

Cloud Studio (云端 IDE)

智能批改



腾讯云

【 版权声明 】

©2013–2026 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

文档目录

智能批改

教师端

智能批改

OJ 自查 (开启)

学生端

关于作业

OJ 自查 (使用)

智能批改

教师端

智能批改

最近更新时间：2026-05-19 18:07:22

智能批改用于老师快速批改编程作业：**配置一次**，支持**单个/批量**批改，并输出可读的评分与建议。

📌 说明：

- 仅支持非公开课程（公开课程不可用）。
- 仅管理角色可配置/批改：所有者 / 老师 / 助教。
- 需要先有课程与章节：参考 [创课/配置课程](#)。

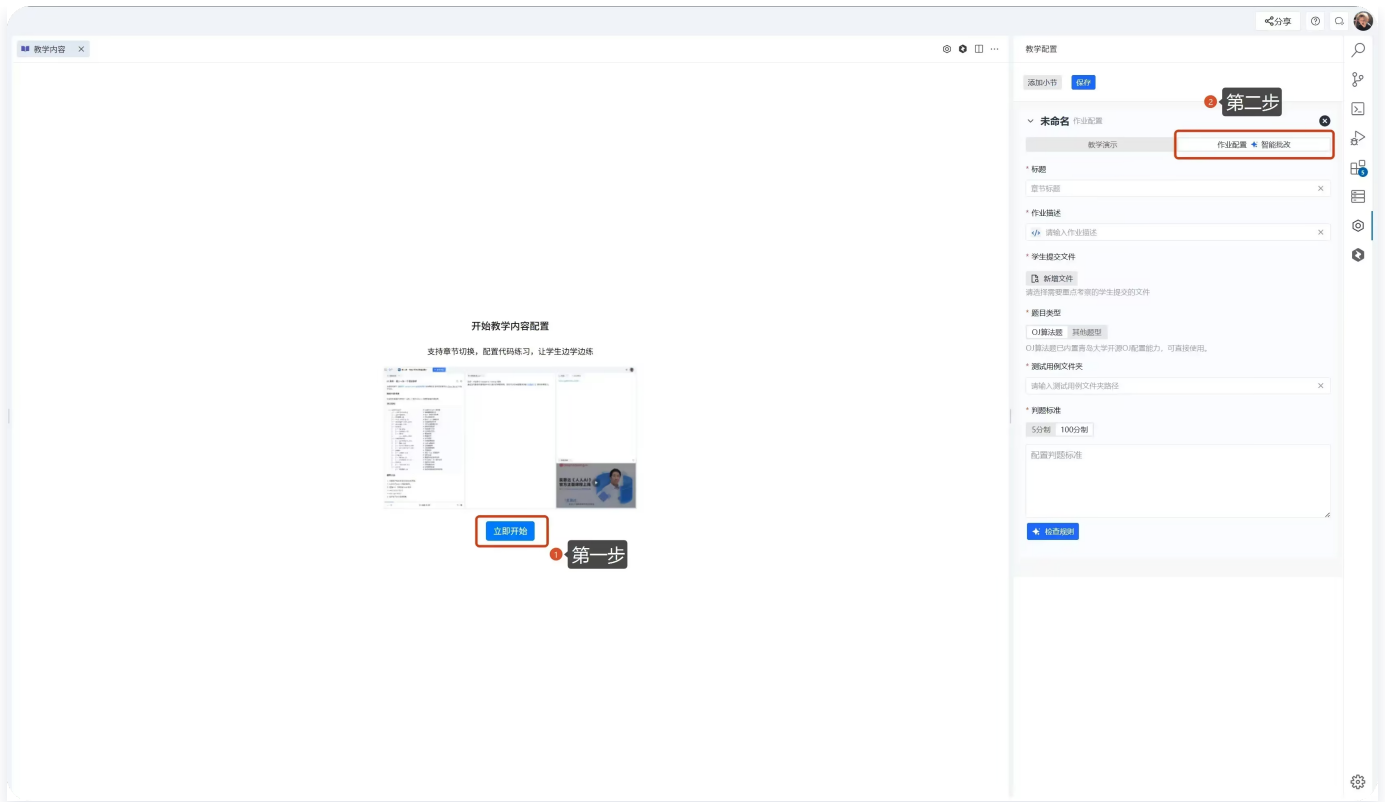
如果只做课堂演示、不需要批改，可以不启用智能批改。

快速上手

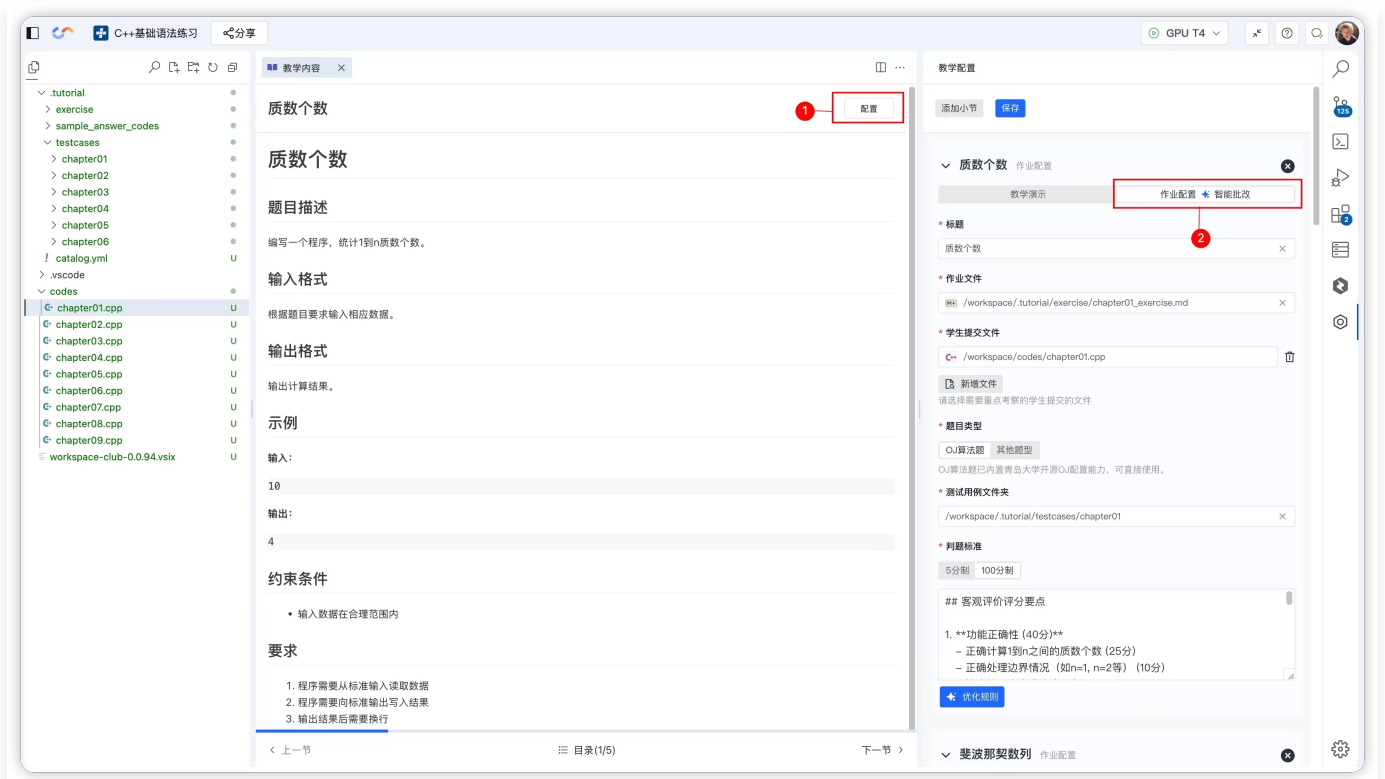
1. 打开课程章节的教学配置
2. 进入「[作业配置&智能批改](#)」
3. 上传作业内容（拖拽或 Git）
4. 配置作业并保存
5. 在学生列表中执行智能批改（单个/批量）
6. 进入学员学习状态页面查看并导出

一、进入「[作业配置&智能批改](#)」

- 首次进入：单击「[立即开始](#)」 > 「[作业配置&智能批改](#)」



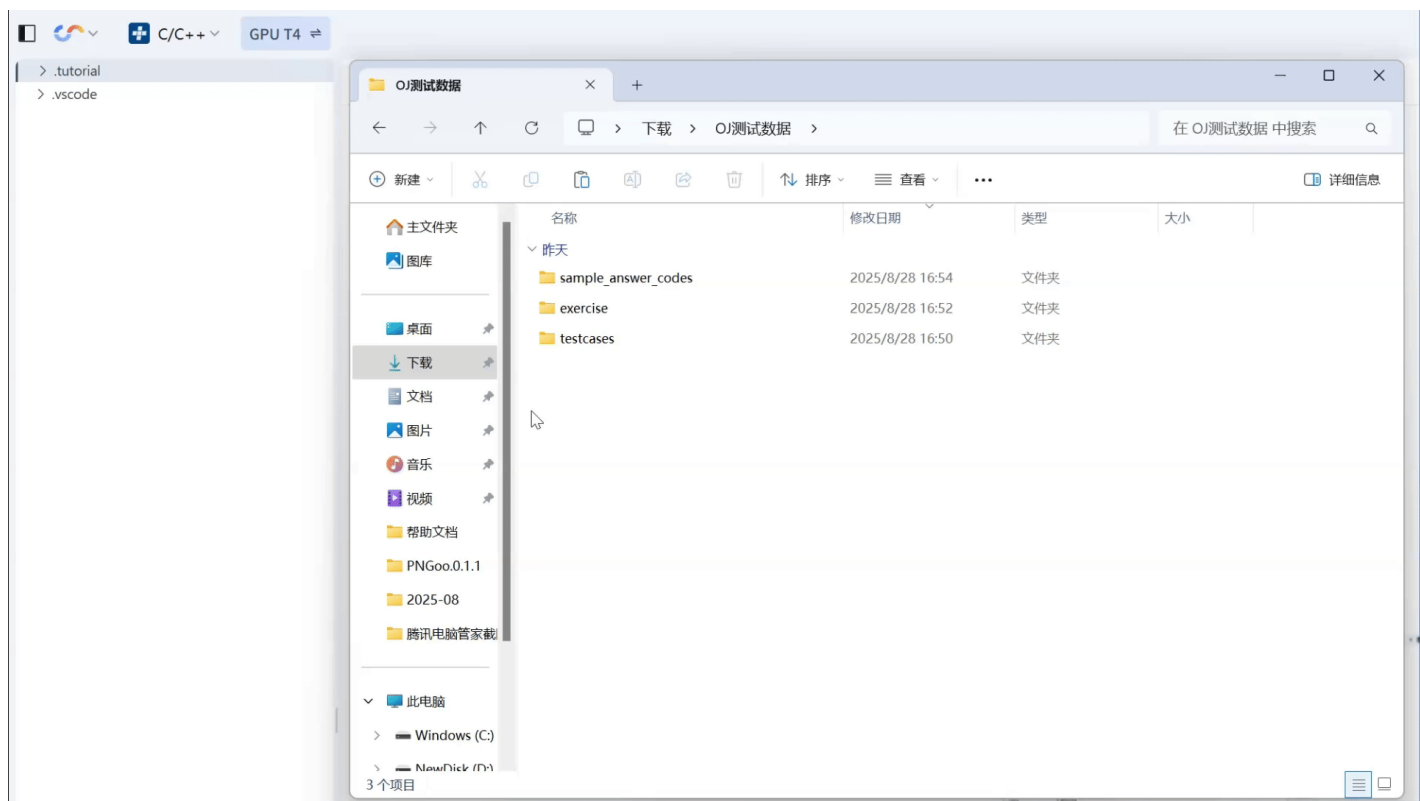
● 已有教学配置: 单击「配置」>「作业配置&智能批改」



二、上传作业内容

方式 1: 拖拽上传

把本地文件夹直接拖拽到工作空间主目录。



方式 2: Git 克隆

在终端从 Git 仓库拉取作业内容（适合版本管理）。

🔔 安全提示（必看）

- 把“不想让学生看到”的文件放进 `.tutorial` 来隐藏。
- 只有当文件在智能批改里被设置为**参考答案**并启用加密时，内容才会被保护；否则学生仍可能通过终端或仓库看到。

三、配置作业

🔔 说明

这里用于设置作业要求与评分规则；判题标准可用 AI 生成，也可手动调整。

1) 配置项说明

配置项	您需要做什么	说明/示例
标题	填作业名称	例: Python 函数进阶实践 – 第 4 章
作业描述	写清要求与提交说明	支持 Markdown / PDF / Word

学生提交文件	设定要提交的文件名（可多文件）	例： <code>report.docx</code>
题目类型	选择 OJ 算法题 / 其他题型	选择后 AI 自动生成基础评分规则
测试用例（OJ 题型专属）	上传测试用例文件夹	文件包含 <code>.in / .out</code> （例： <code>test_cases/</code> ）
运行命令（其他题型扩展）	配置自动运行命令	例： <code>bash data_process.sh</code>
参考答案（其他题型扩展）	上传标准答案文件	例： <code>db_optimize.docx</code>
判题标准	选择分制并调整规则	5 分制/100 分制；可用「优化规则」生成后微调

警告

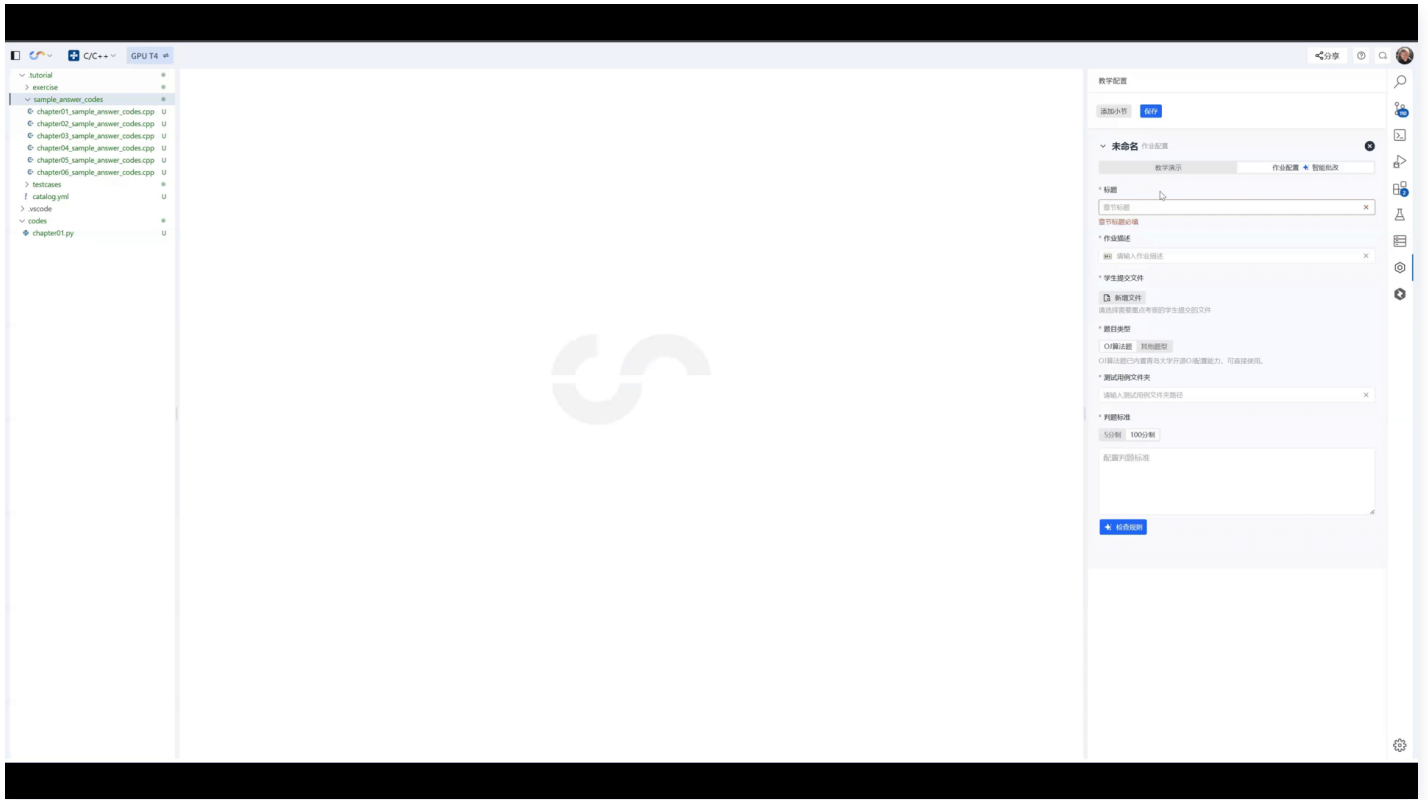
务必单击「保存」完成配置（未保存可能导致配置丢失）

配置逻辑说明：

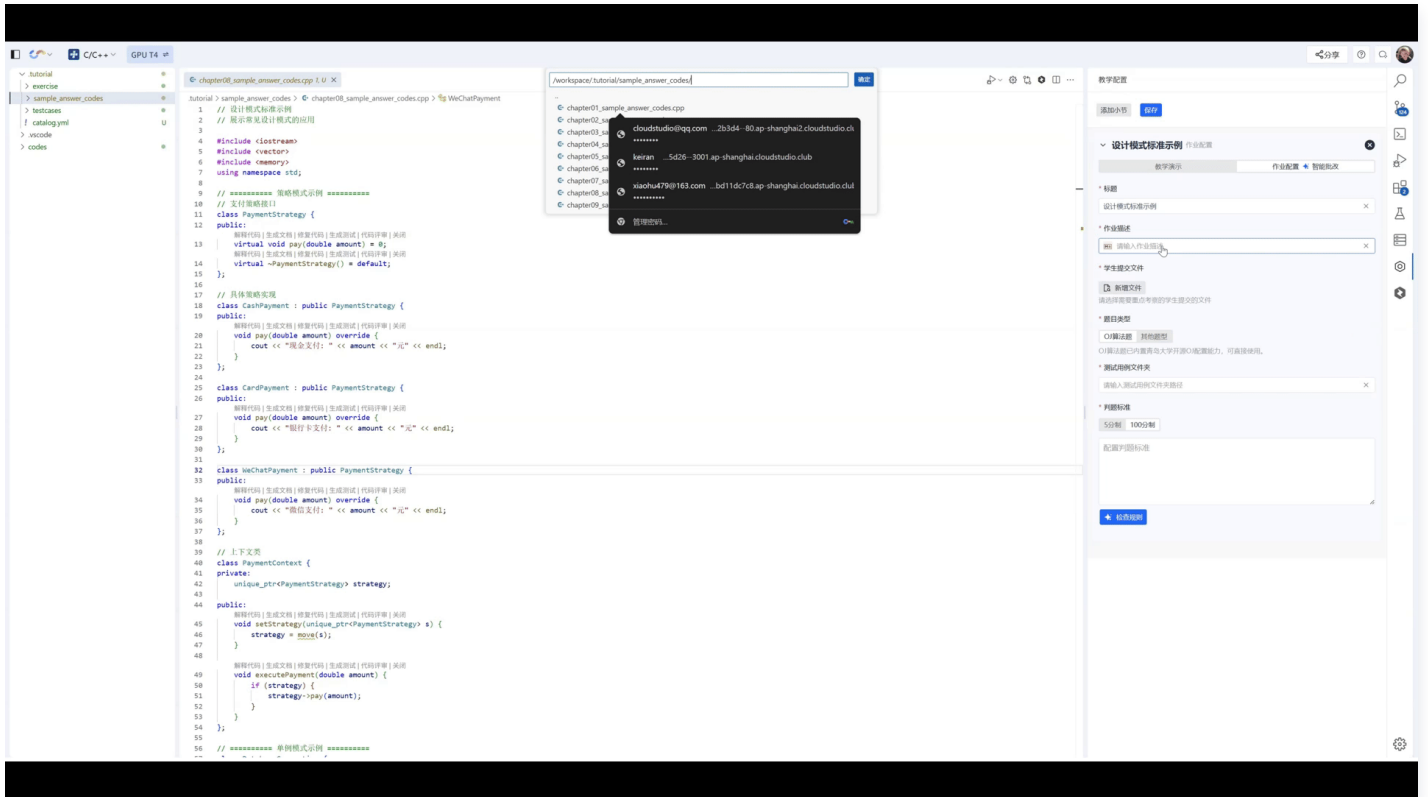
- **OJ 算法题：**通过测试用例自动验证代码正确性，适合编程题；
- **其他题型：**通过运行命令/参考答案辅助评分，适合文档/简答题；
- **判题标准：**AI 生成基础规则后，教师可根据教学需求手动优化（如调整“代码逻辑”“功能实现”等维度权重）。

2) 配置操作示例

OJ 题型



其他题型



四、执行智能批改

批改时间说明

智能批改约需 3-4分钟/题，请耐心等待。批改过程中退出页面不会影响批改进度。

在学生列表中，可选择：

方式一：单独批改

单击学生行右侧「智能批改」



学习中心 / C 基础课程 / C/C 学习进度

全部 2 已学完 2 学习中 0 未学习 0

批量执行智能批改 导出得分

昵称	姓名	学号	学习状态	提交时间	得分			操作
					质数个数(100分)	斐波那契数列(100分)	数组排序(100分)	
Cloud Studio	keiran	20250829	已学完	2025-08-29 11:55	未批改	未批改	未批改	智能批改 查看作业
			已学完	2025-08-29 11:57	未批改	未批改	未批改	智能批改 查看作业

方式二：批量批改

单击右上角「批量执行智能批改」



学习中心 / C 基础课程 / C/C 学习进度

全部 2 已学完 2 学习中 0 未学习 0

批量执行智能批改 导出得分

昵称	姓名	学号	学习状态	提交时间	得分			操作
					质数个数(100分)	斐波那契数列(100分)	数组排序(100分)	
Cloud Studio	keiran	20250829	已学完	2025-08-29 11:55	未批改	未批改	未批改	智能批改 查看作业
			已学完	2025-08-29 11:57	未批改	未批改	未批改	智能批改 查看作业

说明

- 耗时参考：约 3-4 分钟/题（与题目复杂度、提交量有关）。
- 批改过程中退出页面不影响进度。

五、查看结果与进入学生作业

1) 查看得分详情

- 有得分：显示具体分数

● 无得分：显示 - 图标（表示未完成该题目）

学习中心 / C 基础课程 / C/C 学习进度

全部 2 已学完 2 学习中 0 未学习 0

批量执行智能批改 导出得分

昵称	姓名	学号	学习状态	提交时间	得分			操作
					质数个数(100分)	斐波那契数列(100分)	数组排序(100分)	
Cloud Studio	keiran	20250829	已学完	2025-08-29 12:11	95 得分详情	- 点击	70 得分详情	智能批改 确认批改结果
			已学完	2025-08-29 11:57	未批改	未批改	未批改	智能批改 查看作业

2) 评分标准说明

在得分详情中查看主观/客观评价与改进建议。

得分详情

×

质数个数 95 分 [修改得分](#)

得分:

95 分

主观评价:

主观评价 (30分)

算法效率 (10/15分)

使用高效算法 (5/10分)

使用了简单的遍历方法判断质数,而非更高效的埃氏筛法
但实现了一些优化:只检查到sqrt(n),跳过偶数检查

避免不必要的计算 (5/5分)

对偶数进行了特殊处理,减少了不必要的循环
只检查到sqrt(n),大幅减少了计算量

代码质量 (15/15分)

代码结构清晰,逻辑合理 (5/5分)

函数划分合理,分为判断质数和计数两个函数
主函数结构简洁明了

变量命名规范且有意义 (5/5分)

变量名如isPrime、countPrimes、count等命名清晰
参数和返回值命名符合规范

适当的注释说明算法思路 (5/5分)

每个函数都有注释说明其功能
关键步骤有注释说明(如"只需要检查到sqrt(n)")

客观评价:

客观评价 (70分)

测试用例通过情况

测试用例	输入	期望输出	实际输出	结果	CPU时间(ms)	内存使用(KB)	实际运行时间(ms)
1	10	4	4	通过	0	3584	2
2	5	3	3	通过	0	3456	1
3	20	8	8	通过	0	3584	2

功能正确性 (50/50分)

正确计算1到n之间的质数个数 (30/30分)

所有测试用例都通过,结果完全正确

正确处理边界情况 (10/10分)

代码中正确处理了n<2的情况
正确处理了n=2的特殊情况

输出结果完全准确 (10/10分)

所有测试用例的输出与期望完全一致

输入输出处理 (20/20分)

正确从标准输入读取数据 (10/10分)

使用cin正确读取输入数据

输出格式符合要求 (10/10分)

使用cout输出结果并使用endl换行,符合要求

建议:

改进建议

算法效率优化

考虑使用埃氏筛法 (Sieve of Eratosthenes) 来提高算法效率, 特别是当n较大时埃氏筛法的时间复杂度为 $O(n \log \log n)$, 比当前的 $O(n\sqrt{n})$ 更高效
实现示例:

```
int countPrimes(int n) {
    if (n <= 2) return n == 2 ? 1 : 0;

    vector<bool> isPrime(n + 1, true);
    isPrime[0] = isPrime[1] = false;

    for (int i = 2; i * i <= n; i++) {
        if (isPrime[i]) {
            for (int j = i * i; j <= n; j += i) {
                isPrime[j] = false;
            }
        }
    }

    int count = 0;
    for (int i = 2; i <= n; i++) {
        if (isPrime[i]) count++;
    }
    return count;
}
```

代码健壮性

考虑添加输入验证, 确保n为正整数
对于可能的大数据输入, 考虑添加异常处理

性能优化

对于大数据输入, 可以考虑使用更高效的数据结构, 如位图 (bitmap) 来节省内存
可以进一步优化埃氏筛法, 只考虑奇数, 减少一半的空间和时间消耗

总结:

总结

评分项	满分	得分	扣分原因
功能正确性	50	50	无
输入输出处理	20	20	无
算法效率	15	10	使用了简单遍历方法而非埃氏筛法
代码质量	15	15	无
总分	100	95 = 50 + 20 + 10 + 15	

这份质数计数的代码整体表现非常好, 功能完全正确, 所有测试用例都通过。代码结构清晰, 命名规范, 注释充分, 展示了良好的编程风格。

主要的改进空间在于算法效率方面。当前实现使用了简单的遍历方法判断每个数是否为质数, 虽然有一些优化 (如只检查到 \sqrt{n} 和跳过偶数), 但对于大数据输入, 埃氏筛法会更高效。

总的来说, 这是一份高质量的代码实现, 只需在算法选择上做小幅改进, 就能达到更高的性能水平。

3) 查看学生作业

单击「查看作业」进入该学生工作空间。

学习中心 / C 基础课程 / C/C 学习进度

全部 3 已学完 2 学习中 0 未学习 1

批量执行智能批改 导出得分

昵称	姓名	学号	学习状态	提交时间	得分			操作
					质数个数(100分)	斐波那契数列(100分)	数组排序(100分)	
茶饼茶饼	张三	20250811	未学习		-	-	-	-
Cloud Studio	keiran	20250829	已学完	2025-08-29 12:11	智能批改中	智能批改中	智能批改中	智能批改 查看作业
			已学完	2025-08-29 11:57	未批改	未批改	未批改	智能批改 查看作业

六、调整与确认分数（可选）

1. 打开「得分详情」
2. 输入调整后的分数
3. 单击「修改得分」确认

得分详情

质数个数

95 分

修改得分

确认无误后，可单击页面的确认按钮完成评分。

学习中心 / C 基础课程 / C/C 学习进度

全部 3 已学完 1 学习中 0 未学习 2

批量执行智能批改 导出得分

昵称	姓名	学号	学习状态	提交时间	得分			操作
					质数个数(100分)	斐波那契数列(100分)	数组排序(100分)	
茶饼茶饼	张三	20250811	未学习		-	-	-	-
Cloud Studio	keiran	20250829	已学完	2025-08-29 13:44	100 得分详情	-	100 得分详情	智能批改 确认批改结果
			未学习		-	-	-	-

七、导出学习数据

单击「导出得分」生成 CSV。

学习中心 / C 基础课程 / C/C 学习进度

全部 3 已学完 2 学习中 0 未学习 1

批量执行智能批改 导出得分

昵称	姓名	学号	学习状态	提交时间	得分			操作
					质数个数(100分)	斐波那契数列(100分)	数组排序(100分)	
茶饼茶饼	张三	20250811	未学习		-	-	-	-
Cloud Studio	keiran	20250829	已学完	2025-08-29 12:11	智能批改中	智能批改中	智能批改中	智能批改 查看作业
			已学完	2025-08-29 11:57	未批改	未批改	未批改	智能批改 查看作业

常见导出字段包括：学生信息、学习状态、最后提交时间、每题得分等。

常见问题

为什么看不到智能批改？

请确认课程为非公开课程，且您是所有者/老师/助教。

为什么有的学生无法批改/没有分数？

可能未完成该章节学习、未提交、或章节未完成作业/批改配置；以页面提示为准。

批改能重复执行吗？

可以，重新批改会覆盖/更新结果（以页面提示为准）。

智能批改需要多长时间？

平均每道题目需要 3-4 分钟，具体时间取决于题目复杂度和学生提交的代码量。

批改过程中可以退出页面吗？

可以，智能批改在后台进行，退出页面不会影响批改进度。

如何确保批改结果的准确性？

建议教师查看 AI 生成的详细评分报告，必要时可手动调整分数。

相关文档

- [查看学员学习状况](#)
- [成员管理](#)
- [角色介绍](#)

OJ 自查 (开启)

最近更新时间: 2026-05-19 18:07:22

OJ 自查让学生在提交作业前可以主动运行测试用例, 提前验证代码的正确性。教师开启后, 学生在学习界面即可使用。

说明:

- 自查功能仅适用于 OJ 类型作业, 不适用于其他类型作业。
- 章节必须已配置 OJ 类型作业, 才能开启自查。如何配置请参考作业配置。
- 仅管理角色可操作: 所有者 / 老师 / 助教。具体权限说明查看 [角色介绍](#)。

功能价值

价值	说明
即时反馈	学生无需等待教师批改即可了解代码是否通过
鼓励迭代	降低提交门槛, 学生可以反复修改、多次自查, 逐步完善代码
提升作业质量	经过自查后提交的代码通过率更高, 减少教师反复批改的工作量
丰富教学数据	自查次数、通过情况等过程数据帮助教师更全面地了解学生的学习过程

一、进入章节编辑

在课程详情页, 找到需要开启自查的章节, 单击「[编辑章节](#)」进入编辑界面。

编程语法基础课程

秋季班

暂无描述

4 个章节

14人可访问

课程章节

+ 创建章节

- C++基础语法练习**
暂无描述
1 已学完 1 学习中 12 未学习
[编辑章节](#)
- Python基础语法练习**
暂无描述
0 已学完 0 学习中 14 未学习
- All in One**
暂无描述
0 已学完 0 学习中 14 未学习
- C++基础语法练习**
123
0 已学完 0 学习中 14 未学习

课程算力管理

当前课程将使用课程所有者 的算力。
[了解更多](#)

已使用 2.01 机时, 剩余 97.99 机时 2%

[查看使用情况](#)

我的课程算力

已使用 1 机时, 剩余 1.00 机时 50%

使用课程算力

已开启, 在该课程学习将使用课程算力

AI 代码助手设置

对当前课程所有学习记录应用生效

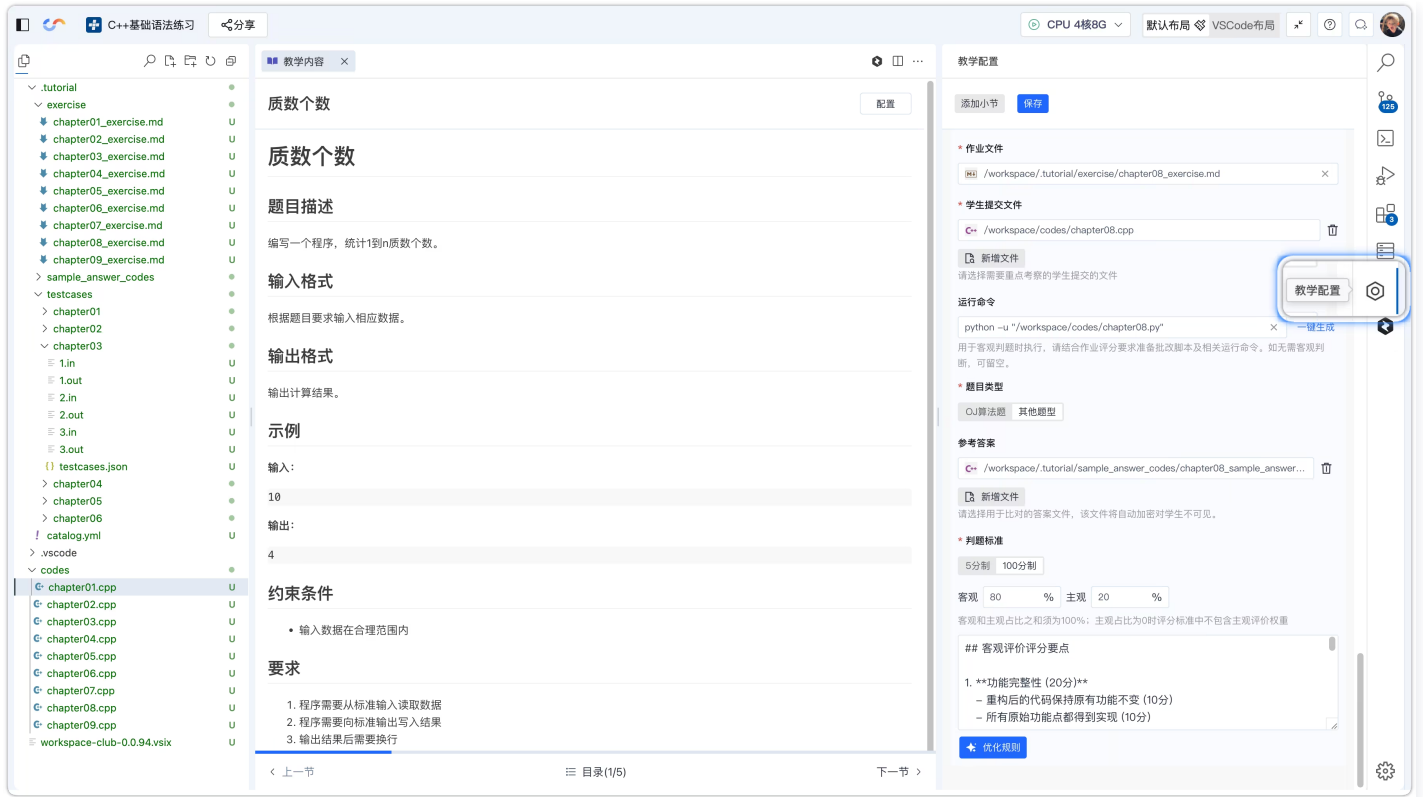
插件整体开关

代码补全

二、开启自查功能

1) 打开教学配置

进入编辑界面后, 单击右侧插件栏的「教学配置」按钮, 展开教学配置面板。



2) 勾选「学员自查」

在教学配置面板顶部，找到「学员自查」选项，勾选前面的复选框即可开启。开启后允许学生在作业过程中调试代码。



● 关闭状态:



● 开启状态:



🔔 重要

开启自查后，请单击「保存」使配置生效。未保存可能导致配置丢失。

三、查看学生自查情况

1) 进入学习进度页

- 方式一：单击章节右侧的「...」菜单，选择「查看学习进度」。



- 方式二：直接单击章节下方的学习状态统计（已学完 / 学习中 / 未学习）进入。



编程语法基础课程
秋季班

编程语法基础课程

暂无描述

4 个章节

编辑 分享 ...

14人可访问

课程章节 + 创建章节

- C++基础语法练习**

暂无描述

1 已学完 1 学习中 12 未学习
- Python基础语法练习**

暂无描述

0 已学完 0 学习中 14 未学习
- All in One**

暂无描述

0 已学完 0 学习中 14 未学习
- C++基础语法练习**

123

0 已学完 0 学习中 14 未学习

课程算力管理

当前课程将使用课程所有者 的算力。
[了解更多](#)

已使用 2.01 机时, 剩余 97.99 机时 2%

查看使用情况

我的课程算力

已使用 1 机时, 剩余 1.00 机时 50%

使用课程算力

已开启, 在该课程学习将使用课程算力

AI 代码助手设置

对当前课程所有学习记录应用生效

插件整体开关

代码补全

2) 查看自查相关数据

在学习进度页面中，可以查看每位学生的详细信息：

AI实训中心 / 编程语法基础课程 / C 基础语法练习学习进度

全部 14 已学完 1 学习中 1 未学习 12

Q 姓名/学号

+ 批量执行智能批改 导出得分

昵称	姓名	学号	学习状态	提交时间	实训总耗时	通过状态	批改次数	操作
			已学完	26-05-14 16:11	258天4小时14分	3/3	11	智能批改 查看作业

其中与自查相关的关键字段：

字段	说明
通过状态	显示学生 OJ 题目的通过情况，如 3/3 表示 3 题全部通过
批改次数	学生在当前章节执行自查的总次数

昵称	姓名	学号	学习状态	提交时间	实训总耗时	通过状态	批改次数	操作
			✔ 已学完	26-05-14 16:11	258天4小时14分	3/3	11	智能批改 查看作业

昵称	姓名	学号	学习状态	提交时间	实训总耗时	通过状态	批改次数	操作
			✔ 已学完	26-05-14 16:11	258天4小时14分	3/3	11	智能批改 查看作业

通过这些数据，教师可以了解学生在提交前的自查情况和代码迭代过程。

相关文档

- [智能批改（作业配置）](#)
- [查看学员学习状况](#)
- [角色介绍](#)

学生端

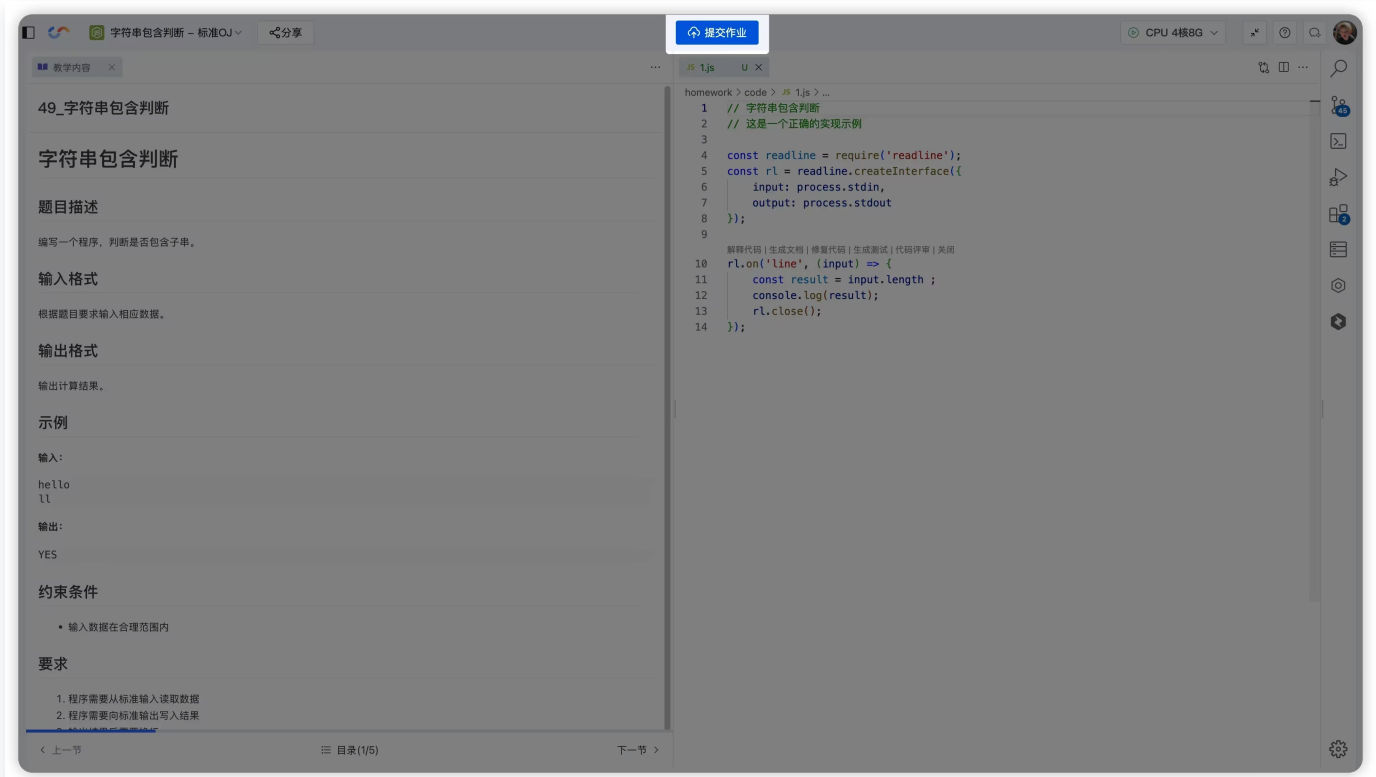
关于作业

最近更新时间：2026-05-19 18:07:22

本页介绍两件事：**如何提交作业**、**如何查看批改结果**。

一、提交作业

1. 按作业要求完成代码/文件
2. 在作业页面顶部单击「**提交作业**」



3. 看到“**提交成功**”提示即表示提交完成

作业已提交

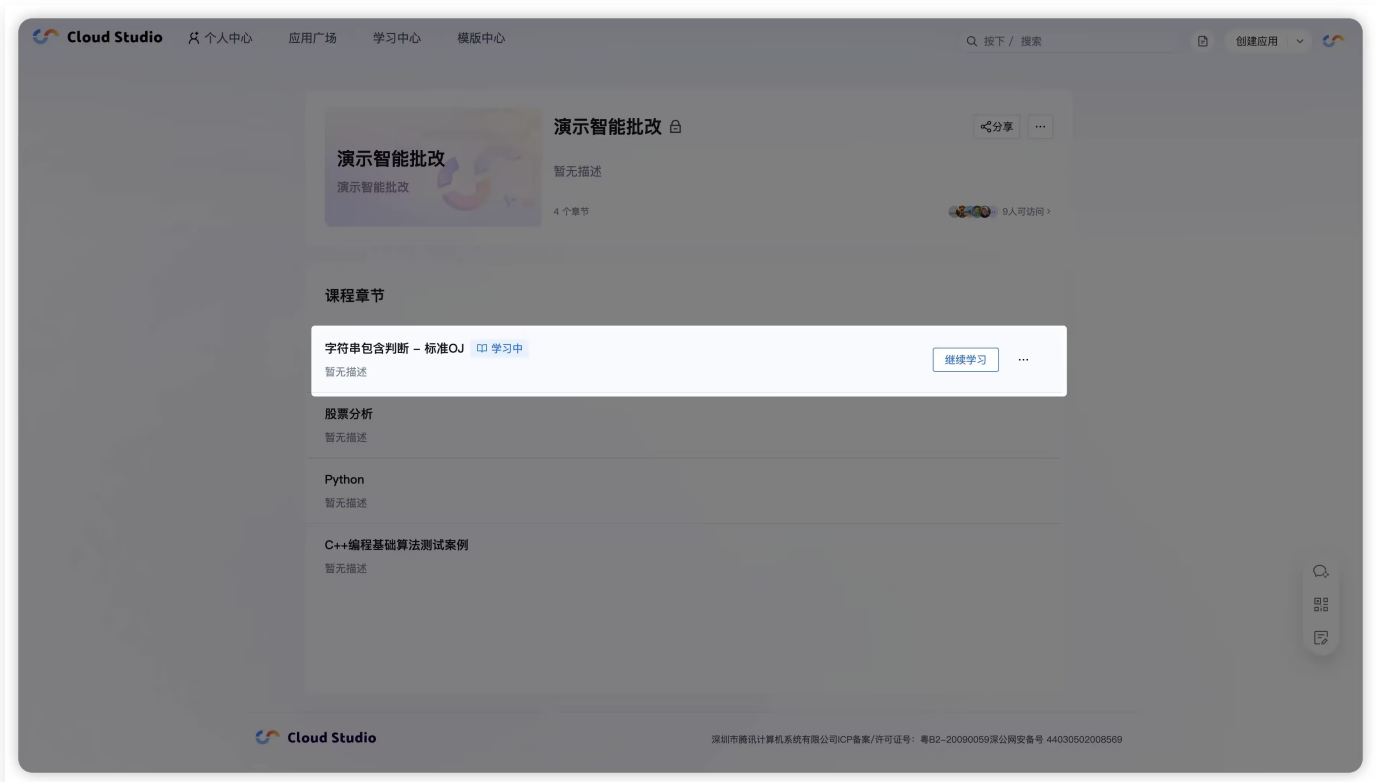
复制链接并关闭当前标签页，方便老师查看作业

```
https://cloudstudio.net/a/30013011113463808/learn?channel=share&sharetype=SUBMIT
```

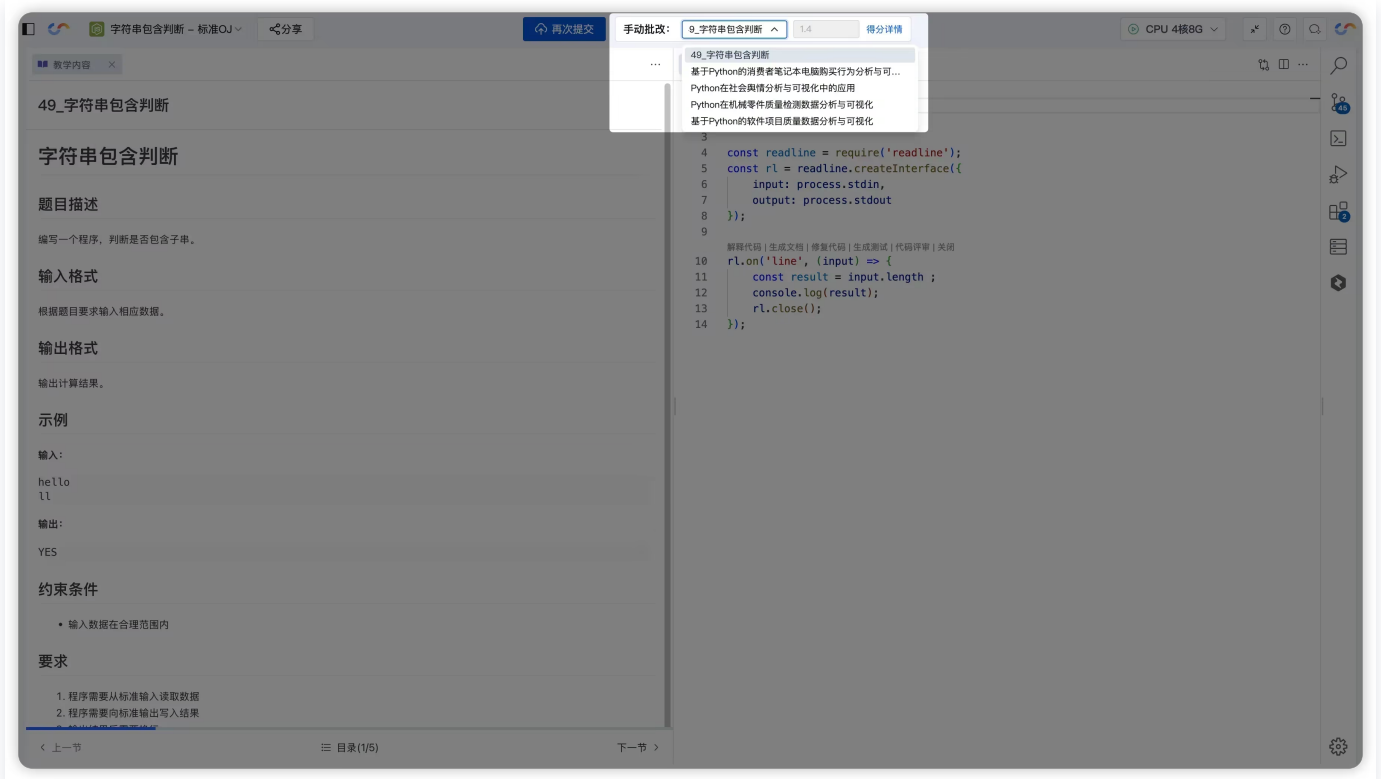
复制链接并关闭

二、查看批改结果

1. 打开作业章节，单击「继续学习」进入作业页面



2. 在顶部栏查看每道题的得分



3. 单击「得分详情」查看扣分原因与改进建议

得分详情



49_字符串包含判断

得分:

1.4 分

主观评价:

主观评价 (40分)

代码结构 (18/25分)

代码逻辑清晰, 结构合理 (10/15分)

代码结构基本合理, 使用了Node.js的readline模块处理输入输出
但代码逻辑与题目要求不符, 没有实现字符串包含判断功能

变量命名规范 (8/10分)

变量命名简单但清晰
使用了符合JavaScript规范的驼峰命名法

代码质量 (7/15分)

代码逻辑清晰, 易于理解 (3/8分)

代码结构简单明了
但实现的功能与题目要求完全不符

变量命名有意义, 符合编程规范 (3/4分)

变量名result、input等命名合理
整体符合JavaScript编程规范

适当的注释说明算法思路 (1/3分)

有基本的文件头注释
缺少关键算法步骤的注释说明
没有实现算法, 因此无法评价算法思路的注释

客观评价:

常见问题

能否再次提交?

若页面显示「再次提交」, 说明支持重新提交; 是否计入最终成绩以老师/课程规则为准。

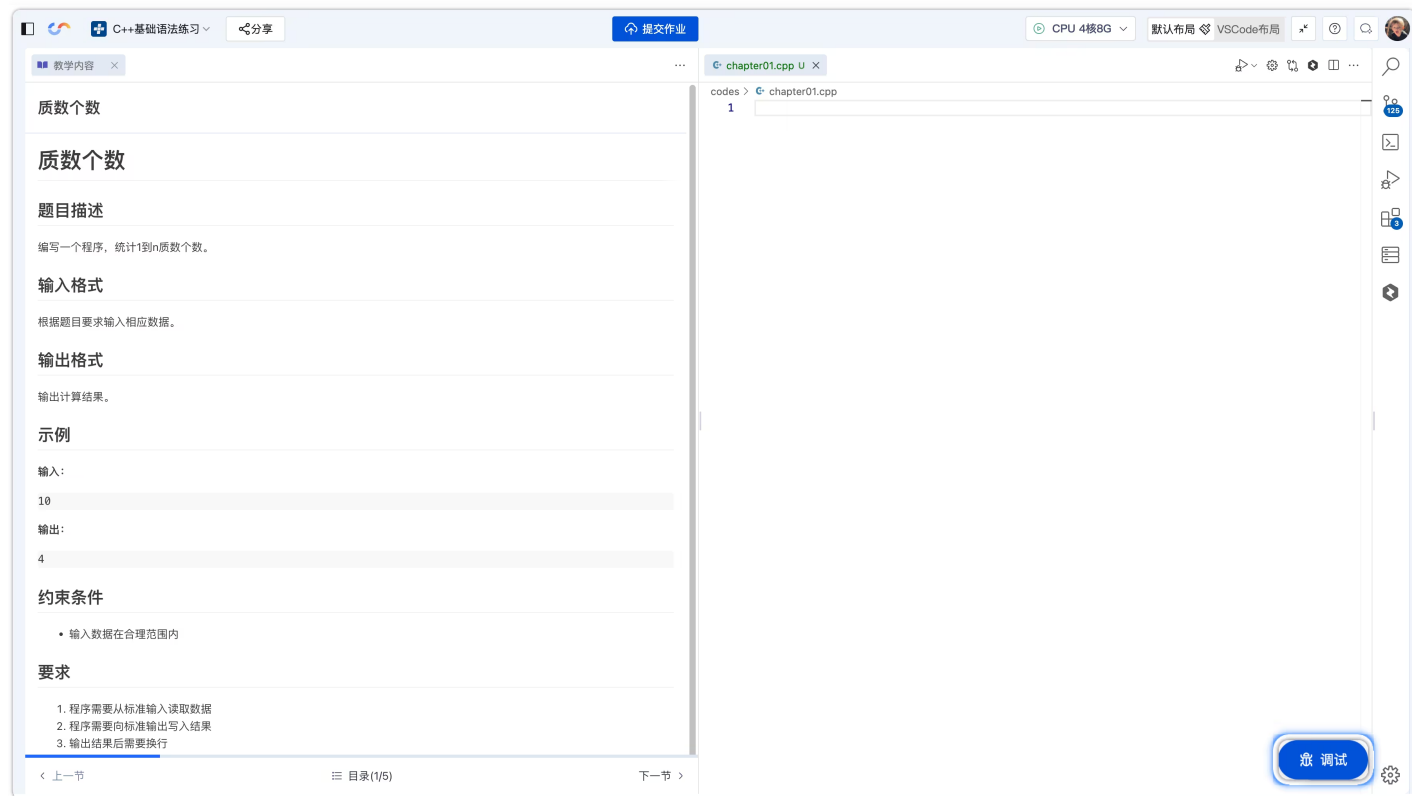
OJ 自查 (使用)

最近更新时间: 2026-05-19 18:07:22

OJ 自查允许学生在提交作业前主动运行测试用例, 提前验证代码是否正确, 避免盲目提交。

一、确认自查功能是否可用

进入课程章节的学习界面后, 查看页面右下角是否出现「调试」图标。如果出现, 说明当前章节已开启自查功能。



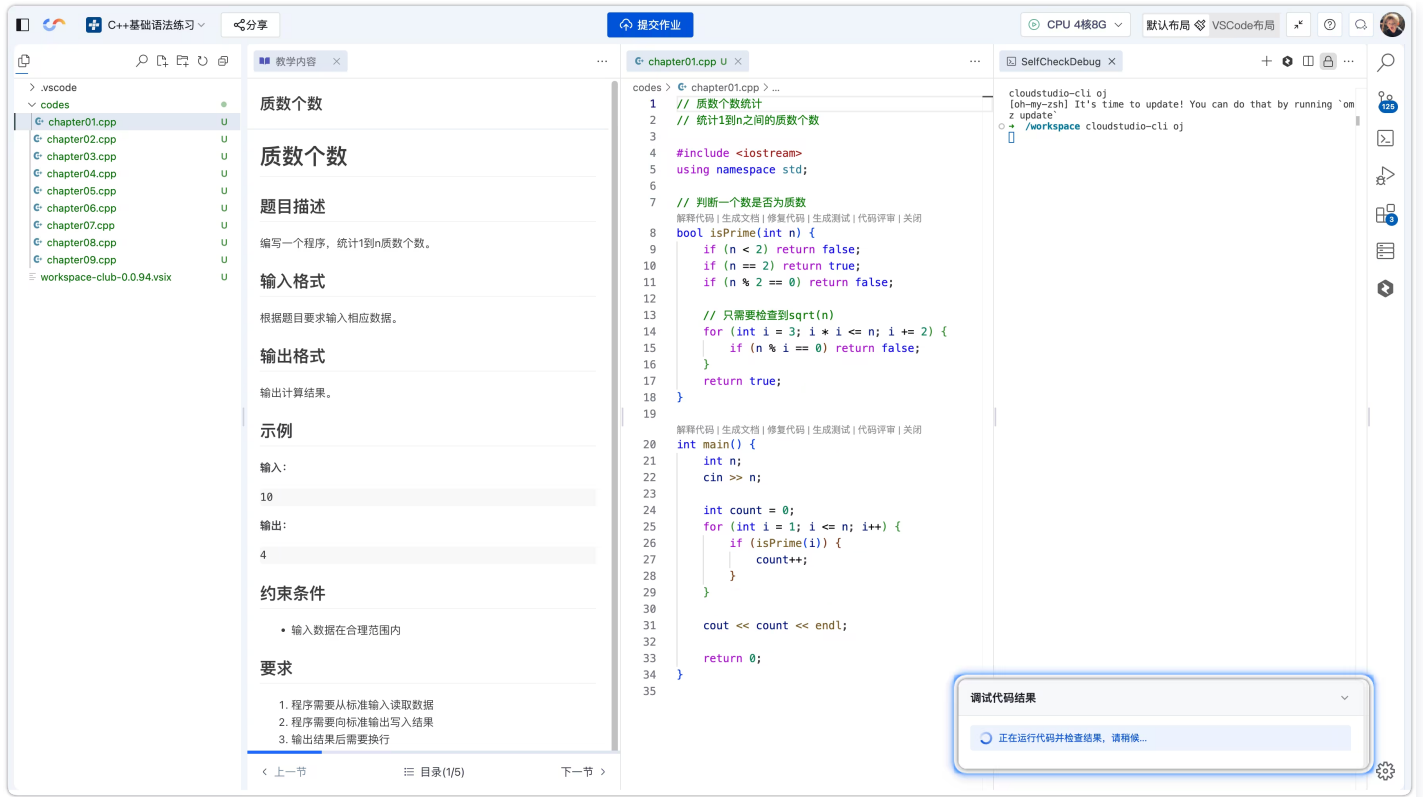
! 说明

自查功能由教师配置开启, 并非所有章节都支持。如果右下角没有该图标, 说明当前章节未启用自查。

二、完成作业并执行自查

1. 按照题目要求完成代码编写
2. 单击右下角的「调试」图标, 触发自查

单击后, 系统会在终端中自动运行测试命令, 右下角弹出「调试代码结果」面板并显示"正在运行代码并检查结果, 请稍候..."。



注意

自查运行期间, 建议不要进行其他操作。

三、查看自查结果

运行完成后, 「调试代码结果」面板会列出所有题目及其结果:

- ✔ **通过**: 该题目测试用例全部通过
- ✘ **未通过**: 该题目存在未通过的测试用例

调试代码结果

题目	结果
质数个数	✔ 通过
斐波那契数列	✘ 未通过
数组排序	✘ 未通过

✘ 存在未通过的题目

重新运行
提交作业

只要存在任意一道未通过的题目, 面板底部会显示红色提示「存在未通过的题目」。

调试代码结果

题目	结果
质数个数	✓ 通过
斐波那契数列	✗ 未通过
数组排序	✗ 未通过

✗ 存在未通过的题目

重新运行 提交作业

四、修改代码并重新自查

如果有题目未通过，可以根据提示修改代码，然后单击面板中的「重新运行」按钮再次自查。

调试代码结果

题目	结果
质数个数	✓ 通过
斐波那契数列	✗ 未通过
数组排序	✗ 未通过

✗ 存在未通过的题目

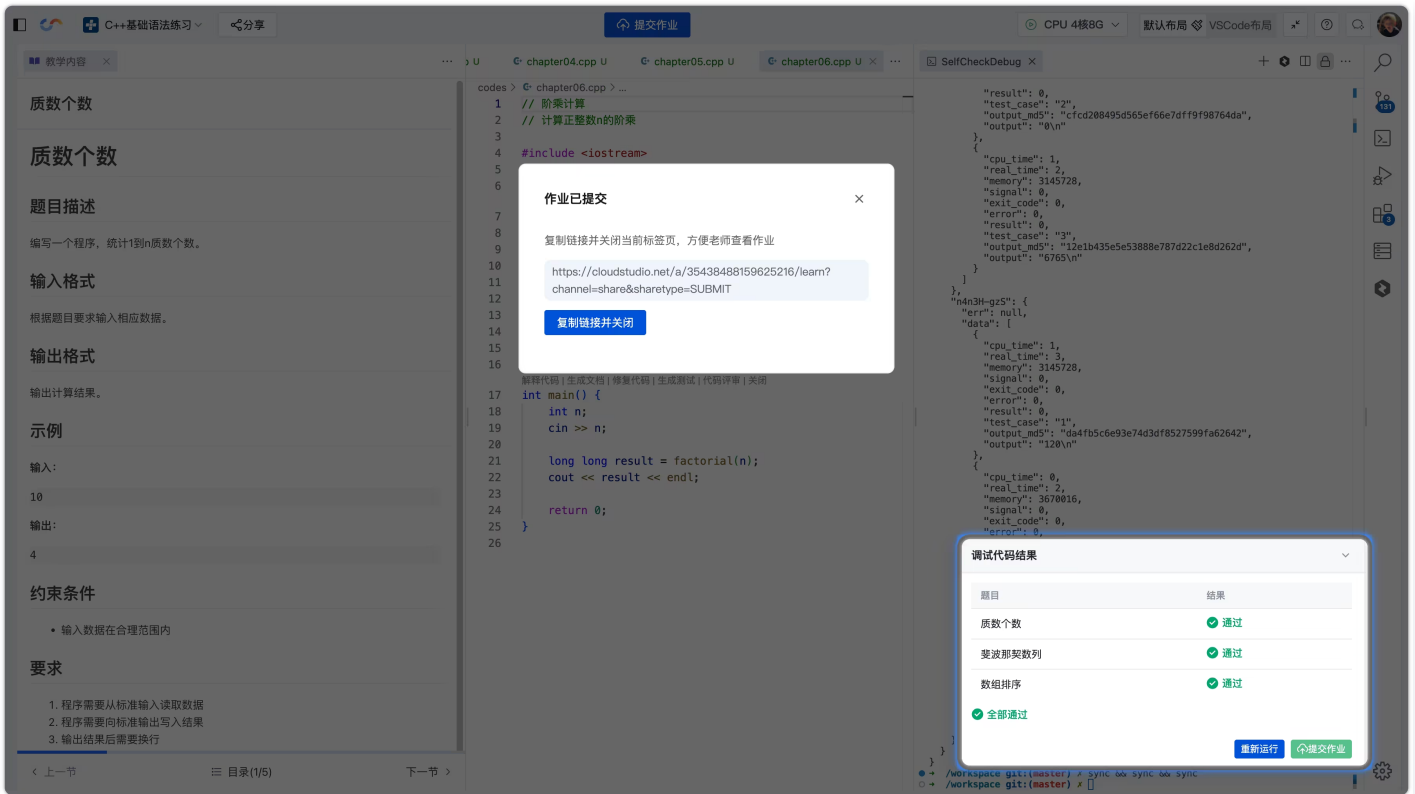
重新运行 提交作业

你可以反复修改和重新自查，直到所有题目都显示为「通过」。

五、提交作业

情况一：自查全部通过

所有题目通过后，单击「提交作业」即可。



情况二：自查未全部通过仍提交

你也可以在自查未全部通过时直接单击「提交作业」。



此时系统会弹窗提醒：「自查未全部通过——当前自查结果存在未通过的题目，提交后将无法再次提交。确定要继续提交吗？」

自查未全部通过



当前自查结果存在未通过的题目，提交后将无法再次提交。确定要继续提交吗？

返回修改

继续提交

- 单击「返回修改」：回到编辑界面继续修改代码
- 单击「继续提交」：确认提交作业

提交成功

提交成功后，页面会弹出「作业已提交」提示，同时提供作业链接，可复制链接并关闭当前标签页。

作业已提交



复制链接并关闭当前标签页，方便老师查看作业

```
https://cloudstudio.net/a/35437536018538496/learn?channel=share&sharetype=SUBMIT
```

复制链接并关闭