328 Odd Even Linked List

Requirements: O(1) space; O(nodes) time.

Solution 1: O(3) space (Generating two new). #Mistake# Not the value of the nodes, but the node number.

```
public ListNode oddEvenList(ListNode head) {

    if(head == null || head.next == null) return head;

    ListNode runner = head;

    ListNode odd = new ListNode(0);
    ListNode even = new ListNode(0);

    ListNode dummyOdd = odd;
    ListNode dummyEven = even;

    while(runner != null) {
    //mistake: value
        if(runner.val%2 != 0) {
            odd.next = runner;
            odd = odd.next;

            runner = runner.next;
        } else {
```

```
            even.next = runner;
            even = even.next;

            runner = runner.next;
        }
    }

    odd.next = dummyEven.next;

    return dummyOdd.next;

}
```

Solution2: O(1) (four pointer pointing to the same linked list)

```
public ListNode oddEvenList(ListNode head) {

    if(head == null || head.next == null) return head;

    ListNode odd = head;
    ListNode even = head.next;
    ListNode evenHead = even;

    //how to decide when to stop
    while(even != null && even.next != null) {
        odd.next = even.next;
        odd = odd.next;
```

```
            even.next = odd.next;
            even = even.next;
        }

        odd.next = evenHead;

        return head;

    }
```