class Solution {

public:

ListNode\* deleteDuplicates(ListNode\* head) {

if (head == nullptr || head->next == nullptr) {

return head; // Empty or single node, no duplicates

}

ListNode\* dummy = new ListNode(0); // Dummy node for easier handling of head

dummy->next = head;

ListNode\* prev = dummy;

ListNode\* curr = head;

while (curr != nullptr) {

bool isDuplicate = false;

while (curr->next != nullptr && curr->val == curr->next->val) {

isDuplicate = true;

curr = curr->next; // Skip all duplicates

}

if (isDuplicate) {

prev->next = curr->next; // Remove the entire duplicate group

} else {

prev = curr; // Move prev forward only if no duplicates were found

}

curr = curr->next;

}

ListNode\* result = dummy->next;

delete dummy;

return result;

}

};