#include

using namespace std;

struct ListNode { int val; ListNode\* next; ListNode(int x) : val(x), next(NULL) {} };

class Solution { public: ListNode\* deleteDuplicates(ListNode\* head) {

ListNode\* dummy = new ListNode(0);

dummy->next = head;

ListNode\* prev = dummy;

while (head != NULL) {

if (head->next != NULL && head->val == head->next->val) {

while (head->next != NULL && head->val == head->next->val) {

head = head->next;

}

prev->next = head->next;

} else {

prev = prev->next;

}

head = head->next;

}

return dummy->next;

}

};

void printList(ListNode\* head) { while (head != NULL) { cout << head->val << " -> "; head = head->next; } cout << "NULL" << endl; }

int main() {

ListNode\* head = new ListNode(1);

head->next = new ListNode(2);

head->next->next = new ListNode(3);

head->next->next->next = new ListNode(3);

head->next->next->next->next = new ListNode(4);

head->next->next->next->next->next = new ListNode(4);

head->next->next->next->next->next->next = new ListNode(5);

Solution sol;

ListNode\* result = sol.deleteDuplicates(head);

cout << "List after removing duplicates: ";

printList(result);

return 0;

}