

Faulty Connection

时间限制：1.0s 内存限制：512M

题目描述

本题为 **multi-pass** 题。判题机会针对同一个测试点多次运行你的程序。

你需要为 `1` 到 `600` 号消息设计一种纠错编码。

对每个消息编号 k ，你需要分配一个长度为 `30` 的整数列表，列表中的数两两不同，且都在 `1` 到 `1000` 之间。

之后，通信信道可能会丢失大部分数字，最后只剩下这 30 个数中的任意两个（顺序任意）。你的目标是让这两个数仍然能唯一确定原始消息编号。

也就是说，你设计的编码必须满足：

- 对于任意消息编号 k ，都有一个长度为 `30` 的互不相同整数序列；
- 从该序列中任取两个不同整数，就必须能唯一反推出消息编号 k 。

交互 / 多次运行规则

你的程序会在同一个测试点上被运行多次。

第一轮：发送

输入为：

- 第一行：`send`
- 第二行：消息编号 k

你需要输出 30 个互不相同的整数，作为消息 k 的编码。

后续轮次：接收

输入为：

- 第一行：`receive`
- 第二行：两个不同的整数 a, b

这两个数一定来自你在第一轮为同一个消息输出的 30 个数中，顺序不保证。

你需要输出原始消息编号 k 。

输入格式

本题输入分两种情况：

情况 1：发送

```
send  
k
```

其中：

- $1 \leq k \leq 600$

情况 2：接收

```
receive  
a b
```

其中：

- $1 \leq a, b \leq 1000$
- $a \neq b$

输出格式

若输入为 `send`

输出 30 个互不相同的整数 x_1, x_2, \dots, x_{30} ，满足：

- $1 \leq x_i \leq 1000$

若输入为 `receive`

输出一个整数 k ，表示对应的原始消息编号。

说明

- 每次运行限时 1 秒。
- 题目提供了测试工具帮助你本地调试。
- `send` 阶段的输出不唯一；只要整体方案正确即可。

样例输入 1

```
send
42
```

样例输出 1

本题为 multi-pass 题，且 `send` 阶段输出不唯一，因此样例输出省略。