Lab-9 1BM19CS024

Date ___/___/___
Page ___
Puffin

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *next;
struct node *prev;
};
struct node *head;
void ins_left()
{
struct node *newnode;
newnode=(struct node*)malloc(sizeof(struct node));
printf("\nEnter the data");
scanf("%d",&newnode->data);
newnode->prev=NULL;
newnode->next=NULL;
if(head==NULL)
{
head=newnode;
}
else
{
newnode->next=head;
head->prev=newnode;
head=newnode;
}}
void ins_end()
{
struct node *newnode,*temp;
```

```
newnode = (struct node *) malloc (sizeof (struct node));
printf("Enter the number: ");
scanf("%d", &newnode -> data);
newnode -> next = NULL;
newnode -> prev = NULL;
if (head == NULL)
{
head = newnode
}
else
{
temp = head;
while (temp -> next != NULL)
temp = temp -> next;
temp -> next = newnode;
newnode -> prev = temp;
}}
void ins_rear()
{
int ele;
struct node *newnode, *temp;
printf("Enter element of list: ");
scanf("%d", &ele);
newnode = (struct node *) malloc (sizeof (struct node));
printf("Enter the new data:");
scanf("%d", &newnode -> data);
newnode -> next = NULL;
newnode -> prev = NULL;
if (head == NULL);
{
```

```
printf("List is empty\n");
return;
}
temp = head;
while (temp->data != ele)
{
  temp = temp->next;
if (temp == NULL)
{
  printf("Element not in list\n");
return;
} }

newnode->next = temp->next
temp->next = newnode;
newnode->prev = temp;
newnode->next->prev = newnode;
}

void del()
{
struct node *temp;
int ele;
if (head == NULL)
{
printf("List is Empty\n");
return;
}
printf("Enter element to be deleted: ");
scanf("%d", &ele);
temp = head;
```

```
while (temp->data != ele)
{
  temp = temp->next;
if (temp == NULL)
{
printf("Element not in 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 *temp;
temp = head;
while (temp != NULL)
{
printf("%d\t", temp->data); }
temp = temp->next; } printf("\n"); }
```