```