

D. Guessing Game

Problem Name	Guessing Game
Time Limit	4 seconds
Memory Limit	1 gigabyte

在隆德老城区，有一条街道，有 N 排房子，索引为 0 到 $N - 1$ 。艾玛住在 在其中一栋房子里，她的朋友安娜和伯特尔想弄清楚是哪一栋。艾玛并不告诉她的朋友她住在哪里，而是和他们玩一个游戏。比赛一开始，安娜和伯特尔只知道街上有多少栋房子。此时，安娜和伯特尔 可以选择一个正整数 K 并就策略达成一致。此后禁止任何通讯。

游戏本身由两个阶段组成。在第一阶段，艾玛选择访问的顺序 房子，这样她的房子就是最后一个参观的房子。然后她带着安娜一一往前走 按照这个顺序，没有提前告诉安娜这个顺序。对于每一个不是艾玛的房子的房子，安娜被允许用粉笔在房子的前门上写一个介于 1 和 K 之间的整数。对于他们访问的最后一所房子，艾玛的房子，艾玛写了一个 1 到 K 之间的整数

在游戏的第二阶段，伯特尔沿着街道从 0 号房子走到 $N - 1$ 号房子，并且 读取安娜和艾玛写在门上的所有数字。他现在想猜猜是哪一个 艾玛住的房子。他有两次猜对的机会，如果猜对了，他和安娜就赢了游戏成功了。否则，艾玛就赢得了比赛。你能设计出一个保证安娜和伯特尔赢得比赛的策略吗？你的策略 将根据 K 的值进行评分（越小越好）。

执行

这是一个多次运行问题，意味着您的程序将被执行多次。首先 当它运行时，它将执行安娜的策略。之后它将实施伯特尔的策略。

输入的第一行将包含两个整数 P 和 N ，其中 P 可以是 1 或 2 （第一阶段或第二阶段）， N 表示房屋的数量。除了示例输入（不用于评分）， N 总是等于 $100,000$ 。

以下输入取决于阶段：

猜数字游戏（5个中的1个）

第一阶段 你的程序应该首先输出一个整数 K 在单独一行中（ $1 \leq K \leq 1,000,000$ ）。然后，还有 $N - 1$ 行，每行包含一个索引 i （ $0 \leq i < N$ ），并输出一个整数 A （ $1 \leq A \leq K$ ），其中 A 是安娜在第 i 个房屋的门上写的数字。除了艾玛所在房屋的索引外，每个索引 i 将恰好出现一次，其顺序由评分系统决定。

第二阶段 你的程序应该读取一行，包含 N 个整数 A_0, A_1, \dots, A_{N-1} ，表示每个房屋门上写的数字。然后，应该输出一行，包含两个整数 s_1 和 s_2 ($0 \leq s_i < N$)，表示猜测的索引。 s_1 和 s_2 可以相等。

实现细节

请注意，在第二阶段运行程序时，程序会重新启动。这意味着你不能在运行之间保存信息在某些变量中。在打印每一行后，请确保刷新标准输出，否则你的程序可能会被判定为超出时间限制。在 Python 中，`print()` 会自动刷新输出。在 C++ 中，`cout << endl;` 除了打印换行符之外还会刷新输出；如果使用 `printf`，请使用 `fflush(stdout)`。

此问题的评分系统可能是自适应的，这意味着它可能会根据你的程序输出来改变其行为，以防止启发式解法通过。它可能会试运行第一阶段，查看你的输出，然后使用从上一次运行中提取的信息来进行真正的第一阶段运行。

你的程序必须是确定性的，即在相同的输入上运行两次时会产生相同的行为。如果你的程序中使用了随机性，请确保使用固定的随机种子。这可以通过将一个硬编码的常量传递给 `srand` (在 C++ 中) 或 `random.seed` (在 Python 中) 来实现，或者如果使用 C++11 的随机数生成器，则可以在构造随机数生成器时指定种子。特别是，你不能在 C++ 中使用 `srand(time(NULL))`。如果评分系统检测到你的程序不是确定性的，它将给出错误答案的判定。

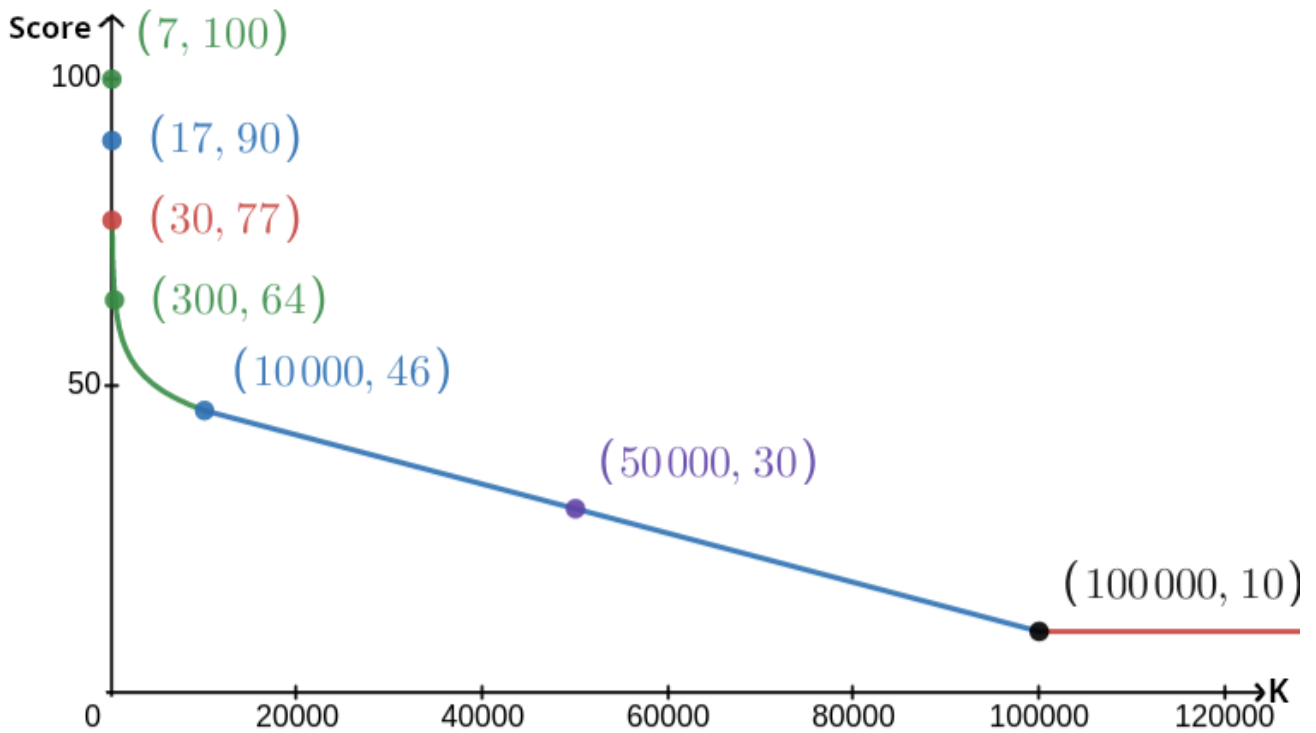
如果你的程序的 (最多3次) 独立运行的总运行时间超过了时间限制，你的提交将被判定为超出时间限制。

评分

您的解决方案将在许多测试用例上进行测试。如果您的解决方案在任何这些测试中失败情况 (例如，给出错误答案 (错误答案)、崩溃 (运行时错误)、超出时间限制 (超出时间限制) 等)，您将获得 0 分和相应的判决。如果您的程序在所有测试用例中成功找到艾玛的房子的索引，您将得到判决接受，分数计算如下。令 K_{max} 为用于的 K 的最大值任何测试用例。根据 K_{max} 值：

	Score
$K_{max} > 99\,998$	10 points
$10\,000 < K_{max} \leq 99\,998$	$10 + \lfloor 40(1 - K_{max}/10^5) \rfloor$ points
$30 < K_{max} \leq 10\,000$	$46 + \lfloor 31(4 - \log_{10}(K_{max})) / (4 - \log_{10}(30)) \rfloor$ points
$7 < K_{max} \leq 30$	$107 - K_{max}$ points
$K_{max} \leq 7$	100 points

评分函数如下图所示



样例测试案例不计入得分，并且你的解决方案不必对其进行处理。

测试工具

为了方便测试你的解决方案，我们提供了一个简单的工具，你可以下载使用。请参阅 Kattis 问题页面底部的“附件”部分。该工具是可选的，你可以自行更改。请注意，Kattis 上的官方评分程序与测试工具不同。

使用示例（其中 $N = 4$ ， $s = 2$ ，其中 s 是写在最后一所访问房屋的数字）：

对于 Python 程序，假设解决方案为 `solution.py`（通常作为 `pypy3 solution.py` 运行）：

```
python3 testing_tool.py pypy3 solution.py <<<"4 2"
```

对于 C++ 程序，首先编译它（例如使用 `g++ -g -O2 -std=gnu++17 -static solution.cpp -o solution.out`），然后运行：

```
python3 testing_tool.py ./solution.out <<<"4 2"
```

测试工具将以随机顺序访问房屋。若要使用特定的顺序，请在测试工具中修改“MODIFY HERE”处。示例交互 样例测试案例不计入得分，并且你的解决方案不必对其进行处理。假设我们有 $N = 4$ ，并且 Emma 住在第 1 所房屋。让 A 是写在房屋上的数字的列表。最初， $A = [0, 0, 0, 0]$ ，其中 0 表示对应房屋上没有写数字。

在第一次运行代码时：给出 $N = 4$ 。您的解决方案的响应结果为 $K = 3$ 。

要求 A_2 。您的解决方案响应为 3。A 现在为 $[0, 0, 3, 0]$ 。

要求 A_0 。您的解决方案响应为 1。A 现在为 $[1, 0, 3, 0]$ 。

要求 A_3 。您的解决方案响应为 2。A 现在为 [1, 0, 3, 2]。

最后，评分者设置 $A_1=2$ ，使得最终 $A=[1,2,3,2]$ 。这标志着第一期的结束阶段。

在代码的第 2 阶段，您的解决方案将传递列表 1 2 3 2。

它响应 1 3。

由于其中一个猜测是房子的正确索引 (1)，安娜和伯特尔赢得了游戏。

grader output	your output
1 4	
	3
2	
	3
0	
	1
3	
	2

grader output	your output
2 4	
1 2 3 2	
	1 3