SE(AI&DS)-Nihaal_Gharat-19

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

        self.val = val
        self.next = next

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


    length = 1
    tail = head
    while tail.next:
        tail = tail.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
    tail.next = head

    return new_head


    if not values:
        return None
    head = ListNode(values[0])
    current = head
    for val in values[1:]:
        current.next = ListNode(val)
        current = current.next
    return head


def print_list(head):
    result = []
    while head:
        result.append(head.val)
        head = head.next
    print("Rotated Linked List:", result)


values = list(map(int, input("Enter linked list elements separated by spaces: ").split()))
k = int(input("Enter the number of rotations: "))


head = create_linked_list(values)


new_head = rotateRight(head, k)


print_list(new_head)
```

```
Enter linked list elements separated by spaces: 1 2 3 4
5 Enter the number of rotations: 3
Rotated Linked List: [3, 4, 5, 1, 2]
```

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

        self.val = val
        self.next = next

def rotateRight(head, k):
    if not head or not head.next or k == 0:
```

```
        return head
```

```
    length = 1
    tail = head
    while tail.next:
        tail = tail.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
    tail.next = head

    return new_head


def create_linked_list(values):
    if not values:
        return None
    head = ListNode(values[0])
    current = head
    for val in values[1:]:
        current.next = ListNode(val)
        current = current.next
    return head


def print_list(head):
    result = []
    while head:
        result.append(head.val)
        head = head.next
    print("Rotated Linked List:", result)


values = list(map(int, input("Enter linked list elements separated by spaces: ").split()))
k = int(input("Enter the number of rotations: "))


head = create_linked_list(values)


new_head = rotateRight(head, k)


print_list(new_head)
```

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

Double-click (or enter) to edit