```
/*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>
#include<string.h>

struct NODE             /* NODE structure for Student Data*/
{
    long long phno;
    float sal;
    char name[20],dept[10],ssn[10],desgn[10];
    struct NODE *next;  /* Node Link to next Employee Data*/
    struct NODE *prev;
};

typedef struct NODE *NODEPTR; /* Node Pointer for NODE structure*/
NODEPTR first = NULL;         /* first Pointer for LIST*/
int count=0;


NODEPTR create_node()
{
    NODEPTR node = (NODEPTR)malloc(sizeof(struct NODE)); /* Create NODE
dynamically*/
    printf("Enter the Employee Ssn\n");
    scanf("%s",node->ssn);
    printf("Enter the Employee Name\n");
    scanf("%s",node->name);
    printf("Enter the Employee Department\n");
    scanf("%s",node->dept);
    printf("Enter the Employee Designation\n");
    scanf("%s",node->desgn);
    printf("Enter the Employee Salary.\n");
    scanf("%f",&node->sal);
    printf("Enter the Employee Phone No.\n");
    scanf("%lld",&node->phno);
    node->next = NULL;  /* Initially NODE next link is set to NULL*/
    node->prev = NULL;  /* NODE prev link is set to NULL*/
    count++;            /* Increment count when NODE is created*/
    return node;        /* return NODE Pointer */
}

void insert_front() /* Function to insert NODE to front */
{
    NODEPTR temp = create_node();
    temp->next = first;
    first=temp;
}
```

```
void delete_front()
{
    NODEPTR temp;
    temp = first;
     if(first->next == NULL)
    {
        first = NULL;
        free(temp);
        count--;
        printf("End node deleted successfully\n");
    }
    else
    if(temp != NULL)
    {
        first = temp->next;
        temp->next = NULL;
        temp->prev =NULL;
        first->prev = NULL;
        free(temp);
        count--;
        printf("Front node deleted successfully\n");
    }
    else
        printf("ALERT!!!:List is Empty\n");
}
void insert_end()
{
    NODEPTR last;
    NODEPTR temp = create_node();
    last = first;
    if(first == NULL)
    {
        first = temp;
    }
    else
    {
        while(last->next != NULL)
        {
            last = last->next;
        }
        last->next = temp;
        temp->prev = last;
    }
}

void delete_end()
{
    NODEPTR last;
    last = first;
     if(first->next == NULL)
    {
        first = NULL;
        free(last);
        count--;
        printf("End node deleted successfully\n");
    }
    else
    if(last != NULL)
```

```
    {
        while(last->next != NULL)
        {
           last = last->next;
        }
        last->next = NULL;
        if(last->prev != NULL)
            (last->prev)->next = NULL;
        free(last);
        count--;
        printf("End node deleted successfully\n");
    }
    else
        printf("ALERT!!!:List is Empty\n");
}

void display()
{
    NODEPTR temp;
    temp = first;
    if(temp == NULL)
    {
        printf("List is Empty\n");
    }
    else
    {
        printf("The List values are ....\n");
         printf("[SSN, Name, Dept, Desgn, Salary, Phone]\n");
        while(temp!= NULL)
        {
            printf("[%s,%s,%s,%s,%.2f,%lld]-->",temp->ssn,temp-
>name,temp->dept,temp->desgn,temp->sal,temp->phno);
            temp = temp->next;
        }
        printf("\nNODE COUNT = %d\n",count);
    }

}

int main()
{
      int  n, ch, i;
      while (1)
      {
            printf("\n**********Doubly Linked List Operations
Menu**********\n");
            printf("1. Create a DLL of N Students Data by using front
insertion\n");
            printf("2. Display the status of DLL\n");
            printf("3. Insertion / Deletion at End of DLL\n");
            printf("4. Insertion / Deletion at Front of DLL\n");
            printf("5. Exit\n");
            printf("Enter your choice:\n");
            scanf("%d", &ch);
            switch (ch)
            {
            case 1: printf("Enter the value of N to create DLL\n");
                 scanf("%d", &n);
```

```
                for(i=1;i<=n;i++)
                {
                    printf("Enter a %d node to insert towards front of
DLL\n",i);
                    insert_end();
                }
                break;
            case 2: display();
                       break;
            case 3: printf("Press 1 to Insert End or 2 to Delete End\n");
                scanf("%d", &ch);
                if(ch == 1)
                {
                    printf("Enter a node to insert towards end of
DLL\n");
                    insert_end();
                }
                else if(ch == 2)
                {
                    delete_end();
                }
                else
                    printf("Invalid Entry\n");
                break;
            case 4: printf("Press 1 to Insert Front or 2 to Delete
Front\n");
                scanf("%d", &ch);
                if(ch == 1)
                {
                    printf("Enter a node to insert to Front of DLL\n");
                    insert_front();
                }
                else if(ch == 2)
                {
                    delete_front();
                }
                else
                    printf("Invalid Entry\n");
                break;
        case 5: exit(0);
            default: printf("Enter the valid choice\n\n");
                  break;
            }
      }
      return 0;
}
```