

DIORPan

登出 ([http://zhengruioi.com/logout?](http://zhengruioi.com/logout?_token=sQjj9hDZ0b4XE2MkPOLMsRsGn3nvyTU7R8DgYha8ch0X1Q6QmGxwwVlsMesQ)[\\_token=sQjj9hDZ0b4XE2MkPOLMsRsGn3nvyTU7R8DgYha8ch0X1Q6QmGxwwVlsMesQ](http://zhengruioi.com/logout?_token=sQjj9hDZ0b4XE2MkPOLMsRsGn3nvyTU7R8DgYha8ch0X1Q6QmGxwwVlsMesQ))

ZROI

21秋季csp7连 day2

## C. 【2021 CSP 7连测 day2】继承类

3:41:22

时间限制: 1000 ms 空间限制: 512 MiB 题目类型: 传统型 答案检查器: 文本比较

❤好评 🏳️差评 [-4]

■ 描述

🕒 提交

➤ 自定义测试

📊 统计 (/contest/974/problem/2000/statistics)

[返回比赛 \(/contest/974\)](/contest/974)

现在发明了一种类似于 C++ 的面向对象编程语言中的类声明。

每个类声明的格式为 " $K : P_1 P_2 \cdots P_K ;$ "

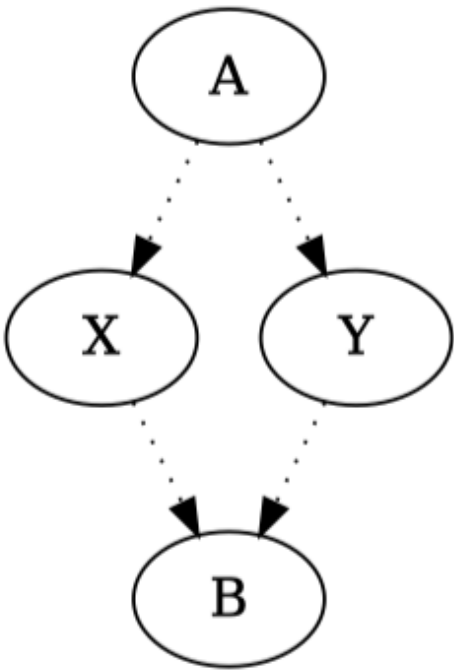
其中  $K$  是要声明的新类的名称,  $P_1$ 、 $P_2$ 、 $\cdots$ 、 $P_K$  类  $K$  继承的类的名称。

例如, "**shape ;**" 是不继承任何其他类的类 "shape" 的声明, 而 "**square : shape rectangle;**" 是继承类 "shape" 和 "rectangle" 的类 "square" 的声明。

如果类  $K_1$  继承类  $K_2$ , 类  $K_2$  继承类  $K_3$ , 依此类推, 直到类  $K_{m-1}$  继承类  $K_m$ , 那么我们说所有类  $K_1$ ,  $K_2$ ,  $\cdots$ ,  $K_{m-1}$  派生自类  $K_m$ 。

编程语言的规则禁止循环定义, 因此不允许从自身派生一个类。换句话说, 类层次结构形成了一个有向无环图。此外, 不允许在类层次结构中出现所谓的菱形。一个菱形由四个不同的类  $A$ 、 $B$ 、 $X$ 、 $Y$  组成, 而且它满足 (如下图所示) :

- 类  $X$  和  $Y$  派生自  $A$ 。
- 类  $B$  派生自  $X$  和  $Y$ 。
- 类  $X$  不是从  $Y$  派生的, 类  $Y$  也不是从  $X$  派生的。



现在你会获得一系列要按顺序处理的类声明，并确定每个类声明是否是正确声明。

正确声明的类被添加到层次结构中，而错误的类被丢弃。声明 “ $K : P_1 P_2 \cdots P_K ;$ ” 如果满足以下条件，则认为正确声明：

- 1. 类  $K$  尚未声明。
- 2. 所有类别  $P_1, P_2, \cdots, P_k$  之前已经声明过。请注意，此条件可确保类永远不会从其自身派生，或者类层次结构中不能存在循环。
- 3. 通过添加继承了  $P_1, P_2, \cdots, P_k$  的类  $K$  以后，类层次结构保持有序，即没有形成任何菱形。

现在需要你分别处理上述声明并确定每个声明的正确性。

## 输入格式

输入的第一行包含整数  $n$  —— 声明的数量。

以下  $n$  行中的每一行都包含一个形式为 “ $K : P_1 P_2 \cdots P_K ;$ ” 的声明，其中  $P_1, P_2, \cdots, P_k$  是类  $K$  继承的一系列类（可以为零个、一个或者多个）。

单个声明中的所有类名  $K, P_1, P_2, \cdots, P_k$  互不相同。每个类名都是最多 10 个英文小写字母的字符串。声明的所有元素（类名和字符 “:”、“;”）都由一个空格分隔。在每个特定的声明中，对于类的数量  $k$ ，保证  $0 \leq k \leq 1000$ 。

## 输出格式

输出包含  $n$  行。如果第  $i$  个声明正确，第  $i$  行必须包含 ok，如果不正确，则必须包含 greska。

## 样例

类层次结构如下图所示，

- 第四个声明是不正确的，因为类 “circle” 已经在第三行中定义了。
- 第六个声明不正确，因为尚未定义类 “object”。
- 第八个声明是正确的，因为现在已经声明了类 “object”，而第六个声明被丢弃了，因此还没有定义类 “runnable”。

- 第十个声明不正确，因为会形成以下菱形：“shape”、“applet”、“square”、“runnable”。

## 输入

```
10
shape : ;
rectangle : shape ;
circle : shape ;
circle : ;
square : shape rectangle ;
runnable : object ;
object : ;
runnable : object shape ;
thread : runnable ;
applet : square thread ;
```

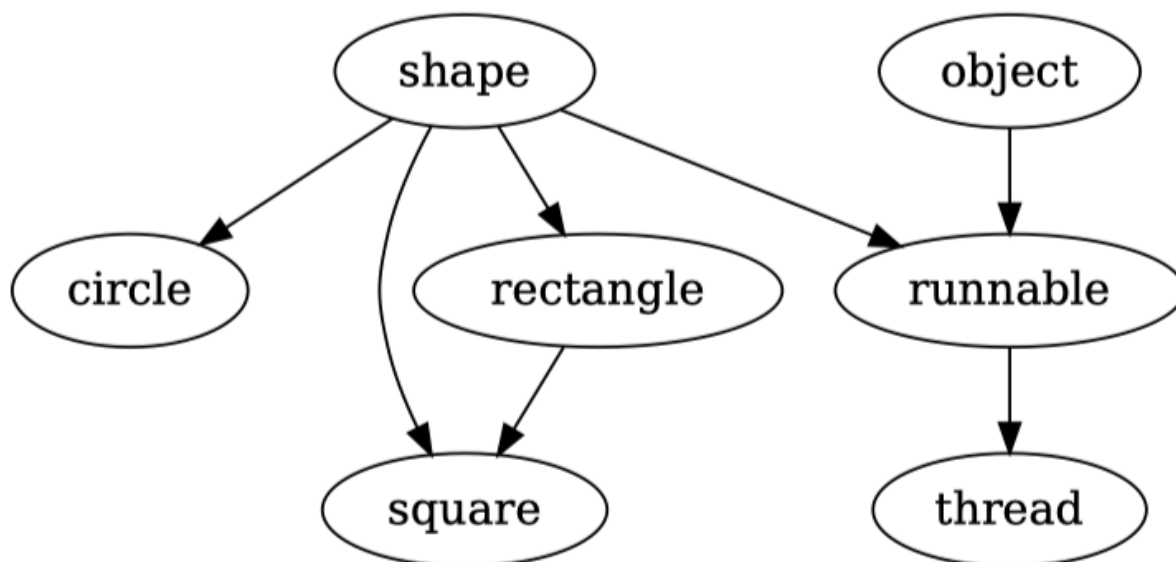
## 输出

```
ok
ok
ok
greska
ok
greska
ok
ok
ok
ok
greska
```

## 样例解释

类层次结构如下图所示，

- 第四个声明是不正确的，因为类“circle”已经在第三行中定义了。
- 第六个声明不正确，因为尚未定义类“object”。
- 第八个声明是正确的，因为现在已经声明了类“object”，而第六个声明被丢弃了，因此还没有定义类“runnable”。
- 第十个声明不正确，否则会形成以下菱形：“shape”、“applet”、“square”、“runnable”。



## 数据范围

对于其中 20% 的数据，保证  $1 \leq n \leq 100$ ，而且仅考虑条件 1 即可确定正确性。

对于其中另外 20% 的数据，保证  $1 \leq n \leq 100$ ，而且仅考虑条件 1, 2 即可确定正确性。

对于其中另外 20% 的数据，保证  $1 \leq n \leq 100$ 。

对于所有数据，保证  $1 \leq n \leq 1000$ 。



(<http://zhengruioi.com/contest/974/problem/2000?locale=zh-cn>)



(<http://zhengruioi.com/contest/974/problem/2000?locale=en>)

Zhengrui Online Judge

Server time: 2021-09-04 18:18:28 | 浙ICP备17047493号 (<http://www.beian.miit.gov.cn/>)