HW 09: Odd Even Linked List (Leetcode-328)

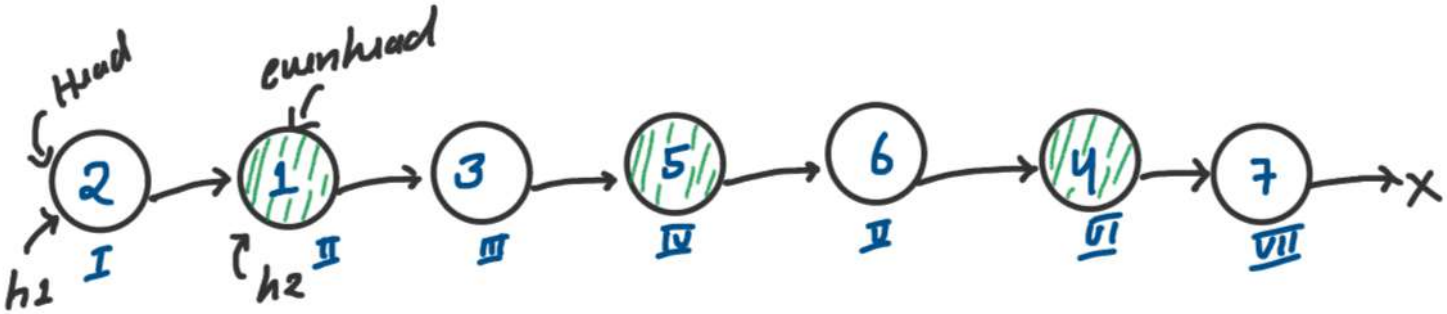
Ex

Input

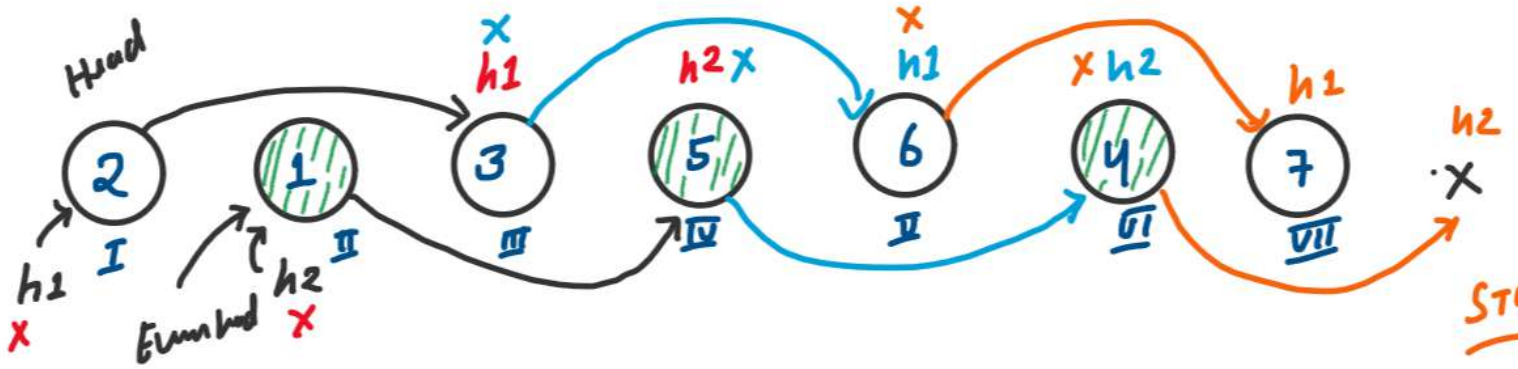
Head

Output

Logic build

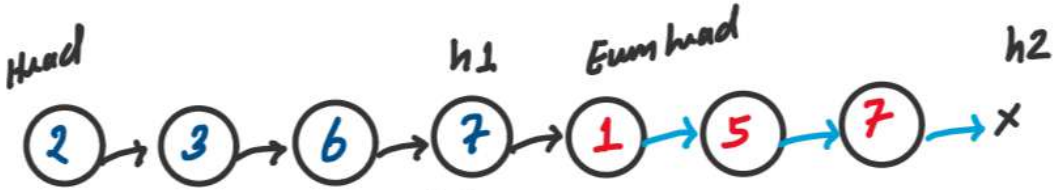



```
Node*  h1 = head;          ← odd Indexed list
Node*  h2 = head->next;    ← Even Indexed List
Node*  evenHead = h2;      ← Save the evenhead to link the odd list
```




```
h1 -> next = h2 -> next;
h2 -> next = h2 -> next -> next;
h1 = h1 -> next;
h2 = h2 -> next;
```

```
h2 == NULL &&
h2 -> next == NULL
```



Re-group the both list

```
{ h1 -> next = evenhead;
  return head; }
```

```cpp
// HW 09: Odd Even Linked List (Leetcode-328)

class Solution {
public:
    ListNode* oddEvenList(ListNode* head) {
        if(head == NULL || head->next == NULL) return head;
        // Odd indexed list
        ListNode* h1 = head;
        // Even indexed list
        ListNode* h2 = head->next;
        // Save h2 for attaching the odd index list
        ListNode* evenHead = h2;

        while(h2 && h2->next){
            h1->next = h2->next;
            h2->next = h2->next->next;

            h1 = h1->next;
            h2 = h2->next;
        }

        // Odd and even indexed list ko regroup krdo
        h1->next = evenHead;
        return head;
    }
};
```

Time complexity: *O(N),*
Where N is number of nodes of the linked list

Space complexity: *O(1),*
Where no extra space used