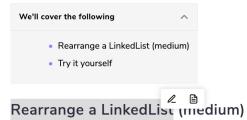


## Problem Challenge 2



Given the head of a Singly LinkedList, write a method to modify the LinkedList such that the **nodes from the second half of the LinkedList are inserted alternately to the nodes from the first half in reverse order**. So if the LinkedList has nodes 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> null, your method should return 1 -> 6 -> 2 -> 5 -> 3 -> 4 -> null.

Your algorithm should not use any extra space and the input LinkedList should be modified in-place.

## Example 1:

```
Input: 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> null
Output: 2 -> 12 -> 4 -> 6 -> 8 -> null
```

## Example 2:

```
Input: 2 -> 4 -> 6 -> 8 -> 10 -> null
Output: 2 -> 10 -> 4 -> 8 -> 6 -> null
```

## Try it yourself #

Try solving this question here:

```
Python3
   class ListNode {
     int value = 0;
     ListNode next;
     ListNode(int value) {
       this.value = value:
   class RearrangeList {
     public static void reorder(ListNode head) {
     public static void main(String[] args) {
      ListNode head = new ListNode(2);
      head.next = new ListNode(4);
       head.next.next = new ListNode(6);
       head.next.next.next = new ListNode(8);
       head.next.next.next = new ListNode(10);
       head.next.next.next.next = new ListNode(12);
       RearrangeList.reorder(head);
       while (head != null) {
         System.out.print(head.value + " ");
         head = head.next;
Run
                                                                                    Save Reset []
```



