High-Dimensional Geometry

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

When a geometric problem is extended to three dimensions or even higher dimensions, its difficulty can increase significantly. To address complex n-dimensional geometric structures, researchers have proposed a model in which the 2^n key positions in space are abstracted as vertices of a graph, and specific geometric constraints between these positions are abstracted as edges in the graph.

You now have a model that has been abstracted into a graph, in which complex geometric features have been transformed into a simple undirected graph containing 2^n vertices with no duplicate edges or self-loops. However, for certain reasons, you cannot directly observe the edges in the graph. You want to know the total number of edges, m.

You may use a "subset detection detector" which works as follows:

- You input a subset S containing k vertices into the detector. The detector requires that the size of each set examined must be **exactly** half the total number of vertices in the graph, i.e., $k = 2^{n-1}$.
- The detector scans these vertices and their associated edges, and returns the total number of edges in the graph where **at least one endpoint** belongs to the set S.

Due to the high cost, the detector can be used at most 2^{n+1} times. Given the limited number of queries, find the total number of edges m in the graph.

Interaction Protocol

First, read an integer n from standard input ($1 \leq n \leq 10$). This means the graph contains 2^n vertices, numbered from 1 to 2^n.

To make a query, output `? s` to standard output, where s is a binary string of length 2^n. The i-th character should be:

- `1`, if vertex i is included in S;
- `0`, otherwise.

You must ensure that the number of `1`s in the string is exactly 2^{n-1}. If you make an invalid query or exceed the query limit, the interactor will terminate your program.

After each query, you need to read an integer d from standard input, which denotes the number of edges that have at least one endpoint in the subset S.

To output the answer, output `! m` to standard output, where m is the total number of edges. After outputting the answer, you must terminate the program immediately.

The interactor is **non-adaptive**. This means that the graph is fixed before the interaction begins and does not change according to your queries.

Note: After each output, you must print a newline and flush the standard output buffer. Use the following methods to flush:

- 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
2 1 0	 ? 1010 ? 0101 ! 1