ADITHI GURIMAJI
IBM19CS005

Lab-9

```
#include<stdio.h>
#include<stdlib.h>
Struct node
{ int data;
   Struct node *next;
   Struct node *prev;
}; Struct node *head=NULL;
void create()
{   Struct node *new_node, *temp;
   new_node=(Struct node *)malloc(sizeof(struct node));
   printf("Enter the item\n");
   scanf("%d", &new_node->data);
   newnode->next=NULL;
   new_node->prev=NULL;
    if(head==NULL)
    head=new_node;
    else
    { temp=head;
      while(temp->next!=NULL)
      temp=temp->next;
      temp->next=newnode;
       newnode->prev=temp;
}   }
void insert_left()
{   struct node *newnode, *temp1, *temp2=NULL;
    temp1=head;
    int ele, ele_bef;
    printf("Enter the element before which data has
     to be inserted\n");
    scanf("%d", &ele_bef);
    printf("Enter the element which has to be inserted\n");
```

```
scanf("%d", &ele);
if(temp1->data==ele_bef)
{  temp1->prev=newnode;
   newnode->next
newnode=(struct node*)malloc(sizeof(struct node));
newnode->data=ele;
if(temp1->data==ele_bef)
{ temp1=>prev=newnode;
   newnode->next=temp1;
   newnode->prev=NULL;
   head=newnode;
} while(temp1->next!=NULL)
  {   if(temp1->next->data==ele_bef)
      {   temp2=temp1->next;
          temp1->next=newnode;
          newnode->prev=temp1;
          newnode->next=temp2;
          temp2->prev=newnode;
          return;
      } else
        temp1=temp1->next;
} if(temp2==NULL)
   { printf("Element is not found in the list\n"); return;
   }}
void inser_right()
{ struct node *newnode, *temp1, *temp2=NULL;
   temp1=head;
   int ele, ele_aft;
   printf("Enter the element after which the data
   has to be inserted:");
   scanf("%d", &ele_aft);
   printf("Enter the element which has to be inserted\n");
```

```
scanf("%d", &ele);
newnode = (struct node*)malloc(sizeof(struct node));
newnode->data = ele;
while (temp1->next != NULL)
{   if (temp1 -> data == ele_aft)
    { temp2 = temp1->next;
      temp1 -> next = newnode;
      newnode -> prev = temp1;
      newnode -> next = temp2;
      temp2 -> prev = newnode;
        return;
    }
    else
    temp1 = temp1->next;
}  if(temp2==NULL)
  if (temp1-> data == ele_aft && temp1->next == NULL)
    { temp1->next=newnode;
      newnode -> prev = temp1;
      newnode->next = NULL; return;
    }
  if (temp2 == NULL)
  printf("Element not found in the list\n"); return;
}
void del()
{ int ele;
  struct node *temp, *del=NULL;
  if(head == NULL)
{ printf("Empty List Can't delete\n"); return; }
  printf("Enter the element to be deleted\n");
  scanf("%d", &ele);
  temp = head;
```

```
if (temp->dat== ele && temp->next == NULL)
{ head=NULL; return; }
if(temp->data ==ele)
{   head= head->next;
    head->prev= NULL;
    return;
}   while ( temp->next!=NULL)
    { if (temp->next->data == ele)
      { del= temp->next;
        if (del->next== NULL)
          temp->next= NULL;
        else {
          temp->next= del->next;
          del->next->prev= temp;
        } }
    else
        temp= temp->next;
    } if (del== NULL)
    { printf (" Element not found in the list\n");
      return; }
}
void display()
{   if(head == NULL)
    { printf ("Nothing to display\n");
      return; }
    struct node *temp;
    temp=head;
    while(temp!= NULL)
    {   printf("%d\t", temp->data);
        temp= temp->next;
    }  printf("\n");
}
```

```
int main()
{ int choice;
  while(1)
  { printf("\n");
    printf("1. Create\n");
    printf("2. Insert to the left of an element\n");
    printf("3. Insert to the right of an element\n");
    printf("4. Delete\n");
    printf("5. Display\n");
    printf("6. Exit\n");
    printf("Enter your choice\n");
    scanf("%d", &choice);
    switch(choice)
    { case 1: create(); break;
      case 2: insert_left(); break;
      case 3: insert_right(); break;
      case 4: del(); break;
      case 5: display(); break;
      case 6: exit(0); break;
      default: printf("Wrong choice\n");
}}}
```