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

typedef struct
{
    char ssn[10];
    char name[10];
    char dept[10];
    char desg[10];
    char ph[10];
    float sal;
} EMPLOYEE;

struct node
{
    char ssn[10];
    char name[10];
    char dept[10];
    char desg[10];
    char ph[10];
    float sal;
```

```c
    struct node *llink;
    struct node *rlink;
};

typedef struct node *NODE;

NODE getnode()
{
    NODE x;

    x = (NODE) malloc(sizeof(struct node));

    if (x == NULL)
    {
        printf("Not Enough Memory\n");
        exit(0);
    }

    return x;
}

void display(NODE head)
{
    NODE cur;

    if (head->rlink == head)
    {
        printf("List is Empty\n");
        return;
    }

    printf("SSN      NAME           DEPT   DESGIN
    PHONE         SAL\n");

    cur = head->rlink;
    while (cur != head)
    {
        printf("%-10s%-10s%-10s%-10s%-14s %.2f\n",
        cur->ssn, cur->name, cur->dept, cur->desg,
        cur->ph, cur->sal);
        cur = cur->rlink;
    }
}

int count_node(NODE head)
{
    NODE cur;
    int count;

    if (head->rlink == head) return 0;

    cur = head->rlink;

    count = 0;
    while (cur != head)
    {
        count++;
        cur = cur->rlink;
    }

    return count;
}

NODE insert_rear(EMPLOYEE emp, NODE head)
{
    NODE last, temp;

    temp = getnode();

    strcpy(temp->ssn, emp.ssn);
    strcpy(temp->name, emp.name);
    strcpy(temp->dept, emp.dept);
    strcpy(temp->desg, emp.desg);
    strcpy(temp->ph, emp.ph);

    temp->sal = emp.sal;

// Get the address of the last node
    last = head->llink;

// Insert the node at the end
    last->rlink = temp;
    temp->llink = last;
    temp->rlink = head;
    head->llink = temp;

    return head;
}

NODE insert_front(EMPLOYEE emp, NODE head)
{
    NODE temp, first;

    temp = getnode();

    strcpy(temp->ssn, emp.ssn);
    strcpy(temp->name, emp.name);
    strcpy(temp->dept, emp.dept);
    strcpy(temp->desg, emp.desg);
    strcpy(temp->ph, emp.ph);
    temp->sal = emp.sal;

// Get the address of the first node
    first = head->rlink;

// Insert the node between head and first node
    temp->rlink = first;
    first->llink = temp;
    head->rlink = temp;
    temp->llink = head;
    return head;
}

NODE delete_rear(NODE head)
{
    NODE last, prev;

// Check for empty list
    if (head->rlink == head)
    {
        printf("List is Empty\n");
        return head;
    }

// Obtain the address of the last node
    last = head->llink;

// Obtain address of the last but one node
    prev = last->llink;

// Make last but one node as the last node
    prev->rlink = head;
    head->llink = prev;

    printf("Item deleted = %s\n", last->ssn);
    free(last);

    return head;
}

NODE delete_front(NODE head)
{
    NODE first, second;
```

```c
//Check for empty list */
    if (head->rlink == head)
    {
        printf("List is Empty\n");
        return head;
    }

    first = head->rlink; // Get the first node
    second = first->rlink; // Get the second node

//Make second node as the first node
    head->rlink = second;
    second->llink = head;

    printf("Item deleted = %s\n",first->ssn);
    free(first);

    return head;
}

void read_employee_details(EMPLOYEE *emp)
{
    printf("SSN : ");
    scanf("%s",emp->ssn);

    printf("Name : ");
    scanf("%s",emp->name);

    printf("Deptment : ");
    scanf("%s",emp->dept);

    printf("Designation : ");
    scanf("%s",emp->desg);

    printf("Phone : ");
    scanf("%s",emp->ph);
    printf("Salary : ");
    scanf("%f",&emp->sal);
}

int main()
{
    int choice, count;
    NODE head;
    EMPLOYEE emp;

    head = getnode();

    head->rlink = head->llink = head;

    for (;;)
    {
        printf("\n\n1:Insert Front 2:Insert Rear\n"
        );
        printf("3:Delete Front 4:Delete Rear\n");
        printf("5:Display 6:Count 7: Exit : ");

        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            read_employee_details(&emp);
            head = insert_front(emp, head);
            break;

        case 2:
            read_employee_details(&emp);
            head = insert_rear(emp, head);
            break;
        case 3:
            head = delete_front(head);
            break;

        case 4:
            head = delete_rear(head);
            break;

        case 5:
            display(head);
            break;

        case 6:
            count  =  count_node(head);
            printf("Number of nodes = %d\n", count);
            break;

        default:
            exit(0);
        }
    }
}

/*

1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 3
List is Empty


1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 4
List is Empty
```

```
1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 5
List is Empty


1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 6
Number of nodes = 0

1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 2
SSN : 001
Name : Jayaprada
Deptment : CSE
Designation : AsstProf
Phone : 1234567890
Salary : 40000


1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 5
SSN  NAME       DEPT DESGIN    PHONE       SAL
001  Jayaprada  CSE  AsstProf  1234567890  40000.00


1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 1
SSN : 002
Name : Jayaprada
Deptment : CSE
Designation : AsstProf
Phone : 1234567890
Salary : 45000
```

```
1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 5
SSN  NAME       DEPT DESGIN    PHONE       SAL
002  Jayaprada  CSE  AsstProf  1234567890  45000.00
001  Jayaprada  CSE  AsstProf  1234567890  40000.00

1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 2
SSN : 003
Name : Jayaprada
Deptment : CSE
Designation : AsstProf
Phone : 1234567890
Salary : 43000


1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 5
SSN  NAME       DEPT DESGIN    PHONE       SAL
002  Jayaprada  CSE  AsstProf  1234567890  45000.00
001  Jayaprada  CSE  AsstProf  1234567890  40000.00
003  Jayaprada  CSE  AsstProf  1234567890  43000.00


1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 1
SSN : 004
Name : Jayaprada
Deptment : CSE
Designation : AsstProf
Phone : 1234567890
Salary : 42000
```

```
1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 5
SSN  NAME       DEPT DESGIN    PHONE       SAL
004  Jayaprada  CSE  AsstProf  1234567890  42000.00
002  Jayaprada  CSE  AsstProf  1234567890  45000.00
001  Jayaprada  CSE  AsstProf  1234567890  40000.00
003  Jayaprada  CSE  AsstProf  1234567890  43000.00


1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 4
Item deleted = 003


1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 5
SSN  NAME       DEPT DESGIN    PHONE       SAL
004  Jayaprada  CSE  AsstProf  1234567890  42000.00
002  Jayaprada  CSE  AsstProf  1234567890  45000.00
001  Jayaprada  CSE  AsstProf  1234567890  40000.00


1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 3
Item deleted = 004

1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 5
SSN  NAME       DEPT DESGIN    PHONE       SAL
002  Jayaprada  CSE  AsstProf  1234567890  45000.00
001  Jayaprada  CSE  AsstProf  1234567890  40000.00
```

```
1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 4
Item deleted = 001

1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 5
SSN  NAME       DEPT DESGIN    PHONE       SAL
002  Jayaprada  CSE  AsstProf  1234567890  45000.00


1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 3
Item deleted = 002


1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 5
List is Empty


1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 3
List is Empty


1:Insert Front 2:Insert Rear
3:Delete Front 4:Delete Rear
5:Display 6:Count 7: Exit : 7

...Program finished with exit code 0
Press ENTER to exit console.
*/
```