REVERSING NODES IN K GROUP IN LINKED LISTS

☆ In this problem, we are given the head of a linked list and a number K. We are supposed to find all groups of size K and reverse them individually

Bruteforce solution is to just find each groups head and reversing the K^{th} Group. We need to make sure to preserve each node before each reversal takes place.

C++ :

```
Node * reverseKGroup (Node * head, int K){
    Node * temp = head ;
    Node * prevNode = nullptr ;
    Node * nextNode = nullptr ;
    while(temp != nullptr) {
        Node * KthNode = findKth(temp, K) ;
        if (KthNode == nullptr) {
            if(prevNode != nullptr)
                prevNode -> next = temp ;
            break ;
        }
        Node * nextN = KthNode -> next ;
        KthNode -> next = nullptr ;
        reverse(temp) ;
        if(temp == head) {
            head = KthNode ;
        } else {
            prevNode->next = KthNode;
        }
        prevNode = temp;
        temp = nextNode;
    }
    return head;
}
Node * findKth(Node * head, int K) {
    K -= 1;
    while(head != nullptr && K>0) {
        K--;
        head = head->next;
    }
    return head;
}

Node * reverse(Node * head) {
    Node * temp = head;
    Node * prev = nullptr;
    while(prev != nullptr) {
        Node * front = temp->next;
        temp->next = prev;
        prev = temp;
        temp = front;
    }
    return prev;
}
```