Problem on SLL: Removing the N^th Node from the End

You are asked to program this method:

public ListNode removeNthFromEnd(**ListNode** head, int n)

Remove the n^th node from the end of the chain starting from head.

Requirement: n ≤ number of nodes in the input chain

https://leetcode.com/problems/remove-nth-node-from-end-of-list/

```
@Test
public void test_1() {
    ListNode input =
        new ListNode(1,
        new ListNode(2,
        new ListNode(3,
        new ListNode(4,
        new ListNode(5, null)))));
    ListUtilities util = new ListUtilities();
    ListNode output = util.removeNthFromEnd(input, 2);
    assertTrue(input == output);
    assertTrue(input.val == 1);
    assertTrue(input.next.val == 2);
    assertTrue(input.next.next.val == 3);
    assertTrue(input.next.next.next.val == 5);
}
```

```
@Test
public void test_2() {
    ListNode input = new ListNode(1, null);
    ListUtilities util = new ListUtilities();
    ListNode output = util.removeNthFromEnd(input, 1);
    assertTrue(output == null);
}
```