

## LAB PROGRAM-9.

```
#include <stdio.h>
#include <stdlib.h>
```

```
struct node
```

```
{
    int data;
    struct node * left;
    struct node * right;
};
```

```
struct node * head = NULL;
```

```
void insert_beg()
```

```
{
    struct node * new_node;
    new_node = (struct node) malloc (sizeof (struct node));
    printf ("Enter the number you want to insert.\n");
    scanf ("%d", &new_node->data);
```

```
new_node->next = NULL;
```

```
new_node->prev = NULL;
```

```
if (head == NULL)
```

```
{
    head = new_node;
```

```
printf ("Element inserted\n");
```

```
}
else
```

```
{
```

```
new_node->next = head;
```

```
head->prev = new_node;
```

```
head = new_node;
```

```
printf ("Element inserted\n");
```

```
}
```

```
void insert-left()
```

```
{  
    int listele;  
    struct node *new-node, *temp;  
    printf("Enter the element in the list : \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 \n");  
        return;  
    }
```

```
temp->prev = new-node->prev;  
temp->prev = new-node;  
new-node->next = temp;  
new-node->prev->next = new-node;
```

```
void insert-right()
```

```
{  
    int listele;  
    struct node *new-node, *temp;  
    printf("Enter the element in the list : \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\n");
        return;
    }
}

```

```

new-node->next = temp->next;
temp->next = new-node;
new-node->prev = temp;
new-node->next->prev = new-node;
}

```

```

void insert_end()
{

```

```

    struct node *new-node, *temp;

```

```

    new-node = (struct node*) malloc (sizeof (struct node));

```

```

    printf("Enter the number you want to insert:");

```

```

    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 ele;
```

```
if (head == NULL)  
{
```

```
printf("Empty list\n"); return;  
}
```

```
printf("Enter the element to be inserted:\n");
```

```
scanf("%d", &ele);
```

```
temp = head;
```

```
while (temp->data != ele)  
{
```

```
temp = temp->next;
```

```
if (temp == NULL)  
{
```

```
printf("Element not found in list\n");
```

```
return;
```

```
}
```

```
if (temp == head)  
{
```

```
head = head->next;
```

```
printf("Element Deleted\n");  
}
```

```
else if (temp->next == NULL)  
{
```

```
temp = temp->prev;
```



```

    temp->next = NULL;
    printf("Element deleted\n");
}

```

```

else
{

```

```

    temp->prev->next = temp->next;
    temp->next->prev = temp->prev;
    printf("Element is deleted");
}

```

```

void display()
{

```

```

    struct node *temp;
    temp = head;

```

```

    if (temp == NULL)
    {

```

```

        printf("Nothing to print");
        return;
    }

```

```

    {

```

```

        printf("Contents of list are:\n");

```

```

        while (temp != NULL)
        {

```

```

            printf("%d\t", temp->data);

```

```

            temp = temp->next;
        }

```

```

    }

```

```

    printf("\n");
}

```

```

int main ()
{

```

```

    {

```

```

        int choice;

```

```

        do
        {

```

```

printf("1. Insert at beginning\n");
printf("2. Insert at the left\n");
printf("3. Insert at the right\n");
printf("4. Insert at the end\n");
printf("5. Delete\n");
printf("6. Display\n");
printf("7. EXIT\n");
printf("Enter your choice : \n");
scanf("%d", &choice);
switch (choice)
{

```

```

    case 1: insert_beg(); break;

```

```

    case 2: insert_left(); break;

```

```

    case 3: insert_right(); break;

```

```

    case 4: insert_end(); break;

```

```

    case 5: del(); break;

```

```

    case 6: display(); break;

```

```

    case 7: printf("EXITING\n"); break;

```

```

    default: printf("Enter correct choice");
}

```

```

}while (choice != 7);

```

```

return 0;
}

```