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 *new_node;
  new_node=(struct node*) malloc(sizeof(struct node
  printf("Enter the data");
  scanf("%d",&new_node->data);
  new_node->next=NULL;
  new_node->prev=NULL;
  if(head==NULL)
    { head=new_node;
    }
  else
    { new_node->next=head;
      head->prev=new_node;
      head=new_node;
    }
}

void insert_right()
{ struct node *new_node,*temp;
  new_node=(struct node*) malloc(sizeof(
    struct node));
  printf("Enter the data\n");
  scanf("%d",&new_node->data);
  new_node->next=NULL;
  new_node->prev=NULL;
```

```
if(head =new_node;
}
else
{ temp = head;
  while (temp -> next != NULL)
    temp = temp -> next;
    temp -> next = new_node;
    new_node -> prev = temp;
  }}
void insert_leftnode()
{
  if(head == NULL)
    printf("List is empty");
  int ele;
  struct node * new_node * temp;
  printf("Enter the element after which you want to
  enter");
  scanf("%d", ele);
  new_node = (struct node*) malloc(sizeof(struct node));
  printf("Enter data");
  scanf("%d", &new_node -> data);
  new_node -> next = NULL;
  new_node -> prev = NULL;
  temp = head;
  if(temp -> data == ele)
  { new_node -> next = head;
    head -> prev = new_node;
    head = new_node;
  }
  else if(temp -> next == NULL)
  {
    printf("Element not found");
  }
```

```
else
{
  while (temp -> next -> data != ele)
  { temp = temp -> next;
    if (temp == NULL)
    { printf (Element not found);
    }}
  new_node -> next = temp -> next;
  temp -> next = new_node;
  new_node -> prev = temp;
  new_node -> next -> prev = new_node
}}
void delete()
{
  struct node * temp;
  int ele;
  if (head == NULL)
  { printf ("List is empty");
    return;
  }
  printf (" Enter the element to delete");
  scanf ("%d", &ele);
  temp = head;
  while (temp -> data != ele)
  { temp = temp -> next;
    if (temp == NULL)
    { printf (Not found);
    }
  }
  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("List is empty");
    }
    else
      { struct node* temp;
        temp = head;
        while(temp!=NULL)
          { printf("%d\t",temp->data);
            temp=temp->next;
          }
      }}
int main()
  { int choice;
    do {
        printf("1. Insert left\n2. Left of a specific node.\n3. Insert Right\n4. Delete a specific value.\n5. Display\n6. Exit");
        scanf("%d",&choice);
```

```
switch(choice)
  { case 1: insert_left(); break;
    case 2: insert_leftnode(); break;
    case 3: insert_right(); break;
    case 4: delete(); break;
    case 5: display(); break;
    case 6: exit(0);
  }
} while(choice != 0);
}
```