



Where's Waldo?

Problem Name	whereswaldo
Time Limit	11 seconds
Memory Limit	1 gigabyte

有一个长度为 N 的隐藏排列 P_0, P_1, \dots, P_{N-1} ，保证以均匀随机的方式生成。排列包含数字 $1, 2, 3, \dots, N$ ，每个数字恰好出现一次，顺序未知。

你可以选择位置 l 和 r ，并以“ $P_l + P_{l+1} + \dots + P_r$ 之间的数字之和是多少？”的形式提问

你的任务是尽可能少地询问问题，以找到 P 中数字 1 的位置。你的得分将取决于所使用的问题数量。

交互

你的程序首先应该读取一行上的两个整数 T 和 N 。 T 是你的程序将在其中进行测试的轮数， N 是 P 的长度。

然后进行 T 轮：

当一轮开始时，你可以开始询问问题。打印一行“ $? a b$ ”来询问位置 a 到 b 之间数字的和（ $0 \leq a \leq b \leq N - 1$ ）。

在每个问题后，你的程序应该读取一个整数，表示该区间中数字的和。一旦你找到了数字 1 的位置，打印一个形如“ $! i$ ”的行，其中 i 是 $P = 1$ 的索引。在打印这个后，下一轮将开始。

确保在询问问题后刷新标准输出，否则你的程序可能会被判断为超时。在Python中，`print()`会自动刷新。在C++中，`cout << endl`除了打印一个换行外，还会刷新；如果使用`printf`，请使用`fflush(stdout)`。

约束和得分

$0 \leq l \leq l + 1 \leq r \leq N - 1$ 你的程序将针对一个单独的测试用例进行测试，其中 $N = T = 1000$ 。每个测试中的排列保证是随机生成的。如果你的解决方案在任何一轮中猜错，你的提交将被判断为错误答案。否则，将根据以下计算得分：得分 = $\min(220 - M/2500, 100)$ 分，其中 M 是你在所有 T 轮中总共询问的问题数。得分将四舍五入到最接近的整数。如果得分变为负数，将视为零分。因此，如果你使用了超过550,000个问题，你将得到0分，如果你使用了300,000个或更少的问题，你将得到100分。在中间，你的得分将线性增长。

测试工具

为了便于测试你的解决方案，我们提供了一个简单的工具，你可以下载。请参见kattis问题页面底部的“附件”部分。该工具是可选的，你可以更改它。请注意，kattis上的官方打分程序与测试工具不同。示例用法（ $T=1000$ ， $N=10$ ）：对于Python程序，假设解决方案为solution.py（通常运行为pypy3 solution.py）：
`python3 testing_tool.py pypy3 solution.py <<<"1000 10"` 对于C++程序，首先编译它（例如`g++ -std=gnu++17 solution.cpp -o solution.out`），然后运行：`python3 testing_tool.py ./solution.out <<<"1000 10"`

示例

在示例测试用例中， $T=2$ ， $N=10$ 。对于这两轮中的第一轮，假设隐藏排列是“6 10 8 7 9 1 2 4 5 3”。第一个问题? 0 9要求所有数字的和，确实是55，第二个问题? 0 4要求 $6 + 10 + 8 + 7 + 9 = 40$ 。

grader output	your output
2 10	
	? 0 9
55	
	? 0 4
40	
	? 5 5
1	
	! 5
	? 0 0
1	
	! 0