Remove Duplicates from Sorted List

Question: Given a sorted linked list, delete all duplicates such that each element appear only once.

For example:

Given 1->1->2, return 1->2.

Given 1->1->2->3->3, return 1->2->3.

Solutions:

```
class ListNode:
    def __init__(self, x):
        self.val = x
        self.next = None


class Solution:
    # @param head, a ListNode
    # @return a ListNode
    def deleteDuplicates(self, head):
        if head == None or head.next == None:
            return head
        p = head
        while p.next:
            if p.val == p.next.val:
                p.next = p.next.next
            else:
                p = p.next
        return head
    def printll(self, node):
        while node:
            print ( node.val )
            node = node.next

if __name__ == '__main__':
    ll1, ll1.next, ll1.next.next = ListNode(2), ListNode(2), ListNode(5)
    Solution().printll( Solution().deleteDuplicates(ll1) )
```