

2016 年 TEG 乐趣编程对抗赛 —— 伪装者

比赛分成两部分：「伪装」和「辩伪」，比赛形式为两两对抗淘汰赛。

第一部分【伪装】

1) 题目内容

根据组委会给出的一对范例，你需要写一个程序。该程序应该能够完成以下操作：依次将多份储存成文件格式的 C++ 代码进行【伪装】，并将【伪装】的代码输出到指定的文件。此处的【伪装】一词指的是：

- 程序输出的代码与输入的代码在功能上保持一致性；
- 您输出的代码被编译后，应该可以在组委会的平台上通过对应题目的所有测试用例；
- 输出的代码与输入的代码在逻辑（如 if、else 顺序），语法选择（如 for 和 while），文本改变（修改变量名）等各个方面差异被你所写的程序最大化。

2) 文件要求

请将你的可执行文件最终名称命名为 **submit**（通过 makefile 生成），接受一个参数 x ，这个参数为待伪装的代码个数。输入的代码都是可以编译、运行的代码。输入的文件相对于可执行文件处于 ./input/ 文件夹下，输入的文件分别为 1.cpp, 2.cpp ... x.cpp 输出的代码文件应该输出到相对于可执行文件处于 ./output/ 文件夹下，输出的命名也分别为 1.cpp, 2.cpp ... x.cpp（也就是与输入的文件名保持一致）。

3) 判分说明

组委会将会选择 N 个测试cpp文件进行测试，伪装后的 N 个代码需要至少有 90% 的比例通过组委会的测试用例，才能进入第二部分的比赛；否则，会被淘汰。

第一部分占总分比例为15%，计分规则：（通过测试用例的数量 $\div N$ ）*15

【第一部分说明】

1. 所有 input 文件夹下的代码，都可以在组委会平台上，在 1s 以内的时间、64M 以内的内存AC（接受）通过；
2. 如果某同学 Part1 的输出结果中，存在某个 cpp 文件不能正常 AC（编译不通过或者不通过对应的测试样例），或缺失某个 cpp 文件，则该题目不得分；
3. 请把代码交到你的 ID 对应的文件夹下，若你的 ID 为 10，那么最后 10/Part1 下的文件应包含三个文件（夹）如下：

`makefile submit.cpp output/`

其中 output 为空文件夹，另外两个为文件，请确保你的代码运行后会把结果写入 output 文件夹中，输出为 1.cpp,2.cpp, ..., x.cpp，如果文件目录、文件名、写入位置写错，或者 make 失败，Part1为 0 分，将直接被淘汰。
4. 不支持引入 `#include <bits/stdc++.h>` 头文件；
5. 1.cpp~x.cpp将依次被参赛队伍的程序 submit 读入处理。

第二部分【辨伪】：

1) 题目说明

你将需要写一个程序。该程序应当能够接受两个被伪装的代码输入，并输出它们是不是【同源】的。此处的【同源】一词指的是：它们实际上来源于同一个程序（输入与输出完全一致）。

2) 文件要求

请将你的可执行文件最终名称命名为 `check`（通过makefile生成），接受一个参数 `x`，这个参数为等待判断的代码对的总数。输入的代码文件相对于可执行文件处于 `./sample1/`、`./sample2/` 文件夹下，每个输入的命名在两个文件夹下均分别为 `1.cpp,2.cpp ... x.cpp` 输出一个 `result.txt` 文件，每行一个字母，Y 表示认为同源，N 表示非同源。

3) 特别说明

比赛为两两对抗淘汰赛，参赛者第二部分 check 程序的输入为对抗对手第一部分打散后的输出（此处的打散为第一部分输出结果间的打散），一共 $N * 90\%$ 组。check 程序需对每一组输入进行判断，如果输出正确，则得1分，直到 $N * 90\%$ 组题目全部判定完成。

4) 计分说明

每一行的输出如果和组委会给定的结果一致则得1分，依次到result.txt文件结束。第二部分占比为85%，计分规则： $(N * 90\% + \text{本队输出正确得分} - \text{对手输出正确得分}) / (N * 90\%) * 85$ 。

【第二部分说明】

1. 请把代码交到你的 ID 对应的文件夹下，若你的 ID 为 10，那么最后 10/Part2 下包含两个文件如下：

`makefile check.cpp`

组委会会执行 make，请确保运行后生成的 result.txt 文件也在你的 10/Part2 目录下（若某同学的 ID 为 10）。如果 make 失败，或程序运行后生成的 result.txt 不在指定位置，Part2直接计0分。

2. 不支持引入 `#include <bits/stdc++.h>` 头文件。

总分说明：

1、总得分 = 第一部分得分 + 第二部分得分。

举例说明：

假设组委会测试文件数量 $N=400$ 。甲和乙互为对抗对手，在第一部分，甲通过组委会平台的测试样例数量为370个，乙通过组委会平台的测试样例数量为360个，甲与乙均进入第二部分辨伪环节。甲得分：

$370/400 * 15 = 13.9$ 分；乙得分： $360/400 * 15 = 13.5$ 分；在第二部分中，甲的 check 程序对乙第一部分打散后的输出程序进行判断，共输出正确 300 道，乙的check程序对甲第一部分打散后的输出程序进行判断，共输出正确 320 道，甲得分 $(360+300-320) / 360 * 85 = 80.3$ 分，乙得分 $(360+320-$

300)/360*85=89.7分。甲总积分=13.9+80.3=94.2分，乙总积分=13.5+89.7=103.2分。因此两两对抗中乙队胜出。

2、若出现两队总分同分情况，以第二部分得分高者胜出；若第二部分得分继续相同，以第二部分本队输出正确数量多者胜出；若每队输出正确数量继续相同，由组委会出题，进行加时赛。

FAQ

1.代码长度限制是多少？

答 单一文件提交的代码最大长度为 32768，即 32 KB。

2.C++ 编译器版本？

答 g++ 4.5.1，**不支持** C++11，我们的判题机器是 64 位 linux。

3.C++ 的堆栈容量是多少？

答 C++ 的堆栈容量是 64MB。