

Follow me for more interesting programming questions and RAW Code visit my GitHub profile:
<https://www.github.com/shubhamthrills> <https://www.linkedin.com/in/shubhamsagar>

Subscribe our YouTube Channel for more videos: <https://www.youtube.com/channel/UCJLMu9mayAibQST1eWwq0cg>

Leetcode May Challenge DAY: 16

1. Python

class Solution:

def oddEvenList(self, head: ListNode) -> ListNode:

if head == None:

return head

even = head.next

odd = head

evenHead = even

while even != None:

odd.next = even.next

if odd.next != None:

even.next = odd.next.next

else:

even.next = None

if odd.next != None:

odd = odd.next

even = even.next

odd.next = evenHead

return head

Follow me for more interesting programming questions and RAW Code visit my GitHub profile:
<https://www.github.com/shubhamthrills> <https://www.linkedin.com/in/shubhamsagar>

Subscribe our YouTube Channel for more videos: <https://www.youtube.com/channel/UCjLMu9mayAibQST1eWwq0cg>

2. C++

class Solution {

public:

```
ListNode* oddEvenList(ListNode* head) {  
    if(head && head->next && head->next->next){  
        ListNode *odd, *even, *t1, *t2;  
        odd = head;  
        even = head->next;  
        t1 = odd;  
        t2 = even;  
        while(t1 && t2){  
            t1->next = t2->next;  
            if(t1->next){  
                t1 = t1->next;  
                t2->next = t1->next;  
                t2 = t2->next;  
            }else{  
                break;  
            }  
        }  
        t1->next = even;  
        return head;  
    }else{  
        return head;  
    }  
};
```

Follow me for more interesting programming questions and RAW Code visit my GitHub profile:
<https://www.github.com/shubhamthrills> <https://www.linkedin.com/in/shubhamsagar>

Subscribe our YouTube Channel for more videos: <https://www.youtube.com/channel/UCJLMu9mayAibQST1eWwq0cg>

3. JAVA

```
public ListNode oddEvenList(ListNode head) {  
    if (head == null || head.next == null) {  
        return head;  
    }  
    ListNode p1 = head, p2 = head.next, pre = p2;  
    while (p2 != null && p2.next != null) {  
        p1.next = p2.next;  
        p1 = p1.next;  
        p2.next = p1.next;  
        p2 = p2.next;  
    }  
    p1.next = pre;  
    return head;  
}
```