Ajith.M.S
1BM19CS010

LAB-9

```
#include<stdio.h>
#include <stdlib.h>
struct node
 {
   int data;
   struct node * next ;
   struct node * prev;
 };
struct node * head=NULL;
void insert left()
{
 struct node* newnode;
newnode= (struct node*) malloc (sizeof(struct node));
printf ("Enter the item \n");
scanf ("%d", &new node->data);
 newnode->next=NULL;
 newnode->prev=NULL;
if (head==NULL)
  {
     head=newnode;
  }
else
 { newnode->next=head;
    head->prev=new-node;
     head=new-node;
 }  }
}
```

```
void insertright()
{
    struct node *newnode, *temp;
    newnode= (struct node*)malloc(sizeof(struct node));
    printf("Enter the item\n");
    scanf("%d", &newnode->data);
    newnode->next =NULL;
    newnode->prev =NULL;
    if(head= NULL)
    {
        head =newnode;
    }
    else
    {
        temp=head;
        while (temp->next !=NULL)
        temp= temp->next;
        temp->next=new_node;
        newnode->prev =temp;
    }
}


void delete()
{
    struct node*temp;
    int ele;
    if (head==NULL)
    {
        printf("Empty list\n");
        return;
    }
```

```
printf ("Enter the element to be deleted");
scanf ("%d ", &ele);
 temp = head;
 while (temp ->data != ele)
 {
   temp = temp ->next;
      if (temp == NULL)
   {
   printf ("Element is not in the list\n");
   return;
   }
if (temp == head)
 {
   head = head->next;
 }
else if (temp ->next == NULL)
 {
   temp = temp ->prev;
         temp-> next = NULL;
 }
else
 {
   temp -> prev -> next = temp ->next;
  temp -> next -> prev = temp ->prev;
  free (temp);
 }
```

```
void display()
{
    if (head==NULL)
    {
     printf("Empty list")
    }
    else
   {
     struct node *temp;
   temp =head;
     while (temp !=NULL)
   {
       printf("%d\t", temp->data);
        temp=temp->next;
         }
      printf("\n");
  }
       }


int main()
  {
    int choice;
  do {
       printf("1.Insert at left .2.Insert at left of speci
              node. 3.Insert at righ. 4.Delete specifi
                value. 5. Display. 6. Exit");
       printf("Enter your choice");
       switch (choice),
   {
   Case1: insert left(); break;
   Case2: Insert lefttop(); break
  Case3: Insertright(); break;
  Case4:  delete(); break;
  Case5:  display(); break;
  Case6: exit(0);
   }
}while(choice!=6);

}
```