Lab-8 1BM19CS024

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *next;
};
struct node *head=NULL;
void ins_rear1(int ele)
{
struct node *newnode,*temp;
newnode=(struct node*)malloc(sizeof(struct node));
newnode->data=ele;
newnode->next=NULL;
if(head==NULL)
{
head=newnode;
}
else
{
temp=(struct node*)malloc(sizeof(struct node));
temp=head;
while(temp->next!=NULL)
{
temp=temp->next;
}
temp->next=newnode;
}
}
```

```
void del-front()
{
struct node *temp;
temp=(struct node *) malloc(sizeof(struct node));
if(head==NULL)
{
printf("List is Empty\n");
}
else
{
temp=head;
head=temp->~~hea~~ next;
printf("%d is removed\n", temp->data);
free(temp);
}
}
void del-rear()
{
struct node *temp, *run;
temp=(struct node *)malloc(sizeof(struct node));
run=(struct node *)malloc(sizeof(struct node));
temp=head->next
if(run==NULL)
{
printf("List is Empty\n");
}
else
{
while(temp->next!=NULL)
```

```
{
rem = temp;
temp = temp -> next;
}
printf ("%d is deleted \n", temp -> data);
rem -> next = NULL;
free(temp);
}
}
void del(int ele)
{
struct node *temp, *del;
temp = (struct node*) malloc(sizeof(struct node));
del = (struct node*) malloc (sizeof(struct node));
del = NULL;
if( head -> data = = ele)
{
del = head;
head = head -> next;
del -> next = NULL;
}
else
{
temp = head;
while (temp ! = NULL)
{
if (temp -> next -> data = = ele)
{
del = temp -> next;
```

Date......./......./.......
Page.......
Puffin

```
temp->next=del->next;
del->next=NULL;
break;
}
else
{
temp=temp->next;
}
}
}
if(del==NULL)
{
printf("\nElement not found\n");
}
}
void sort(struct node *h);
{
int j,i,a;
struct node *temp1;
struct node *temp2;
for(temp1=h;temp1!=NULL;temp1=temp1->next)
{
for(temp2=temp1;temp2!=NULL;temp2=temp2->next)
{
if(temp2->data<temp1->data)
{
a=temp1->data;
temp1->data=temp2->data;
temp2->data=a;
}}}}
```

```
void rev(struct node *h)
{
int i,j,a;
struct node *temp1, *temp2;
for(temp1=h; temp1!=NULL; temp1=temp1->next)
{
for(tem2=temp1; temp2!=NULL; temp2=temp2->next)
{
a=temp1->data;
temp1->data=temp2->data;
temp2->data=a;
}}}
void conc
{
struct node *temp, *run;
temp=(struct node *)malloc(sizeof(struct node));
run=(struct node *)malloc(sizeof(struct node));
run=head;
if(head1==NULL && head2==NULL)
{
printf("Both lists empty\n");
}
else
 {
temp=head1;
while(temp->next!=NULL)
{
temp=temp->next;
}
```

Date ..../..../....
Page ....
Puffin

```
temp->next=head2;
}
printf("\n The concatenated list - \n");
while(run!=NULL)
{
printf("%d \n", run->data);
run=run->next;
}}
```

Scanned with CamScanner