PROBLEM

Remove every kth node

Easy Accuracy: 29.88% Submissions: 70K+ Points: 2

Done with winning Geekbits? Now win GfG Bag, GfG T-shirt & much more just by writing your experiences. Start Writing, Start Winning.

Given a singly linked list having **n** nodes, your task is to remove every **k**th node from the linked list.

Example 1:

Input:
n = 8
linked list: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8
k = 2
Output:
1 -> 3 -> 5 -> 7
Explanation:
After removing every 2nd node of the linked list, the resultant linked list will be: 1 -> 3 -> 5 -> 7.

Example 2:

Input:
n = 10
linked list: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 10
k = 3
Output:
1 -> 2 -> 4 -> 5 -> 7 -> 8 -> 10
Explanation:
After removing every 3rd node of the linked list, the resultant linked list will be: 1 -> 2 -> 4 -> 5 -> 7 -> 8 -> 10.

Your Task:
The task is to complete the function **deleteK()** which takes **head** of linked list and integer **k** as input parameters and delete every **k**th node from the linked list and return its head.

Expected Time Complexity : O(n)
Expected Auxiliary Space : O(1)

Constraints:
$1 <= n <= 10^5$
-10^9 <= elements of linked list <= 10^9
1 <= k <= n

CODE

```
#Your task is to complete this function
#Your function should return the new head pointer
'''
class node:
    def __init__(self,x):
        self.data = x
        self.next = None
'''


class Solution:
    def deleteK(self, head, k):
        if k == 1:
            return None
        temp = head
        i = 1
        while temp is not None and temp.next is not None:
            if i == k-1:
                temp.next = temp.next.next
                i = 0
            i+=1
            temp = temp.next
        return head
        #code here
```