Escaping from Trap

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

This is an interactive problem.

Panda, an adventurous child, has accidentally activated a trap while exploring ruins. The trap is a **regular polygon** with N vertices, numbered $1, 2, \ldots, N$ in a counterclockwise direction. The trap's center (i.e., the centroid of the polygon) must be destroyed if he wants to escape from the trap. (A regular polygon is a convex polygon that is both equilateral, meaning all sides have the same length, and equiangular, meaning all interior angles are equal.)

Panda doesn't know the side length of the polygon or his exact location. **He might be inside, on the boundary, or outside the regular polygon**. His only tool is a measurement method: he can choose any two polygon vertices, a and b, and his tool will tell him the area of the triangle formed by his current position and these two vertices. Time is short, and he has **at most** 5 measurement attempts. Your task is to help him calculate the distance from his current position to the polygon's centroid so that he can summon a ring of fire to destroy the trap.

Interaction Protocol

This problem contains multiple sets of interactive tests. First, your program should read an integer T from standard input ($1 \leq T \leq 10^4$), indicating that there are T sets of interactive tests.

At the beginning of each test set, your program should read an integer N from standard input ($4 \leq N \leq 200$), representing the number of vertices of the regular polygon. It is guaranteed that the circumradius of this regular polygon (the distance from the polygon's centroid to any vertex) is a real number in the range $[1, 100]$, and the distance from Panda's position to the centroid of the polygon (i.e., the answer) will not exceed 10000. All hidden parameters remain fixed during each test case, including the circumradius of the polygon, Panda's position, and the exact coordinates of the polygon's vertices.

For a measurement query, output a line to standard output in the form of `? x y`, where x, y are two integers from 1 to N, indicating the indices of the points on the regular polygon you want to query. The interaction system will return a non-negative real number with at least 10 valid digits, representing the area of the triangle formed by these two vertices and Panda's position. Please note that if your program makes more than 5 queries, the interactor will output a line `-1` and stop the interaction, and your solution will receive `Wrong Answer`.

After at most 5 queries, output a line to standard output in the form of `! d` to give the answer, where d is a non-negative real number representing the distance from Panda to the centroid of the regular polygon. Your output will be considered correct if the absolute or relative error between your output p and the interaction system's computed standard answer q does not exceed 10^{-4}, i.e., $\min\left(\left|\frac{p-q}{q}\right|, |p-q|\right) \leq 10^{-4}$. In this case, the interactor will output a line `Correct` and continue to the next set of data for interaction; otherwise, the interactor will output a line `Wrong` and stop the interaction, and your solution will receive `Wrong Answer`.

Note: Each output must be followed by a newline, and the standard output buffer must be flushed. To flush the buffer, you can:

- For C or C++, use `fflush(stdout)` or `cout.flush()`.
- For Java, use `System.out.flush()`.
- For Python, use `stdout.flush()`.

Example

standard input	standard output
1 4 1.000000000 3.000000000 3.000000000 1.000000000 Correct	 ? 1 2 ? 2 3 ? 3 4 ? 1 4 ! 1.000000000