1. Partition List

Medium

Given a linked list and a value *x*, partition it such that all nodes less than *x* come before nodes greater than or equal to *x*.

You should preserve the original relative order of the nodes in each of the two partitions.

**Example:**

Input: head = 1->4->3->2->5->2, x = 3  
Output: 1->2->2->4->3->5

**Solution**

新建两个链表，尾插然后合并

/\*\*  
 \* Definition for singly-linked list.  
 \* struct ListNode {  
 \* int val;  
 \* ListNode \*next;  
 \* ListNode(int x) : val(x), next(NULL) {}  
 \* };  
 \*/  
class Solution {  
public:  
 ListNode\* partition(ListNode\* head, int x) {  
 if(head == NULL)return head;  
 ListNode \*head1 = new ListNode(0);  
 ListNode \*head2 = new ListNode(0);  
 ListNode \*p = head;  
 ListNode \*q1 = head1, \*q2 = head2, \*tail = q1;  
 while(p != NULL){  
 if(p->val < x){  
 addNum(q1, p->val);  
 q1 = q1->next;  
 tail = q1;  
 }else{  
 addNum(q2, p->val);  
 q2 = q2->next;  
 }  
 p = p->next;  
 }  
 tail->next = head2->next;  
 return head1->next;  
 }  
 void addNum(ListNode \*p, int x){  
 ListNode \*newnode = new ListNode(x);  
 p->next = newnode;  
 }  
};