LAB 9

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

struct node {
    int data;
    struct node *next;
    struct node * prev;
};
struct node * head = NULL;

void insert_left() {
    int listele;
    struct node *new_node, * temp;
    printf("Enter element in list: ");
    scanf("%d", &listele);
    new_node = (struct node*) malloc (sizeof(struct node));
    printf("Enter new node 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 not in list");
            return;
        }
```

```
    if (temp -> prev == NULL) {
        temp -> prev = new_node;
        new_node -> prev = NULL;
        new_node -> next = temp;
        head = new_node;
    }
    else {
        new_node -> prev = temp -> 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 element in list ");
    scanf("%d", &listele);
    new_node = (struct node*) malloc(sizeof(struct node));
    printf("Enter new node 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 not in list");
            return;
        }
    }
    if(temp->next == NULL) {
        temp->next = new_node;
        new_node->prev = temp;
    }
    else {
        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 element:");
    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");
        return;
    }
    printf("Enter element to delete: ");
    scanf("%d", &ele);
    temp = head;
    while(temp -> data != ele){
        temp = temp -> next;
        if(temp == NULL){
            printf("Element not in list");
            return;
        }
    }
    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 *temp;
    temp = head;
```

```
    if (head == NULL) {
        printf ("Empty list");
        return;
    }
    while (temp != NULL) {
        printf ("%d \t", temp->data);
        temp = temp -> next;
    }
}
int main() {
    int ch;
    do {
        printf ("1. Insert left \n 2. Insert right \n 3. Create \n 4. Delete \n 5.
            Display \n 6. Exit \n Enter choice : ");
        scanf ("%d", &ch);
        switch (ch) {
            case 1: insert_left(); break;
            case 2: insert_right(); break;
            case 3: insert_end(); break;
            case 4: del(); break;
            case 5: display(); break;
            case 6: exit(0);
        }
    } while (ch != 6);
}
```