

Doubly LinkedList:

```
import java.util.Scanner;

public class DoubleLinkedList {
    static class Node {
        int data;
        Node next, previous;
        Node(int data) {
            this.data = data;
            this.next = null;
            this.previous = null;
        }
    }

    Scanner input = new Scanner(System.in);
    Node head, temp = null;

    public void create(int data) {
        Node newNode = new Node(data);
        newNode.previous = null;
        newNode.next = null;
        if (head == null) {
            head = temp = newNode;
        } else {
            temp.next = newNode;
            newNode.previous = temp;
            temp = newNode;
            temp.next = null;
        }
    }

    public void addLeft(int data) {
        Node newNode = new Node(data);
        newNode.previous = null;
        newNode.next = null;
        head.previous = newNode;
        newNode.next = head;
        head = newNode;
    }

    public void addRight(int data) {
```

```

Node newNode = new Node(data);
newNode.previous = null;
newNode.next = null;
temp.next = newNode;
newNode.previous = temp;
temp = newNode;
}

public void insertPosition() {
    int i = 1;
    temp = head;
    System.out.print("Enter position: ");
    int position = input.nextInt();

    System.out.print("Enter data: ");
    int data = input.nextInt();
    Node newNode = new Node(data);

    if (position < 0) {
        System.out.println("Invalid Position");
    } else {
        while (i < position - 1) {
            temp = temp.next;
            ++i;
        }
        newNode.previous = temp;
        newNode.next = temp.next;
        temp.next = newNode;
        newNode.next.previous = newNode;
    }
}

public void deleteLeft() {
    if (head == null) {
        System.out.println("Empty list");
    } else {
        head = head.next;
        head.previous = null;
    }
}

public void deleteRight() {
    if (head == null) {
        System.out.println("Empty List");
    }
}

```

```

    } else {
        temp.previous.next = null;
        temp = temp.previous;
    }
}

public void deletePosition() {
    int i = 1;
    temp = head;
    System.out.print("Enter Position: ");
    int position = input.nextInt();
    if (position < 0) {
        System.out.println("Invalid Position");
    } else {
        while (i < position - 1) {
            temp = temp.next;
            ++i;
        }
        temp.previous.next = temp.next;
        temp.next.previous = temp.previous;
    }
}

```

```

public void display() {
    Node curNode = head;
    if (curNode == null) {
        System.out.println("Empty List");
    } else {
        while (curNode != null) {
            System.out.print(curNode.data + " ");
            curNode = curNode.next;
        }
        System.out.println();
    }
}

```

```

public static void main(String[] args) {
    DoubleLinkedList myList = new DoubleLinkedList();
    myList.create(2);
    myList.addLeft(1);
    myList.addRight(4);
    myList.addRight(5);
    myList.addRight(6);
}

```

