```java
/*Author: Bochen (mddboc@foxmail.com)
Last Modified: Tue Apr 10 22:28:44 CST 2018*/

/*Given a linked list, remove the nth node from the end of list and return its head.

        For example,

        Given linked list: 1->2->3->4->5, and n = 2.

        After removing the second node from the end, the linked list becomes
        1->2->3->5.
        Note:
        Given n will always be valid.
        Try to do this in one pass.*/

import java.util.*;
import java.lang.Math;
import java.lang.System;
import java.lang.Integer;


public class Main {

    public static void main(String[] args) {
        int[] nums = {5, 5, 3, 5, 1, -5, 1, -2};

        Solution solution = new Solution();
        List<List<Integer>> receive = solution.fourSum(nums, 4);


        System.out.println("haha");
    }

}


class ListNode {
    int val;
    ListNode next;

    ListNode(int x) {
        val = x;
    }
}

class Solution {
    public ListNode removeNthFromEnd(ListNode head, int n) {

        ListNode psudoHeadNode = new ListNode(0);
        psudoHeadNode.next = head;

        ListNode frontNode = psudoHeadNode, backNode = psudoHeadNode;

        for (int i = 0; i < n; i++) {
            frontNode = frontNode.next;
        }

        while (frontNode.next != null) {

            frontNode = frontNode.next;
            backNode = backNode.next;
        }

        frontNode = backNode.next.next;
        backNode.next = frontNode;

        return psudoHeadNode.next;
    }
}
```