ONE LINE ADDITION TO PREVIOUS PROBLEM (RECURSION).

Given a linked list of size **N**. The task is to reverse every **k** nodes (where k is an input to the function) in the linked list.

**Input:**  
First line of input contains number of testcases T. For each testcase, first line contains length of linked list and next line contains the linked list elements.

**Output:**  
For each testcase, there will be a single line of output which contains the linked list with every k group elements reversed.

**User Task:**  
The task is to complete the function **reverse**() which should reverse the linked list in group of size **k**.

**Expected Time Complexity**: O(n)  
**Expected Auxilliary Space**: O(1)

**Example:**  
**Input:**  
2  
8  
1 2 2 4 5 6 7 8  
4  
5  
1 2 3 4 5  
3

**Output:**  
4 2 2 1 8 7 6 5  
3 2 1 5 4

CODE LOGIC -> USE PREVIOUS APPROACH AND RECURSION

CODE->

struct node \*reverse (struct node \*head, int k)

{

struct node \* prev=NULL, \*curr=head, \*second = NULL;

int counter = k;

while(curr!=NULL&&counter --)

{

second = curr->next;

curr->next= prev;

prev = curr;

curr = second;

}

if(head!=NULL)

head->next= reverse(second , k);

return prev;

// Complete this method

}