

```

class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next

def deleteDuplicates(head):
    # If the list is empty or has only one node, return it
    if not head or not head.next:
        return head

    current = head

    # Iterate through the list
    while current and current.next:
        # If current node value equals the next node value, skip the next
node
        if current.val == current.next.val:
            current.next = current.next.next
        else:
            current = current.next # Move to the next node

    return head

# Helper function to convert a list into a linked list
def create_linked_list(arr):
    if not arr:
        return None
    head = ListNode(arr[0])
    current = head
    for val in arr[1:]:
        current.next = ListNode(val)
        current = current.next
    return head

# Helper function to print the linked list
def print_linked_list(head):
    result = []
    current = head
    while current:
        result.append(current.val)
        current = current.next
    print(result)

# Example 1:
head1 = create_linked_list([1, 1, 2])
new_head1 = deleteDuplicates(head1)
print_linked_list(new_head1) # Output: [1, 2]

# Example 2:
head2 = create_linked_list([1, 1, 2, 3, 3])
new_head2 = deleteDuplicates(head2)
print_linked_list(new_head2) # Output: [1, 2, 3]

```

Output:

[1, 2]

[1, 2, 3]