Rotate List

Question: Given a list, rotate the list to the right by k places, where k is non-negative.

For example:

Given 1->2->3->4->5->NULL and k = 2,

return 4->5->1->2->3->NULL

Solutions:

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

    def to_list(self):
        return [self.val] + self.next.to_list() if self.next else [self.val]

class Solution:
    # @param head, a ListNode
    # @param k, an integer
    # @return a ListNode
    def rotateRight(self, head, k):
        if head == None:
            return None
        temp = head
        for i in range(0,k):
            if temp.next == None:
                temp = head
            else:
                temp = temp.next
        newLast = head
        while temp.next != None:
            temp = temp.next
            newLast = newLast.next
        temp.next = head
        newHead = newLast.next
        newLast.next = None
        return newHead

if __name__ == "__main__":
    n1 = ListNode(1)
    n2 = ListNode(2)
    n3 = ListNode(3)
    n4 = ListNode(4)
    n5 = ListNode(5)
    n1.next = n2
    n2.next = n3
    n3.next = n4
    n4.next = n5
    r = Solution().rotateRight(n1, 2)
    print ( r.to_list() )
```