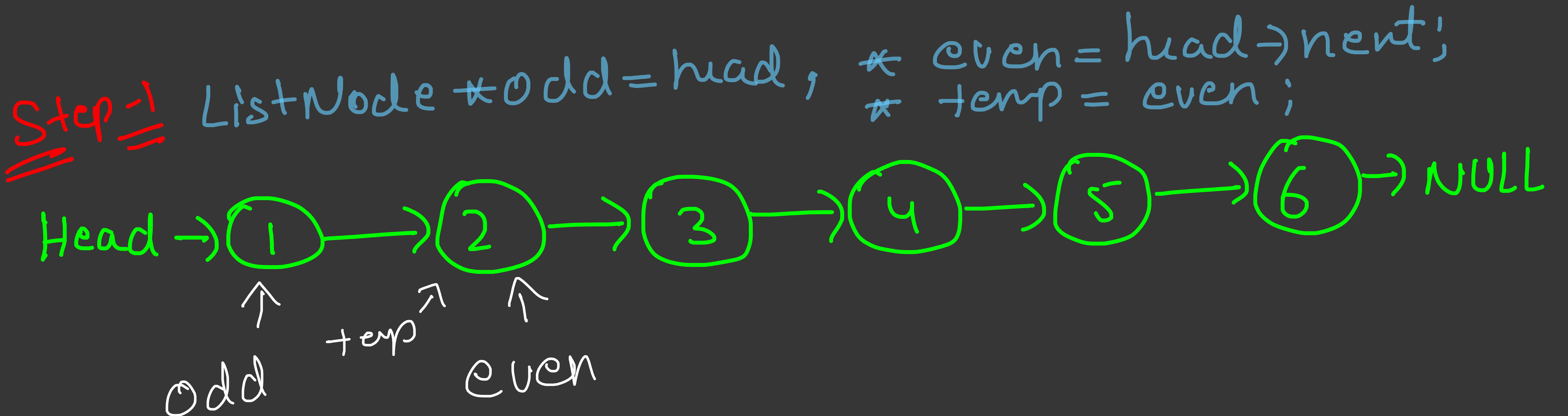
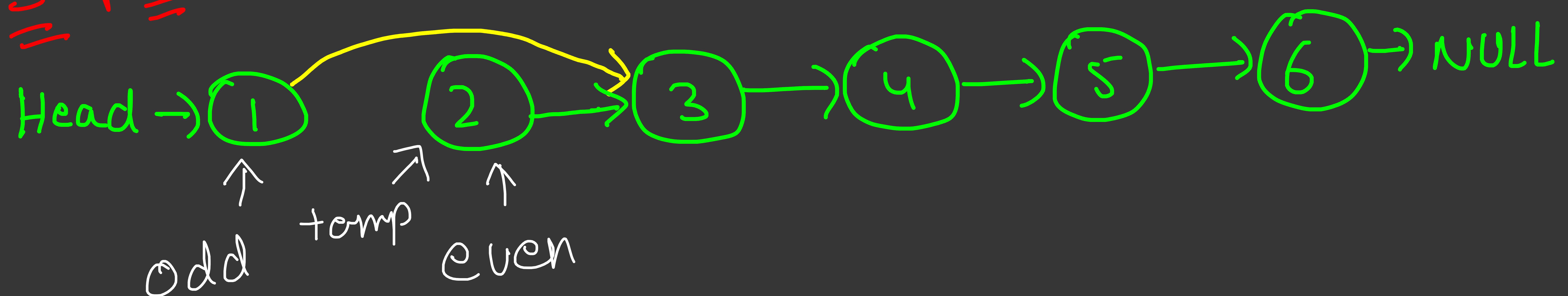


Odd Even linked List



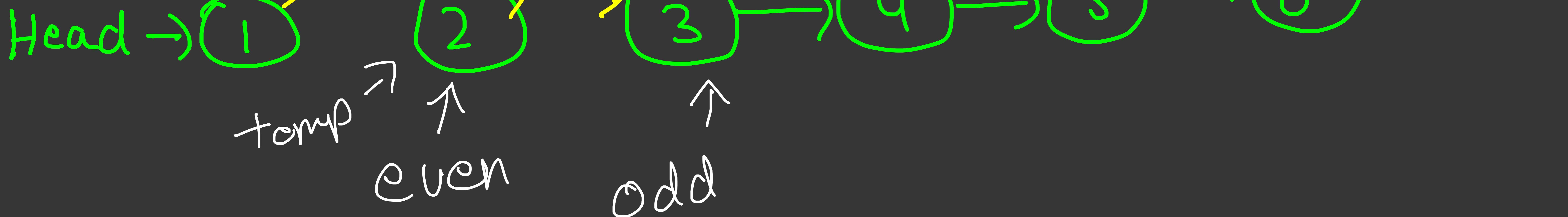
Step -2 odd->next = even->next



$\text{odd} = \text{odd} \rightarrow \text{next}$
 $\text{even} \rightarrow \text{next} = \text{odd} \rightarrow \text{next}$

$\text{stop} = 3$

$=$

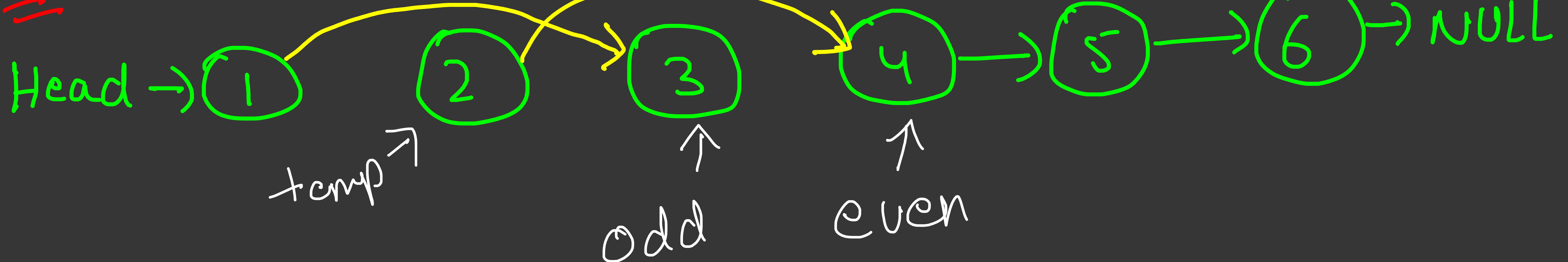


$\text{even} = \text{even} \rightarrow \text{next};$

$\text{stop} = 4$

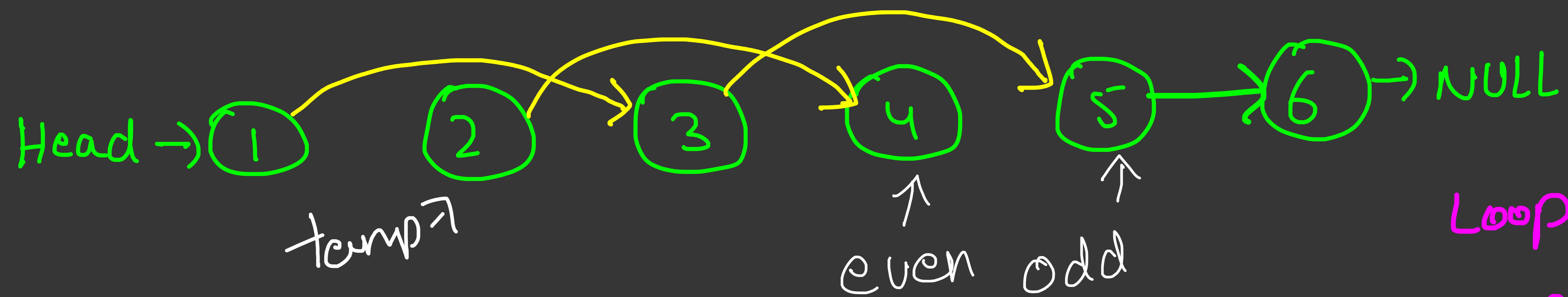
$=$

$=$

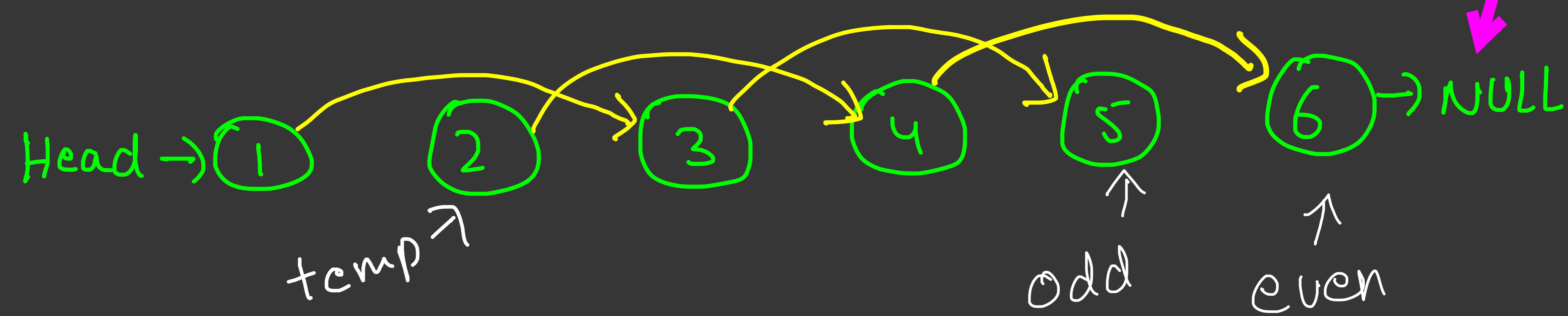


Step.5 Repeat Step ②, ③ & ④

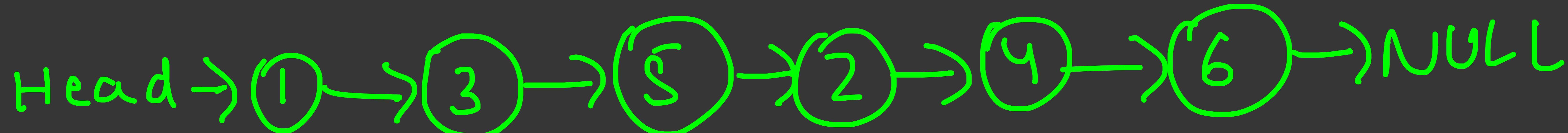
until ($\text{even} \neq \text{NULL}$ & $\text{even} \rightarrow \text{next} \neq \text{NULL}$)



Loop terminates
 $\text{even} \rightarrow \text{next} = \text{NULL}$



Step.6 $\text{Head} \rightarrow \text{next} = \text{temp}$
return `Head`;



Source Code:-

```
1  class Solution {
2  public:
3      ListNode* oddEvenList(ListNode* head) {
4          if(head==NULL or head->next==NULL) return head;
5          ListNode*odd=head, *even=head->next, *temp=even;
6          while(even!=NULL and even->next!=NULL){
7              odd->next = even->next;
8              odd=odd->next;
9              even->next=odd->next;
10             even=even->next;
11         }
12         odd->next=temp;
13         return head;
14     }
15 }
```

Don't Forget

to

Upvote 

