

# MCS – 253P ADVANCED PROGRAMMING AND PROBLEM SOLVING

## LAB 5 Program(Odd Even LinkedList)

Aswin Sampath  
saswin@uci.edu





53844684


Date: 04/11/2023

Question:

[Description](#) [Editorial](#) [Solutions \(6K\)](#) [Submissions](#)

### 328. Odd Even Linked List

Medium  9.3K  491  

 Companies

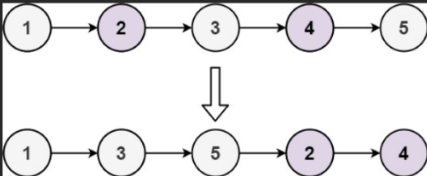
Given the `head` of a singly linked list, group all the nodes with odd indices together followed by the nodes with even indices, and return *the reordered list*.

The **first** node is considered **odd**, and the **second** node is **even**, and so on.

Note that the relative order inside both the even and odd groups should remain as it was in the input.

You must solve the problem in  $O(1)$  extra space complexity and  $O(n)$  time complexity.

**Example 1:**



**Input:** `head = [1,2,3,4,5]`  
**Output:** `[1,3,5,2,4]`

# Code

```
1 class Solution {
2 public:
3     ListNode* oddEvenList(ListNode* head) {
4         if (!head || !head->next) {
5             return head; // No change needed for empty or single-node lists
6         }
7
8         ListNode* oddHead = head;
9         ListNode* evenHead = head->next;
10        ListNode* odd = oddHead;
11        ListNode* even = evenHead;
12
13        while (even && even->next) {
14            odd->next = even->next;
15            odd = odd->next;
16            even->next = odd->next;
17            even = even->next;
18        }
19
20        odd->next = evenHead; // Connect the odd and even parts
21
22        return oddHead;
23    }
24 };
25
```

# Output:

The screenshot displays a coding platform interface with a submission for the problem "725. Split Linked List in Parts". The submission is by user "Aswin" on Nov 04, 2023, at 09:53, using C++.

**Submission Details:**

- Status: Accepted
- Runtime: 8 ms (Beats 59.62% of users with C++)
- Memory: 11.06 MB (Beats 8.81% of users with C++)

**More challenges:**

- 725. Split Linked List in Parts

**Related Tags:**

Select related tags 0/5

**Code Snippet:**

```
class Solution {
public:
    ListNode* oddEvenList(ListNode* head) {
        if (!head || !head->next) {
            return head; // No change needed for empty or single-node lists
        }

        ListNode* oddHead = head;
        ListNode* evenHead = head->next;
        ListNode* odd = oddHead;
        ListNode* even = evenHead;

        while (even && even->next) {
            odd->next = even->next;
            odd = odd->next;
            even->next = odd->next;
            even = even->next;
        }

        odd->next = evenHead; // Connect the odd and even parts

        return oddHead;
    }
};
```