```java
/*
    Question 4
    Creator: Wahid Bawa
    Purpose: To add the new sortedInsert method
*/
public class LList {
    private LNode head, tail;

    public LList() {
        head = null;
        tail = null;
    }

    public void add(int val) {
        LNode tmp = new LNode(val, tail, null);
        if (tail != null) {
            tail.setNext(tmp);
        }
        if (head == null) {
            head = tmp;
        }
        tail = tmp;
    }

    public String toString() { // displays the parts of the linked list
        String ans = "";
        LNode tmp = head;
        while (tmp != null) {
            ans += tmp.getVal() + (tmp.getNext() == null ? "" : "-");
            tmp = tmp.getNext();
        }
        return ans;
    }

    public void sortedInsert(int val) { // this inserts val into list in ascending order
        LNode tmp = head;
        if (tmp == null) {
            add(val);
        }
        while (tmp != null) {
            if (tmp == head) {
                if (val <= head.getVal()) {
                    LNode n = new LNode(val, null, head);
                    head.setPrev(n);
                    head = n;
                    break;
                }
            }
            if (tmp == tail) { // this is for the special case for last element
                if (val >= tmp.getVal()) {
                    add(val);
                    break;
                }
            }
            else if (tmp.getVal() <= val && val <= tmp.getNext().getVal()) { // this sees if the prev val is lower and next val is higher
                LNode n = new LNode(val, tmp, tmp.getNext()); // inserts the node inbetween
                tmp.getNext().setPrev(n);
                tmp.setNext(n);
                break;
            }
            tmp = tmp.getNext();
        }
    }
}
```