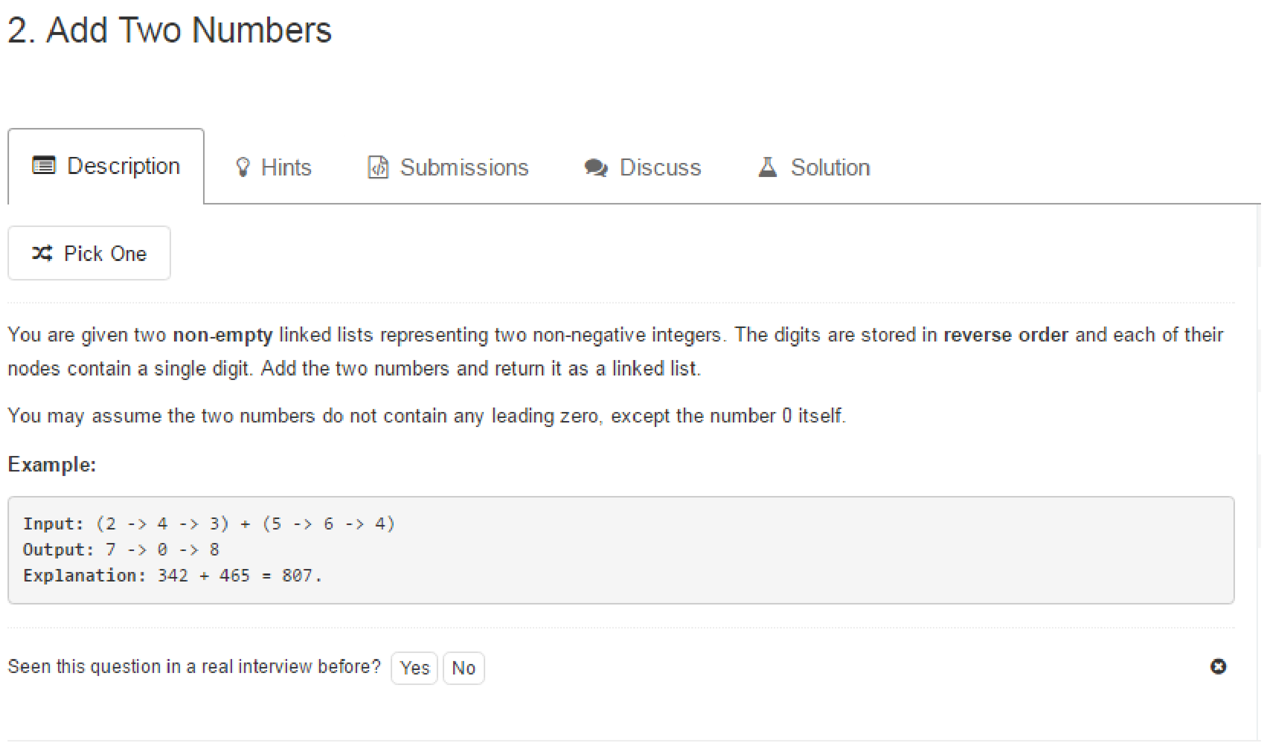


2. Add Two Numbers

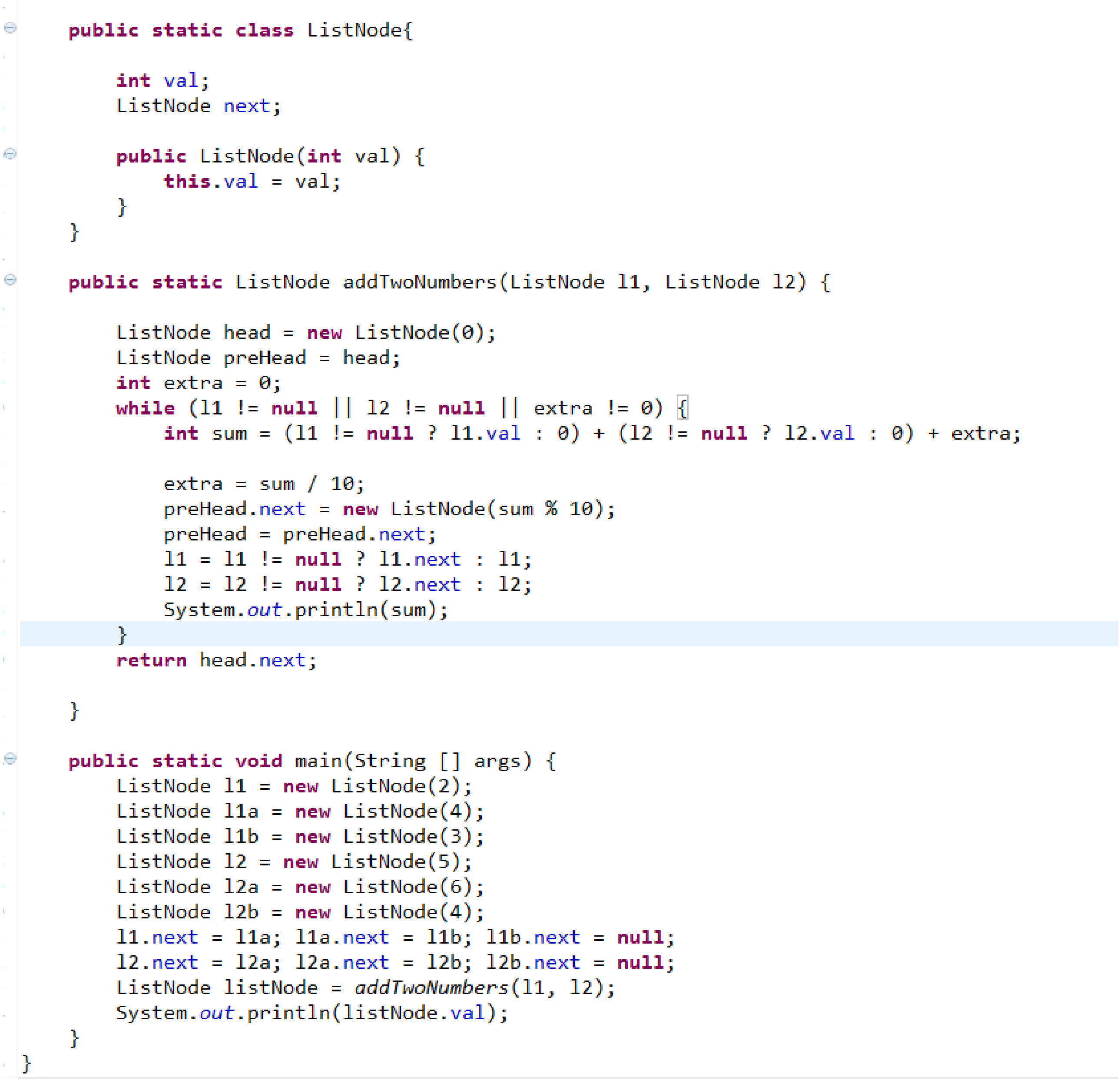
You are given two **non-empty** linked lists representing two non-negative integers. The digits are stored in **reverse order** and each of their nodes contain a single digit. Add the two numbers and return it as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

Example:

```
Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)
Output: 7 -> 0 -> 8
Explanation: 342 + 465 = 807.
```

Seen this question in a real interview before?



```java
    public static class ListNode{

        int val;
        ListNode next;

        public ListNode(int val) {
            this.val = val;
        }
    }

    public static ListNode addTwoNumbers(ListNode l1, ListNode l2) {

        ListNode head = new ListNode(0);
        ListNode preHead = head;
        int extra = 0;
        while (l1 != null || l2 != null || extra != 0) {
            int sum = (l1 != null ? l1.val : 0) + (l2 != null ? l2.val : 0) + extra;

            extra = sum / 10;
            preHead.next = new ListNode(sum % 10);
            preHead = preHead.next;
            l1 = l1 != null ? l1.next : l1;
            l2 = l2 != null ? l2.next : l2;
            System.out.println(sum);
        }
        return head.next;

    }

    public static void main(String [] args) {
        ListNode l1 = new ListNode(2);
        ListNode l1a = new ListNode(4);
        ListNode l1b = new ListNode(3);
        ListNode l2 = new ListNode(5);
        ListNode l2a = new ListNode(6);
        ListNode l2b = new ListNode(4);
        l1.next = l1a; l1a.next = l1b; l1b.next = null;
        l2.next = l2a; l2a.next = l2b; l2b.next = null;
        ListNode listNode = addTwoNumbers(l1, l2);
        System.out.println(listNode.val);
    }
}
```