OJ 新生使用说明

OJ 地址： https://www.huznuoj.com/

> 本说明面向第一次接触 Online Judge（OJ）的同学。
>
> 请按照“注册账号 → 调整界面 → 找到训练 → 阅读题目 → 编写与测试代码 → 正式提交 → 查看评测结果”的顺序完成操作。
>
> **请注意：C 语言基础训练位于顶部“比赛”栏目中，但这些训练仅用于学习与练习，不作为排名依据。对应场次结束后，仍可在“题库”中继续完成题目。**
>
> 第一次使用本站时，建议完整阅读本说明，并先熟悉一次“打开题目 → 在线运行 → 正式提交 → 查看评测结果”的流程。

---

1. 什么是 OJ?

OJ（Online Judge，在线评测系统）用于自动检查程序是否正确。

完成一道题时，你需要：

1. 阅读题目，理解输入和输出要求；
2. 编写程序；
3. 将代码提交到 OJ；
4. OJ 会使用多组测试数据运行你的程序；
5. 只有程序在规定时间和内存限制内，对所有测试数据都输出正确答案，才算通过该题。

因此，**“样例运行正确”不等于“这道题已经通过”**。样例只是帮助你理解题意，最终结果以正式提交后的评测结果为准。

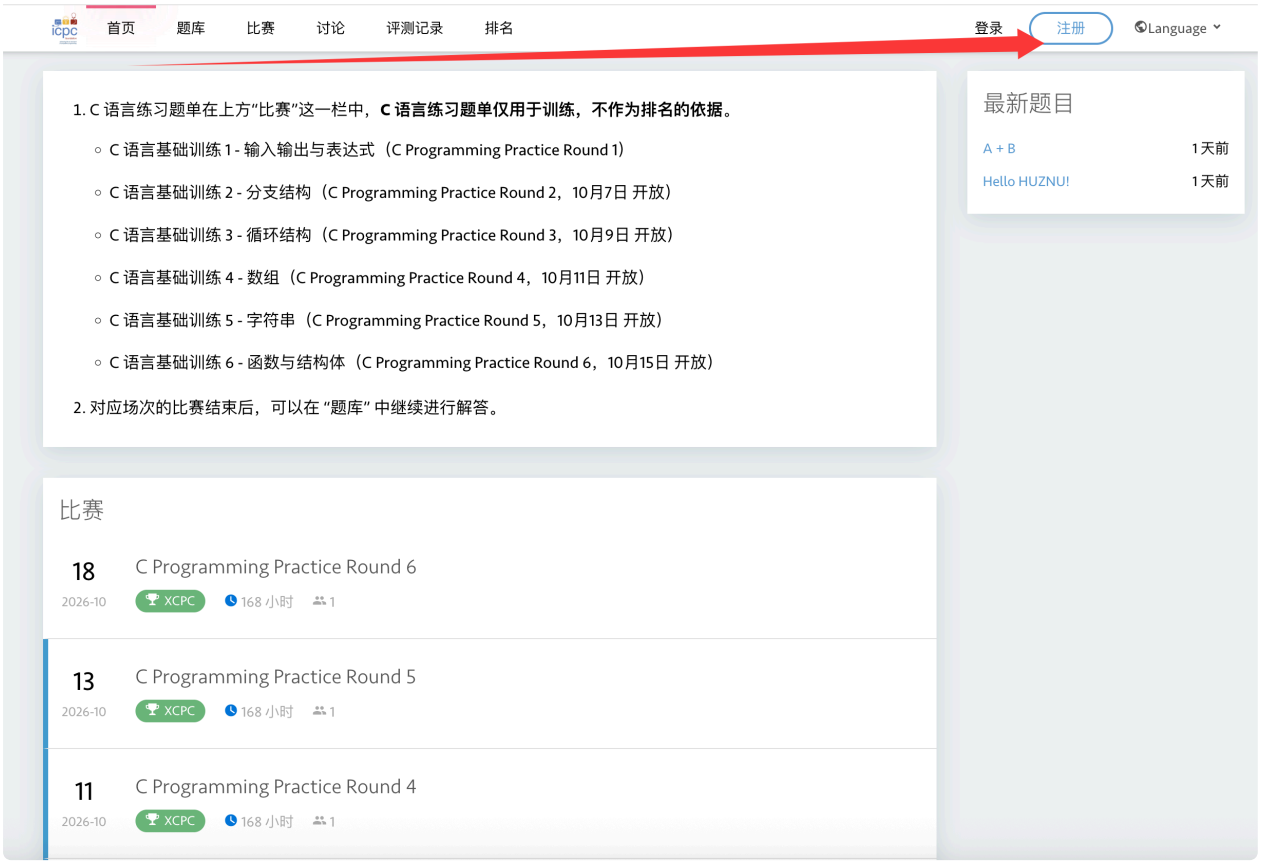
---

2. 注册账号

2.1 进入注册页面

打开 HUZNU OJ 首页，在页面右上角找到 **“注册”**，点击进入注册页面。

如果你已经有账号，可以直接点击旁边的 **“登录”**，不需要重复注册。



2.2 填写邮箱并发送验证邮件

进入注册页面后：

1. 在“电子邮件”一栏填写自己能够正常接收邮件的邮箱；
2. 点击 **“发送验证邮件”**；
3. 等待系统发送注册邮件。

请检查垃圾邮件或广告邮件文件夹，不要连续、频繁地重复点击发送。



2.3 打开邮箱完成验证

打开收到的注册邮件，点击邮件中的注册链接。

请直接点击系统发送的注册链接，不要把注册链接转发给其他人。注册链接仅用于完成你自己的账号注册。



2.4 设置用户名和密码

打开注册链接后，按照页面要求填写：

- 用户名；

- 密码；
- 重复密码。

确认两次密码一致后，点击 **“注册”**。

用户名格式随意，建议使用长期稳定、便于识别的名称。

注册完成后，即可使用该账号登录 OJ。



3. 将界面语言设置为简体中文

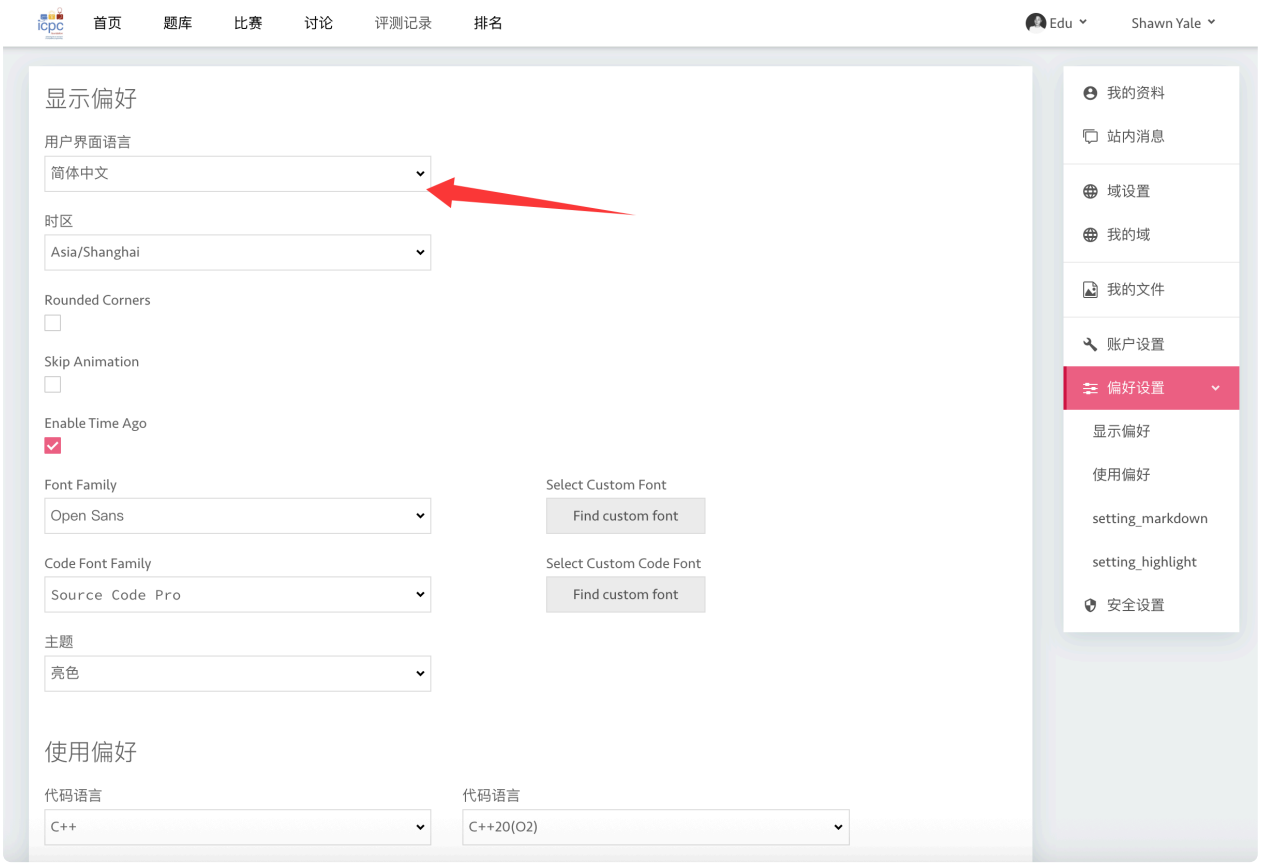
首次登录会自动跳转 **偏好设置**。

建议将 **“用户界面语言”** 设置为 **“简体中文”**。

时区建议使用 **Asia/Shanghai**。其他显示选项如果没有特殊需求，可以暂时保持默认设置。

修改完成后，下滑，点击 **“保存所有修改”**。

如果你修改设置后部分页面没有立即变化，可以刷新页面后重新查看。



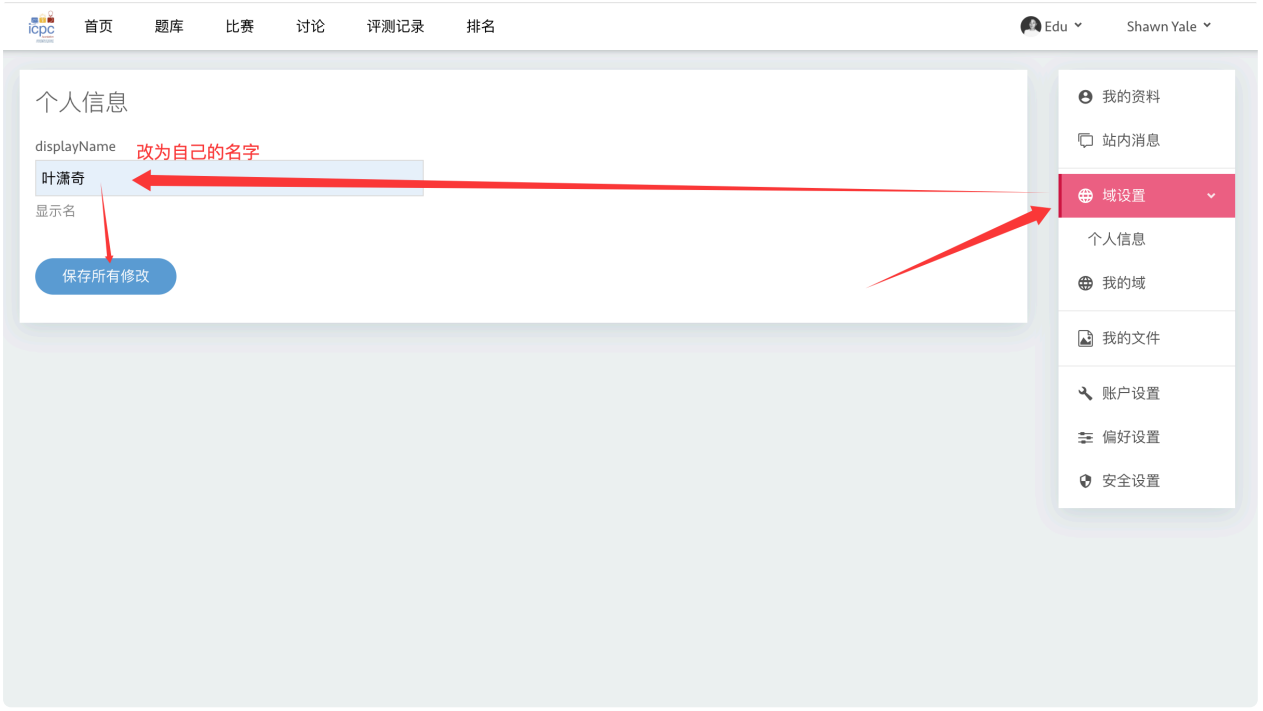
4. 将显示名修改为自己的姓名

为了方便识别，请将自己的显示名设置为本人姓名。

在菜单中进入：**域设置**（或将鼠标移动到右上角自己的用户名，点击“@ Edu”）。

找到 `displayName`，将其中的内容修改为自己的姓名，然后点击 **“保存所有修改”**。

这里修改的是 OJ 页面上显示的名称。请不要随意填写难以辨认的昵称、乱码或与本人无关的名称。



5. 认识 OJ 的主要栏目

登录后，页面顶部通常可以看到以下栏目：

- **首页**：查看各种信息；
- **题库**：查找和练习已经公开的题目；
- **比赛**：进入正在进行或即将开始的训练、比赛；
- **讨论**：查看或参与题目、比赛相关讨论 / 题解；
- **评测记录**：查看提交的代码及评测结果（默认查看自己的，可以点击右上角“重置”查看所有人的代码）；
- **排名**：查看对应的排名信息（每晚12点刷新）。

对于刚开始学习 C 语言的同学，最常使用的是 **“比赛”**、**“题库”** 和 **“评测记录”**。

6. 如何使用题库

6.1 进入题库

点击页面顶部的 **“题库”**，即可进入题目列表。

题库中的每一行对应一道题。通常可以看到题号、题目名称、通过情况以及 Rating 等信息。

6.2 Rating 是什么?

题库中的 **Rating** 用来表示题目的大致难度。

本站当前最低 Rating 为 **800**。一般情况下，Rating 越高，题目越难。

对于刚开始学习 C 语言的同学，不需要过度关注 Rating。优先按照课程安排和 C 语言基础训练的顺序完成题目即可。

6.3 使用题库筛选功能

题库右侧提供了筛选功能，可以按照以下条件查找题目：

- 题目分类；
- Rating 区间；
- 做题状态，如“全部”“未 AC”“已 AC”等。

例如，当你只想查看尚未通过的题目时，可以选择 **“未 AC”**。

如果右侧筛选区域显示异常、布局错乱或没有正常加载，可以先尝试强制刷新页面：

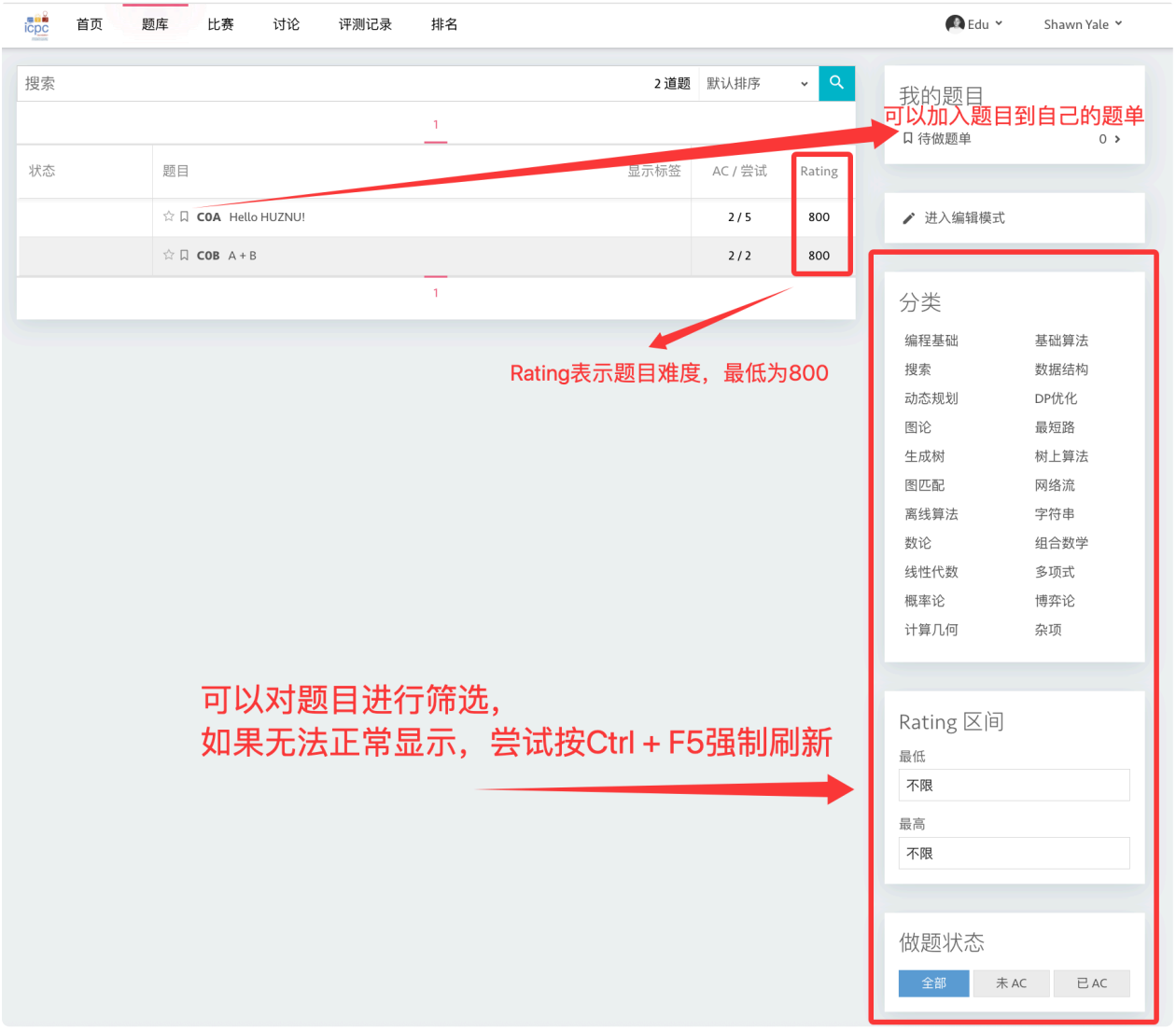
- Windows：通常可使用 `Ctrl + F5`；
- macOS（Chrome 等浏览器）：通常可使用 `Command + Shift + R`。

6.4 将题目加入自己的题单

在题目列表中，可以点击“加入待做题单”，将题目加入自己的题单；之后可以在右侧的 **“我的题目”** 中查看。

这个功能适合保存：

- 暂时不会、准备以后再做的题目；
- 需要复习的题目；
- 自己认为比较典型的题目。



7. 如何参加 C 语言基础训练

7.1 进入“比赛”页面

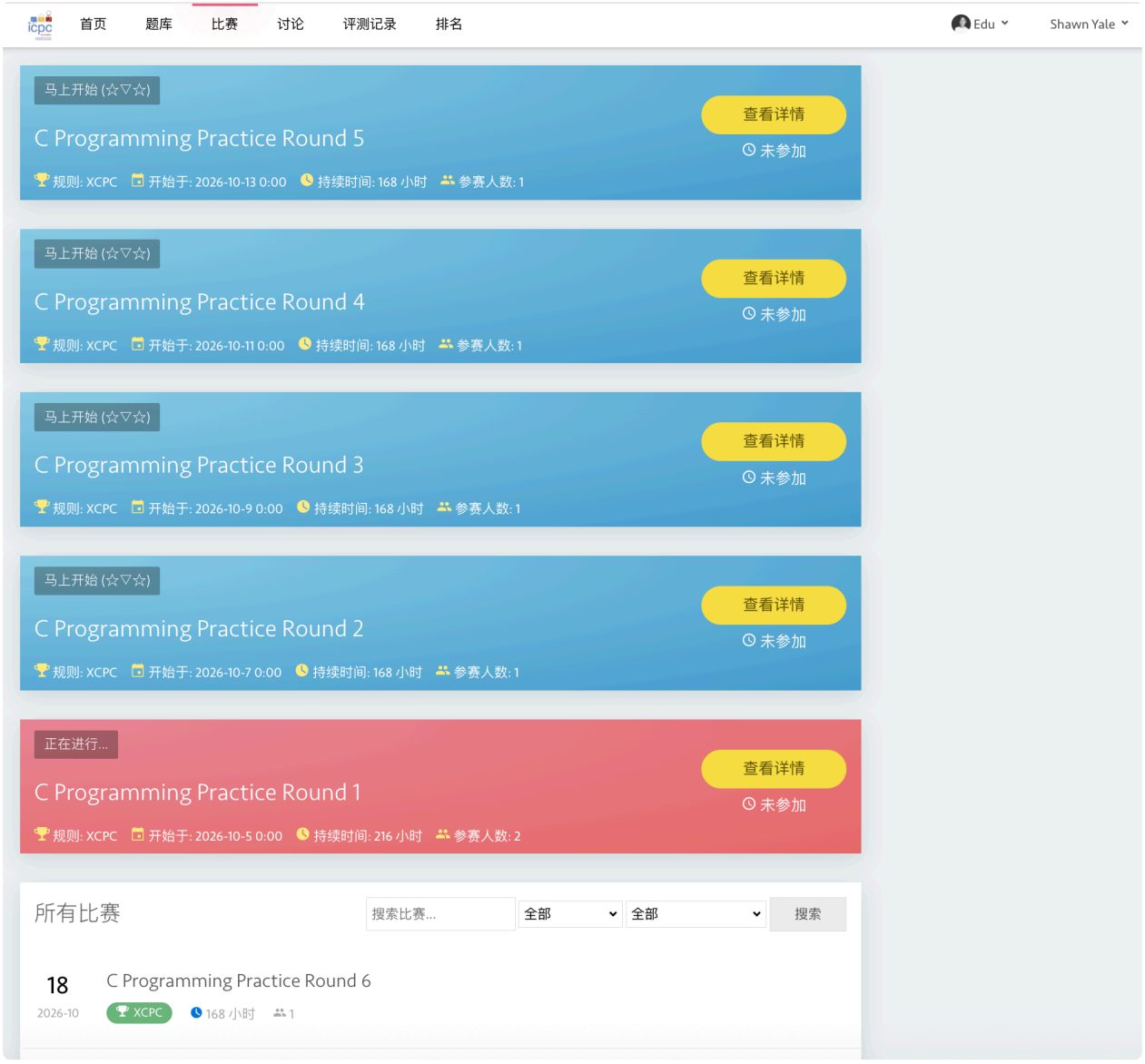
点击顶部的 **“比赛”**。

你会看到 C 语言基础训练的各个场次，例如：

- C Programming Practice Round 1；
- C Programming Practice Round 2；
- ……
- C Programming Practice Round 6。

蓝色卡片表示即将开始的场次，红色卡片表示正在进行的场次。

点击对应场次右侧的 **“查看详情”** 进入比赛页面。



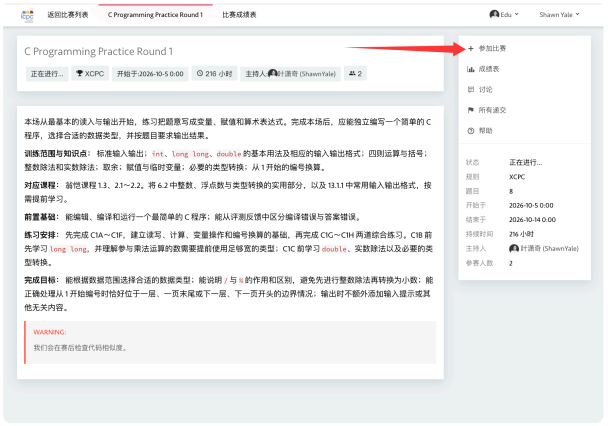
7.2 参加训练

第一次进入某一场训练时，右侧会看到 **“参加比赛”**。

点击它即可加入本场训练。

加入成功后，页面会显示 **“已参加”**。

这些 C 语言练习场主要用于训练，不必因为“比赛”两个字感到紧张。按自己的学习进度完成即可。



7.3 打开题目列表

加入训练后，在右侧点击 **“题目列表”**。

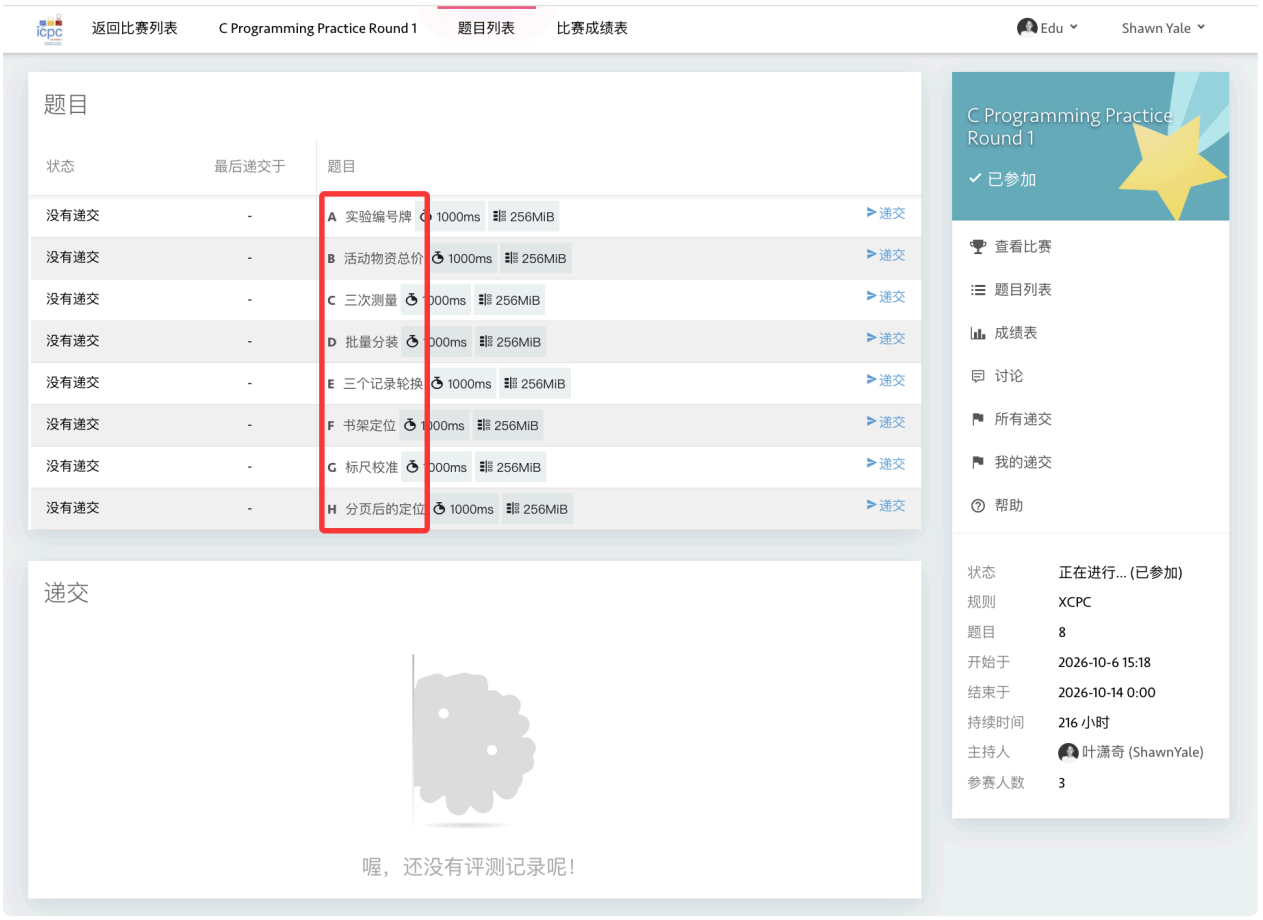
进入后可以看到本场训练包含的所有题目。



题目通常按照 A、B、C、D……排序。对于基础训练，建议按照题号从前往后完成，不要一开始随机跳题。

列表中还会显示每道题的时间限制、内存限制以及当前提交状态。

如果显示 **“没有递交”**，表示你还没有向 OJ 正式提交过这道题。



8. 题目页面由哪些部分组成?

点击题目名称后，就会进入题目详情页。



一道标准题目通常包含以下内容：

8.1 题意

告诉你要解决什么问题。

不要看到题目后立刻开始写代码。先确认自己能够用一句话说明：

“程序读入什么，需要计算什么，最后输出什么。”

8.2 输入格式

告诉你程序会从标准输入中读到什么数据。

例如：

```
3 12
```

表示 OJ 在运行程序时会自动把这些数据提供给你的程序。

不要把样例中的 `3 12` 直接写死在代码里。

正确做法是使用 `scanf` 等输入函数读取数据。

8.3 数据范围

例如：

```
1 <= g, s <= 100
```

数据范围会帮助你判断：

- 应该使用什么数据类型；
- 程序需要处理多大的数据；
- 某种写法是否可能超时。

在目前的 C 语言基础训练中，首先要学会根据数据范围正确选择 `int`、`long long`、`double` 等类型。

8.4 输出格式

告诉你最终必须输出什么。

OJ 会按照题目要求检查输出，因此不要随意增加额外文字。

例如题目只要求输出：

```
12
```

就不要输出：

```
The answer is 12
```

除非题目明确要求这样输出。

8.5 样例输入与样例输出

样例用于帮助你理解题意，并方便你在提交前进行测试。

样例通过只能说明程序通过了这一组数据，不能证明程序一定正确。

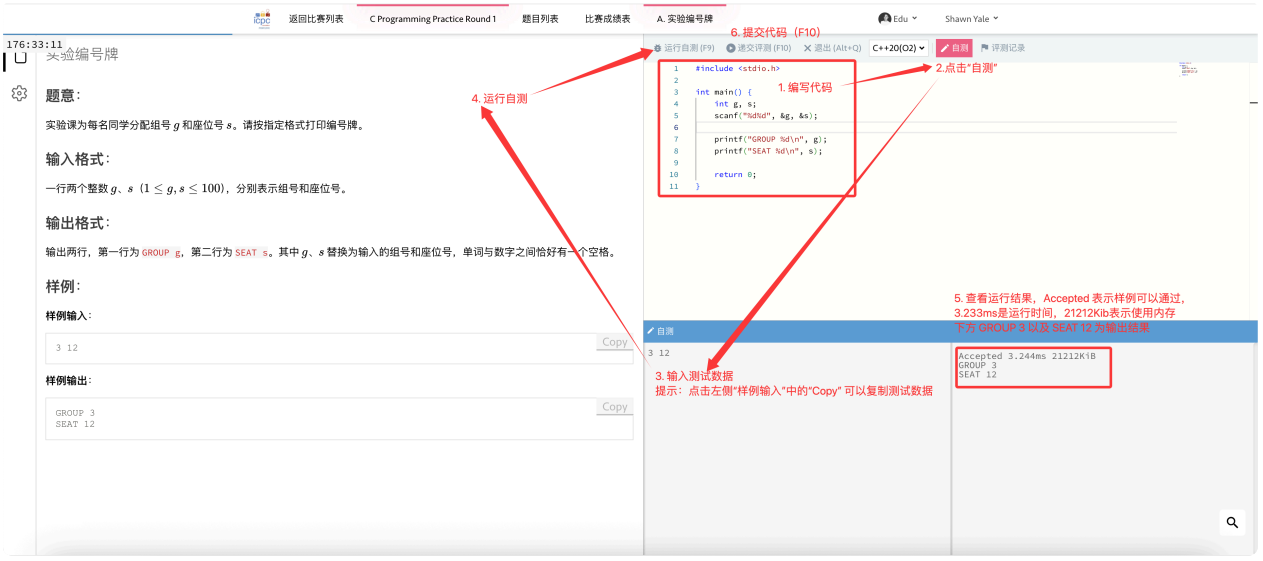
9. 编写、运行和测试代码

注意：有本地编译器的同学可以在本地自行操作，不需要使用在线编辑模式。

9.1 进入在线编辑模式

在题目页面右侧可以点击 **“进入在线编辑模式”**，也可以按照页面提示使用快捷键进入。

进入后，页面会同时显示题目、代码编辑器、测试输入和运行结果区域。



9.2 编写代码

一个最基本的 C 程序可以写成：

```c
#include <stdio.h>

int main(void)
{
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d\n", a + b);
```

```
    return 0;
}
```

注意：

- 输入使用标准输入，例如 `scanf`；
- 输出使用标准输出，例如 `printf`；
- 不需要输出“请输入一个整数”之类的提示语；

9.3 输入测试数据

在测试输入区域填入你希望测试的数据。

最开始可以直接使用题目给出的样例输入。题目样例旁边通常有 **Copy** 按钮，可以先复制，再粘贴到测试输入区域。

例如题目的样例输入是：

```
3 12
```

那么测试时就在输入区域填入：

```
3 12
```

而不是把它写进代码。

9.4 点击运行

点击编辑器上方的 **“运行”** 按钮。

系统会使用你填写的测试数据运行当前代码，并在结果区域显示程序输出以及运行状态。

如果运行结果显示 **Accepted**，表示这一次自定义测试正常完成。

请再次注意：

这里的 Accepted 只代表当前测试数据通过，不代表正式评测已经 AC。

9.5 对照输出结果

运行后，需要自己检查程序实际输出是否与题目的期望输出一致。

特别注意：

- 是否多输出了说明文字；
- 是否少输出或多输出了一项数据；
- 换行是否合理；

- 大小写是否正确；
- 题目要求固定单词时是否拼写正确。

确认样例正确后，还应该自己构造几组简单数据进行测试，例如最小值、最大值、特殊边界值等。

10. 提交代码

10.1 “运行”和“提交”有什么区别?

运行：

只使用你自己填写的测试数据运行一次程序，主要用于调试。

提交：

将代码交给 OJ，系统会使用多组你看不到的正式测试数据进行评测。

因此正确流程应当是：

写代码 → 用样例运行 → 自己补充测试 → 确认无误 → 正式提交。

10.2 提交代码

确认代码无误后，点击页面上的 **“提交”** 按钮。截图中的在线编辑界面也标出了提交入口及对应快捷键。

提交后通常会显示评测结果，也可以点击顶部『评测记录』查看自己的提交。

11. 怎么看评测结果

11.1 最重要的结果：Accepted / AC

如果正式评测结果显示：**Accepted**

通常简称 **AC**，表示你的程序已经通过该题全部测试数据。

只有看到正式评测的 **Accepted / AC**，这道题才算真正完成。

11.2 常见评测状态

状态	常见缩写	含义	首先检查什么
Accepted	AC	答案正确	已通过，无需修改
Wrong Answer	WA	输出答案错误	题意、公式、边界、输出格式

状态	常见缩写	含义	首先检查什么
Compile Error	CE	编译失败	语法错误、变量名、括号、分号、语言选择
Runtime Error	RE	运行时错误	数组越界、除以 0、非法访问等
Time Limit Exceeded	TLE	超过时间限制	死循环、算法过慢
Memory Limit Exceeded	MLE	超过内存限制	数组过大、内存使用过多

出现 WA、CE 等结果是正常的学习过程。根据报错或失败原因修改代码，然后再次提交即可。