```c
/*
Q8. 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.
*/

/*
We are creating 8 functions:
1. main() function.
2. getnode() function
3. read() function
4. CreateDLL() function.
5. displaycount() function
6. Insertionfront() function
7. Deletionfront() function
8. Deletionend() function
*/

#include<stdio.h>
#include<stdlib.h>

struct node                  //structure to store employee details
{
    char ssn[10],name[10],dept[15],desig[10];
    int phno, sal;
    struct node *next;
    struct node *prev;
};

typedef struct node *NODE;         // Renaming struct node as NODE

NODE temp;
NODE FIRST=NULL;
NODE END=NULL;

void main()
{
    int ch;
    while(1)
        {
            printf("1 - Create DLL of N Employees\n");
            printf("2 - Display DLL\n");
            printf("3 - Insertion at front\n");
            printf("4 - Insertion at end\n");
            printf("5 - Deletion at front\n");
            printf("6 - Deletion at end\n");
            printf("7 - Exit\n");

            printf("Enter Your Choice: ");
            scanf("%d",&ch);

            switch(ch)
            {
                case 1:CreateDLL();
                       break;

                case 2:displaycount();
                       break;

                case 3:Insertionfront();
```

```
                        break;

                case 4:Insertionend();
                        break;

                case 5:Deletionfront();
                        break;

                case 6:Deletionend();
                        break;

                case 7:return;
                default: printf("Invalid Choice\n");

            }
        }
}//end of main

/*
Creating a node x with both left and right links.
Initialize them with NULL values.
A single node will not have a address of preceding element or the next element.
Hence both the values are initialised to NULL.
*/

NODE getnode()
{
    NODE x;
    x=(NODE)malloc(sizeof(struct node));
    x->next=NULL; //next node address
    x->prev=NULL; //previous node address.
    return x;
}

/*
Based on the number of employees, we create that many nodes.
For each node, we need to put in all the employee details.
All this is done using read() function.
temp is used generate nodes.
*/

void read()                         // read details of employee
{
    temp=getnode();

    printf("Enter SSN:");
    scanf("%s",temp->ssn);
    printf("Enter Name:");
    scanf("%s",temp->name);
    printf("Enter Dept:");
    scanf("%s",temp->dept);
    printf("Enter Designation:");
    scanf("%s",temp->desig);
    printf("Enter Phno:");
    scanf("%d",&temp->phno);
    printf("Enter Salary:");
    scanf("%d",&temp->sal);

    return;
}

/*
Creating a DLL of 'n' Employees by using endinsertion.
First we check whether the list empty or not.
If its empty, then the new node that we inserted will be first node.
Otherwise we find the last node and insert the new node after that.
```

```c
*/

void CreateDLL()
{
    int n;
    int i=1;
    printf("Enter the number of employees\n");
    scanf("%d",&n);
    while(i<=n)
        {
            printf("Enter the details of number %d employee\n", i++);
            read();
            if(FIRST==NULL)
                {
                    FIRST=temp;
                    END=temp;
                }
            else
                {
                    END->prev=temp;
                    temp->next=END;
                    END=temp;
                }
        }                           // end of while statement
}                                  //  end of create() function

/*
Display the status of DLL and count the number of nodes in it.
First we check wheter the list is empty or not.
If empty, we say 'No employee data'.
Otherwie we display all nodes in the list.
*/
void displaycount()
{
    temp=FIRST;
    int count=0;

    if(FIRST==NULL) //
        {
            printf("No employee data\n");
        }
    else
        {
            while(temp!=NULL) //
            {
                count++;
                printf("Employee details:\n");
                printf("%s\t%s\t%s\t%s\t%d\t%d\n",
                       temp->ssn, temp->name, temp->dept,
                       temp->desig, temp->phno, temp->sal);

                temp=temp->prev;
            }

            printf("Employee count is %d\n",count);

        }                           // end of else statement.
return;
}                                  // end of display() function.

/*
Performing Insertion at front of DLL.
First we check for empty list.
If empty we set new node as first node.
Otherwise if list is already present, we insert the node at front of list.
*/
```

```c

void Insertionfront()
{
    printf("Enter the details of the employee\n");
    read();

    if(FIRST==NULL)
        FIRST=temp;

    else                          //Inserting at front of the list.
        {
            temp->prev=FIRST;
            FIRST->next=temp;
            FIRST=temp;
        }
}                               // end of insertionfront() function.

/*
Performing Insertion at the end of DLL.
First we check for empty list.
If empty, newnode itself will be first and last node.
Otherwise if list already present,we find the last node & insert after that.
*/

void Insertionend() //Perform Insertion at End of DLL
{
    printf("Enter the details of the new employee\n");
    read();
    if(FIRST==NULL)           // check for empty list
        {
            FIRST=temp;
            END=temp;
        }
    else        // otherwise find the last node and insert the new node
        {
            END->prev=temp;
            temp->next=END;
            END=temp;
        }
    return ;
}                                // end of insertionend() function.

/*
Deleting the node from the front of the DLL.
First we check if the list is empty.
If not we check, if the list has only one node.
If only one node, we delete that node & initialise list to NULL.
Deletion of the list is done based on unique number ssn.
Otherwise we go delete first node from the list.
*/
void Deletionfront()
{
    temp = FIRST;
    if(FIRST == NULL)                       // check for empty list
        printf("List is empty\n");

    else if(FIRST == END)              // check for single node in list
        {
            printf("deleted employee is %s\n", temp->ssn);
            FIRST = NULL;
            END = NULL;
            free(temp);
        }
        else                  // otherwise delete node from front of DLL
            {
                printf("deleted employee is %s\n", temp->ssn);
```

```
                FIRST = FIRST->prev;
                FIRST->next = NULL;
                free(temp);
            }

        return;
}                                   // end of deletefront() function.
/*
Deleting the node from the end of the DLL.
First we check if the list is empty.
If not, we check if the list has only one node.
If only one node, we delete that node & initialise list to NULL.
Deletion of the list is done based on unique number ssn.
Otherwise we go delete last node from the list.
*/
void Deletionend()
{
    temp = END;
    if(FIRST==NULL)                     // check for empty list
        printf("List is empty\n");

    else if(FIRST==END)                 // check for single node in list
        {
            printf("deleted employee is %s\n", temp->ssn);
            FIRST=NULL;
            END=NULL;
            free(temp);
        }
        else                        // otherwise delete end node from DLL
            {
                printf("deleted employee is %s\n", temp->ssn);
                END = END->next;
                END->prev = NULL;
                free(temp);
            }
        return ;
}                               // end of deleteend() function.
```

OUTPUT:

```
1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 2
No employee data

1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 5
List is empty

1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 6
List is empty

1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 1
Enter the number of employees
3
Enter the details of employee 1
Enter SSN:111
Enter Name:ABC
Enter Dept:ISE
Enter Designation:PROF
Enter Phno:9870
Enter Salary:5000
Enter the details of employee 2
```

```
Enter SSN:222
Enter Name:XYZ
Enter Dept:CSE
Enter Designation:PRIN
Enter Phno:3456
Enter Salary:6000
Enter the details of employee 3
Enter SSN:333
Enter Name:PQR
Enter Dept:ECE
Enter Designation:HODD
Enter Phno:6750
Enter Salary:4000

1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 2
Employee details:
111     ABC     ISE     PROF    9870    5000
Employee details:
222     XYZ     CSE     PRIN    3456    6000
Employee details:
333     PQR     ECE     HODD    6750    4000
Employee count is 3

1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 3
Enter the details of the employee
Enter SSN:444
Enter Name:STR
Enter Dept:MEE
Enter Designation:ATTD
Enter Phno:7658
Enter Salary:3000

1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
```

```
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 2
Employee details:
444     STR     MEE     ATTD    7658    3000
Employee details:
111     ABC     ISE     PROF    9870    5000
Employee details:
222     XYZ     CSE     PRIN    3456    6000
Employee details:
333     PQR     ECE     HODD    6750    4000
Employee count is 4

1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 4
Enter the details of the new employee
Enter SSN:555
Enter Name:KLM
Enter Dept:CVV
Enter Designation:INST
Enter Phno:5432
Enter Salary:2000

1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 2
Employee details:
444     STR     MEE     ATTD    7658    3000
Employee details:
111     ABC     ISE     PROF    9870    5000
Employee details:
222     XYZ     CSE     PRIN    3456    6000
Employee details:
333     PQR     ECE     HODD    6750    4000
Employee details:
555     KLM     CVV     INST    5432    2000
Employee count is 5
```

```
1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 5
deleted employee is 444

1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 2
Employee details:
111     ABC     ISE     PROF    9870    5000
Employee details:
222     XYZ     CSE     PRIN    3456    6000
Employee details:
333     PQR     ECE     HODD    6750    4000
Employee details:
555     KLM     CVV     INST    5432    2000
Employee count is 4

1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 6
deleted employee is 555

1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 2
Employee details:
111     ABC     ISE     PROF    9870    5000
Employee details:
222     XYZ     CSE     PRIN    3456    6000
```

```
Employee details:
333     PQR     ECE     HODD    6750    4000
Employee count is 3

1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 2
Employee details:
111     ABC     ISE     PROF    9870    5000
Employee details:
222     XYZ     CSE     PRIN    3456    6000
Employee details:
333     PQR     ECE     HODD    6750    4000
Employee count is 3

1 - Create DLL of N Employees
2 - Display DLL
3 - Insertion at front
4 - Insertion at end
5 - Deletion at front
6 - Deletion at end
7 - Exit
Enter Your Choice: 7
```