

## Problem F. Known Problem

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

*This is an interactive problem.*

Little Anya is a competitive programmer. She also wants to write problems for the contests!

Anya came to the contest coordinator and said that she wants to make a simple problem: calculate the sum of two numbers. Surely, the coordinator responded that this problem is already present.

Anya thought for a moment and proposed to sum not just two numbers, but 10 000 of them. Unfortunately, the coordinator answered, again, that a similar problem is already used in the practice session.

Anya thought for a moment more and proposed to sum 100 000 000 numbers. The coordinator considered it for a while, but then noted that the tests will be too large for the contest server to handle.

Then Anya suggested to make the problem interactive, and instead of storing the tests, generate them on the fly. The coordinator got interested and asked how exactly will the tests be generated.

Right then, Anya remembered RANDU, one of the first widespread pseudorandom number generators, and proposed the following. Let the jury start with an integer  $s$  from 0 to  $2^{31} - 1$  which is kept secret but fixed in advance for every test. To obtain the next number  $x$  in the sequence, change  $s$  into  $(s \cdot 65\,539) \bmod 2^{31}$ , and after that, give the number  $x = s \bmod 1\,000\,000$  to the contestant. The contestant's task is to sum all the values of  $x$  we will give to them.

The coordinator considered it again and agreed, but asked to add to the interaction protocol a possibility to give the answer earlier than all numbers are generated. Anya couldn't help but wonder: why would one need that? But also agreed.

Solve a generalized version of Anya's problem. While receiving a sequence of  $n$  pseudorandom numbers generated as outlined above, find the sum of all these numbers.

### Interaction Protocol

First, an integer  $n$  is given on a separate line ( $1 \leq n \leq 100\,000\,000$ ). After that, the contestant asks from 0 to  $n$  questions, and finally gives the answer.

To ask the next question, print “?  $t$ ” on a separate line, where  $t$  is the sum of the elements of the sequence received so far.

To give the answer, print “!  $a$ ” on a separate line, where  $a$  is the sum of all  $n$  elements of the sequence. After that, terminate the program gracefully.

To prevent output buffering, flush the output buffer after each printed line: this can be done by using, for example, `fflush (stdout)` in C or C++, `System.out.flush ()` in Java, `flush (output)` in Pascal, or `sys.stdout.flush ()` in Python.

### Example

standard input	standard output	secret number
4	? 0	( $s = 238\,572\,887$ )
5	? 5	( $s = 5$ )
327695	? 327700	( $s = 327\,695$ )
966125	? 1293825	( $s = 1\,966\,125$ )
847495	! 2141320	( $s = 8\,847\,495$ )