

# oGyx!语言

## 题目描述

以下是一个典型的 oGyx!语言 程序。

```
1 { vars
2   a: int
3   b: int
4   c: array[int, 1..2]
5   d: int
6 }
7 # 以上变量默认值均为 0
8 # 变量名只可是英文字母
9
10 # gyxsay 语句可以输出一个数字，随后跟一个空格
11 :gyxsay 2
12 # 输出 2 和一个空格（以下不再提到空格）
13
14 # set 语句可以为变量赋值
15 # 运算符只支持加减号即可
16 :set a, 1
17 :set b, 2
18 :gyxsay a + b
19 # 上一条语句将输出 3
20
21 # 以下的判断语句均使用以下的格式：
22 # 操作符，表达式，表达式
23 # 例如 eq, a, 1 即 C 语言中 a == 1
24 # 所有操作符包括：lt (<) ; gt (>) ; le (<=) ; ge (>=) ; eq (==) ; neq (!=)
25
26 # 为了 xx Gyx, 因此他将 if 改成了 gif
27 { gif eq, a, 1
28   :set d, 3
29   :gyxsay 1
30 }
31 # 输出 1
32 # 以上是 gif 语句，无需支持 else
33
34 # 为了 yy Gyx, 因此他将 for 改成了 gor
35 # gor i = 1 to d 如下
36 { gor i, 1, d
37   :gyxsay i
38 }
```

```

39 # 输出 1 2 3
40
41 # 为了 zz Gyx, 因此他将 while 改成了 ghile
42 :set i, 1
43 { ghile le, i, 2
44     :gyxsay i
45     :set c[i], i + 1
46     :gyxsay c[i]
47     :set i, i + 1
48 }
49 # 输出 1 2 2 3
50
51 # 数组不会出现嵌套, 即只有 a[i]、a[i + 2]而不会有类似于 a[i + b[i]] 这样的
52 # oGyx!语言的最后一行, 一定是一个换行
53

```

你的任务是写一个 oGyx!语言 的解释器，实现输入 oGyx!语言 的程序，解释之执行后输出执行结果。

## 输入格式

输入文件全部为 oGyx!语言 程序，最后一行保证是个空行。

请处理输入的时候，一直读到 EOF 为止。

## 输出格式

该 oGyx!语言 程序的执行结果。

具体上，是该 oGyx!语言 程序所有 :gyxsay 语句的输出。

## 样例

见上文。

## 提示

对程序做出以下保证：

1. 输入数据一定是合法的 oGyx!语言 程序，注释、代码、缩进风格（四个空格）样式与上述样例完全相同，但不保证逗号、运算符前面的空格数量和有无一定相同；

2. 变量名在 10 个字符以下，仅包含小写英文，数组最大大小为 1000，变量最多 50 个，所有表达式的运算结果，包括变量的值一定在 int 范围内；（但数组可能是类似于 [2001...3000] 的，下标可能范围为 0 到  $10^8$ ）
3. 所有指令小写；
4. for 语句执行过程中，循环变量、初始值、结束值不会被循环中的代码改变；
5. 该程序最多 500 行。