Delete duplicate-value nodes from a sorted linked list

This challenge is part of a tutorial track byMyCodeSchool

You're given the pointer to the head node of a sorted linked list, where the data in the nodes is in ascending order. Delete as few nodes as possible so that the list does not contain any value more than once. The given head pointer may be null indicating that the list is empty.

For now do not be concerned with the memory deallocation. In common abstract data structure scenarios, deleting an element might also require deallocating the memory occupied by it. For an initial intro to the topic of dynamic memory please consult: http://www.cplusplus.com/doc/tutorial/dynamic/

Input Format
You have to complete the `Node* RemoveDuplicates(Node* head)` method which takes one argument - the head of the sorted linked list. You should NOT read any input from stdin/console.

Output Format
Delete as few nodes as possible to ensure that no two nodes have the same data. Adjust the `next` pointers to ensure that the remaining nodes form a single sorted linked list. Then `return` the head of the sorted updated linked list. Do NOT print anything to stdout/console.

Sample Input

```
1 -> 1 -> 3 -> 3 -> 5 -> 6 -> NULL
NULL
```

Sample Output

```
1 -> 3 -> 5 -> 6 -> NULL
NULL
```

Explanation
1. 1 and 3 are repeated, and are deleted.
2. Empty list remains empty.