|  |
| --- |
|  |
|  |  |

暑期算法训练报告

专业班级： 11J191

学 号： 20191000645

姓 名： 邓咏霖

指导老师： 张剑波

**中国地质大学地理与信息工程学院**

**2020年 8月**

**第1题**

【需求规格说明】

翻转一个链表

**样例 1:**

输入: 1->2->3->null

输出: 3->2->1->null

**样例 2:**

输入: 1->2->3->4->null

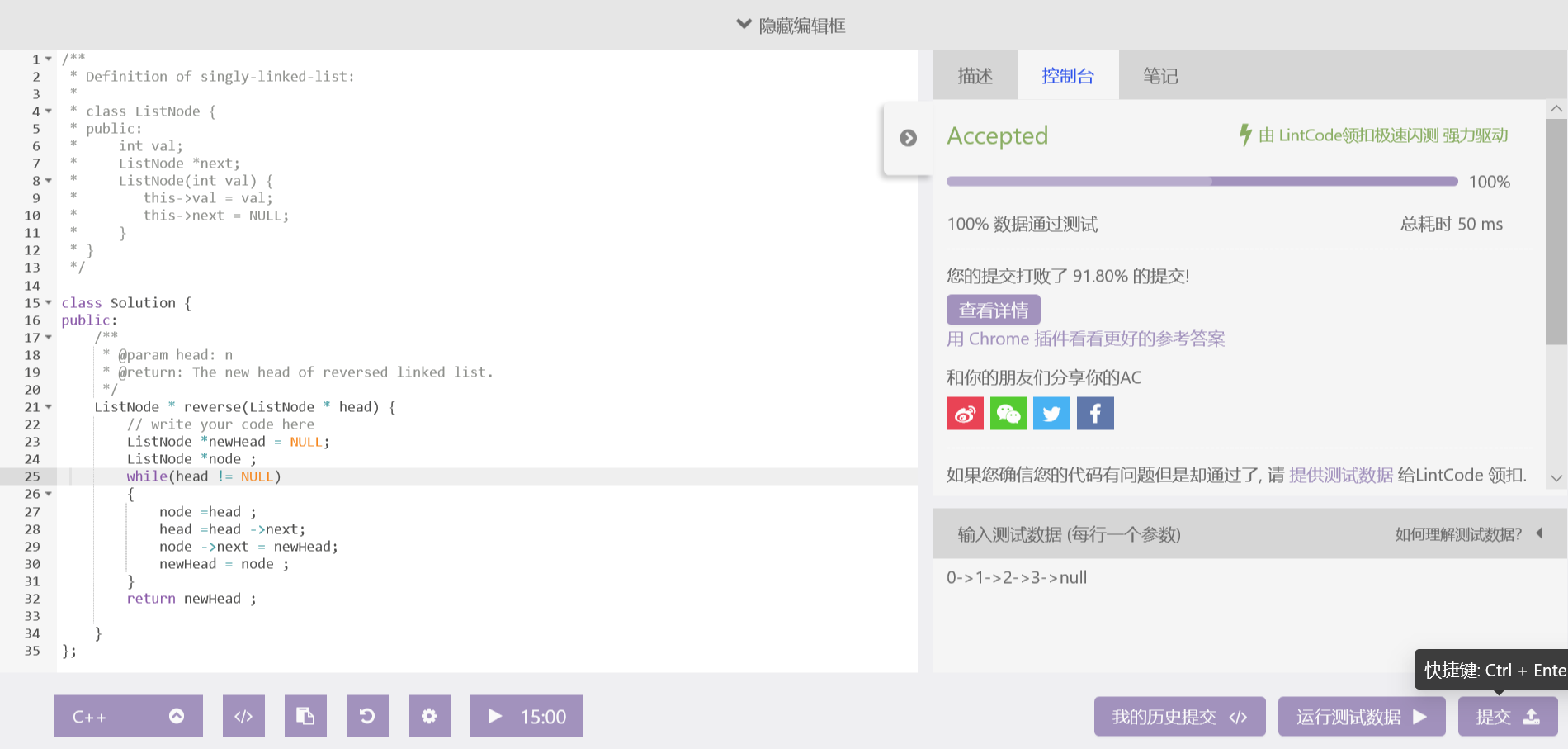
输出: 4->3->2->1->null

【算法设计】

**设计思想**：

对原链表做头删操作，然后再对新链表做头插，再利用循环迭代实现整个链表的反转

【附录】



**第2题**

【需求规格说明】

给定一个排序链表，删除所有重复的元素每个元素只留下一个。

**样例**

样例 1:

输入: null

输出: null

样例 2:

输入: 1->1->2->null

输出: 1->2->null

样例 3:

输入: 1->1->2->3->3->null

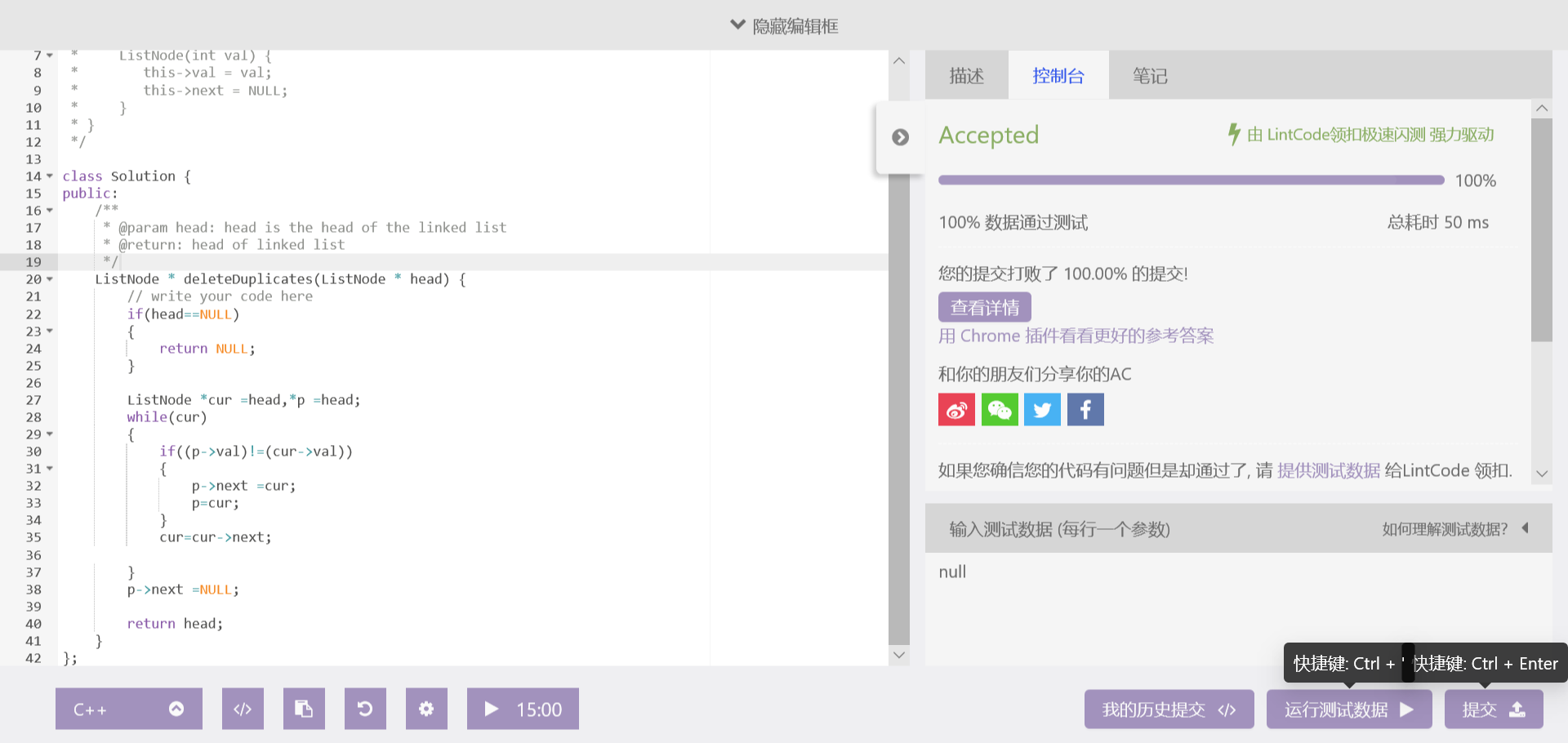
输出: 1->2->3->null

【算法设计】

**设计思想**：

1. 先分类讨论当链表为空的情况，返回NULL
2. 定义两个指针，分别为cur和p;用cur来跟踪链表中每次循环到的节点，用p来进行对重复元素的删除操作。跳出循环后，让p->next =NULL,完成对整个链表的删除。

【附录】



**第3题**

【需求规格说明】

将两个排序链表合并为一个新的排序链表

### 样例

样例 1:

输入: list1 = null, list2 = 0->3->3->null

输出: 0->3->3->null

样例2:

输入: list1 = 1->3->8->11->15->null, list2 = 2->null

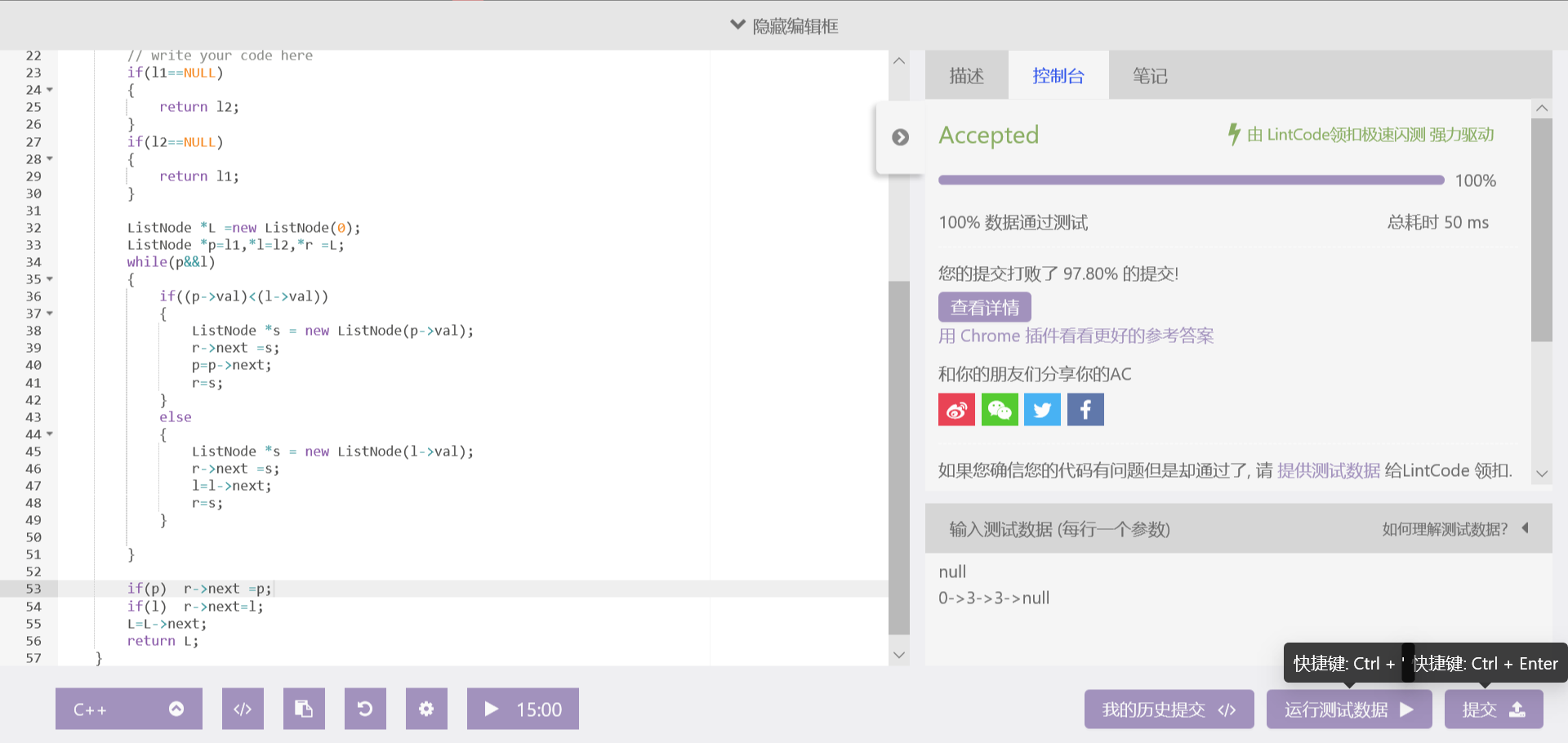
输出: 1->2->3->8->11->15->null

【算法设计】

**设计思想**：

1. 先讨论两种特殊情况：当l1或者l2为NULL的情况的时候
2. 创建一个新的头节点，利用循环每次插入两个链表中较小的数，直至其中一个链表迭代到表尾。
3. 判断哪个链表不为空，将这个链表剩下的节点插入到新的链表中。

【附录】



**第4题**

【需求规格说明】

找到单链表倒数第n个节点，保证链表中节点的最少数量为n。

### 样例

Example 1:

Input: list = 3->2->1->5->null, n = 2

Output: 1

Example 2:

Input: list = 1->2->3->null, n = 3

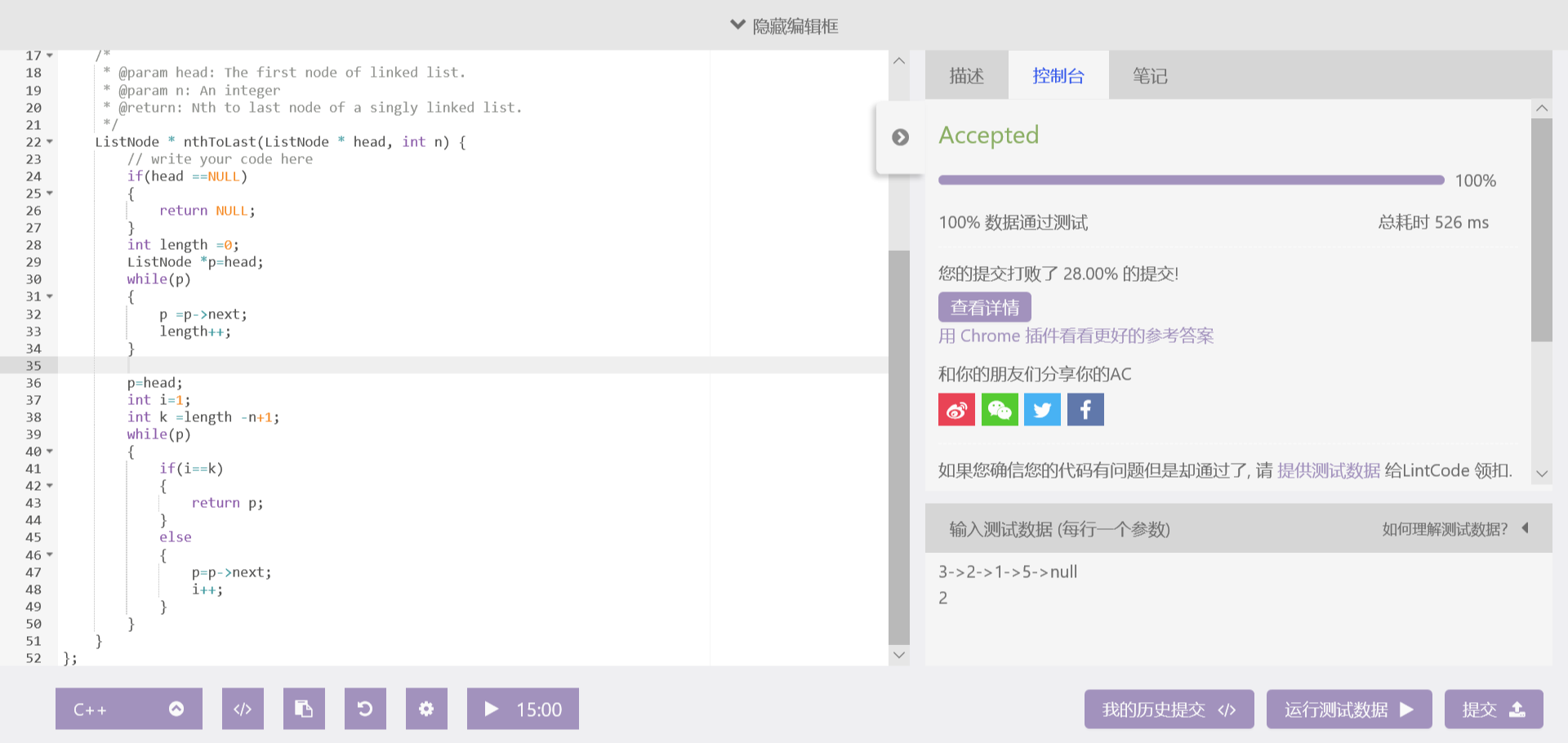
Output: 1

【算法设计】

**设计思想**：

1. 先讨论链表为空的情况，返回NULL
2. 定义一个新的指针p和一个int类型的变量length，对P进行迭代的同时对count实现加一，统计链表中节点的数量。
3. 倒数第N个元素正向数是length-n+1个元素，然后迭代找到这个元素后返回。

【附录】



**第5题**

【需求规格说明】

给定一个链表，删除链表中倒数第n个节点，返回链表的头节点。

### 样例

Example 1:

Input: list = 1->2->3->4->5->null， n = 2

Output: 1->2->3->5->null

Example 2:

Input: list = 5->4->3->2->1->null, n = 2

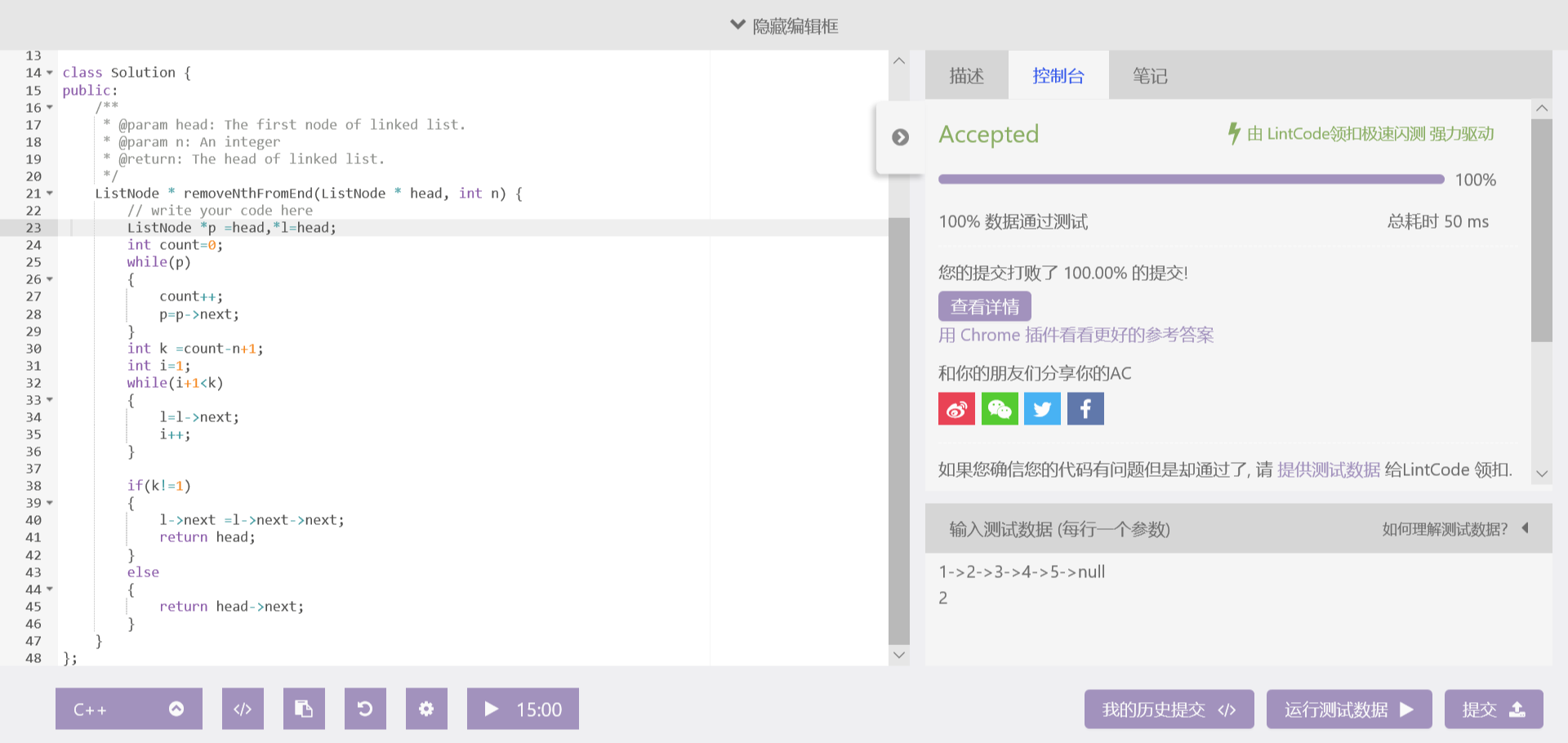
Output: 5->4->3->1->null

【算法设计】

**设计思想**：

1. 定义一个新的指针p指向head,定义一个int类型的变量count，用来统计链表中节点的数量，利用循环遍历实现。
2. 算出倒数第n个节点正向为第count-n+1个节点。
3. 定义一个新的指针l指向head,利用循环得到被删节点的前一个节点。
4. 分类进行讨论，如果被删节点是第一个节点，则直接返回head->next;如果不是，实现删除后返回头指针。

【附录】



**第6题**

【需求规格说明】

在链表中插入一个节点。

### 样例

**样例 1：**

输入：head = 1->4->6->8->null, val = 5

输出：1->4->5->6->8->null

**样例 2：**

输入：head = 1->null, val = 2

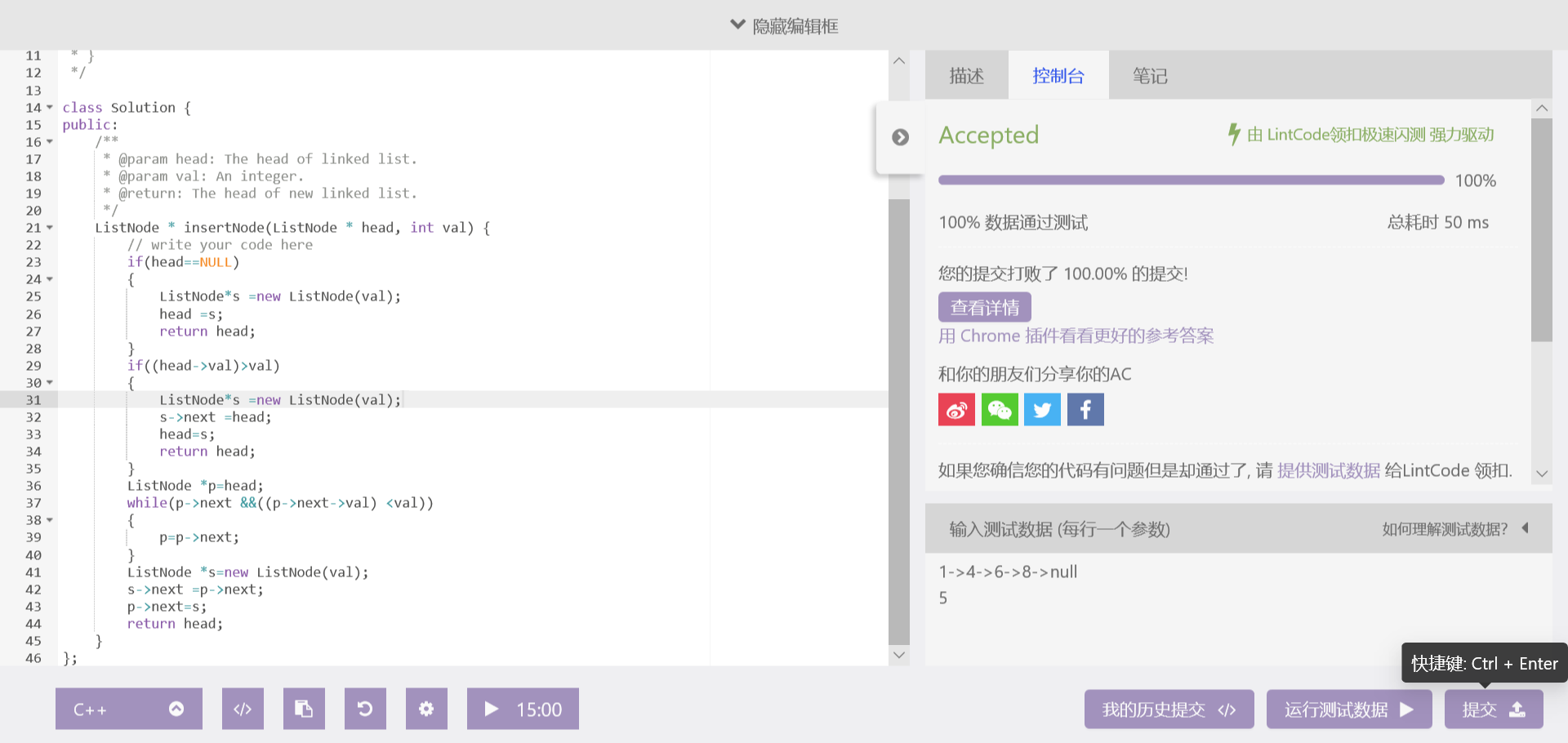
输出：1->2->null

【算法设计】

**设计思想**：

1. 先讨论链表为空的情况和讨论插入的节点比头节点数值小的情况
2. 定义一个新的指针p指向head，利用循环找到比插入节点的数值大的节点的前一个位置，进行插入；或者循环到表尾进行插入。
3. 返回头指针。

【附录】



**第7题**

【需求规格说明】

给定一个单链表中的一个等待被删除的节点(非表头或表尾)。请在在 O(1) 时间复杂度删除该链表节点。

### 样例

**样例 1：**

输入：

1->2->3->4->null

3

输出：

1->2->4->null

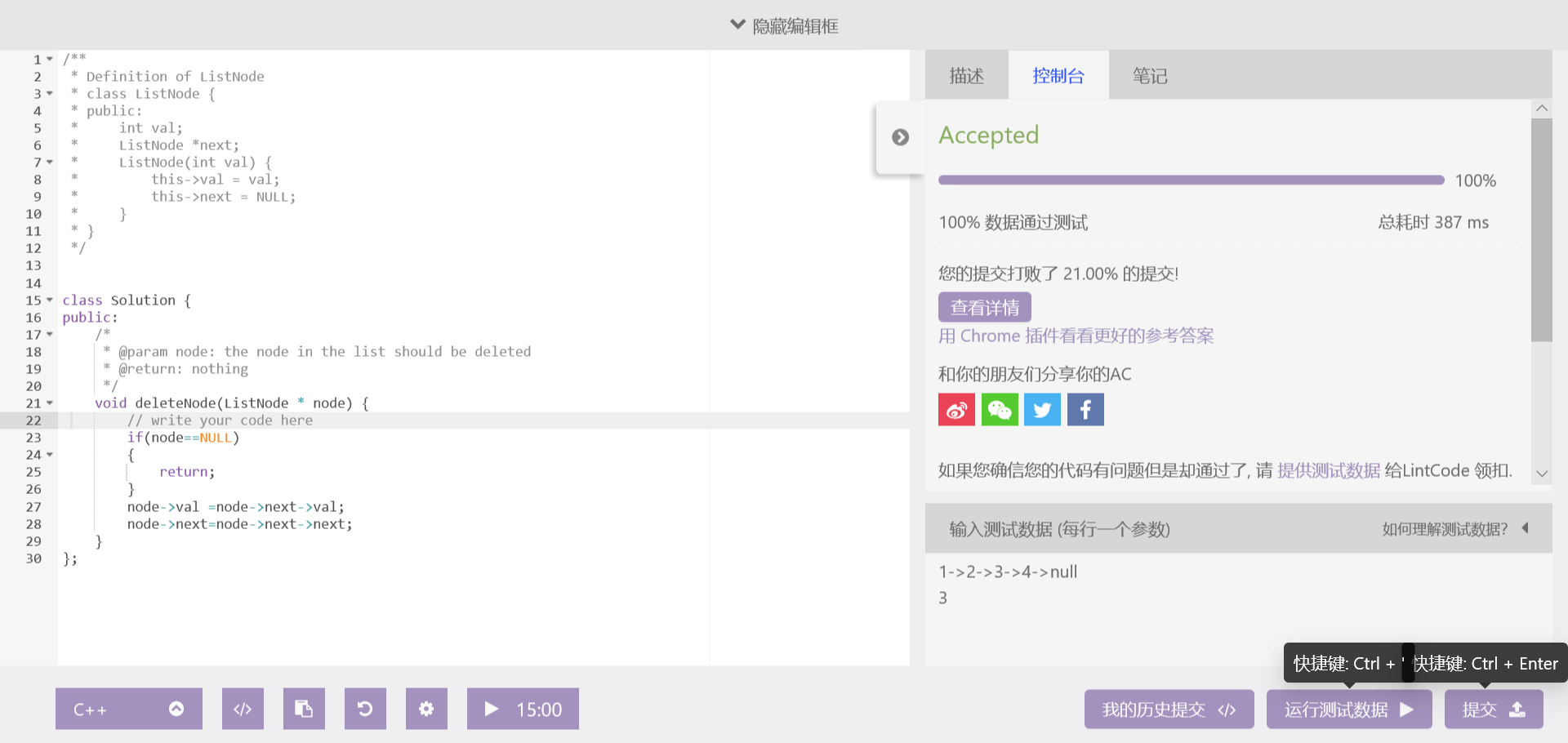
【算法设计】

**设计思想**：

如果利用循环找到删除的位置，再进行删除，时间复杂度为O（n）,而题目要求的是O（1），所以可以将下一个节点的数值赋值到当前节点，然后删除下一个节点。

还应该讨论当node==NULL的情况，则应该直接返回。

【附录】



**第8题**

【需求规格说明】

删除链表中等于给定值 val 的所有节点。

### 样例

**样例 1：**

输入：head = 1->2->3->3->4->5->3->null, val = 3

输出：1->2->4->5->null

**样例 2：**

输入：head = 1->1->null, val = 1

输出：null

【算法设计】

**设计思想**：

删除链表中指定值的节点，可以利用循环找到该节点的前一个节点的位置，然后进行删除。但如果要删除的节点为第一个节点，则删除的操作会有所不同，可以为链表增加一个头节点，从而使得删除的操作一样。返回的时候，直接返回头节点的地址域。

【附录】



**第9题**

【需求规格说明】

给出两个链表形式表示的数字,写一个函数得到这两个链表相乘乘积。

### 样例

**样例 1:**

输入：9->4->6->null,8->4->null

输出：79464

解释：946\*84=79464

**样例 2:**

输入：3->2->1->null,1->2->null

输出：3852

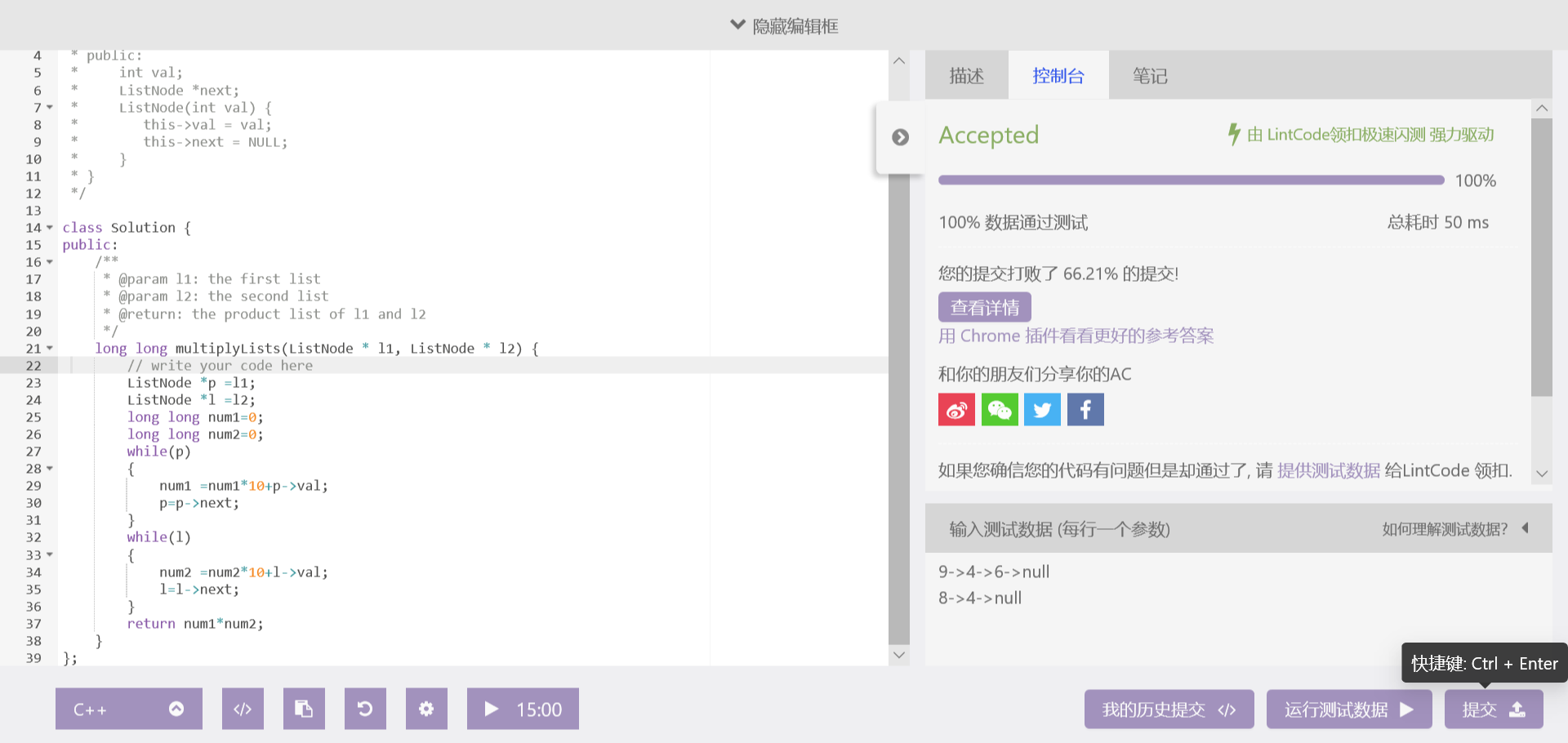
解释：321\*12=3852

【算法设计】

**设计思想**：

利用循环，取出l1,l2中的数值，返回两个数的乘积

【附录】



**第10题**

【需求规格说明】

给出一个链表，并将链表的值以**倒序**存储到数组中。

### 样例

**样例1**

输入: 1 -> 2 -> 3 -> null

输出: [3,2,1]

**样例2**

输入: 4 -> 2 -> 1 -> null

输出: [1,2,4]

### 注意事项

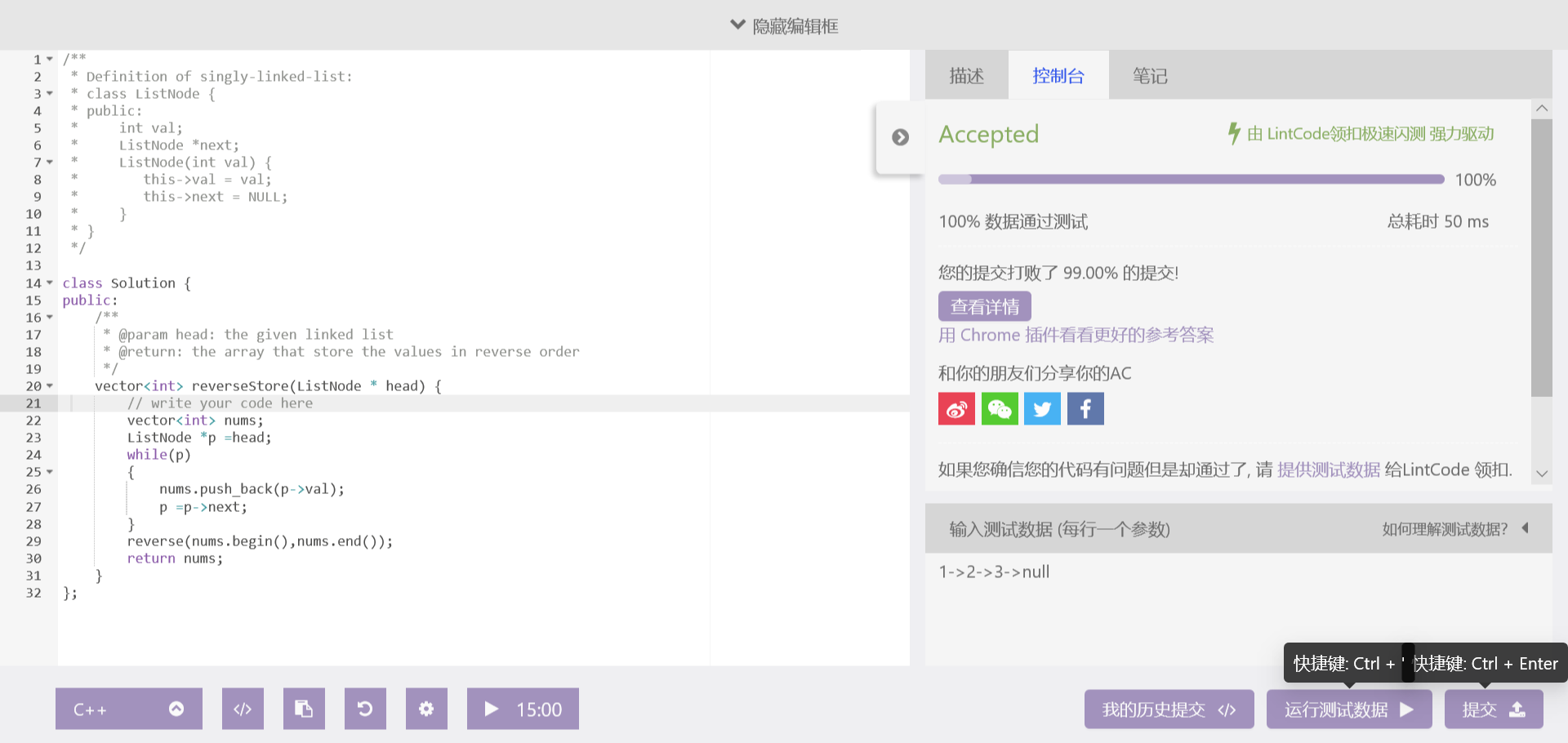
* 您**不能改变**原始链表的结构。
* ListNode 有两个成员变量：ListNode.val 和 ListNode.next

【算法设计】

**设计思想**：

利用循环，将链表中节点的数据存放到vector中，然后利用STL中的reverse函数反转数组。

【附录】



**第11题**

【需求规格说明】

给定一个带有头结点 head 的非空单链表，返回链表的中间结点。

如果有两个中间结点，则返回第二个中间结点。

### 样例

**样例 1:**

输入：1->2->3->4->5->null

输出：3->4->5->null

**样例 2:**

输入：1->2->3->4->5->6->null

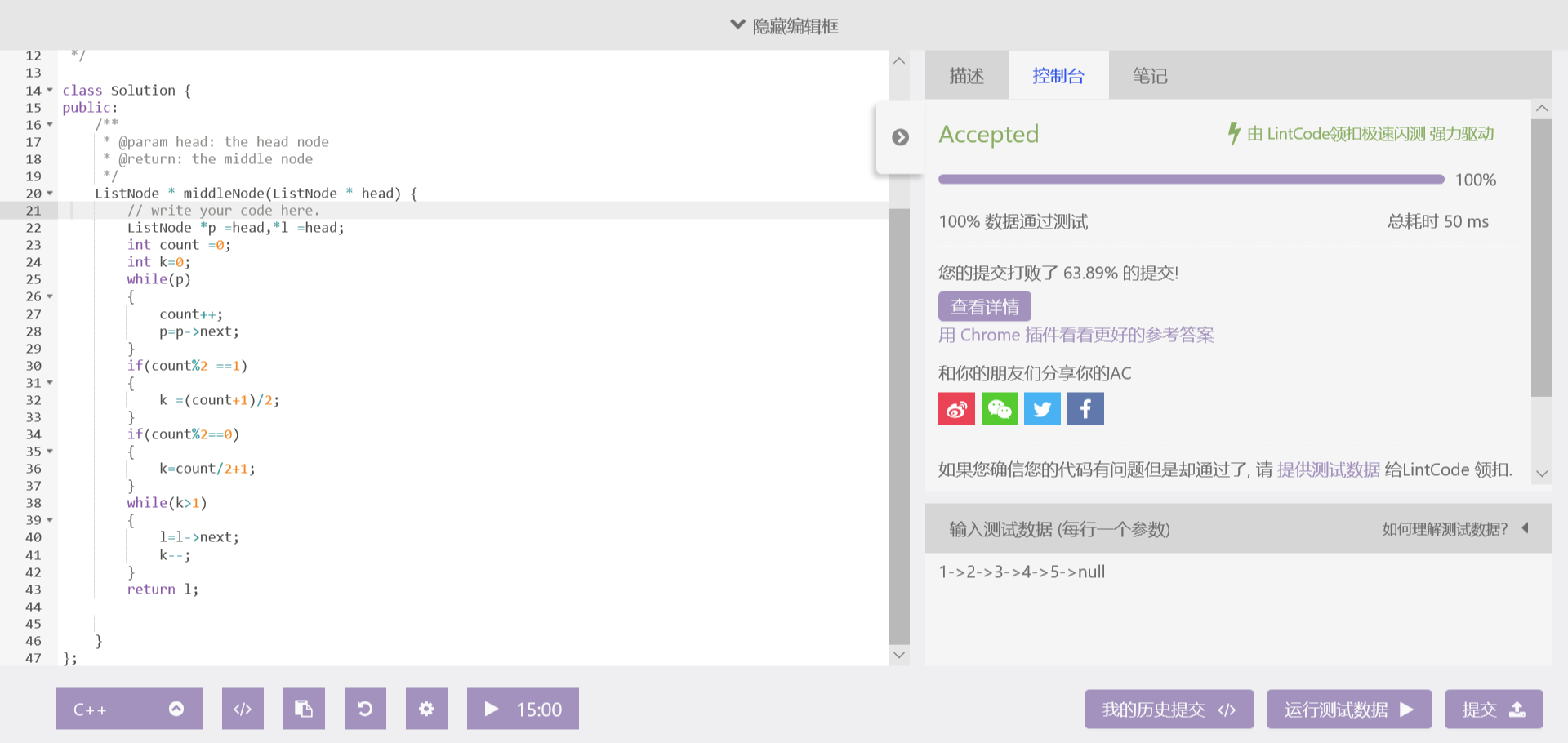
输出：4->5->6->null

【算法设计】

**设计思想**：

1. 定义一个新指针p指向head,定义一个int类型的变量count，初始化为0，遍历链表，指针后移的同时使count加一，得到链表中节点的数量。
2. 分情况得到，当count为奇数和偶数的时候中间节点在第几个位置。
3. 遍历找到中间节点的位置后返回。

【附录】



**第12题**

【需求规格说明】

给一个链表，两两交换其中的节点，然后返回交换后的链表。

### 样例

**样例 1：**

输入：1->2->3->4->null

输出：2->1->4->3->null

**样例 2：**

输入：5->null

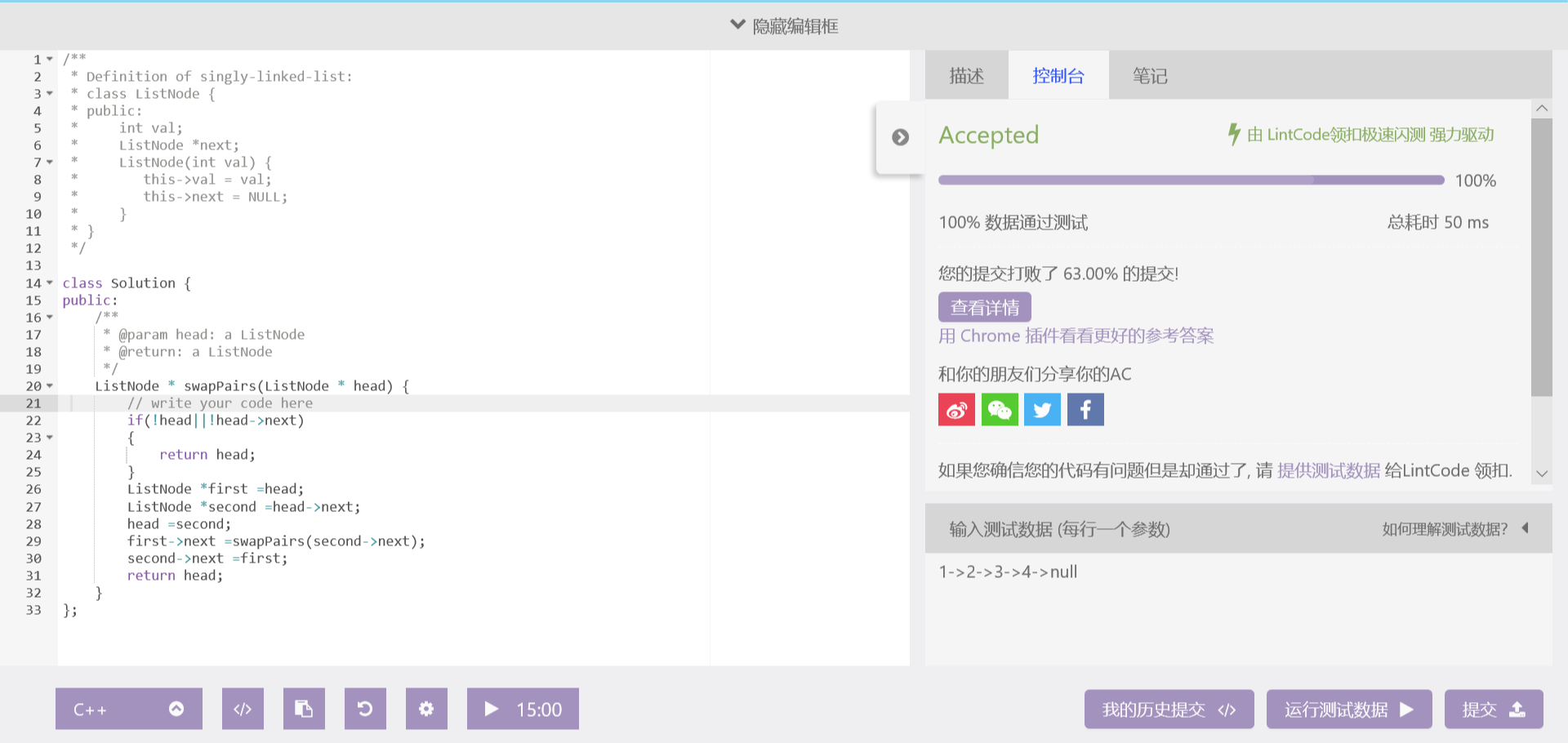
输出：5->null

【算法设计】

**设计思想**：

1. 利用递归实现对整个链表的节点两两交换
2. 先分类讨论当链表为空或者只有一个节点的情况，直接返回头节点

【附录】



**第13题**

【需求规格说明】

给定一个排序的整数数组（升序）和一个要查找的整数target，用O(logn)的时间查找到target第一次出现的下标（从0开始），如果target不存在于数组中，返回-1。

### 样例

样例 1:

输入:[1,4,4,5,7,7,8,9,9,10]，1

输出: 0

样例解释:

第一次出现在第0个位置。

样例 2:

输入: [1, 2, 3, 3, 4, 5, 10]，3

输出: 2

样例解释:

第一次出现在第2个位置

样例 3:

输入: [1, 2, 3, 3, 4, 5, 10]，6

输出: -1

样例解释:

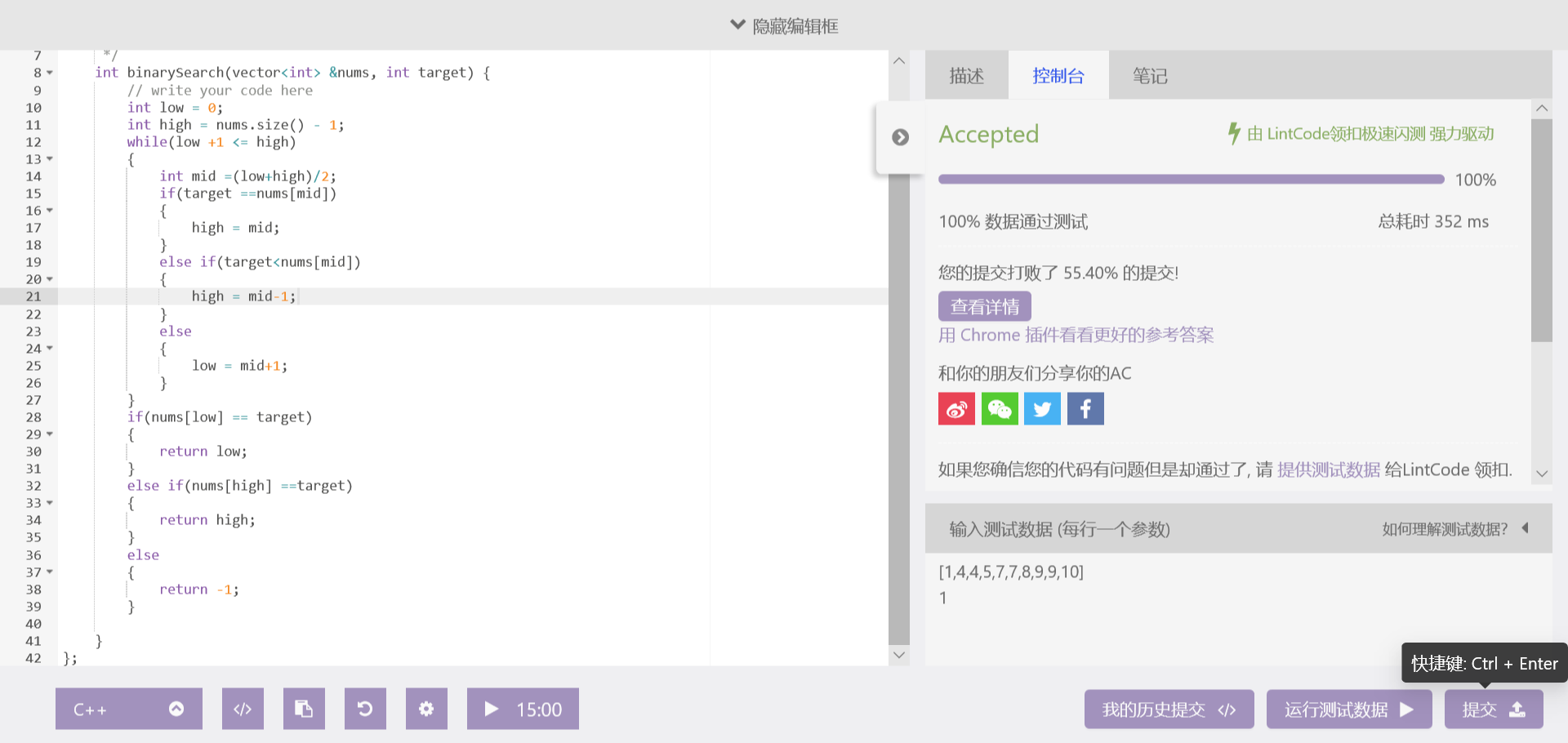
没有出现过6， 返回-1

【算法设计】

**设计思想**：

利用二分法对数据进行查找。本题重点是当出现重复的数字的时候，返回的是第一次出现的位置。所以循环结束的条件应该是二分到只有两个元素的情况，然后跳出循环，判断看哪个位置的值等于要查找的值，则返回该索引。

【附录】



**第14题**

【需求规格说明】

给定一个**旋转**排序数组，在原地恢复其排序。（升序）

### 样例

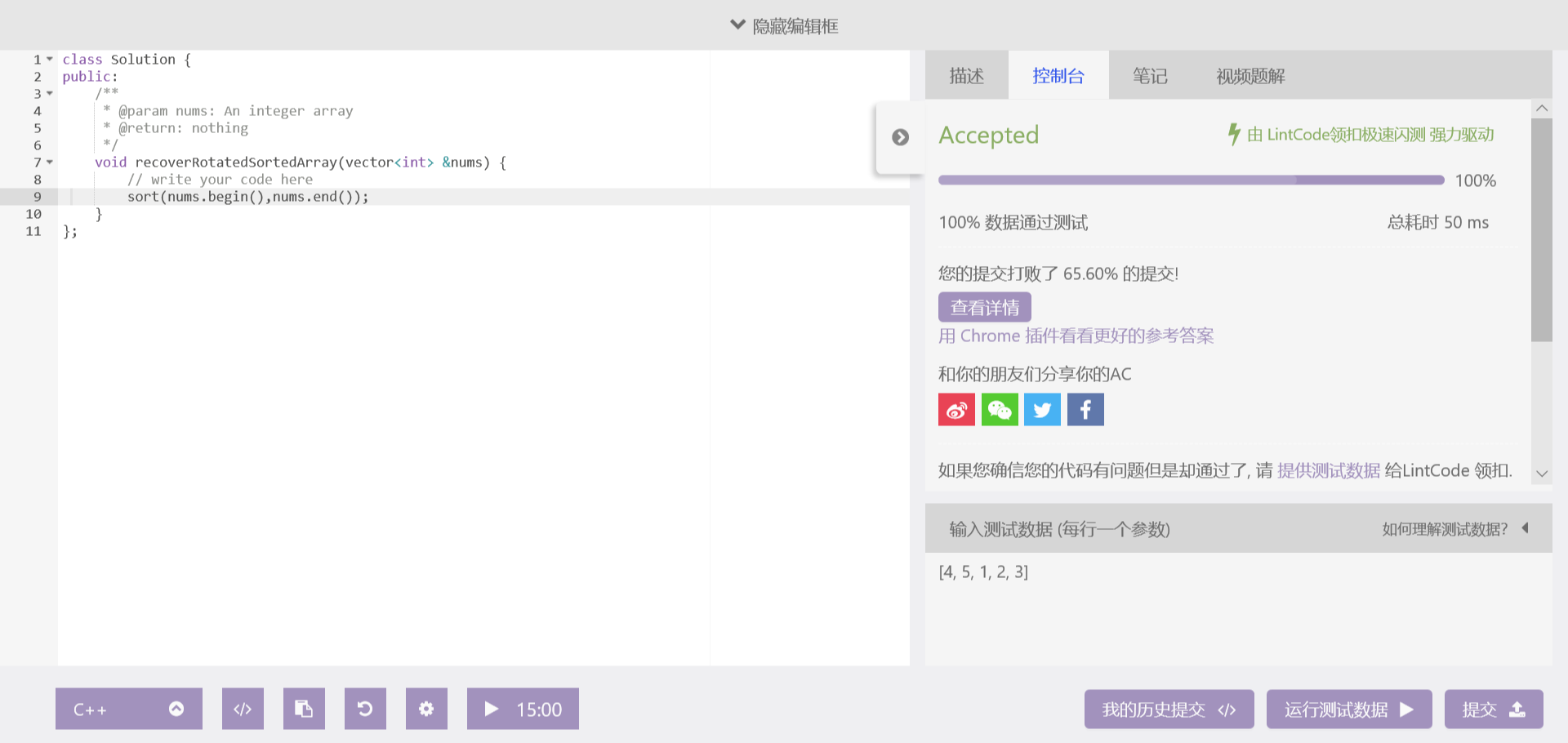
**Example1:**  
[4, 5, 1, 2, 3] -> [1, 2, 3, 4, 5]  
**Example2:**  
[6,8,9,1,2] -> [1,2,6,8,9]

【算法设计】

**设计思想**：

直接利用STL中的sort函数对vector进行排序。

【附录】



**第15题**

【需求规格说明】

给一个整数数组，找到两个数使得他们的和等于一个给定的数 target。

你需要实现的函数twoSum需要返回这两个数的下标, 并且第一个下标小于第二个下标。注意这里下标的范围是 0 到 n-1。

### 样例

Example1:

给出 numbers = [2, 7, 11, 15], target = 9, 返回 [0, 1].

Example2:

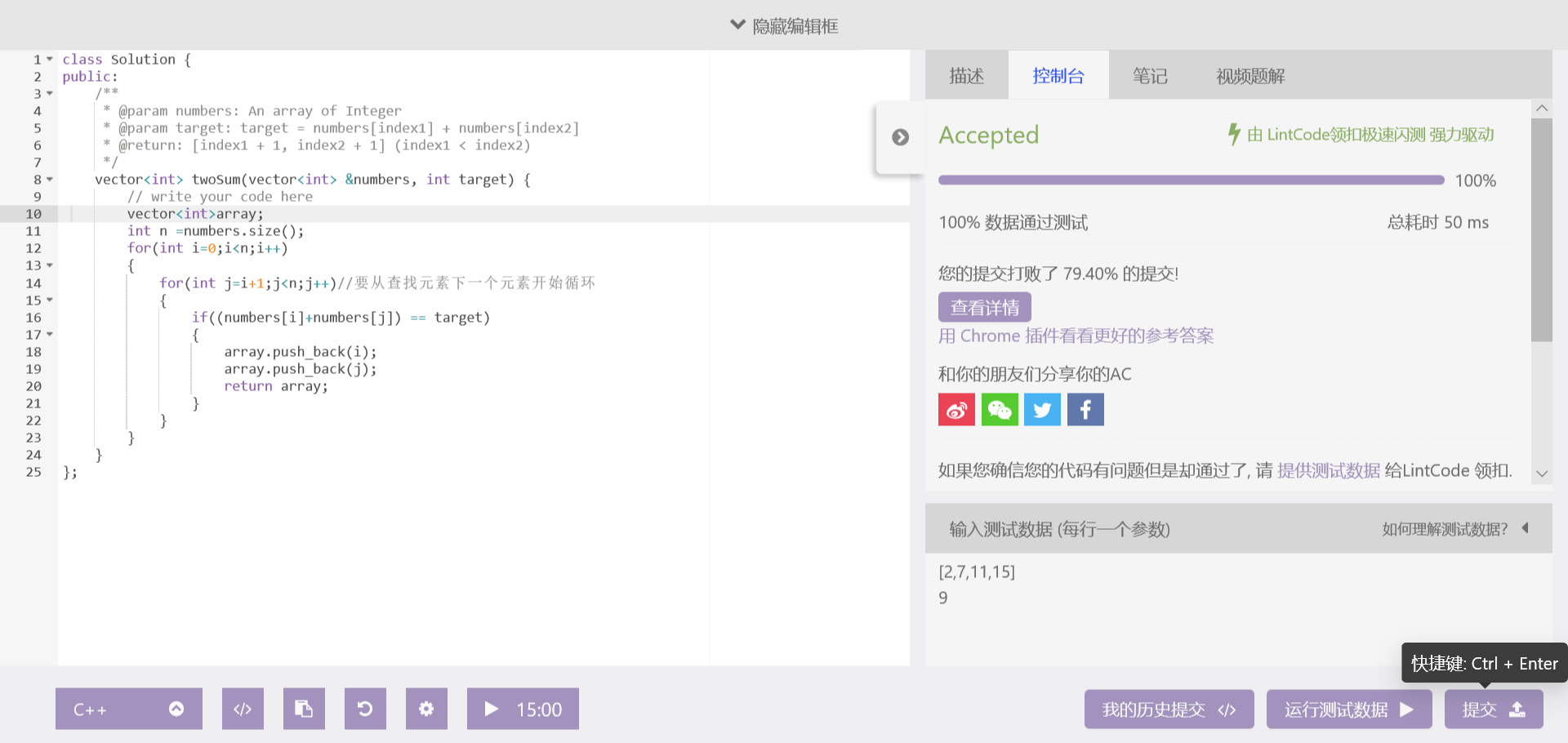
给出 numbers = [15, 2, 7, 11], target = 9, 返回 [1, 2].

【算法设计】

**设计思想**：

利用两个循环嵌套来查找两个数的和是否等于给定的数，找到后插入到vector中，然后返回。

【附录】



**第16题**

【需求规格说明】

给定一个排序数组和一个目标值，如果在数组中找到目标值则返回索引。如果没有，返回到它将会被按顺序插入的位置。

你可以假设在数组中无重复元素。

### 样例

**[1,3,5,6]**，5 → 2

**[1,3,5,6]**，2 → 1

**[1,3,5,6]**， 7 → 4

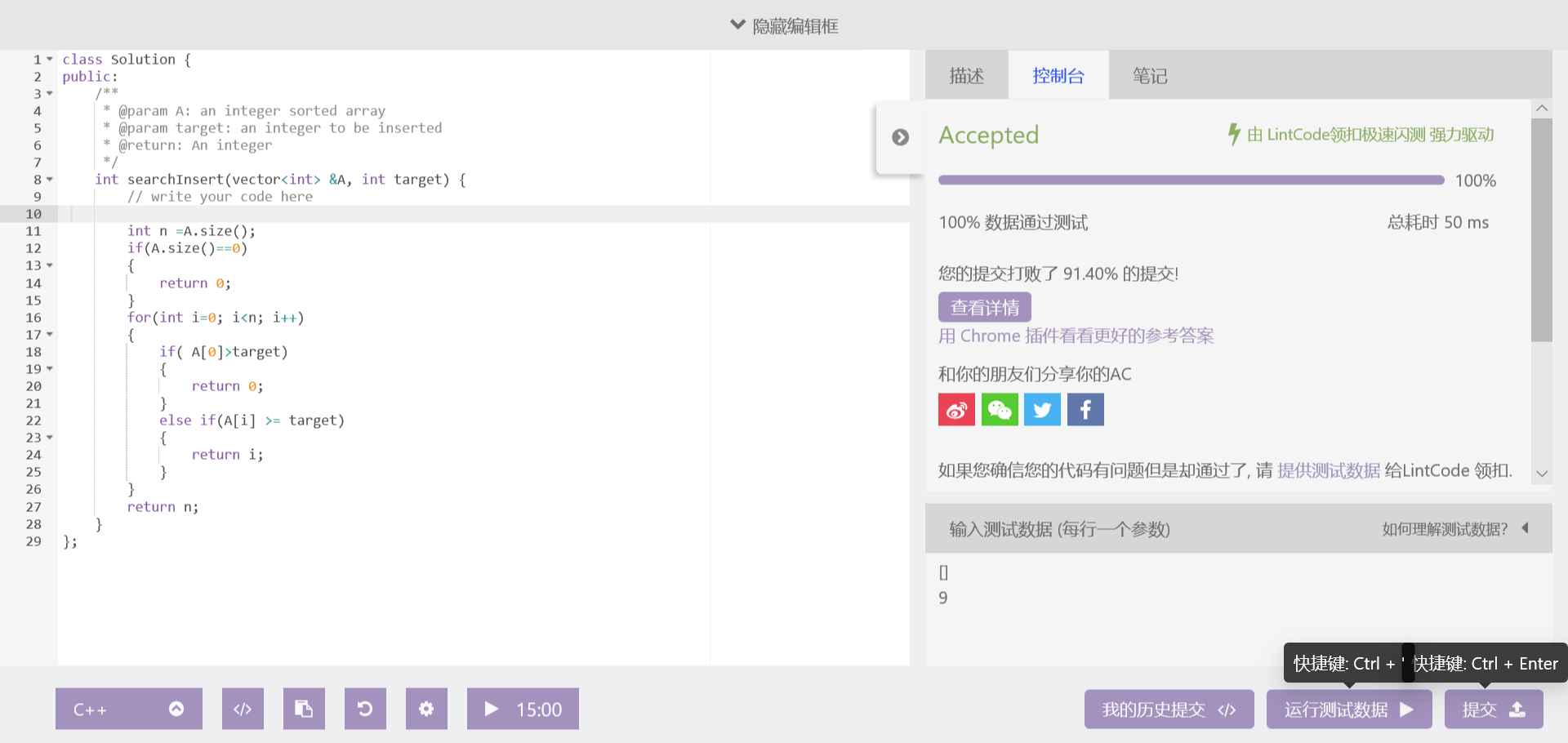
**[1,3,5,6]**，0 → 0

【算法设计】

**设计思想**：

1. 先讨论当数组没有元素的情况，返回0
2. 利用循环寻找插入或者相等的位置，如果数组第一个元素大于目标值，直接返回0；当循环的索引值等于或者大于目标值的时候，即找到要返回的索引值，然后进行返回。

【附录】



**第17题**

【需求规格说明】

合并两个排序的整数数组A和B变成一个新的数组。

### 样例

**样例 1:**

输入：[1, 2, 3] 3 [4,5] 2

输出：[1,2,3,4,5]

解释:

经过合并新的数组为[1,2,3,4,5]

**样例 2:**

输入：[1,2,5] 3 [3,4] 2

输出：[1,2,3,4,5]

解释：

经过合并新的数组为[1,2,3,4,5]

### 注意事项

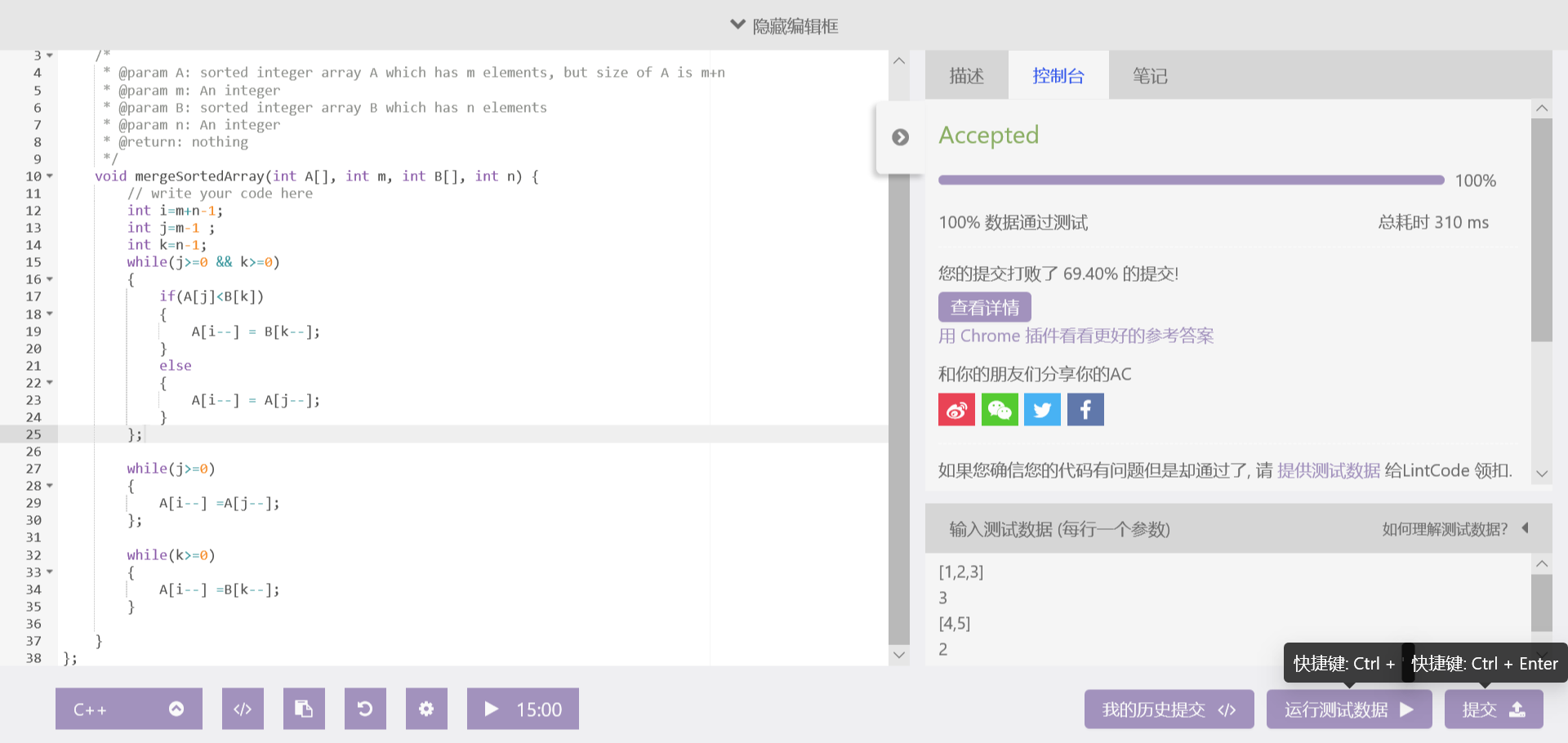
你可以假设A具有足够的空间（A数组的大小大于或等于m+n）去添加B中的元素。

【算法设计】

**设计思想**：

对两个已排好序的数组进行归并，利用循环迭代两个数组中的元素，将更小的数放入新数组中，直至其中一个数组循环到最后。判断哪个数组没有全部迭代，将该数组剩下的元素添加到新数组中。然后将新数组中的元素赋值到A数组中，进行返回。

【附录】



**第18题**

【需求规格说明】

给定一个未排序的整数数组，找到其中位数。

中位数是排序后数组的中间值，如果数组的个数是偶数个，则返回排序后数组的第N/2个数。

### 样例

**样例 1:**

输入：[4, 5, 1, 2, 3]

输出：3

解释：

经过排序，得到数组[1,2,3,4,5]，中间数字为3

**样例 2:**

输入：[7, 9, 4, 5]

输出：5

解释：

经过排序，得到数组[4,5,7,9]，第二个(4/2)数字为5

### 挑战

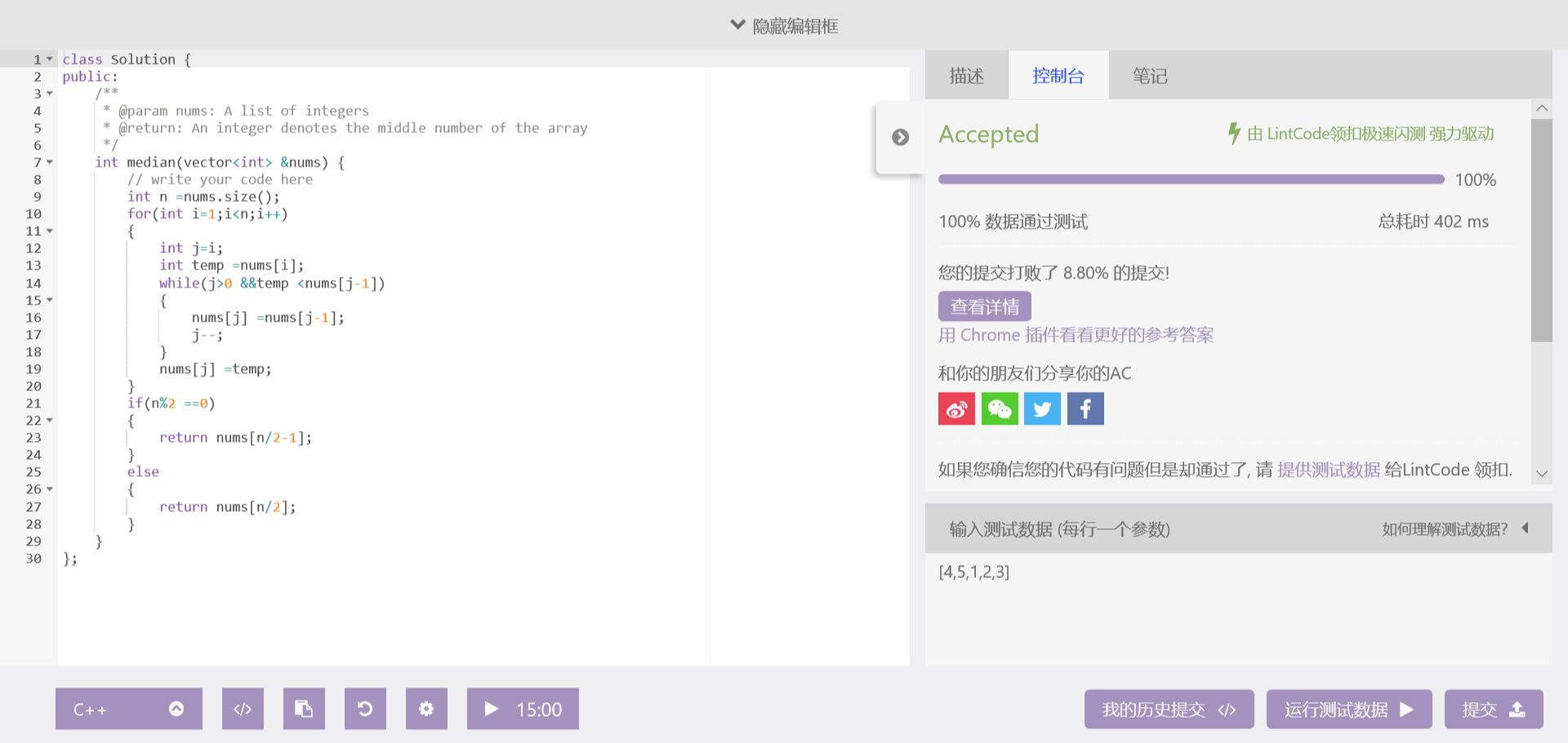
时间复杂度为O(n)

【算法设计】

**设计思想**：

先对数组进行排序，然后返回其中位数。如果数组元素个数n为奇数，则返回第（n+1）/2个元素，如果数组元素个数n为偶数，则返回第n/2个元素。

【附录】



**第19题**

【需求规格说明】

给定一个排序数组，在原数组中“删除”重复出现的数字，使得每个元素只出现一次，并且返回“新”数组的长度。

不要使用额外的数组空间，必须在不使用额外空间的条件下原地完成。

### 样例

**样例 1:**

输入: []

输出: 0

**样例 2:**

输入: [1,1,2]

输出: 2

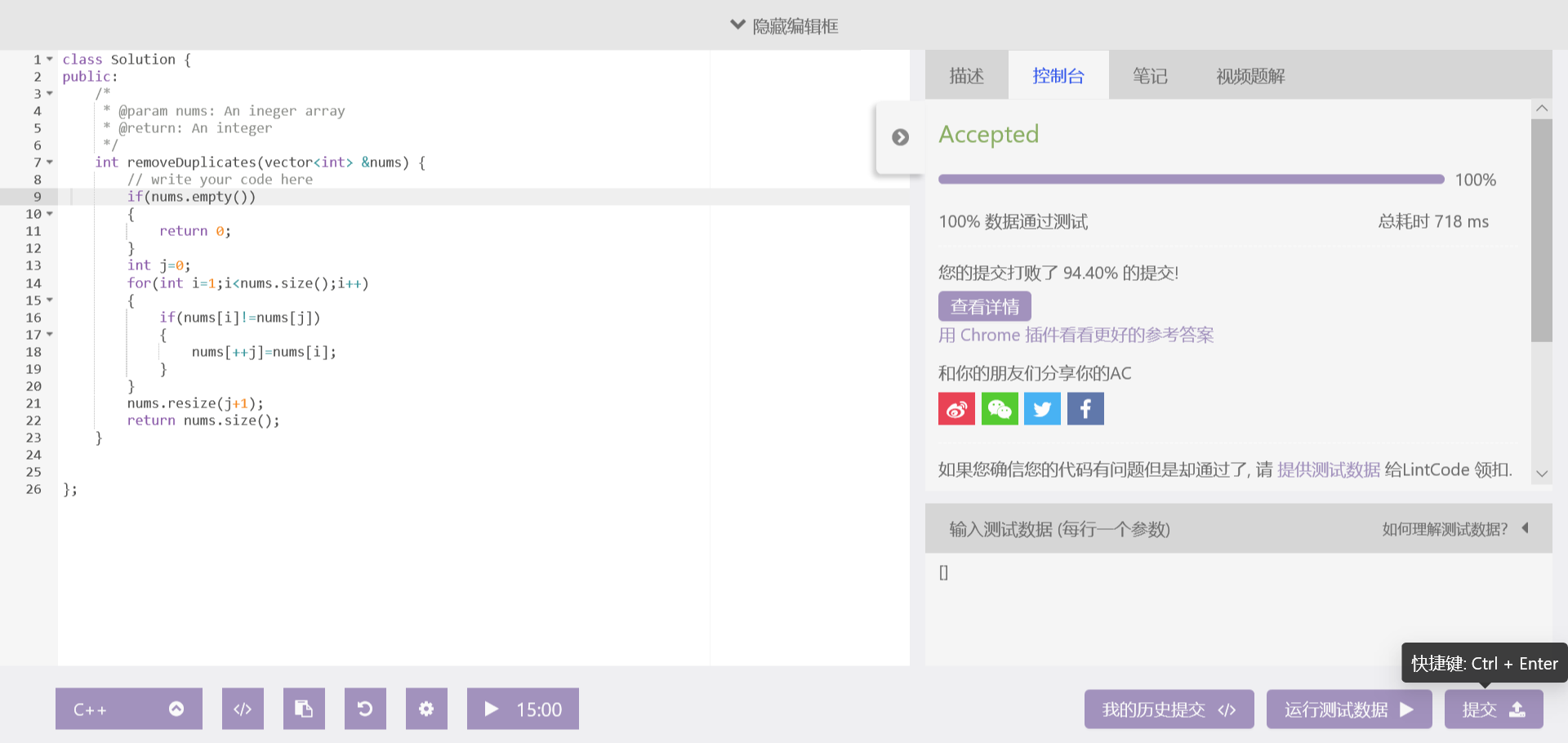
解释: 数字只出现一次的数组为: [1,2]

【算法设计】

**设计思想**：

1. 先讨论数组元素为0的情况，返回0
2. 定义两个int类型的变量，i初始化为1，用来遍历数组：j初始化为0，用来将不重复的数值覆盖到数组中。当前后两个元素不一样的时候，实现覆盖。然后重新设定数组的大小。

【附录】



**第20题**

【需求规格说明】

给你一个排序数组，删除其中的重复元素，使得每个数字最多出现两次，返回新的数组的长度。  
如果一个数字出现超过2次，则这个数字最后保留两个。

### 样例

样例 1:

输入: []

输出: 0

样例 2:

输入: [1,1,1,2,2,3]

输出: 5

样例解释:

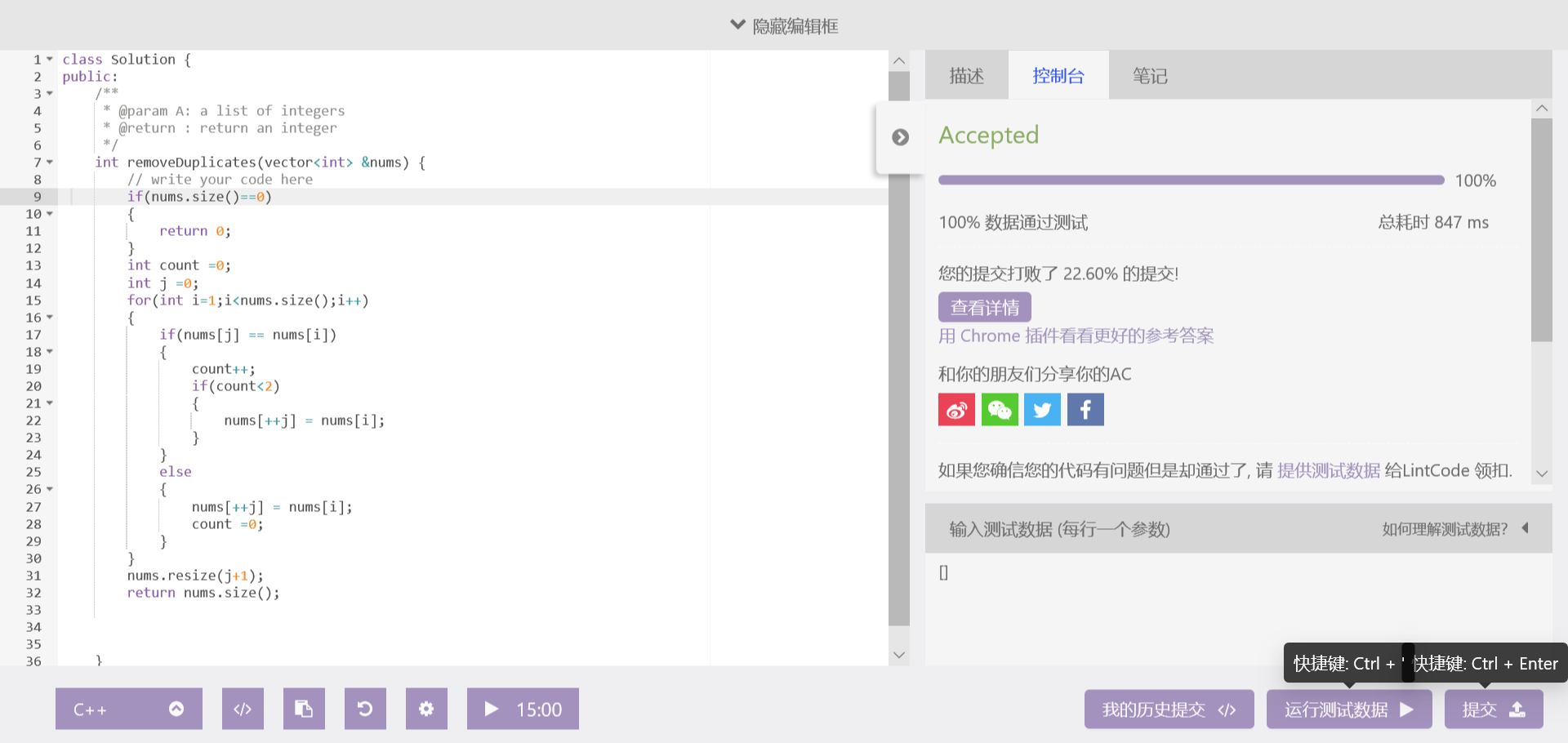
长度为 5， 数组为：[1,1,2,2,3]

【算法设计】

**设计思想**：

这题与上一题相比，增加了count用作统计重复数值的数量，当count=1时候，将元素覆盖到数组中；当前后两个数值不一样的时候，将元素覆盖到数组中，并且将count归0；当count>=2,不覆盖数组。

【附录】



**第21题**

【需求规格说明】

给定一个数组和一个值，在原地删除与值相同的数字，返回新数组的长度。

元素的顺序可以改变，并且对新的数组不会有影响。

### 样例

Example 1:

Input: [], value = 0

Output: 0

Example 2:

Input: [0,4,4,0,0,2,4,4], value = 4

Output: 4

Explanation:

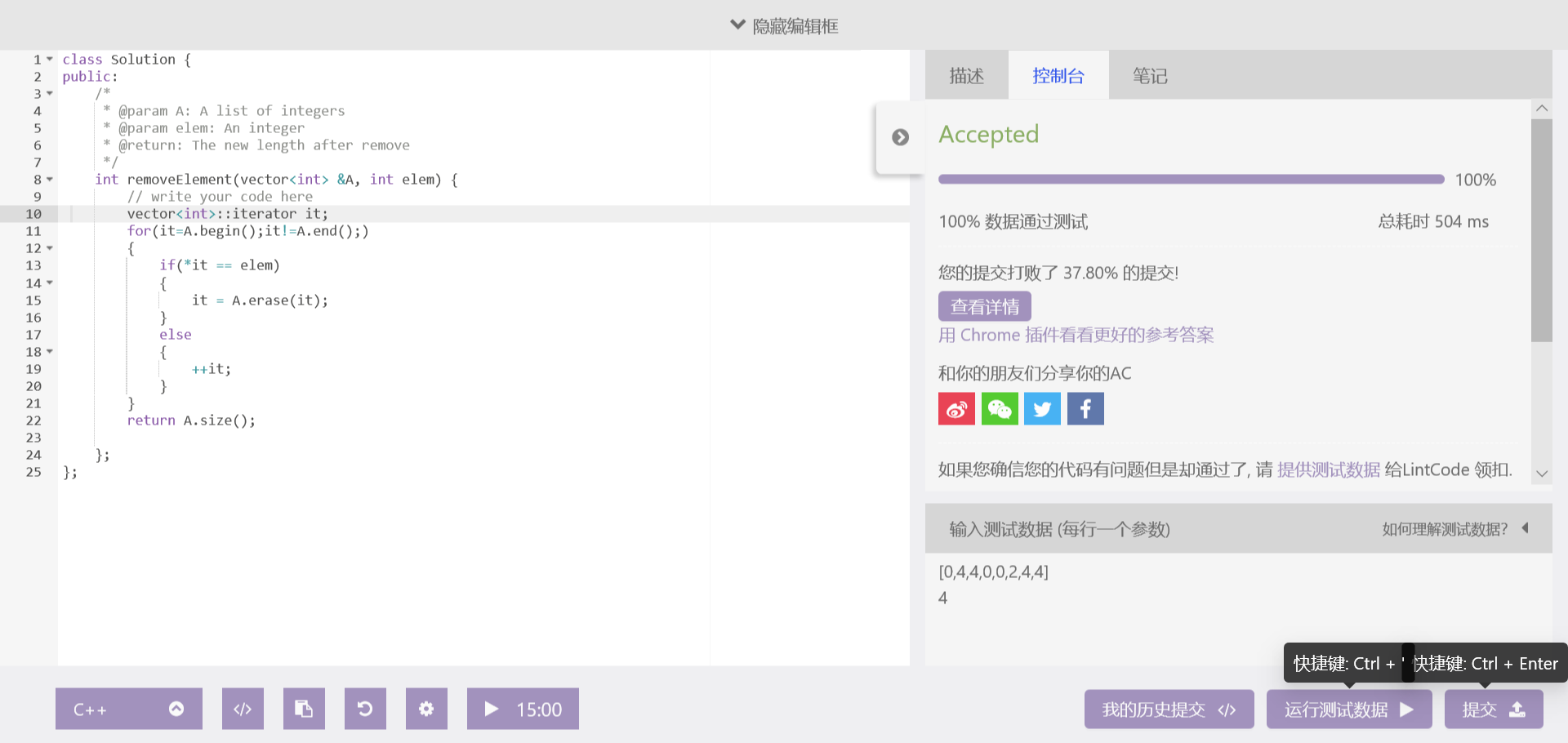
the array after remove is [0,0,0,2]

【算法设计】

**设计思想**：

利用循环找到需要删除的元素的位置，利用erase函数进行删除，要注意的是，erase函数返回的是下一个位置的迭代器，所以用it来接收。

【附录】



**第22题**

【需求规格说明】

给定一个非负数，表示一个数字数组，在该数的基础上+1，返回一个新的数组。

该数字按照数位高低进行排列，最高位的数在列表的最前面。

### 样例

**样例 1：**

输入：[1,2,3]

输出：[1,2,4]

**样例 2：**

输入：[9,9,9]

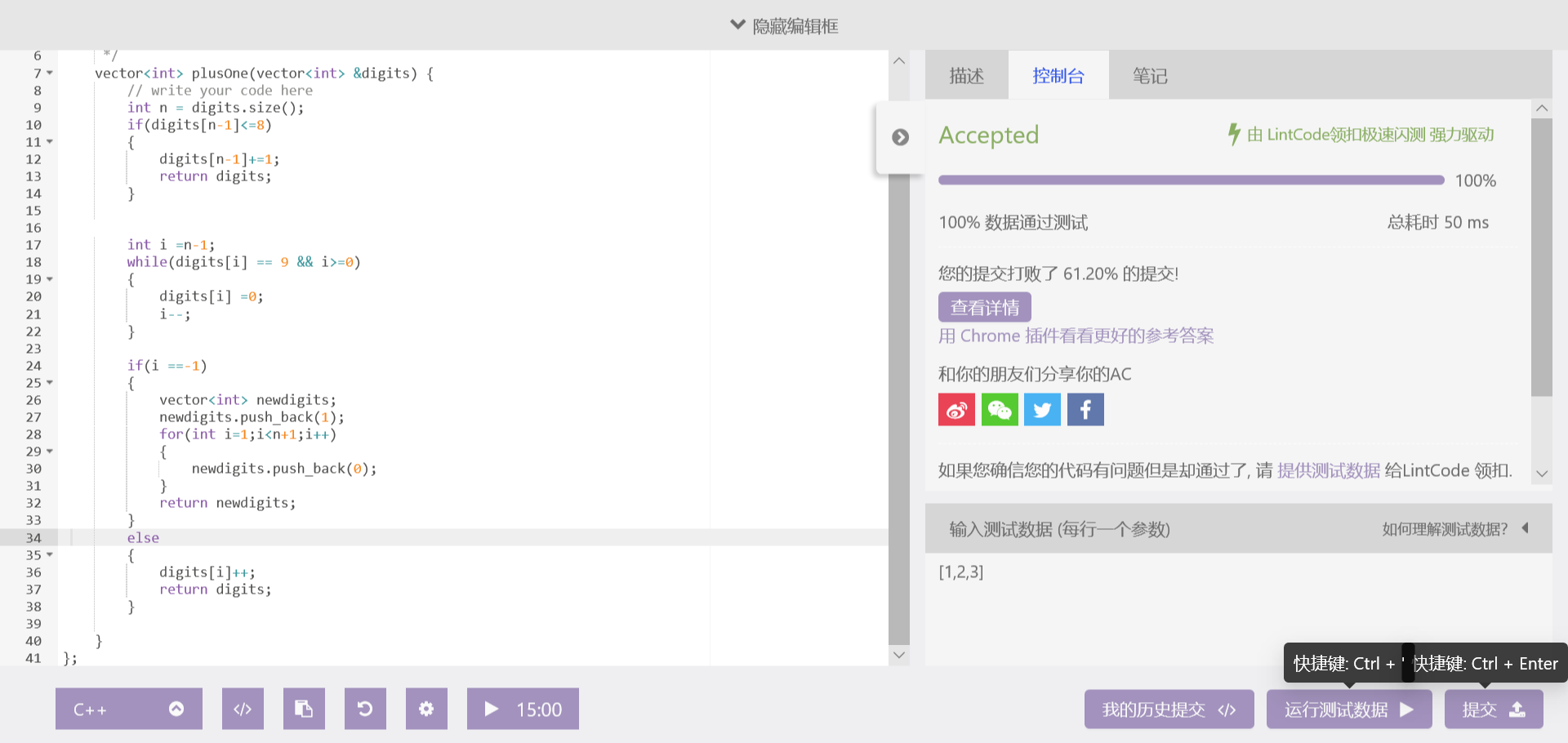
输出：[1,0,0,0]

【算法设计】

**设计思想**：

分类进行讨论，如果数组中最后一个元素<9,则直接对数组最后一个元素实现+1；若数组最后一个元素=9，利用循环进行遍历，当遇到不为9的元素或者循环至数组开头的时候跳出循环，如果遇到不为9的元素，则该元素后的数值均变0，该元素加一后返回；若循环至数组开头跳出循环，则建立一个新的数组，大小比原数组大一，然后该数组第一个元素为1，后面的元素均为0，然后返回。

【附录】



**第23题**

【需求规格说明】

给出一个链表，并将链表的值以**倒序**存储到数组中。

样例

**样例1**

输入: 1 -> 2 -> 3 -> null

输出: [3,2,1]

**样例2**

输入: 4 -> 2 -> 1 -> null

输出: [1,2,4]

注意事项

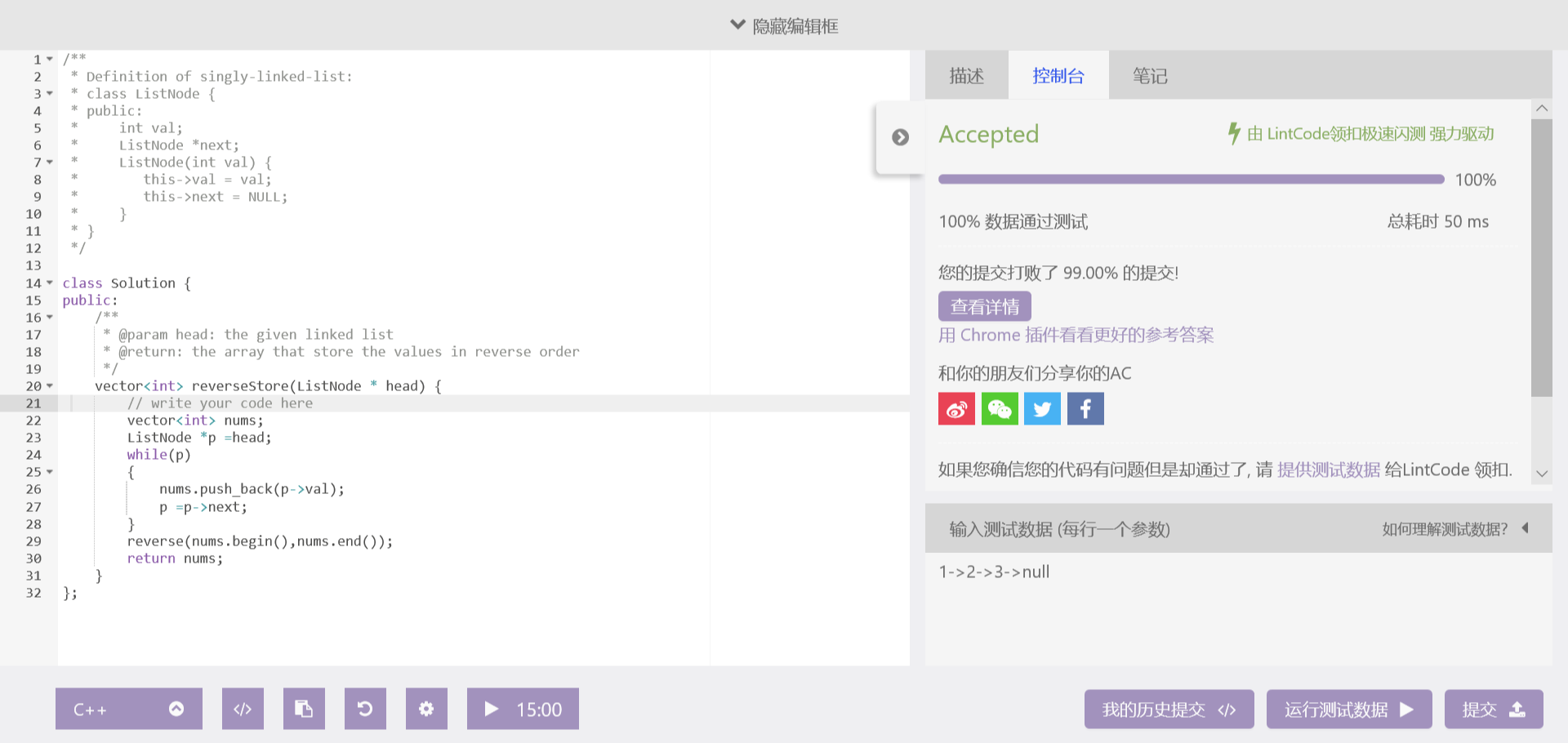
* 您**不能改变**原始链表的结构。
* ListNode 有两个成员变量：ListNode.val 和 ListNode.next

【算法设计】

**设计思想**：

新建一个数组存储链表中的数值，然后利用reverse函数反转数组。

【附录】



**第24题**

【需求规格说明】

给定一个整数数组，你需要找到一个**连续子数组**，如果你只按升序对这个子数组进行排序，那么整个数组也将按升序排序。

你需要找到**最短的**这样的子数组并输出它的长度。

### 样例

输入: [2, 6, 4, 8, 10, 9, 15]

输出: 5

解释: 你需要对[6, 4, 8, 10, 9]按升序排列从而整个数组也变为升序。

### 注意事项

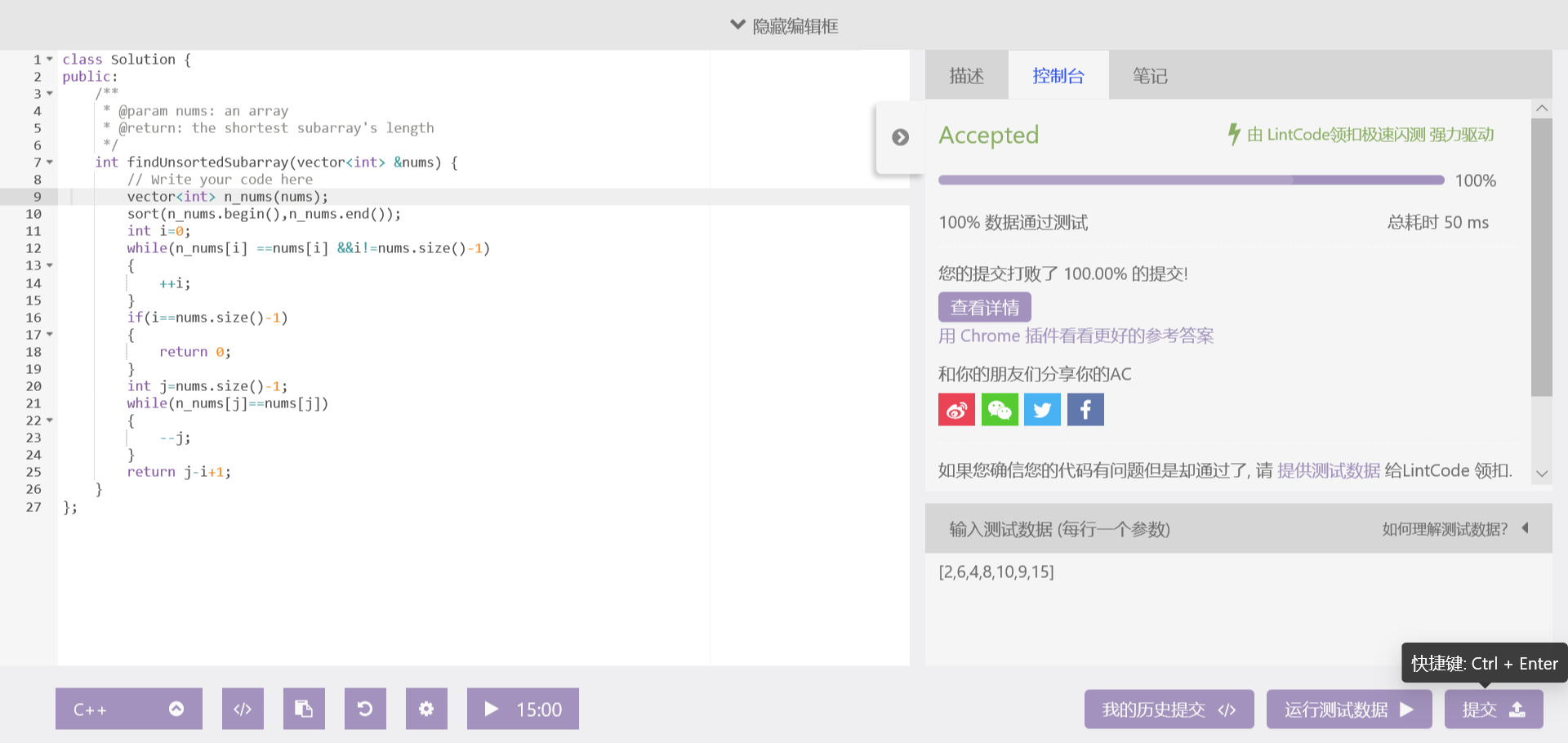
1. 输入的数组长度范围为[1, 10,000]。
2. 输入的数组可能会包含重复元素，本题升序的含义为<=。

【算法设计】

**设计思想**：

将该数组排序，将排序好的数组与原数组进行比较，然后分别从首位两边进行迭代，直至找到与原数组不相同的数值为止。

【附录】



**第25题**

【需求规格说明】

给定一个非空数组，返回此数组中第三大的数。如果不存在，则返回数组中最大的数。要求算法时间复杂度必须是O(n)。

### 样例

**样例 1:**

输入: num = [3, 2, 1]

输出: 1

解释: 第三大的数是 1.

**样例 2:**

输入: num = [1, 2]

输出: 2

解释: 第三大的数不存在, 所以返回最大的数 2 .

**样例 3:**

输入: num = [2, 2, 3, 1]

输出: 1

解释: 注意，要求返回第三大的数，是指第三大且唯一出现的数。

存在两个值为2的数，它们都排第二。

【算法设计】

**设计思想**：

重点是对不同情况下的分类讨论。

【附录】



**第26题**

【需求规格说明】

给定一个整数数组（下标从 0 到 n-1， n 表示整个数组的规模），请找出该数组中的最长上升连续子序列。（最长上升连续子序列可以定义为从右到左或从左到右的序列。）

### 样例

**样例 1：**

输入：[5, 4, 2, 1, 3]

输出：4

解释：

给定 [5, 4, 2, 1, 3]，其最长上升连续子序列（LICS）为 [5, 4, 2, 1]，返回 4。

**样例 2：**

输入：[5, 1, 2, 3, 4]

输出：4

解释：

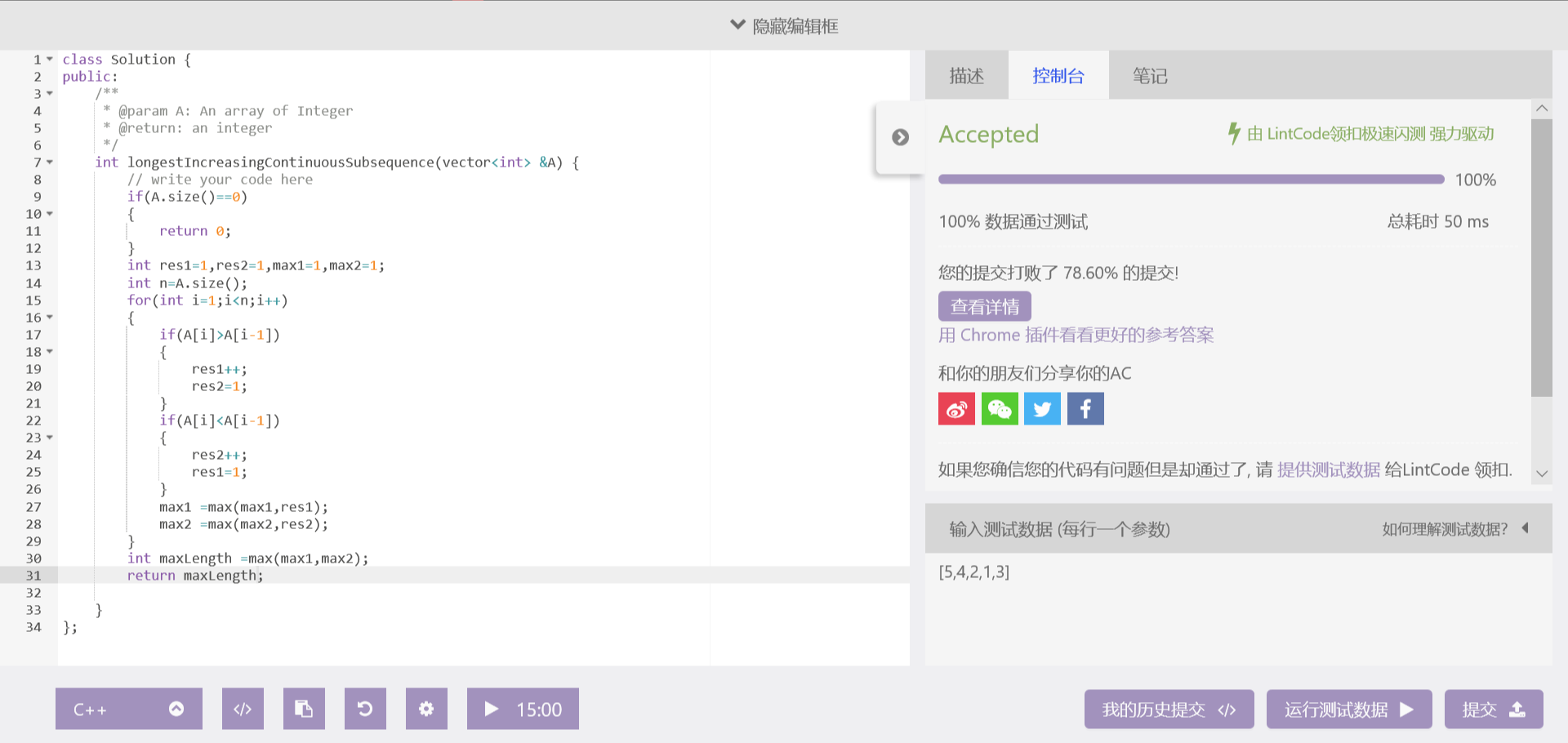
给定 [5, 1, 2, 3, 4]，其最长上升连续子序列（LICS）为 [1, 2, 3, 4]，返回 4。

【算法设计】

**设计思想**：

1. 先讨论数组中没有元素的情况，返回0
2. 用一个循环统计递增和递减的数列，并且一旦递增（减）断开，对应的计数器就要重置。
3. 用max函数来记录每次循环后最长的上升或下降子列的数目。

【附录】



**总结**

此次的作业对于我来说有较高的难度，但正应如此，我从中受益匪浅，在间，我通过学习慕课浙江大学的数据结构，学会了链表的基本操作和基本运用，熟悉了对链表的创建，删除，插入，反转和修改，并且能较为熟练的利用递归对多个链表进行操作，排序，合并等。作业中的数组题对算法知识有一定的要求，在刚开始做题的时候，难度较大，但是在与同学和老师，以及谷歌的帮助下，获取了非常多的新知识，对数组下标索引的运用和理解，以及各种排序算法有了更深的认识。