1. Remove Duplicates from Sorted List II

Medium

Given a sorted linked list, delete all nodes that have duplicate numbers, leaving only *distinct* numbers from the original list.

**Example 1:**

Input: 1->2->3->3->4->4->5  
Output: 1->2->5

**Example 2:**

Input: 1->1->1->2->3  
Output: 2->3

**Solution**

Noting : in example 1, 3 and 4 are all removed

/\*\*  
 \* Definition for singly-linked list.  
 \* struct ListNode {  
 \* int val;  
 \* ListNode \*next;  
 \* ListNode(int x) : val(x), next(NULL) {}  
 \* };  
 \*/  
class Solution {  
public:  
 ListNode\* deleteDuplicates(ListNode\* head) {  
 ListNode \*res = new ListNode(0), \*p = res, \*q = head;  
 res->next = head;  
 //p = res, q = head;  
 while(q != NULL && q->next != NULL){  
 if(q->val != q->next->val){  
 p = p->next;  
 q = q->next;  
 }else{  
 while(q->next != NULL && q->val == q->next->val)q = q->next;  
 q = q->next;  
 p->next = q;  
 }  
 }  
 return res->next;  
 }  
};