EDGE

ANVITHA
1BM19CS021

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
  int data;
  struct node *next;
  struct node *prev;
};
struct node *head = NULL
void insertleft()
{
  struct node *new_node;
  new_node = (struct node *) malloc (sizeof(struct
                                          node
  printf("Enter the no:");
  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->next->prev = new_node;
    head = new_node;
  }
}
```

```
void del()
{
  struct node *temp;
  int elem;
  if (head == NULL)
  {
    printf("Empty list\n");
    return;
  }
  printf("Enter the elem to be deleted");
  scanf("%d", &elem);
  temp = head;
  while (temp->data != elem)
  {
    temp = temp->next;
    if (temp == NULL)
    {
      printf("Element is not list");
      return;
    }
  }
  if (temp == head)
  {
    head = head->next;
  }
  else if (temp->next == NULL)
  {
    temp = temp->next;
    temp->next = NULL;
  }
  else
  {
    temp->prev->next = temp->next;
    temp->next->prev = temp->prev;
```

```
void insert_between1()
{
  int listele;
  struct node * new_node, * temp;
  printf("Enter the element in the list");
  scanf("%d", &listele);
  new_node = (struct node*) malloc (sizeof (struct node));
  printf("Enter the new data");
  scanf("%d", &new_node -> data);
  new_node -> next = NULL;
  new_node -> prev = NULL;
  if (head == = NULL)
  {
   printf("Empty list"); return;
  }
  temp = head;
  while (temp -> data != listele)
  {
    temp = temp -> next;
    if (temp == = NULL)
    {
      printf("Element is not in the list");
      return;
    }
  }
  new_node -> prev = temp -> prev;
  temp -> prev = new_node;
  new_node -> next = temp;
  new_node -> prev = new_node;
}
```

```
void insert_between()
{
  int listele;
  struct node *new_node, *temp;
  printf("Enter the element in the list");
  scanf("%d", &listele);
  new_node = (struct node *) malloc (sizeof(
                                      node
  printf("Enter the new node data\n");
  scanf("%d", &new_node->data);
  new_node->next = NULL;
  new_node->prev = NULL;
  if (head == NULL)
  {
  printf("Empty list\n"); return;
  }
  temp = head;
  while (temp->data != listele)
  {
    temp = temp->next;
    if (temp == NULL)
    {
    printf("Element is not in list");
    return;
    }
  }
  new_node->next = temp->next;
  temp->next = new_node;
  new_node->prev = temp;
  new_node->next->prev = new_node;
}
```

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

int main()
{
  int choice;
  while(1)
  {
    printf("1. Insert at beg
            2. Insert a node before given node
            3. Insert a node after give node
            4. delete
            5. display
            6. Exit ");
    printf("Enter your choice: \n");
    scanf("%d", &choice);
    switch(choice)
    {
      case 1: insertleft(); break;
      case 2: insert_between1(); break;
      case 3: insert_between2(); break;
      case 4: del(); break;
      case 5: display(); break;
      case 6: exit(0);
    }
```