

328. Odd Even Linked List

题目描述：

将一个链表的奇数位（不是节点的val，而是节点的位置）的节点放在列表前部，偶数位节点放在后面，且保持其奇数节点部分和偶数节点部分位置相对不变。

例如：1->2->3->4->5->6->NULL

转换之后为：1->3->5->2->4->6->NULL

8->2->3->6->7->1->4->NULL

转换之后为：8->3->7->4->2->6->1->NULL

解题思路：

将遍历链表，将所有节点的下一跳设置为下下跳。记录第一个偶数节点和当前位置的最后一个奇数节点。

代码：

```

/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* oddEvenList(ListNode* head) {
        if(head == NULL || head->next == NULL)
            return head;
        ListNode* second = head->next;
        ListNode* now = head;
        ListNode* lastOdd = head;
        int count = 0;
        while(now->next != NULL){
            count++;
            if(count % 2 == 1)
                lastOdd = now;
            ListNode *ne = now->next;
            now->next = ne->next;
            now = ne;
        }
        if(count%2 == 0)
            lastOdd = now;
        lastOdd->next = second;
        return head;
    }
};

```