8. **Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, PhNo**

a. Create a DLL of N Employees Data by using end insertion.
b. Display the status of DLL and count the number of nodes in it
c. Perform Insertion and Deletion at End of DLL
d. Perform Insertion and Deletion at Front of DLL
e. Demonstrate how this DLL can be used as Double Ended Queue.
f. Exit

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
char name[20],ssn[20],dept[20],desig[20],phno[12];
float sal;
struct node *llink,*rlink;
};
typedef struct node *dlist;
/****************************/
dlist getnode()
{
 dlist x;
x=(dlist)malloc(sizeof(struct node));
if(x==NULL)
{
 printf("not enough memory\n");
exit(0);
}
return x;
}
/***************************/

dlist insertf(dlist first)
{
dlist temp;
temp=getnode();
temp->llink=temp->rlink=NULL;
printf("ENTER SSN NAME DEPT DESIGNATION SALARY PHNO\n");
scanf("%s%s%s%s%f%s",temp->ssn,temp->name,temp->dept,temp->desig,&temp-
>sal,temp->phno);
if(first==NULL)
{
 return temp;
}
temp->rlink=first;
first->llink=temp;
first=temp;
return first;
}
/**************************/




dlist insertr(dlist first)
{
 dlist temp,cur;
 temp=getnode();
temp->rlink=temp->llink=NULL;
printf("ENTER SSN NAME DEPT DESIGNATION SALARY PHNO\n");
scanf("%s%s%s%s%f%s",temp->ssn,temp->name,temp->dept,temp->desig,&temp-
>sal,temp->phno);
if(first==NULL)
{
 return temp;
}
cur=first;
while(cur->rlink!=NULL)
cur=cur->rlink;
cur->rlink=temp;
temp->llink=cur;
return first;
}
/************************/

display(dlist first)
{
dlist cur;
int count=0;
if(first==NULL)
{
 printf("list is empty\n");
 return;
}
cur=first;
printf("sSN\tNAME\tDEPT\tDESIGNATION\tSALARY\tPHONE NUMBER\n");
while(cur!=NULL)
{
  printf("%s\t%s\t%s\t%s%f\t%s\n",cur->ssn,cur->name,cur->dept,cur-
>desig,cur->sal,cur->phno);
 cur=cur->rlink;
count++;
}
printf("\n NUMBER OF NODES=%d\n",count);
}
/****************************/
dlist deleter(dlist first)
{
 dlist cur,prev;
 if(first==NULL)
{
 printf("Double linked list is empty\n");
 return first;
}
cur=first;
prev=NULL;
if(first->rlink==NULL)
{
 printf("DELETED NODE\n%s\t%s\t%s\t%s\t%f\t%s\n",cur->ssn,cur->name,cur-
>dept,cur->desig,cur->sal,cur->phno);
 free(cur);
 first=NULL;
 return first;
}
```

```
while(cur->rlink!=NULL)
{
 prev=cur;
 cur=cur->rlink;
}
 printf("DELETED NODE\n %s\t%s\t%s\t%s\t%f\t%s\n",cur->ssn,cur->name,cur-
>dept,cur->desig,cur->sal,cur->phno);
 prev->rlink=NULL;
 free(cur);
 return first;
}
/***************************/
dlist deletef(dlist first)
{
dlist cur,prev;
 if(first==NULL)
{
 printf("list is empty\n");
 return first;
}
cur=first;
printf("DELETED NODE\n %s\t%s\t%s\t%s\t%f\t%s\n",cur->ssn,cur->name,cur-
>dept,cur->desig,cur->sal,cur->phno);
first=first->rlink;
free(cur);
return first;
}
/**************************/
main()
{
dlist first;
 int n,i,c;
 first=NULL;
  for(;;)
{
 printf("\n1--->INSERT REAR\n2--->INSERT FRONT\n3--->DELETE REAR\n4--
>DELETE FRONT\n5--->DISPLAY\n");
 printf("enter your choice\n");
 scanf("%d",&c);
 switch(c)
{
 case 1: printf("ENTER NUMBER OF EMPLOYEES\n");
       scanf("%d",&n);
      for(i=0;i<n;i++)
       first=insertr(first);
         break;
 case 2: first=insertf(first);
       break;
 case 3: first=deleter(first);
       break;
 case 4: first=deletef(first);
       break;
 case 5: display(first);
       break;
 default: exit(0);
}
}
}
```