ROTATING A LINKED LIST

☆ In this problem, we are given an integer K and a singly linked list and our job is to rotate the linked list K times.

Obvious solution is to first find the length of linked list and trace its end. Any rotation is only possible when the end points to the head and K-1th element from head points to null.

C++ :

```
Node * rotateLL(Node * head, int K) {
    int c = 1;
    Node * temp = head;
    while(temp -> next != nullptr) {
        c++;
        temp = temp -> next;
    }
    if(K % c == 0) {
        return head;
    }
    K = K % c;
    temp -> next = head;
    int t = c - K;
    temp = head;
    while(t > 0) {
        temp = temp -> next;
```

```
        t--;
    }
    head = temp->next;
    temp->next = nullptr;
    return head;
}
```