

Echo Judge 自定义答案校验器使用说明

首先，我们衷心的感谢您对 Echo Judge 的信任。

如果您在使用传统比较方式的基础上想扩展答案比较的方法，请您仔细阅读本篇文章。

交互器的编写方法

Echo Judge 提供了交互测评模式，在此模式下出题人被要求编写一个交互器以完成对选手的交互信息进行应答，下面我们将介绍 Echo Judge 交互器 的编写方法。

在运行您的交互器之前，我们将会通过命令行提供五个入口参数，您可以在交互器中获取这些参数的值，它们分别是。

`argv[1]` 输入文件

`argv[2]` 自定义校验器输入文件

`argv[3]` 测试点分值

`argv[4]` 得分输出文件

`argv[5]` 错误报告输出文件

以上的五个入口参数将在出题人提交题目时确定，当且仅当题目发生改变时入口参数会更改。

Echo Judge 会在选手程序开始运行时将 输入文件 中的内容 输入进选手程序 中，同时您也可以通过文件读取的方式获取这些内容。与此不同的是，自定义校验器输入文件 中的内容不会被选手程序得到，这仅供您的 自定义校验器 使用。

值得注意的是，如果题目在提交时并未指定 输入文件 或者自定义 校验器输入文件 那么它们将会通过一个长度为 4 的字符串 **null** 传入您的程序。

测试点分值内容为当前测试点的 最高分值，您可以在得分输出文件中输出本测试点选手得分，若您输出的得分高于测试点分值，将会以 自定义校验器崩溃 作为结果返还给选手，且选手在本测试点的得分将记作 **0**。

您可以将您想显示给选手的信息输出至 错误报告输出文件 中，只要您错误报告中含有任何错误信息，无论当前测试点的选手程序是否正确，错误报告都会被返还给选手。

当然，您所生成的 得分情况 以及 错误报告 会在当前测试点测试结束时被 删除。

交互器与选手程序交互的方法

您的交互器将可以通过 标准输入输出流 与选手程序进行交互。

由于系统默认在缓冲区被 充满 后发送缓冲区内容，因此请您在输出信息后 手动刷新缓冲区。

您可以使用以下的方法手动刷新缓冲区。

C++

```
fflush(stdout);
```

java:

```
System.out.flush();
```

Python:

```
stdout.flush()
```

Pascal:

```
flush(output)
```

使用其他语言请参见语言文档。

交互器的注意事项

您的交互器可以随时通过 **标准错误输出流** 输出长度为 **15** 的字符串 **System_Shutdown** 来停止选手程序的运行
输出结束信息后请 **刷新标准错误输出流**。

Echo Judge 不会检测选手交互语句的格式是否合法，若有需要，请您在交互器中对选手交互语句进行检查。

若您的交互器并未处理这一特殊情况且选手程序所输出的交互语句格式不合法，那将会导致选手程序以及您的交互器均因 **超时** 结束运行，超时时限见下方交互器终止运行条件。

当且仅当出现以下情况时，选手的程序会被终止运行

- 选手程序正常退出
- 选手程序因访问无效内存等错误崩溃
- 选手程序运行超时
- 交互器通过标准错误输出流输出长度为 **15** 的字符串 **System_Shutdown**
- 交互器结束运行

当且仅当出现以下情况时，交互器会被终止运行

- 交互器正常退出
- 交互器因访问无效内存等错误崩溃
- 交互器运行超时 (选手程序限制时间 + **5s**)

使用过程中遇到任何错误，请您联系开发组

我们将提供以下的联系方式：

- 大喊大叫
- 群发邮件

不保证您所反馈的信息会被全部接收