Problem on SLL: Shifting the List to the Right

You are asked to program this method:

public Node<S.> shiftedToRightBy(**Node<S.>** head, **int** n)

Return the same chain, with nodes being shifted to the right by **n** positions.

Assumptions: head is not null and n ≥ 0

```
@Test
public void test2() {
    ListUtilities<String> util = new ListUtilities<>();
    Node<String> n4 = new Node<>("Lists", null);
    Node<String> n3 = new Node<>("Linked", n4);
    Node<String> n2 = new Node<>("Love", n3);
    Node<String> n1 = new Node<>("I", n2);

    Node<String> output = util.shiftedToRightBy(n1, 2);
    assertTrue(output == n3);
    assertTrue(output.getNext() == n4);
    assertTrue(output.getNext().getNext() == n1);
    assertTrue(output.getNext().getNext().getNext() == n2);
    assertNull(output.getNext().getNext().getNext().getNext());

    assertTrue(output.getElement().equals("Linked"));
    assertTrue(output.getNext().getElement().equals("Lists"));
    assertTrue(output.getNext().getNext().getElement().equals("I"));
    assertTrue(output.getNext().getNext().getNext().getElement().equals("Love"));
}
```