

D. 人口普查 (census)

时间限制: 1 秒

空间限制: 128 MiB

关于 Cesenatico 一个鲜为人知的事实是，这里住着一个由 N 位女性信息学家组成的秘密协会。这个协会非常隐秘；成员之间互不相识。每位成员都有一个唯一的 ID：一个非负整数 I 。

成员之间唯一的沟通方式是间接的，通过在镇上不同地点用粉笔写下的数字来传达。每隔 100 年，协会会进行一次人口普查来统计成员的人数。普查结束后，每位成员都应该知道协会的总人数。

人口普查持续多日。每一日，还在参与流程的成员都需要选择并仅执行一个动作：**读取 (read)**、**写入 (write)**，或者**停止 (stop)** 参与。

- 若成员选择 **读取**，她会选定一个地点 P 。在白天，她会访问该地点并读取那里写下的数字。
- 若成员选择 **写入**，她会选定一个地点 P 和一个数字 V 。在傍晚，她会访问该地点并将那里的数字改为 V 。由于天色已晚，在写下新数字之前，她无法读取旧数字。
- 若成员选择 **停止**，她在随后的日子里将不再采取任何行动（即不再参与流程）。

如果某成员看到另一位成员写下数字，她可能会认出对方。因此，协会严禁两名或以上的成员在同一天选择在同一个地点写入。（对于读取操作则没有此限制，因为读取可以隐蔽地进行。）

如果一位或多位成员在某天读取某个地点，而另一位成员打算在同一天向该地点写入新数字，所有的读取操作都会发生在写入操作之前。

协会该如何规划普查流程，使得所有人得知正确成员人数所需的天数是最少的？

实现详情

⇒ 这是一个交互题，你的程序会有未知数量 ($1 \leq N \leq 100$) 的实例 (instances) 同时运行。每个实例模拟协会中的一名成员。

共有 10^{18} 个地点。地点 P 必须满足 $0 \leq P < 10^{18}$ 。最初，所有地点上写入的值均为 $V = 0$ 。

地点上写入的新值 V 必须始终是一个整数，满足 $0 \leq V \leq 10^9$ 。在大多数子任务中， V 只能为 0 或 1。更多详情请参阅“评分”部分。

当你的程序的一个实例启动时，它应首先输入一行包含两个整数 I 和 M ($0 \leq I \leq M - 1$)：该实例所代表的成员的唯一 ID，以及可能的 ID 总数。在每个测试数据中，所有实例将获得相同的 M 值和不同的 I 值。注意，可能存在未分配给任何成员的 ID。

然后，对于普查过程中的每一天，你的程序应选择它想执行的动作并相应地输出一行：

动作	含义
$r\ P$	读取地点 P 。 输出此行后，你的程序应输入一行，其中包含当前写在 P 处的数值。
$w\ P\ V$	写入地点 P 处的新值 V 。 如果多个实例在同一天写入同一个 P ，你将得到 <i>Not correct</i> 的判定。 除了样例和子任务 3 外，你必须输出 $0 \leq V \leq 1$ ；详情请参阅“评分”部分。
$!\ N$	回答并停止：报告有 N 名成员并停止参与普查。 回答后， 你的程序应正常退出 。（注意，你的程序的其他实例可能会继续运行几天。）

如果你的程序的任何实例回答了错误的 N 值、违反了协议（protocol）、使用了超过 500 天，或超过了（每个进程的）时间/内存限制，你的提交将被判定为 *Not correct*。

否则，你的程序在测试数据上将被判定为 *(Partially) Correct*，并根据 D 值（任何实例回答所用的最大天数）进行评分。要获得满分，你需要用 $D \leq 61$ 和 $V \leq 1$ 解出每一个测试数据。详情请参阅“评分”部分。

刷新输出。 如果你没有使用提供的模板，请确保在输出每一行后刷新标准输出，否则你的程序可能会被判定为 *Not correct*。在 Python 中，如果你使用 `input()` 读取行，这会自动完成。在 C++ 中，`cout << endl`；在输出换行符之外还会进行刷新；如果使用 `printf`，请使用 `fflush(stdout)`。

约束条件

- $1 \leq N \leq 100$ 。
- $1 \leq M \leq 100\,000$ 。
- 你最多可以使用 500 天。

评分方式

你的程序将在分成若干子任务的测试数据上进行测试。要获得某个子任务的分数，你必须正确解出该子任务中所有的测试数据。

- **子任务 0 [0 分]**: 样例（你可以写入任何整数 $0 \leq V \leq 1\,000\,000\,000$ ）。
- **子任务 1 [11 分]**: $M \leq 100$ ，且 N 名成员的 ID 为 $0, 1, \dots, N-1$ 。
- **子任务 2 [12 分]**: $1 \leq N \leq 2$ 。
- **子任务 3 [22 分]**: $M \leq 8000$ ，且你可以写入任何整数 $0 \leq V \leq 1\,000\,000\,000$ 。
- **子任务 4 [55 分]**: 无额外限制。

在子任务 1、2 和 4 中，每次写入操作时你只能写入 $V = 0$ 或 $V = 1$ 。

设 X_s 为子任务 s 的最高分（如上所示）， D_s 为你的程序在子任务 s 的测试中所使用的最大天数。那么：

$$\text{score}_s = \begin{cases} X_s & \text{如果 } D_s \leq 61 \\ X_s \cdot (0.2 + 0.8 \cdot 1.01^{(60-D_s)}) & \text{如果 } 61 < D_s \leq 500 \\ 0 & \text{如果 } 500 < D_s. \end{cases}$$

score_s 的值按子任务四舍五入到最接近的整数，你的总分是这些分数的总和。要获得本题的满分，你需要每一个测试数据都满足 $D \leq 61$ 且 $V \leq 1$ 。

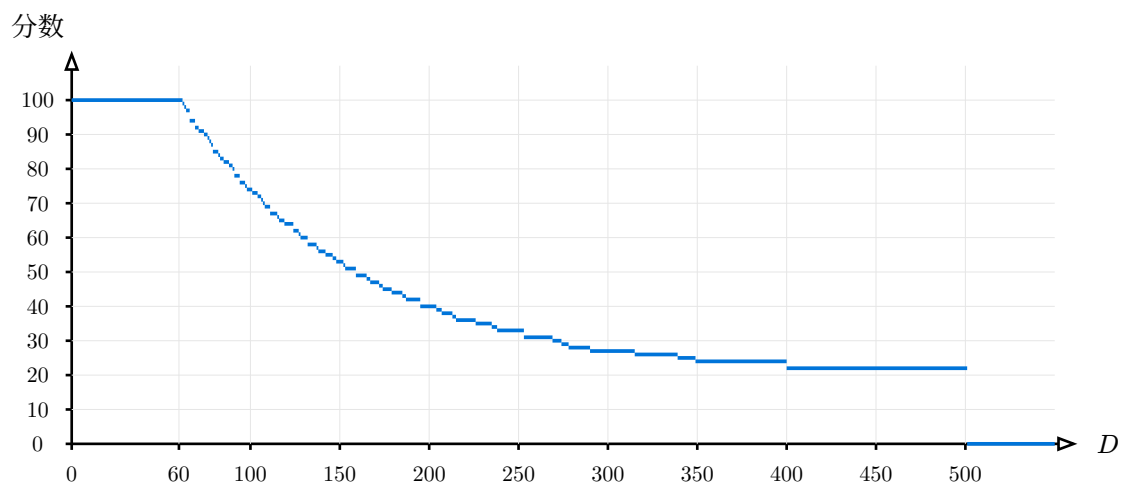


图 1 总分，假设每个子任务都以相同的最大 D 解决。

样例

第一个样例。每对列显示评测程序与一个实例之间的交互。

评测机	实例 0	评测机	实例 1	评测机	实例 2	评测机	实例 3	评测机	实例 4
0 100		1 100		2 100		3 100		4 100	
	w 12 1		w 50 1		w 99 0		w 7 1		r 5
								0	
	r 50		r 7		r 12		w 1 1		! 5
1		1		1					
	! 5		r 1		w 0 0		! 5		
		1							
		! 5			! 5				

第二个样例。

评测程序	实例 0	评测程序	实例 1
0 8000		3 8000	
	w 0 0		w 2 1
	w 1 1		r 1
		0	
	r 2		r 2
1		1	
	! 2		r 1
		1	
			! 2

样例解释

第一个样例。协会有 $N = 5$ 名成员，ID 分别为 0, 1, 2, 3, 4，且 $M = 100$ （适用于子任务 1、3 和 4）。实例 i 对应 ID 为 i 的成员。上述交互仅是一种可能合法的操作序列，不代表这是一个高效或合理的策略；它仅用于演示交互的原理。

第二个样例。协会有 $N = 2$ 名成员，ID 分别为 0 和 3，且 $M = 8000$ （适用于子任务 2、3 和 4）。第一天，ID 为 0 的成员在地点 0 写入 0（无变化），ID 为 3 的成员在地点 2 写入 1。

地点	0	1	2	3	4	...
数值	0	0	1	0	0	...

第二天，ID 为 0 的成员在地点 1 写入 1，ID 为 3 的成员读取了同一个地点。注意，读取发生在白天，在傍晚的写入之前。因此，ID 为 3 的成员看到的仍然是 0。

地点	0	1	2	3	4	...
数值	0	1	1	0	0	...

第三天，他们都读取了地点 2，那里写着 1。

第四天，ID 为 0 的成员回答有 2 名成员（正确），同时 ID 为 3 的成员读取了地点 1 处的 1。ID 为 0 的成员在此之后立即退出，不再参与接下来的日子。

最后，在第 $D = 5$ 天，剩余的成员也正确回答了 $N = 2$ 。

测试

为了方便测试你的解法，我们提供了一个简单的工具，你可以从 CMS 下载。使用该工具是可选的。请注意，CMS 上的官方评测机与此测试工具不同。

要使用该工具，你需要一个输入文件。你可以使用提供的样例输入 `census.input0.txt` 和 `census.input1.txt`，或者创建自己的。输入文件应以成员人数 N 和可能的 ID 数 M 开头一行，接着是一行包含 N 个数字，指定协会成员的 ID。

对于 Python 程序，假设为 `census.py`（通常运行命令为 `pypy3 census.py`），按如下方式运行测试工具：

```
python3 testing_tool.py pypy3 census.py < census.input0.txt
```

对于 C++ 程序，首先编译你的解决方案：

```
g++ -DEVAL -std=gnu++20 -O2 -pipe -static -s -o census census.cpp
```

然后运行测试工具：

```
python3 testing_tool.py ./census < census.input0.txt
```

注意，本题中标准输出用于与评测程序交互，因此不应将其用于调试。相反，你可以使用标准错误输出（stderr）。在 C++ 中，你可以使用 `cerr << msg << endl`；在 Python 中，你可以使用 `print(msg, file=sys.stderr)`。

测试工具将读取并呈现这些 stderr 消息，以及你所有程序实例执行的查询。请注意，由于技术原因，这些讯息可能稍微不同步。