

## Problem A. Alien Invasion

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 512 mebibytes

*This is an interactive problem.*

Recently, aliens from planet *Nibiru*, also known as *Not-Taking-It*, have decided to invade the Earth. As the chief of the Nibiru Special Forces headquarters, you are planning to land the first group of spaceships in Byteland, the most peaceful country of the Earth.

You know that the territory of Byteland has a shape of a polygon with  $n$  vertices on a plane (yes, Byteland is not very large, so we can view the surface around Byteland as part of the plane). Each vertex of this polygon represents a town, and each side represents a part of the border of Byteland. Of course, the border cannot intersect itself, and two sides of the polygon can have a common point only if they are neighboring (in this case, there is exactly one common point).

Your task is to find the exact area of Byteland in order to calculate the number of spaceships to be located there. However, you do not know the exact shape of Byteland, since it would take too much time to discover. Fortunately, the Nibiru Intelligence Agency has sent spies in each of the  $n$  towns of Byteland. Each spy has a huge signal lamp. You also know that the spies are numbered 1 through  $n$  such that the spies with neighboring numbers (in other words, numbers that differ either by 1 or by  $n - 1$ ) are located in towns that are directly connected by a part of the border.

You can perform the following *special operation*. At first, you tell each spy to switch his lamp either on or off. Then, you send an astronaut to the Earth, she sees all lighted lamps and tells you the least area of a convex set containing all these lamps (in other words, the area of the convex hull of these lamps). As it is difficult to keep orientation in space, the astronaut can tell nothing else about positions of the towns.

Note that if you perform a *special operation* more than  $\frac{n(n-1)}{2}$  times, the Bytelandian police will notice something strange and the invasion will fail. Find the exact area of Byteland without making too much noise!

### Interaction Protocol

At first, you get one integer  $n$  ( $3 \leq n \leq 200$ ), the number of vertices of the polygon representing Byteland.

Then you make at most  $\frac{n(n-1)}{2}$  *special operations*. A query describing the special operation should have the form of two lines:

?  $k$

$a_1 a_2 \dots a_k$

where  $k$  ( $3 \leq k \leq n$ ) is the number of spies that have to switch on their lamps and  $a_1, \dots, a_k$  ( $1 \leq a_i \leq n$ ,  $a_i \neq a_j$  for  $i \neq j$ ) are the indices of towns where these spies are located. All spies in all other towns will switch off their lamps.

After making a query, you should read one line with a single non-negative number from the input: the area of the convex hull built on lamps that are lit up (in square meters).

When you think you know the area  $x$  of Byteland, print

!  $x$

on a single line. Printing the answer does not count as a special operation. The answer should be accurate, neither any errors nor leading zeros are allowed. After that, your program should terminate gracefully.

Remember to put a newline character and flush the output buffer after making a query or printing an answer.

It is guaranteed that the polygon does not change during the program execution, and the area of the polygon is strictly greater than zero. Also, it is guaranteed that there exists a Cartesian coordinate system on a plane containing Byteland such that both coordinates of all vertices of the polygon (in meters) are integers in the range  $[-10^9, 10^9]$ .

## Example

standard input	standard output
3	? 3 1 2 3
0.5	! 0.5

## Note

For the sample test, the Bytelandian towns are located in points  $(0, 0)$ ,  $(0, 1)$  and  $(1, 0)$  for some Cartesian coordinate system.

Note that blank lines in sample input and output are printed only for clarity and do not exist in reality.