

IOI2004 中国国家队选拔赛

练习赛

题目名称	逻辑等价	素性测试	石头剪子布
目录	day0/logic	day0/prime	day0/soko
可执行文件名	logic	prime	soko
输入文件名	logic.in	prime1.in~prime3.in	---
输出文件名	logic.out	prime1.out~prime3.out	---
测试点个数	4	3	3
满分	40	30	30
运行限制	1s		1s
	64M		64M

竞赛时间：2004 年 5 月 8 日下午 15:00–17:00

逻辑表达式

(logic)

【背景描述】

HURRICANE 小组想要判断两个逻辑表达式是否等价。等价的意思是它们的真值表相同。你能帮助他们吗？

【任务描述】

你的任务是，判断两个逻辑表达式是否等价。

【输入格式】: (logic.in)

输入数据第一行是数据个数 n ，以下 n 行，每行包括两个表达式。表达式是由 $a..z$ 这 26 个字母与 $'|' \& '^' '(' ')' \sim$ 等运算符组成的。其中 $'|' \& '^' '(' ')' \sim$ 分别表示‘或者’，‘并且’，‘异或’和‘非’。优先级从高到低分别是 $'()' \sim \& '^' |$ 。你需要忽略任何其它字符。两个表达式之间可能没有分隔，你需要自己判断他们的分隔处。

每个表达式最多出现10个不同变量（字母），一个表达式的运算符不超过100个，表达式的长度不超过1000位。

【输入样例】

```
3
a ^b&(b|a)~b^ a
a^b&(b|a)(a^(b&(b|a)))
~~~~Z~~Z
```

【输出格式】: (logic.out)

输出 n 行，分别对应每一个输入，如果两个表达式等价，则输出‘Yes’，否则输出‘No’。

【输出样例】

```
No
Yes
Yes
```

【运行限制】

运行时间	1 秒
内存使用	64M

【评分方法】

本题目一共有四个测试点，每个测试点的分数为总分数的 25%。对于每个测试点来说，如果你给出的答案正确，那么你将得到该测试点全部的分数，否则得 0 分。

素性测试（提交答案）

(prime)

【背景描述】

HURRICANE 小组的成员们最近开始学习数论,对其中讲到的素数非常感兴趣。他们按照某些规律找出了一些数字,不过他们并不知道这些数字是不是素数。现在请你帮他们判断一下。

【任务描述】

你的程序必须根据我们给出的 3 个输入,提供符合题意的 3 个输出:

- 每个输入中包括一个整数,为需要你判断的数。
- 对于每个输入,你需要判断它是不是素数。如果是,那么在输出文件中输出 PRIME;如果不是,输出 COMPOSITE。

【输入格式】: (*prime1.in~prime3.in*)

我们提供的输入文件为 *prime1.in~prime3.in*。每个文件只包括一个整数,为需要判断的数。

【输入样例】

3

【输出格式】: (*prime1.out~prime10.out*)

你需要对于我们的输入文件分别完成 *prime1.out~prime3.out*。每个文件只包括一行,为你的判断。

【输出样例】

PRIME

【评分方法】

这是一道提交答案的题目,对于提供的 3 个输入文件,你需要提交相应的输出文件。我们将按照上述的格式对你提交的输出进行检查,如果不符合格式,那么得 0 分,否则如果答案不正确,也得 0 分,否则得满分。

石头·剪子·布 (交互式)

(soko)

【背景描述】

HURRICANE 小组最近制作出来了一个猜拳机,不过你很不服气,要和这个猜拳机进行比赛。

【任务描述】

这是一道交互式的题目,你的程序需要和交互库交互并按照约定完成所有库函数的调用:

- 你需要在执行任何其他的操作之前初始化交互库;
- 然后通过调用库函数来完成你对数据的输入及处理,其中包括:
 - 从交互库获得猜拳的次数;
 - 给出你出的是石头、剪子还是布;
 - 从交互库获得他出的是石头、剪子还是布;
- 在猜拳结束的任务后,将由程序库终止你程序的执行。你的程序不得自行退出。

【交互方式】(库函数)

我们提供的交互库中包括了如下的函数,现将它们的功能说明如下:

- **init**: 初始化函数。你需要在你程序的一开始调用它,且仅能调用一次。调用后交互库会完成相应的初始化工作。
- **getcount**: 你可以调用这个函数来获得猜拳的次数;
- **getjob**: 你从这个函数返回交互库的猜拳,只有在调用 *answer* 后才能调用一次这个函数,0 为石头,1 为剪子,2 为布。
- **answer(i)**: *i* 为你的程序的猜拳,0 为石头,1 为剪子,2 为布。

【测试说明】

运行程序时测试库会生成 *soko.log*,记录你的程序的每一次对库函数的调用。*soko.log* 会忠实记录你的程序的调用,即使它们是非法的。你可以根据 *soko.log* 来判断你的程序的正确性。

【语言相关的说明】

对于使用 PASCAL 的选手来说，我们提供的 sokolib.ppu 单元包括如下的函数定义：

- `procedure init;`
- `function getcount:longint;`
- `function getjob:longint;`
- `procedure answer(ans:longint);`

你可以通过 `uses sokolib;` 语句来使用这个单元并调用其中的函数。

对于使用 C/C++ 的选手来说，我们提供的 sokolib.h/.o 库包括如下函数定义：

- `void init();`
- `long getcount();`
- `long getjob();`
- `void answer(long ans);`

你需要建立一个工程，然后将 sokolib.o 及你的源程序加入到该工程中。然后就可以通过 `#include "sokolib.h"` 预编译指令来包含这个头文件并使用并调用其中的函数。

【运行限制】

运行时间	1 秒
内存使用	64M

【评分方法】

本题目一共有三个测试点，每个测试点的分数为总分数的 33%。在测试过程中，如果有如下的情况发生：

- 以任何方式进行直接文件操作；
- 非法调用测试库；
- 使测试库非法退出；
- 自行退出程序；
- 运行时间超过最大允许时间；

那么得 0 分；否则你的交互结果将被用来做进一步的评审：

- 如果你的程序赢测试库的次数大于总次数的一半，那么得满分；
- 如果你的程序赢测试库的次数小于总次数的一半，那么你的得分按照如下公式计算：

$$score = \frac{2 * win_count}{match_count}$$

`win_count` 为你赢的次数，`match_count` 为总共猜拳的次数。