DEV KANSAL
1BM19CS046

```
#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 (size of (struct node));
  printf ("Enter the item");
  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 item\n");
  Scanf("%d",& new_node->data);
  new_node->next=NULL;
  new_node->prev=NULL;
  if(head==NULL)
    { head=new_node;
  }else
  { temp=head;
  while(temp->next=NULL)
    temp=temp->next
    temp->next=new_node;
    new_node->prev=temp;
  }}
  void del()
  { Struct node * temp;
    int el;
      if(head==NULL)
    { printf("Empty list\n");
        return;
      }
      Printf("Enter the element to be deleted\n");
      Scanf("%d",& el);
       temp=head;
```

```
while (temp-> data!=ele)
  { temp = temp-> next;
    if (temp==NULL)
    { printf("\n Element is not in the list\n");
      break;
    }}
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;
  }}
Void display()
{ struct node *ptr;
  Ptr= head;
  while(ptr!=NULL)
  {printf("\n %d\t", ptr-> data);}
      Ptr =ptr-> next;
  }printf("\n");
}
```

```
int main()
{ int choice;
  while(1)
  { printf("1. insert at the left \n2. Insert at the
        right \n3. Delete \n4. Display");
  printf("\n5. Exit\n");
  printf("Enter your choice\n");
  scanf("%d", &choice);
  switch(choice)
  { case 1: insert_lyt(); break;
      case 2: insert_right(); break;
      case 3: del(); break;
    case 4: display(); break;
    case 5: exit(0);
  }
}
}
```