

```

//CIRCULAR LINKED LIST
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node*link;
}*h,*last;
void addbeg()
{
    int item;
    struct node *new;
    new=(struct node *) malloc (sizeof(struct node *));
    printf("Enter the data: ");
    scanf("%d",&item);
    new->data=item;
    new->link=NULL;
    if(h->link==NULL)
    {
        h->link=new;
        new->link=h->link;
        last=new;
    }
    else
    {
        new->link=h->link;
        h->link=new;
        last->link=new;
    }
}
void addbtw()
{
    int item,key;
    struct node *new,*ptr;
    new=(struct node *) malloc (sizeof(struct node *));
    printf("Enter the data: ");
    scanf("%d",&item);
    printf("\nEnter the key: ");
    scanf("%d",&key);
    new->data=item;
    new->link=NULL;
    ptr=h->link;
    while(ptr->data!=key&&ptr->link!=NULL)
    {
        ptr=ptr->link;
    }
    if(ptr->link==NULL)
    {
        printf("Insertion not possible");
    }
    else
    {
        new->link=ptr->link;
    }
}

```

```

    ptr->link=new;
}
}
void addend()
{
    int item;
    struct node *new;
    new=(struct node *)malloc(sizeof(struct node *));
    printf("\nEnter the data: ");
    scanf("%d",&item);
    new->data=item;
    new->link=last->link;
    last->link=new;
    last=new;
}
void disp()
{
    struct node *ptr;
    ptr=h->link;
    if(ptr==NULL)
    {
        printf("List Empty\n");
    }
    else
    {
        while(ptr!=last)
        {
            printf("%d\t",ptr->data);
            ptr=ptr->link;
        }
        printf("%d\t",ptr->data);
    }
}
void delbeg()
{
    struct node *ptr;
    ptr=h->link;
    if(ptr==NULL)
    {
        printf("\nList Empty");
    }
    else
    {
        h->link=ptr->link;
        last->link=h->link;
        free(ptr);
    }
}
void delend()
{
    struct node *ptr,*tmp;
    ptr=h->link;
    if(ptr==NULL)

```

```

{
    printf("\nList Empty");
}
else
{
    while(ptr!=last)
    {
        tmp=ptr;
        ptr=ptr->link;
    }
    tmp->link=last->link;
    last=tmp;
    free(ptr);
}
}
void delbtw()
{
    int key;
    printf("\nEnter the key: ");
    scanf("%d",&key);
    struct node *ptr,*tmp;
    ptr=h->link;
    if(ptr==NULL)
    {
        printf("\nList Empty");
    }
    else
    {
        tmp=h;
        while(ptr->data!=key&&ptr->link!=NULL)
        {
            tmp=ptr;
            ptr=ptr->link;
        }
        if(ptr->link==NULL)
        {
            printf("\nDeletion in between not possible");
        }
        else
        {
            tmp->link=ptr->link;
            free(ptr);
        }
    }
}
void main()
{
    int ch;
    h=(struct node *)malloc(sizeof(struct node *));
    do
    {
        printf("\n1:Add beginning\n2:Add between\n3:Add ending\n4:Display\n5:Del beginning\n6:Del ending\n7:Del
between\n8:EXIT\n");
    }

```

```

printf("Enter the choice: ");
scanf("%d",&ch);
switch(ch)
{
    case 1:addbeg();break;
    case 2:addbtw();break;
    case 3:addend();break;
    case 4:disp();break;
    case 5:delbeg();break;
    case 6:delend();break;
    case 7:delbtw();break;
    case 8:exit(0);break;
    default:printf("\nInvalid choice");
}
}while(ch<=8);
}

```

/*OUTPUT:

```

1:Add beginning
2:Add between
3:Add ending
4:Display
5:Del beginning
6:Del ending
7:Del between
8:EXIT
Enter the choice: 4
List Empty

```

```

1:Add beginning
2:Add between
3:Add ending
4:Display
5:Del beginning
6:Del ending
7:Del between
8:EXIT
Enter the choice: 1
Enter the data: 30

```

```

1:Add beginning
2:Add between
3:Add ending
4:Display
5:Del beginning
6:Del ending
7:Del between
8:EXIT
Enter the choice: 4
30
1:Add beginning

```

2:Add between
3:Add ending
4:Display
5:Del beginning
6:Del ending
7:Del between
8:EXIT
Enter the choice: 1
Enter the data: 10

1:Add beginning
2:Add between
3:Add ending
4:Display
5:Del beginning
6:Del ending
7:Del between
8:EXIT
Enter the choice: 4
10 30

1:Add beginning
2:Add between
3:Add ending
4:Display
5:Del beginning
6:Del ending
7:Del between
8:EXIT
Enter the choice: 2
Enter the data: 20

Enter the key: 10

1:Add beginning
2:Add between
3:Add ending
4:Display
5:Del beginning
6:Del ending
7:Del between
8:EXIT
Enter the choice: 4
10 20 30
1:Add beginning
2:Add between
3:Add ending
4:Display
5:Del beginning
6:Del ending
7:Del between
8:EXIT
Enter the choice: 3

Enter the data: 50

1:Add beginning

2:Add between

3:Add ending

4:Display

5:Del beginning

6:Del ending

7:Del between

8:EXIT

Enter the choice: 4

10 20 30 50

1:Add beginning

2:Add between

3:Add ending

4:Display

5:Del beginning

6:Del ending

7:Del between

8:EXIT

Enter the choice: 5

1:Add beginning

2:Add between

3:Add ending

4:Display

5:Del beginning

6:Del ending

7:Del between

8:EXIT

Enter the choice: 4

20 30 50

1:Add beginning

2:Add between

3:Add ending

4:Display

5:Del beginning

6:Del ending

7:Del between

8:EXIT

Enter the choice: 7

Enter the key: 30

1:Add beginning

2:Add between

3:Add ending

4:Display

5:Del beginning

6:Del ending

7:Del between

8:EXIT

Enter the choice: 4

20 50

1:Add beginning

2:Add between

3:Add ending

4:Display

5:Del beginning

6:Del ending

7:Del between

8:EXIT

Enter the choice: 6

1:Add beginning

2:Add between

3:Add ending

4:Display

5:Del beginning

6:Del ending

7:Del between

8:EXIT

Enter the choice: 4

20

1:Add beginning

2:Add between

3:Add ending

4:Display

5:Del beginning

6:Del ending

7:Del between

8:EXIT

Enter the choice: 8

user@user-WIV68B55-0113:~\$

*/