

---

## HW机考攻略

### 【刷题网址】

牛客——<https://www.nowcoder.com/ta/huawei>

力扣——<https://leetcode-cn.com/problemset/all>

剑指offer——<https://www.nowcoder.com/ta/coding-interviews?page=1>

### 【刷题题型】

#### 1.入门题（5题）

- (1) 输入处理（重要）： HJ5.进制转换 [stoi一把梭](#)
- (2) 排列组合： （牛客搜索)NC61.两数之和 [hash表](#)
- (3) 快速排序： HJ3.明明的随机数 [简单题](#)
- (4) 哈希表： HJ10.字符个数统计 [如题所示哈希表](#)
- (5) 递归： NC68.跳台阶 [DP](#)

（带\*题目与第一第二道题目难度相近，以下题目基本覆盖大部分知识点）

#### 2.字符串操作（8题）

- (1) HJ17.坐标移动 [算是给后面的BFS/DFS做铺垫吧](#)
- (2) HJ20.密码验证合格程序 [多种情况综合考量](#)
- (3) \*HJ23.删除字符串中出现次数最少的字符 [hash table](#)
- (4) \*HJ33.整数与IP地址间的转换 [位运算](#)
- (5) HJ101.输入整型数组和排序标识[简单的](#)
- (6) \*HJ106.字符串逆序 [简单的](#)
- (7) [leetcode 1839 最长子字符串](#) [大道至简，走一步看一步，别被吓到了](#)
- (8) [HJ46 截取字符串](#) [简单的](#)

#### 3.排序（5题）

- (1) HJ8.合并表记录 [这道题可以用很妙得思路来做](#)
- (2) \*HJ14.字符串排序 [简单题](#)
- (3) HJ27.查找兄弟单词 [思路清晰](#)
- (4) \*NC37.合并区间 [思路其实是清晰的](#)
- (5) \*HJ68.成绩排序 [思路清晰，注意排序规则](#)

#### 4.栈（2题）

- (1) NC52.括号序列 [遇到左括号直接入栈，遇到右括号判断栈顶](#)
- (2) \*leetcode 1614.括号的最大嵌套深度 [遇到左括号++，遇到右括号就--，](#)

#### 5.排列组合（2题）

- (1) \*leetcode 面试题08.08.有重复字符串的排列组合 [殊途同归，回溯法yyds](#)
- (2) leetcode 77.组合 [组合问题回溯法有了大致的了解](#)

#### 6.双指针（3题）

- (1) \*leetcode 674.最长连续递增序列 [动态规划](#)

---

(2) NC17.最长回文子串

(3) NC28.最小覆盖子串

## 7.深搜（1题）

(1) HJ41.称砵码 将复杂的问题简化，老老实实一个个加

## 8.二叉树（2题）

(1) \*leetcode 剑指offer 32 — II.从上到下打印二叉树 II 层次遍历，华为要求自行输入输出，那可能硬考这种题目的概率不高

(2) leetcode 剑指offer 32 — III.从上到下打印二叉树 III

## 9.其他（12题）

(1) \*HJ108.求最小公倍数 注意计算的公式，其实是简单题

(2) \*HJ28.素数伴侣 匈牙利算法，这道题目有点难

(3) \*HJ60.查找组成一个偶数最接近的两个素数 简单题 直接暴力破解

(4) \*leetcode 994.腐烂的橘子 BFS，主要还是不熟悉这个模板，有点难

(5) leetcode 204.计数质数 暴力法会超时，但是可以混过一些样例，骗些分，标签还给了埃氏筛

(6) HJ25. 数据分类处理 这道题目思路还是清晰的，但是把代码写出来还是有一定的难度

(7) HJ29 字符串加解密 常规题目，大小之间的转换

(8) 剑指 Offer 61. 扑克牌中的顺子 草稿纸上刷刷写刷

(9) HJ43 迷宫问题 有点难，主要是回溯法还不是很熟

(10) leetcode 322. 零钱兑换，也可搜索 动态规划相关题型（技术面试高频考点） dp问题，来来回回看代码随想录

(11) 火锅问题 csdn搜索 不看这个了，考到了认了

(12) 岛屿问题 leetcode 200 （技术面试高频考点） BFS/DFS

## 【注意事项】

1. 摄像头一定要开，避免离开座位，左右晃头，以免系统误判作弊
2. 可以用本地IDE调试，不能用在线IDE，考试期间会录屏，屏幕只允许在考试页面和本地IDE之间切换
3. 练习时要注意语言的数据结构，编程语言的高级函数的使用等
4. 编程的时候就用牛客网的 IDE 去答题，因为答题的时候不会提示错误，用自己的会

以上题目熟练掌握后进行模拟考试，模拟套卷链接：<https://www.nowcoder.com/test/1088888/summary>

## HW机考在线练习和考试指南

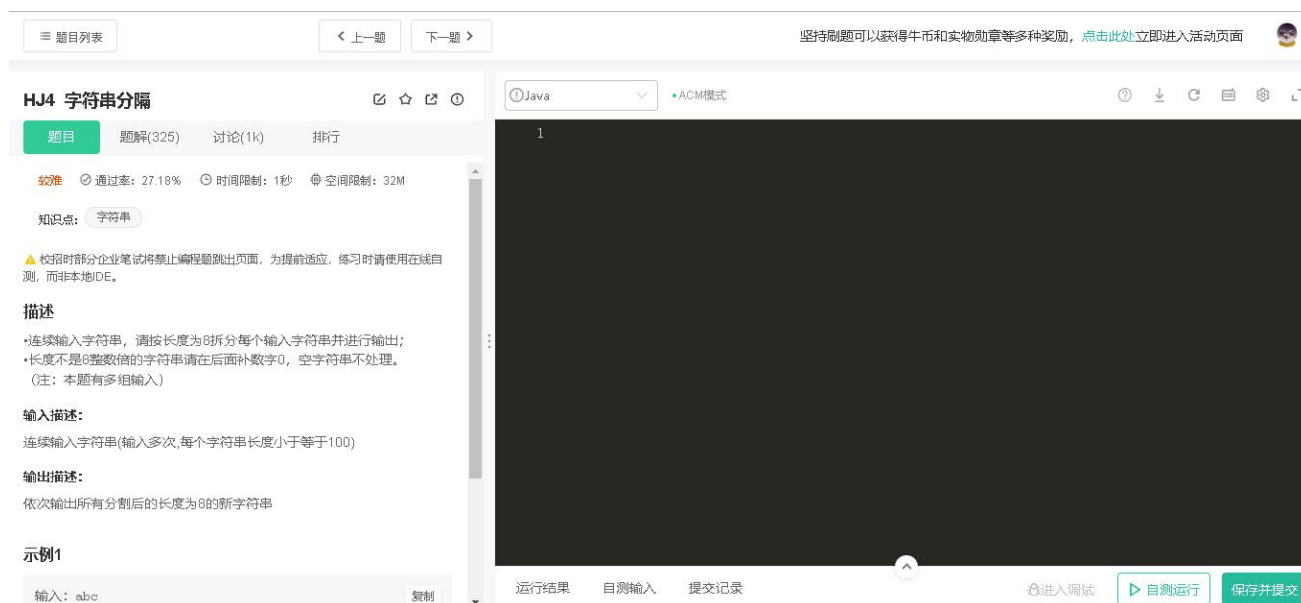
### 一、在线练习

考试前一定要在线训练，练习地址：<https://www.nowcoder.com/ta/huawei>，直接裸考很容易挂，考试题型都涵盖在训练题当中，难、中、易题型都练习一下，充分熟悉题型和答题逻辑，如下图：



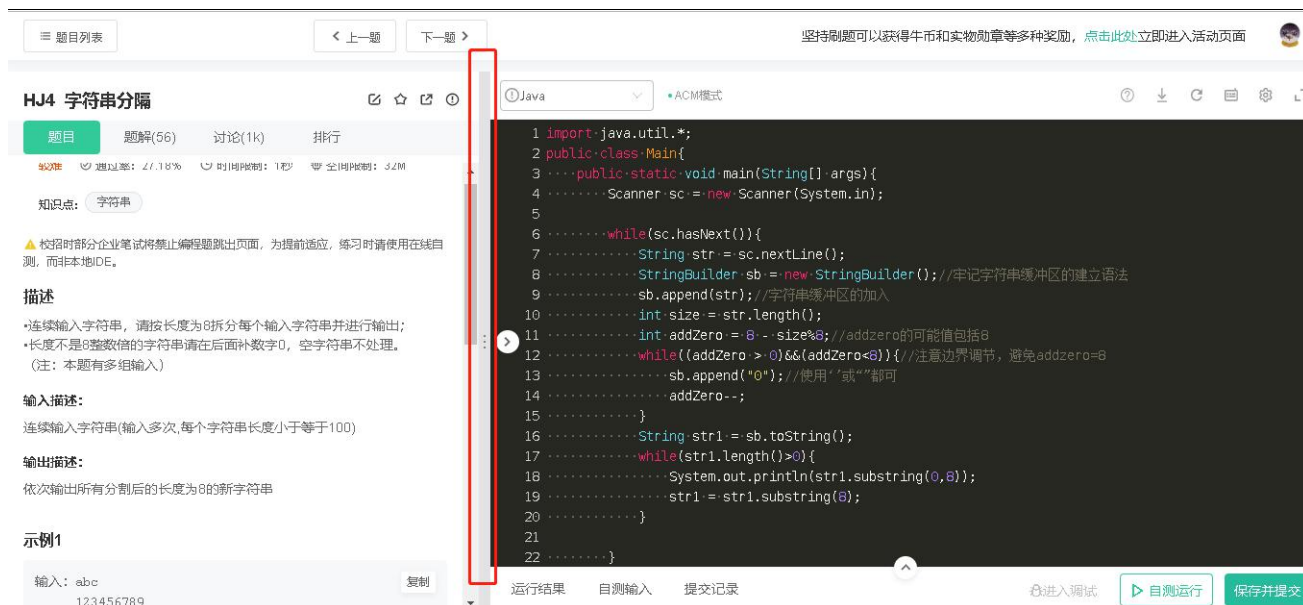
### 二、使用方法（以Java为例）

在模拟练习时，如下图所示：

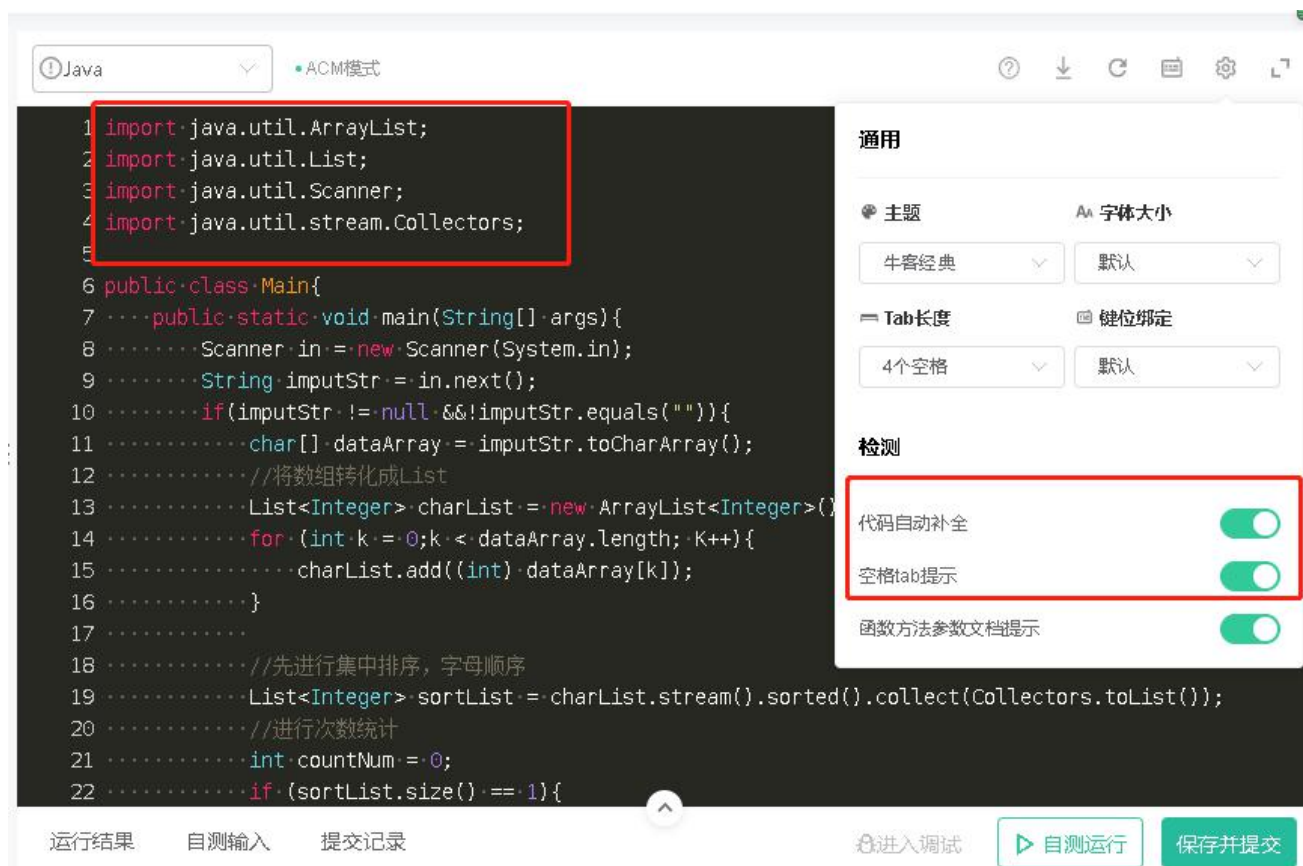


### 1、程序开发

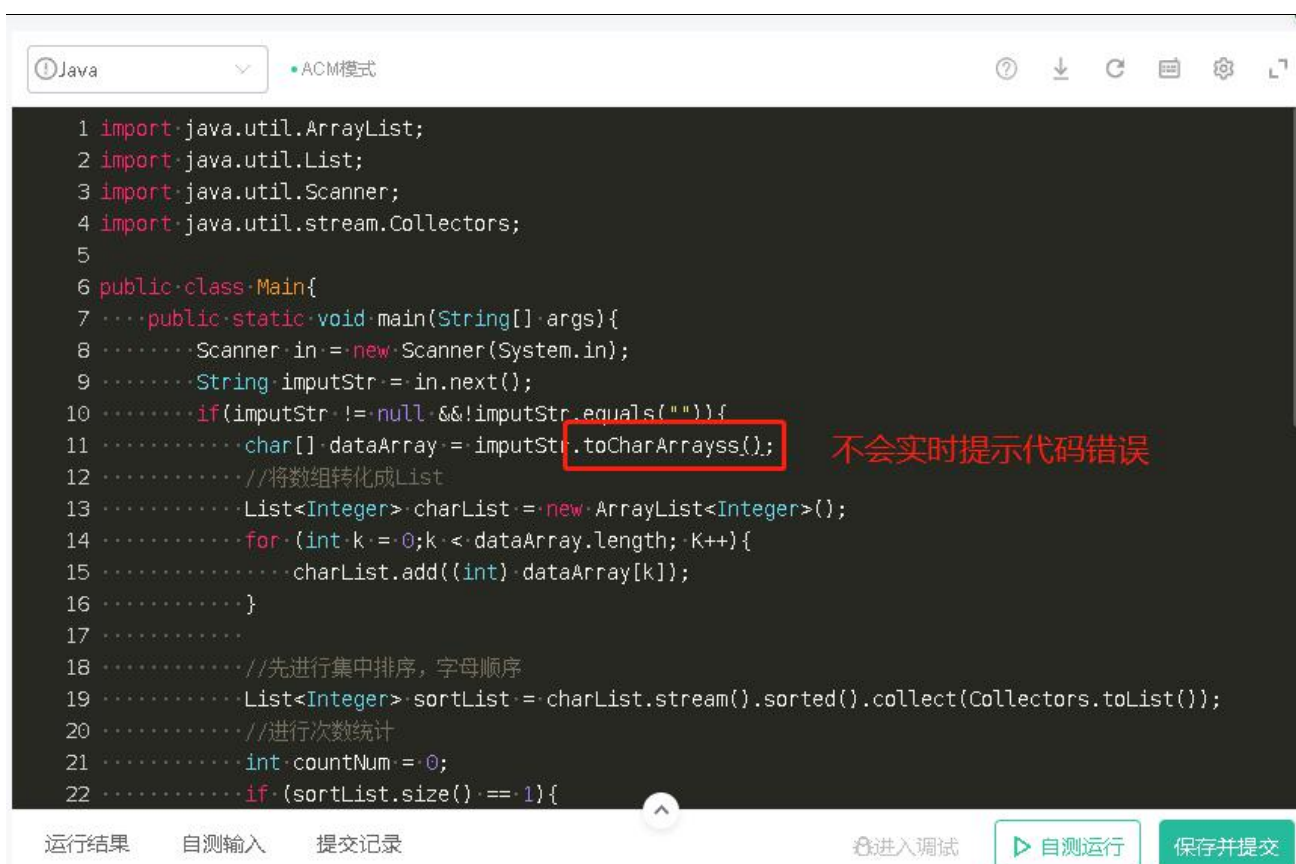
（1）程序开发时不要紧张，先将工具调整为最合适的宽度（当鼠标悬停在下图红框处鼠标变为“\*”时直接拖拽左侧的文本框即可，也可点击右上角的全屏，进行全屏编码），保证编码舒适性，如下图：



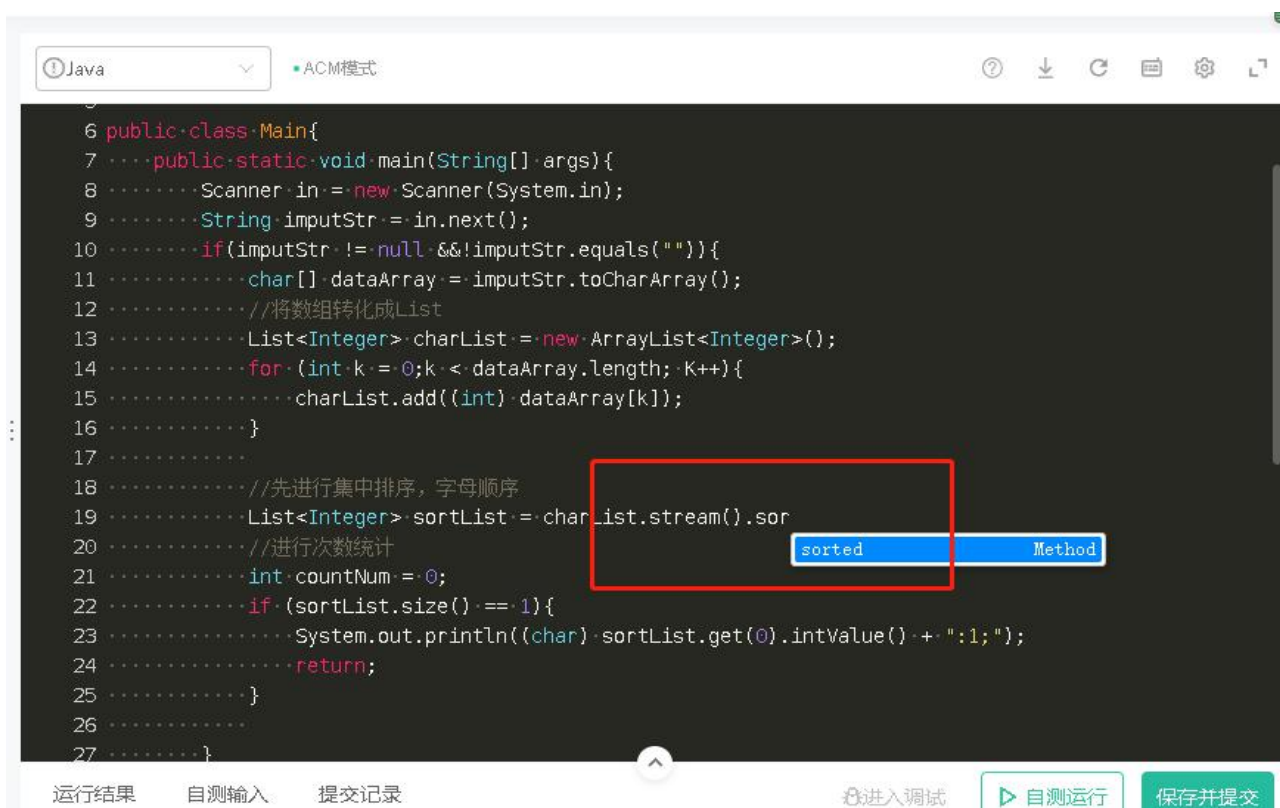
(2) 编码时，注意 import 所需要的包，代码工具无法自动提示相关类的引入，代码补全及 tab 空格提示已默认开启：



(3) 代码错误提示：除了括号匹配外，其他错误如拼写错误一概不会给出提示，直到运行调试时才会  
在结果框（可理解为控制台 console）中显现，此处建议，编码当达到一定量后做代码自测，以免因无代码报错功能而导致拼写错误有多处时很难找到错误原因。具体详情请见 2、程序调试



(4) 针对 java 的代码补全功能并不强大，很多提示无法得出，请勿过分依赖于代码补全功能，应早做准备，熟记字符串及数组等常用方法名称，例如 jdk1.8 的排序方法在代码提示中就无法得出，如下图：





其他 jdk1.8 lambda 表达式也是如此, Collectors 以及 Comparator 等常用工具类的引用等需要手动引入, 代码补全不会提示(List 及 ArrayList 的引入均不会自动提示)。

(5) 牛客ACM编程模式, 你的代码需要处理输入输出, 如下图:



```
1 import java.util.*;
2 public class Main{
3     ... public static void main(String[] args){
4         ... Scanner sc = new Scanner(System.in);
5
6         ... while(sc.hasNext()){
7             ... String str = sc.nextLine();
8             ... StringBuilder sb = new StringBuilder();//牢记字符串缓冲区的建立语法
9             ... sb.append(str);//字符串缓冲区的加入
10            ... int size = str.length();
11            ... int addZero = 8 - size%8;//addzero的可能值包括8
12            ... while((addZero > 0) && (addZero < 8)){//注意边界调节, 避免addzero=8
13                ... sb.append("0");//使用' '或" "都可
14                ... addZero--;
15            ... }
16            ... String str1 = sb.toString();
17            ... while(str1.length() > 0){
18                ... System.out.println(str1.substring(0,8));
19                ... str1 = str1.substring(8);
20            ... }
21
22            ... }
```

输入

输出

运行结果 自测输入 提交记录 进入调试 自测运行 保存并提交

## 2、程序调试

(1) 当代码完成一部分时, 如需调试, 请点击右下角的自测运行, 打开自测页面, 如下图:



```
1 import java.util.*;
2 public class Main{
3     ... public static void main(String[] args){
4         ... Scanner sc = new Scanner(System.in);
5
6         ... while(sc.hasNext()){
7             ... String str = sc.nextLine();
8             ... StringBuilder sb = new StringBuilder();//牢记字符串缓冲区的建立语法
9             ... sb.append(str);//字符串缓冲区的加入
10            ... int size = str.length();
11            ... int addZero = 8 - size%8;//addzero的可能值包括8
12            ... while((addZero > 0) && (addZero < 8)){//注意边界调节, 避免addzero=8
13                ... sb.append("0");//使用' '或" "都可
14                ... addZero--;
15            ... }
16            ... String str1 = sb.toString();
17            ... while(str1.length() > 0){
18                ... System.out.println(str1.substring(0,8));
19                ... str1 = str1.substring(8);
20            ... }
21
22            ... }
```

Java ACM模式

运行结果 自测输入 提交记录 进入调试 自测运行 保存并提交

(2) 下图中自测页面文本框解释如下图:

```
8 .....StringBuilder sb = new StringBuilder();//牢记字符串缓冲区的建立语法
9 .....sb.append(str);//字符串缓冲区的加入
10 .....int size = str.length();
11 .....int addZero = 8 - size%8;//addzero的可能值包括8
12 .....while((addZero > 0) && (addZero < 8)){//注意边界调节,避免addzero=8
13 .....sb.append("0");//使用'或"都可
14 .....addZero--;
15 .....}
16 .....String str1 = sb.toString();
17 .....while(str1.length() > 0){
18 .....System.out.println(str1.substring(0,8));
```

运行结果 自测输入 提交记录 进入调试 自测运行 保存并提交

输入数据

输入框，入参，完成输入之后点击“自测运行”

载入示例 1

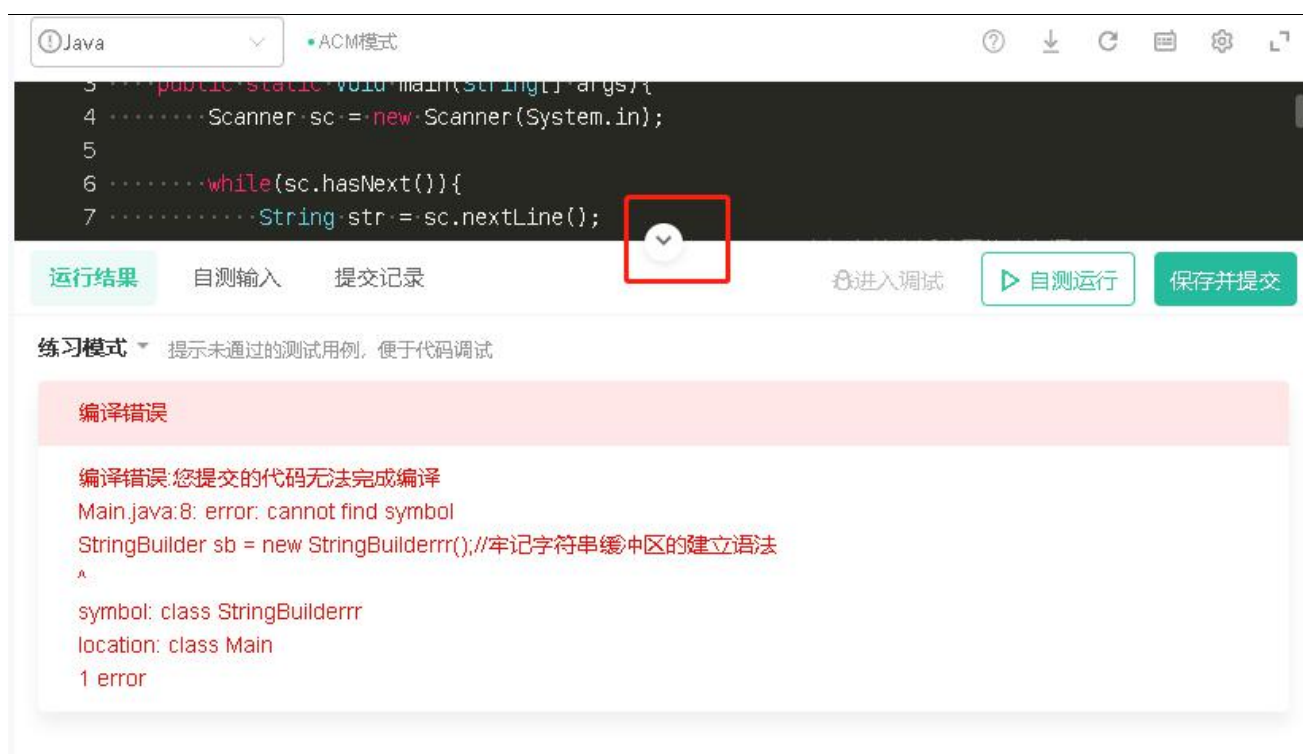
自测示例如下:

运行结果 自测输入 提交记录 进入调试 自测运行 保存并提交

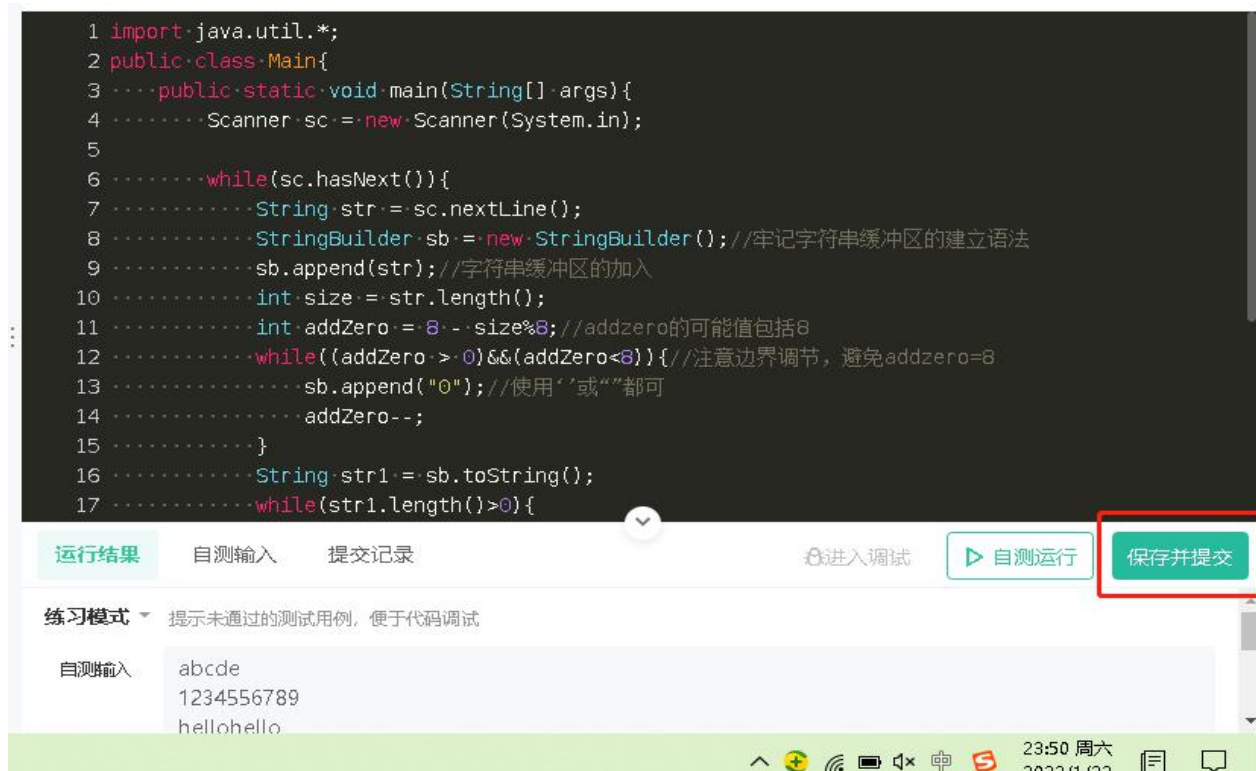
练习模式 提示未通过的测试用例, 便于代码调试

自测输入	实际输出
abcde	abcde000
1234556789	12345567
hellohello	89000000
	hellohel
	lo000000

(3) 自测输出框可以拉升, 方便查询输出及错误信息, 根据输出及错误信息调整代码即可。



(4) 程序基本开发完成后记得点击“保存并提交”，否则答案不作保存，不限制提交运行次数（以提交中最高得分计分），调试完成点击【提交本题型】按钮，最后交卷。



### 3、牛客判卷系统状态

等待评测： 评测系统还没有评测到这个提交，请稍候；

正在评测： 评测系统正在评测，稍候会有结果；



---

**编译错误：**您提交的代码无法完成编译，点击“编译错误”可以看到编译器输出的错误信息；

**答案正确：**恭喜！您通过了这道题；

**运行错误：**您提交的程序在运行时发生错误，可能是空指针；

**部分正确：**您的代码只通过了部分测试点，继续努力！

**格式错误：**您的程序输出的格式不符合要求（比如空格和换行与要求不一致）；

**答案错误：**您的程序未能对评测系统的数据返回正确的结果；

**运行超时：**您的程序未能在规定时间内运行结束；

**内存超限：**您的程序使用了超过限制的内存；

**异常退出：**您的程序运行时发生了错误；

**返回非零：**您的程序结束时返回值非 0，如果使用 C 或 C++ 语言要保证 `int main` 函数最终 `return 0`；

**浮点错误：**您的程序运行时发生浮点错误，比如遇到了除以 0 的情况；

**段错误：**您的程序发生段错误，可能是数组越界，堆栈溢出（比如，递归调用层数太多）等情况引起；

**多种错误：**您的程序对不同的测试点出现不同的错误；

**内部错误：**请仔细检查你的代码是否有未考虑到的异常情况，例如非法调用、代码不符合规范等。

### 三、考试提示

- 1、 在线训练和正式考试，请在本地 IDE 上进行编写，然后再粘贴复制到笔试页面，保存并调试。
- 2、 考试时**不用紧张**，保持正常编码习惯即可，遇到无法提示又无把握的实现时，改用其他方式代替。
- 3、 训练和考试的代码**均需要编写输入和输出代码**，编码一段代码后记得进行调试，保证代码正确性。
- 4、 编码工具删除行 `ctrl+d` 依旧可以使用，`ctrl+c`，`+v`，`+z` 等均可正常使用，**其他快捷键建议尽量莫用**，尤其是有保存习惯的同学，**不要用 `ctrl+s`**，否则会打开页面保存键，算弹出当前页面一次。
- 5、 请安装使用最新版 **chrome** 浏览器作答，考试需开启摄像头，请确保电脑带有摄像头，考试中途请不要查看其它网站。请关闭所有可能屏幕提示的软件，包括**QQ、OUTLOOK、杀毒软件**等，只要屏幕失焦都算是退出，**弹窗出现即认为跳出页面**。
- 6、 答题前请注意查看示例(计入考试时长)，代码完成后请务必点击**【保存并提交】**按钮(可以多次，以最高得分记分)，否则答案不做保存，调试完成点击**【提交本题型】**按钮，最后交卷。
- 7、 考试时间 150 分钟，满分 400 分，及格 150 分。卷面共 3 道题，2 道一星难度，一题 100 分，1 道二星难度，一题 200 分，考试语言可自行选择。**考试时间合理把握**，不要太纠结第一、二道题，十分钟没啥思路就放弃，如果耽误久了，第3道题肯定没时间做，（一定要给第3道题留出一个小时的时间），等第3道做完回头可以再看下之前不会的。第三道虽然题目复杂，但是给分相对宽松，更容易拿分，只要不点击最后的总提交都可返回重新修改前两题；
- 8、 考试时允许使用草稿纸，请提前准备纸笔。如果需要上厕所，请在考试前提前准备好，避免考试中途离开座位，出现离开监控的情况。
- 9、 考试期间如遇到断电、断网、死机等问题，可以关闭浏览器重新打开试卷链接即可继续做题，遇到问题请及时反馈给 HR。