Reverse alternate nodes in Link List

Given a linked list, you have to perform the following task:

1. Extract the alternative nodes starting from second node.
2. Reverse the extracted list.
3. Append the extracted list at the end of the original list.

Note: Try to solve the problem without using any extra memory.

Example 1:

```
Input:
LinkedList = 10->4->9->1->3->5->9->4
Output:
10 9 3 9 4 5 1 4
Explanation:
Alternative nodes in the given linked list are 4,1,5,4.
Reversing the alternative nodes from the given list,
and then appending them to the end of the list results
in a list 10->9->3->9->4->5->1->4.
```

Example 2:

```
Input:
LinkedList = 1->2->3->4->5
Output:
1 3 5 4 2
Explanation:
Alternative nodes in the given linked list are 2 and 4.
Reversing the alternative nodes from the given list,
and then appending them to the end of the list results
in a list 1->3->5->4->2.
```

Your Task:
You don't have to read input or print anything. Your task is to complete the function **rearrange**() which takes the head of the linked list as input and rearranges the list as required.

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(1)

Constraints:
1 <= N <= 10^5
0 <= Node_value <= 10^9