

无线电测向运动

这是一道交互题。

无线电测向运动，也称为无线电定向寻找或狐狸狩猎，是一项结合无线电技术与户外导航的运动。参与者使用特制的接收器寻找隐藏的无线电发射器，这项运动考验了方向感和无线电操作技能。

某大学开设了无线电测向运动这一门体育课。期末考试的时候，老师会将同学们带到当地某旅游景点。景点地图可以看作是一个 n 个点的环（ n 为奇数），顺时针依次编号为 $1, 2, \dots, n$ 。老师事先在两个不同的点各埋了一个发射器，然后给了你一个无线电接收器。

这个接收器比较特殊，每次接收信号需要手动按一次接收器上的按钮，然后接收器会告诉你这两个发射器距离你当前所在的点的最短距离之和（距离定义为经过的边数）。期末考核通过的标准是：在使用不超过 40 次接收器的情况下，找到这两个发射器的位置。

现在，请你写一个交互程序，从而能顺利通过期末考试。

交互协议

首先从标准输入中读取一个正整数 T ($1 \leq T \leq 10^3$)，表示测试数据组数。

对于每一组数据，首先从标准输入中读取一个正整数 n ($3 \leq n \leq 10^9$ ，且 n 是奇数)，表示环的点数。

然后开始交互过程。每一次交互，你可以向标准输出中输出以下两种格式的数据：

- **? x**: 表示你在编号为 x 的点使用一次接收器，之后你的程序需要从标准输入中读取一个整数 $dist$ ，表示两个发射器到点 x 的最短距离之和。你需要保证 $1 \leq x \leq n$ 。
 - 对于每一组数据，**? x** 操作应不超过 40 次。如果你询问的次数超过了限制，交互器将立即结束，你会接收到「答案错误」的评测结果。
- **! x y**: 表示你的答案，即这两个发射器分别所在的点（顺序无所谓）分别在 x 和 y 。此时：
 - 如果你的输出格式无误且答案正确，那么交互器将立即进入到下一组数据；
 - 否则，如果你的输出格式有误或者答案不正确，交互器将立即结束，你会接收到「答案错误」的评测结果。

注意：每次输出都需要输出换行并且刷新缓冲区，否则你可能会得到意想不到的除了答案正确之外的结果。为了刷新缓冲区，你可以：

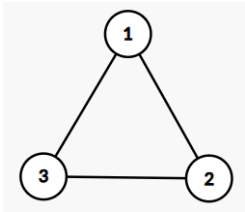
- 对于 C 或 C++，使用 `fflush(stdout)` 或 `cout.flush()`。
- 对于 Java，使用 `System.out.flush()`。
- 对于 Python，使用 `stdout.flush()`。

样例

standard input	standard output
2	
3	
1	? 1
1	? 2
2	? 3
5	! 1 2
4	? 5
3	? 1
	! 2 3

注释

第一组样例如下图，隐藏点为 1 和 2：



- 询问 ? 1，对应的最短距离之和为 $0 + 1 = 1$ ；
- 询问 ? 2，对应的最短距离之和为 $1 + 0 = 1$ ；
- 询问 ? 3，对应的最短距离之和为 $1 + 1 = 2$ ；
- 最后，输出 ! 1 2，答案正确。