**Delete Middle of Linked List (Easy)**

**Problem Statement:**

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

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

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

**Solution:**

class Solution {

Node deleteMid(Node head) {

Node low = head;

Node high = head;

high = high.next.next;

while(high!=null && high.next!=null) {

low = low.next;

high = high.next.next;

}

low.next = low.next.next;

return head;

}

}