Shruti Gauchandra_16

Assignment - 03

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None

    def insertion(self, data):
        new_node = Node(data)
        if not self.head:
            self.head = new_node
        else:
            current = self.head
            while current.next:
                current = current.next
            current.next = new_node

    def traversal(self):
        current = self.head
        while current:
            print(current.data, end=" ")
            current = current.next
        print()

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

        # Compute the length
        length = 1
        tail = self.head
        while tail.next:
            tail = tail.next
            length += 1

        # Make it circular
        tail.next = self.head

        # Find the new tail
        k = k % length
        if k == 0:
            tail.next = None
            return

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

        # Set new head and break the circular link
        self.head = new_tail.next
        new_tail.next = None

# Example usage
ll = LinkedList()
ll.insertion(1)
ll.insertion(2)
ll.insertion(3)
ll.insertion(4)
ll.insertion(5)
print("For Example 1")
print("Original List 1:")
ll.traversal()

ll.rotateRight(2)

print("Rotated List 1:")
```

```
l1.traversal()

l2 = LinkedList()
l2.insertion(0)
l2.insertion(1)
l2.insertion(2)
print("For Example 2")
print("Original List 2 :")
l2.traversal()

l2.rotateRight(4)

print("Rotated List 2:")
l2.traversal()
```

```
For Example 1
Original List 1:
1 2 3 4 5
Rotated List 1:
4 5 1 2 3
For Example 2
Original List 2 :
0 1 2
Rotated List 2:
2 0 1
```