PROBLEM

Delete Middle of Linked List

Easy Accuracy: 54.52% Submissions: 120K+ Points: 2

Given a singly linked list, delete **middle** of the linked list. For example, if given linked list is 1->2->3->4->5 then linked list should be modified to 1->2->4->5.

If there are **even** nodes, then there would be **two middle** nodes, we need to delete the **second middle element**. For example, if given linked list is 1->2->3->4->5->6 then it should be modified to 1->2->3->5->6.

If the input linked list has **single** node, then it should return **NULL.**

Example 1:

Input:
LinkedList: 1->2->3->4->5
Output:
1 2 4 5

Example 2:

Input:
LinkedList: 2->4->6->7->5->1
Output:
2 4 6 5 1

Your Task:
The task is to complete the function **deleteMid**() which takes head of the linkedlist and return head of the linkedlist with **middle element deleted** from the linked list. If the linked list is **empty** or contains **single** element then it should return **NULL**.

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

Constraints:
$1 <= n <= 10^5$
$1 <= value[i] <= 10^9$

CODE

```
#User function Template for python3

'''
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
'''
class Solution:
    def deleteMid(self,head):
        c = 0
        ptr = head
        while ptr!= None:
            ptr = ptr.next
            c = c+1
        temp = head
        if c ==1:
            head = head.next
            return head
        else:
            for i in range(c//2):
                prev = temp
                temp = temp.next
                if i ==c//2-1:
                    prev.next = temp.next
            return head
        '''
        head:  head of given linkedList
        return: head of resultant llist
        '''

        #code here
```