Put Even position nodes after Odd position nodes★

Given a singly linked list, group all odd position nodes together followed by the even position nodes.

You should try to do it in place. The program should run in O(1) space complexity and O(nodes) time complexity.

Sample Test Cases:

Input: 1→ 2→ 3→ 4→ 5→ NULL

Output: 1→ 3→ 5→ 2→ 4→ NULL

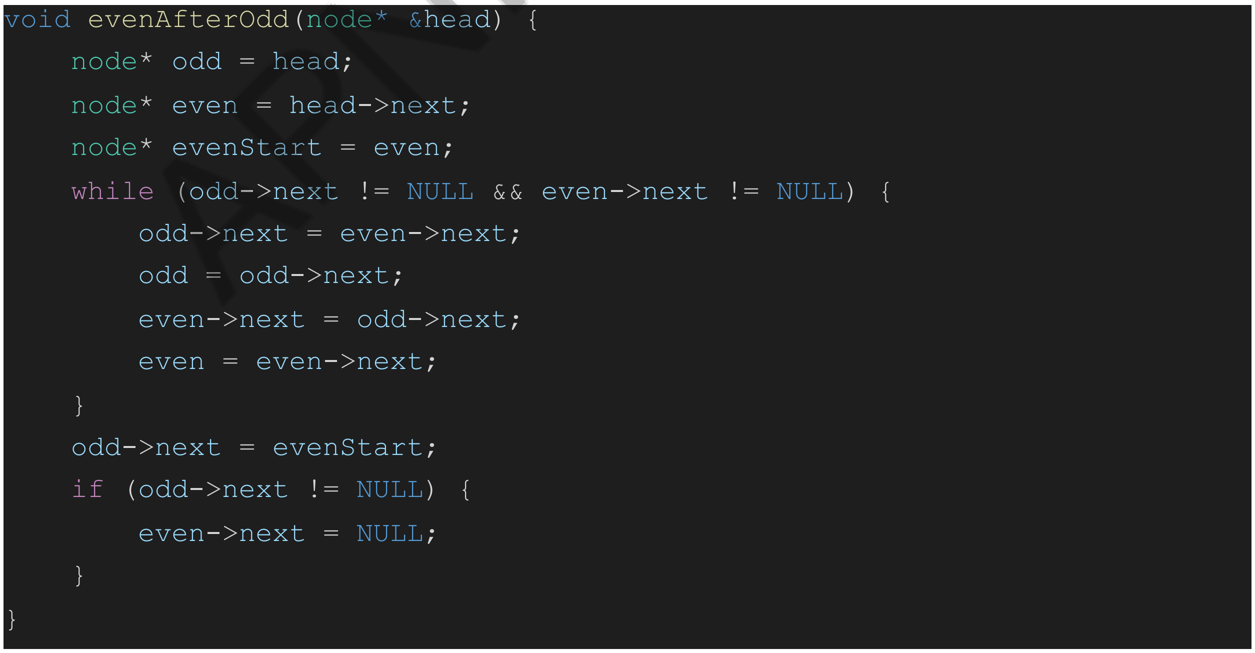
Input: 2 → 1→ 3→ 5→ 6→ 4→ 7→ NULL

Output: 2 → 3→ 6→ 7→ 1→ 5→ 4→ NULL

Idea:

- Initialize odd to head, even to head->next
- Insert odd nodes to ‘odd’ LL, and even nodes to ‘even’ LL
    - Even->next will be an odd node, so it should be joined with odd LL.
    - Odd->next will be an even node, so it should be joined with even LL.
- Finally, merge the two Linked List.

Code:



```
void evenAfterOdd(node* &head) {
    node* odd = head;
    node* even = head->next;
    node* evenStart = even;
    while (odd->next != NULL && even->next != NULL) {
        odd->next = even->next;
        odd = odd->next;
        even->next = odd->next;
        even = even->next;
    }
    odd->next = evenStart;
    if (odd->next != NULL) {
        even->next = NULL;
    }
}
```