SE(AI&DS) - Bhavya Damani_10

```
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next

def rotateRight(head: ListNode, k: int) -> ListNode:
    if not head or not head.next or k == 0:
        return head

    length = 1
    last = head
    while last.next:
        last = last.next
        length += 1

    k = k % length
    if k == 0:
        return head

    new_tail = head
    for _ in range(length - k - 1):
        new_tail = new_tail.next

    new_head = new_tail.next
    new_tail.next = None
    last.next = head

    return new_head

def list_to_linked(lst):
    if not lst:
        return None
    head = ListNode(lst[0])
    curr = head
    for val in lst[1:]:
        curr.next = ListNode(val)
        curr = curr.next
    return head

def linked_to_list(head):
    result = []
    while head:
        result.append(head.val)
        head = head.next
    return result

user_list = list(map(int, input("Enter space-separated elements of the linked list: ").split()))
k = int(input("Enter the number of rotations: "))

head = list_to_linked(user_list)
rotated_head = rotateRight(head, k)

print("Rotated List:", linked_to_list(rotated_head))
```

```
Enter space-separated elements of the linked list: 0 1 2
Enter the number of rotations: 4
Rotated List: [2, 0, 1]
```

Start coding or generate with AI.