

## Singly LinkedList:

```
public class LinkedList { //Singly Linked List
    static class Node {
        int data;
        Node next;

        public Node(int data) {
            this.data = data;
            this.next = null;
        }
    }

    Node head = null;

    public void addFirst(int data) {
        Node newNode = new Node(data);
        if (head == null) {
            head = newNode;
            return;
        }
        Node temp = head;
        head = newNode;
        head.next = temp;
    }

    public void addLast(int data) {
        Node newNode = new Node(data);
        if (head == null) {
            head = newNode;
            return;
        }
        Node curNode = head;
        while (curNode.next != null) {
            curNode = curNode.next;
        }
        curNode.next = newNode;
    }

    public void delLeft() {
        if (head == null) {
            System.out.println("Empty List");
            return;
        }
        head = head.next;
    }
}
```

```

public void delRight() {
    if (head == null) {
        System.out.println("Empty List");
        return;
    }
    Node secondLastNode = head;
    Node lastNode = head.next;
    while (lastNode.next != null) {
        lastNode = lastNode.next;
        secondLastNode = secondLastNode.next;
    }
    secondLastNode.next = null;
}

public void search() {
    Node p = head;
    int s, position = 0;
    Scanner input = new Scanner(System.in);
    System.out.print("Which element you want to remove: ");
    s = input.nextInt();

    while (p != null) {
        if (p.data == s) {
            System.out.println("element found");
            break;
        }
        ++position;
        p = p.next;
    }
    delete(position);
}

public void delete(int position) {
    Node temp = head;
    if (head == null) {
        return;
    }
    if (position == 0) {
        head = temp.next;
        return;
    }

    for (int i = 0; temp != null && i < position - 1; i++) {
        temp = temp.next;
    }

    if (temp == null || temp.next == null) {
        return;
    }

```

```

    }
    temp.next = temp.next.next;
}

public void display() {
    Node curNode = head;
    if (head == 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) {
    LinkedList myList = new LinkedList();
    myList.addFirst(5);
    myList.addLast(6);
    myList.addFirst(4);
    myList.addLast(7);
    myList.addLast(8);
    myList.display();
    myList.delRight();
    myList.delLeft();
    myList.search();
    myList.display();
}
}

```