**8. Develop 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 ssn[25],name[25],dept[10],designation[25];  
         int sal;  
         long int phone;  
         struct node \*llink;  
         struct node \*rlink;  
};  
typedef struct node\* NODE;  
  
NODE first = NULL;  
int count=0;  
  
  
NODE create()  
{  
          NODE enode;  
          enode = (NODE)malloc(sizeof(struct node));  
          if( enode== NULL)  
          {  
                    printf("\nRunning out of memory");  
                    exit(0);  
           }  
           printf("\nEnter the ssn,Name,Department,Designation,Salary,PhoneNo of the employee: \n");  
         scanf("%s %s %s %s %d %ld", enode->ssn, enode->name, enode->dept, enode->designation, &enode->sal, &enode->phone);  
           enode->llink=NULL;  
           enode->rlink=NULL;  
count++;  
           return enode;  
}  
  
NODE insertfront()  
{  
         NODE temp;  
         temp = create();  
          if(first == NULL)  
         {  
                 return temp;  
          }  
          temp->rlink = first;  
          first->llink = temp;  
          return temp;  
}  
  
void display()  
{  
          NODE cur;  
          int nodeno=1;  
          cur = first;  
          if(cur == NULL)  
                      printf("\nNo Contents to display in DLL");  
          while(cur!=NULL)  
         {  
        printf("\nENode:%d||SSN:%s|Name:%s|Department:%s|Designation:%s|Salary:%d|Phone no:%ld", nodeno, cur->ssn, cur->name,cur->dept, cur->designation, cur->sal, cur->phone);  
                   cur = cur->rlink;  
                    nodeno++;  
         }  
          printf("\nNo of employee nodes is %d",count);  
}  
  
NODE deletefront()  
{  
         NODE temp;  
         if(first == NULL)  
         {  
                   printf("\nDoubly Linked List is empty");  
return NULL;  
         }  
         if(first->rlink== NULL)  
        {  
                  printf("\nThe employee node with the ssn:%s is deleted", first->ssn);  
                 free(first);  
                 count--;  
                  return NULL;  
         }  
         temp = first;  
         first = first->rlink;  
         temp->rlink = NULL;  
         first->llink = NULL;  
         printf("\nThe employee node with the ssn:%s is deleted",temp->ssn);  
         free(temp);  
         count--;  
         return first;  
}  
  
NODE insertend()  
{  
          NODE cur, temp;  
          temp = create();  
  
          if(first == NULL)  
          {  
                    return temp;  
          }  
         cur= first;  
         while(cur->rlink!=NULL)  
        {  
                  cur = cur->rlink;  
         }  
  
         cur->rlink = temp;  
         temp->llink = cur;  
          return first;  
}  
  
NODE deleteend()  
{  
         NODE prev,cur;  
         if(first == NULL)  
        {  
                 printf("\nDoubly Linked List is empty");  
                 return NULL;  
        }  
  
        if(first->rlink == NULL)  
        {  
                   printf("\nThe employee node with the ssn:%s is deleted",first->ssn);  
                   free(first);  
                   count--;  
                   return NULL;  
        }  
  
         prev=NULL;  
         cur=first;  
  
        while(cur->rlink!=NULL)  
        {  
                  prev=cur;  
                  cur = cur->rlink;  
         }  
  
         cur->llink = NULL;  
         printf("\nThe employee node with the ssn:%s is deleted",cur->ssn);  
         free(cur);  
         prev->rlink = NULL;  
         count--;  
         return first;  
}  
  
void deqdemo()  
{  
       int ch;  
       while(1)  
      {  
              printf("\nDemo Double Ended Queue Operation");  
      printf("\n1:InsertQueueFront\n 2: DeleteQueueFront\n 3:InsertQueueRear\n4:DeleteQueueRear\n 5:DisplayStatus\n 6: Exit \n");  
              scanf("%d", &ch);  
  
              switch(ch)  
             {  
                     case 1: first=insertfront();  
                                 break;  
                    case 2: first=deletefront();  
                               break;  
                   case 3: first=insertend();  
                               break;  
                   case 4: first=deleteend();  
                              break;  
                    case 5: display();  
                               break;  
                   default : return;  
           }  
     }  
}  
  
void main()  
{  
    int ch,i,n;  
    while(1)  
    {  
        printf("\n\n~~~Menu~~~");  
        printf("\n1:Create DLL of Employee Nodes");  
        printf("\n2:DisplayStatus");  
         printf("\n3:InsertAtEnd");  
         printf("\n4:DeleteAtEnd");  
         printf("\n5:InsertAtFront");  
         printf("\n6:DeleteAtFront");  
         printf("\n7:Double Ended Queue Demo using DLL");  
         printf("\n8:Exit \n");  
         printf("\nPlease enter your choice: ");  
        scanf("%d",&ch);  
  
        switch(ch)  
        {  
         case 1 : printf("\nEnter the no of Employees:   ");  
                   scanf("%d",&n);  
                   for(i=1;i<=n;i++)  
                   first = insertend();  
                   break;  
  
         case 2:  display();  
                   break;  
  
          case 3: first = insertend();  
                  break;  
  
          case 4: first = deleteend();  
                  break;  
  
          case 5: first = insertfront();  
                  break;  
  
          case 6: first = deletefront();  
                break;  
  
          case 7: deqdemo();  
                  break;  
  
          case 8 : exit(0);  
        default: printf("\nPlease Enter the valid choice");  
        }  
    }  
}