Problem E. Number Guessing

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Alice has a secret integer y, which is selected from $[1, 10^{18}]$. Bob wants to get the number, so he asks Alice some questions. In each question, Bob gives an integer x in $[1, 10^{18}]$ to Alice, and Alice returns 0 if $y < x$, 1 if $y = x$, and 2 otherwise.

Everything sounds great, but the communication between Alice and Bob has been tampered with by Eve! Eve wrote a random number generator $\mathcal{H}$ that generates pseudorandom numbers between 0 and $n - 1$. Every time Bob gives the number, Eve will call $\mathcal{H}$ to generate a pseudorandom number x, and gives Bob the bitwise exclusive-or sum of x and the result that Alice returns.

Bob found that the result returned by Alice has been tampered with by someone because he is getting conflicting results. By some means, Bob obtained $\mathcal{H}$'s source code, which is shown below.

```
Algorithm 1: random number generator $\mathcal{H}$
  Data: constant $n$, constant $P$, seed of generator $seed$
  Result: a number $x$ selected from $[0, n-1]$
  $seed \leftarrow (seed \times n) \mod P$;
  $result \leftarrow seed \mod n$;
  return $result$;
```

Fig. 1: How $\mathcal{H}$ works.

The constants P and n are specified in the code, so Bob also knows their values. But he knows nothing about the *seed*, except that he knows the tamperer, Eve, will not modify the value during the interaction. Could you help Bob get the value of Alice's secret number in 100 queries?

Input

The first line contains a single integer t, the number of test cases ($1 \le t \le 100$). Descriptions of the test cases follow.

Each test case starts by a line containing two integers n and P: the two constants in $\mathcal{H}$. It is guaranteed that $3 \le n \le 4$, $10 \le P \le 10^{18}$, and P is a prime number.

Interaction Protocol

To ask a question, print a single line formatted as "`?` x" ($1 \le x \le 10^{18}$). Then, you should read a single line with the answer. In each test case, you can ask at most 100 questions.

If the answer is a non-negative integer, it is the number that Bob received, and you can continue guessing. If the answer is -1, it means you exceeded the number of queries or made an invalid query. Exit immediately after receiving -1, and you will see a "`Wrong Answer`" verdict. Otherwise, you can get an arbitrary verdict because your solution will continue to read from a closed stream.

If you have got the secret number y, print a single line formatted as "`!` y". Then, you should read a single line with the answer.

If the answer is -1, it means your guess is wrong, and you should exit immediately. If the answer is 1, it means your guess is correct, and you should read and solve the next test case. Note that printing this secret number does not count as one of the 100 queries.

It is guaranteed that the interactor is deterministic, which means that the interactor does not modify the values of y and *seed* dynamically based on your queries.

Do not forget to print the end-of-line character and flush the output every time your solution prints a line. Otherwise, you will likely get an "`Idleness Limit Exceeded`" verdict.

Example

standard input	*standard output*
1	
3 998244353	
	0
? 4	
	0
? 2	
	1
? 1	
	1
! 1	

Note

In the example, we have $seed = 0$, so Bob always receives the correct answer from Alice. It is reasonable for Bob to directly use binary search in this test case, but it does not apply to other possible situations.