Remove Nth Node from End of List

Question: 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.

Solutions:

```
class ListNode:
    def __init__(self, x):
        self.val = x
        self.next = None


class Solution:
    def getlength(self,head):
        res = 0
        while(head):
            res += 1
            head = head.next
        return res


    def removeNthFromEnd(self, head, n):
        """
        :type head: ListNode
        :type n: int
        :rtype: ListNode
        """
        if self.getlength(head)==n:
            return head.next

        node = head
        for i in range(self.getlength(head)-n-1):
            node = node.next
        node.next = node.next.next
        return head

    def printll(self, node):
        while node:
            print ( node.val )
            node = node.next

if __name__ == '__main__':
    ll1, ll1.next, ll1.next.next = ListNode(0), ListNode(1), ListNode(5)
    Solution().printll( Solution().removeNthFromEnd(ll1,2) )
```