82. Remove Duplicates from Sorted List II

Total Accepted: **73128** Total Submissions: **270336** Difficulty: **Medium**

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

For example,

Given 1->2->3->3->4->4->5, return 1->2->5.

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

```
/**
 * 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) {
        if(head==NULL) return head;
        ListNode dummy(INT_MIN);
        dummy.next = head;
        ListNode *prev = &dummy;
        ListNode *cur = head;
        while(cur != NULL)
        {
            bool duplicate = false;
            while(cur->next!=NULL && cur->val == cur->next->val)
            {
                duplicate = true;
                ListNode *temp = cur;
                cur = cur->next;
                delete temp;
            }
            //delete the last elem
            if(duplicate)
            {
                ListNode *temp = cur;
                cur = cur->next;
                delete temp;
                continue;
            }
            prev->next = cur;
            cur = cur->next;
            prev = prev->next;
        }
        prev->next = cur;
        return dummy.next;
    }
};
```