```java
import java.util.Scanner;

class Circular
{
    class Node {
        int data;
        Node next;

        Node(int d) {
            data = d;
            next = null;
        }
    }
    Node root,last;
    Circular()
    {
        root=last=null;
    }
    public void insertLeft(int e)
    {
        Node n=new Node(e);
        if(root==null)
        {
            root=last=n;
            last.next=root;
        }
        else
        {
            n.next=root;
            root=n;
            last.next=root;
        }
    }
    public void deleteLeft()
    {
        if(root==null)
        {
            System.out.println("Empty List");
        }
        else
        {
            Node t=root;
            root=root.next;
            last.next=root;
            System.out.println(t.data+ "Removed ");
        }
    }
    public void insertRight(int e)
    {
        Node n=new Node(e);
        if(root==null)
        {
            root=last=n;
            last.next=root;
        }
        else
        {
            last.next=n;
            last=n;
            last.next=root;
        }
    }
    public void deleteRight()
    {
        if(root==null)
        {
            System.out.println("Empty List");
        }
        else
        {
            Node t=root;
            Node t2=root;
        while(t!=last)
        {
            t2=t;
            t=t.next;

        }
        last=t2;
        last.next=root;
        System.out.println(t.data+ "Remove");
        }
    }
    public void printList()
    {
        if(root==null)
        {
            System.out.println("Empty List");
        }
        else
        {
            Node t=root;
            do
            {
                System.out.println(t.data);
                t=t.next;
            }while(t!=root);
        }
    }
}
public class CircularList {
    public static void main(String args[]) {
        int val, ch;
        Scanner s = new Scanner(System.in);
        Circular c = new Circular();
        do {
            System.out.println("\n1.Insert Left \n2.Delete Left \n3.Insert Right \n4.Delete Right \n5.Print List \nEnter choice :");
            ch = s.nextInt();
        switch (ch) {
            case 1:
                System.out.println("Insert Left Node");
                val = s.nextInt();
                c.insertLeft(val);
                break;
            case 2:
                System.out.println("Delete Left Node");
                c.deleteLeft();
                break;
            case 3:
                System.out.println("Insert Right Node");
                val = s.nextInt();
                c.insertRight(val);
                break;
            case 4:
                System.out.println("Delete Right Node");
                c.deleteRight();
                break;
            case 5:
                System.out.println("Print LinkList");
                c.printList();
                break;

        }

        }while(ch!=0);
    }
}

```

```java
import java.util.Scanner;

class Dll
{
    class Node
    {
        int data;
        Node left,right;
        Node(int e)
        {
            data=e;
            left=null;
            right=null;
        }
    }
    Node root,last;
    Dll()
    {
        root=last=null;
    }
    public void insertLeft(int e)
    {
        Node n=new Node(e);
        if(root==null)
        {
            root=last=n;
        }
        else
        {
            root.left=n;
            n.right=root;
            root=n;
        }
    }
    public void deleteLeft()
    {
        if(root==null)
        {
        System.out.println("Empty List");
        }
        else
        {
            Node t=root;
            root=root.right;
            root.left=null;
            System.out.println(t.data+ "Remove");
        }
    }
    public void insertRight(int e)
    {
        Node n=new Node(e);
        if(root==null)
        {
            root=last=n;
        }
        else
        {
            last.right=n;
            n.left=root;
            last=n;
        }
    }
    public void deleteRight()
    {
        if(root==null)
        {
        System.out.println("Empty List");
        }
        else
        {
            Node t=last;
            last=last.left;
            last.right=null;
            System.out.println(t.data+ "Remove");
        }
    }
    public void printStart() {
        Node t=root;
        while(t!=null)
        {
            System.out.println(t.data);
            t=t.right;
        }
    }
    public void printLast() {
        Node t=last;
        while(t!=null)
        {
            System.out.println(t.data);
            t=t.left;
        }
    }

}
public class DoublyList {
    public static void main(String args[]) {
        int value, ch;
        Scanner s = new Scanner(System.in);
        Dll d = new Dll();
        do {
            System.out.println("\n1.Insert Left \n2.Delete Left \n3.Insert Right \n4.Delete 
Right \n5.Print Start \n6.Print Last \nEnter choice :");
            ch = s.nextInt();
            switch (ch) {
            case 1:
                System.out.println("Insert Left Node");
                value = s.nextInt();
                d.insertLeft(value);
                break;
            case 2:
                System.out.println("Delete Left Node");
                d.deleteLeft();
                break;
            case 3:
                System.out.println("Insert Right Node");
                value = s.nextInt();
                d.insertRight(value);
                break;
            case 4:
                System.out.println("Delete Right Node");
                d.deleteRight();
                break;
            case 5:
                System.out.println("Print LinkList Start");
```

```java
                    d.printStart();
                    break;
                case 6:
                    System.out.println("Print LinkList Last");
                    d.printLast();
                    break;
                }
            }while(ch!=0);

}
}

```