Problem
Guess the Number (Deluxe ver.)
Time limit: 2 seconds

This is a multi-pass, interactive problem. Remember to flush the output buffer after every print. To flush your output, you can use:

- `fflush(stdout)` or `cout.flush()` in C/C++;
- `System.out.flush()` in Java and Kotlin;
- `sys.stdout.flush()` in Python.

Bob needs to guess n secret integers, each between 0 and 100 (inclusive). After guessing each number correctly, he records it by writing an 8-character binary string on a note. Later, he must recover these numbers from the notes. However, each note may have been flipped, reversing the recorded string. Therefore, Bob must use an encoding such that for any number, both the encoded string and its reverse decode to the same original number.

Interaction Protocol

Your solution is executed twice on each test case. In either of the runs, your solution will be evaluated as an interactive procedure.

First Run

The first line of input contains two integers r ($r = 1$) and n ($1 \le n \le 100$).

Then, for each of the n secret numbers:

- You may ask queries to determine the current secret number. To make a query, output a line starting with `?`, followed by a space and an integer x ($0 \le x \le 100$). After flushing, read an integer $s \in \{0, 1\}$:
  - If $s = 1$, the secret number is x;
  - If $s = 0$, it is not.
- Once you have determined the number, output a line starting with `!`, followed by a space and an 8-character string consisting only of `0` and `1`. This string encodes the number and will be stored as a note. Flush the output and proceed to the next secret number, or terminate immediately if there are no more secret numbers.

You may ask at most 100 queries per number. Recording the note does not count as a query.

Second Run

Your program is restarted for the second run.

The first line of input contains two integers r ($r = 2$) and n.

Then, for each of the n notes:

- Read a line containing an 8-character binary string. This string is either the original encoding of a secret number or its reverse.
- Output a single integer y ($0 \le y \le 100$) — the original number recorded on this note. Flush the output and proceed to the next note, or terminate immediately if there are no more notes.

The n strings are exactly those output in the first run, but each may appear as either the original string or its reverse, and in arbitrary order.

Evaluation of Correctness

Your solution passes the test case if:

- In the first run, every secret number is correctly identified (i.e., you eventually output `!` with a valid encoding of the secret number);
- In the second run, every output number matches the original number that was encoded in the corresponding note.

A testing tool is provided to help you develop and test your solution.

Read	Sample 1, Pass 1	Write
`1 2`		
	`? 0`	
`1`		
	`! 00000000` `? 0`	
`0`		
	`? 100`	
`1`		
	`! 00001111`	

Read	Sample 1, Pass 2	Write
`2 2` `11110000`		
	`100`	
`00000000`		
	`0`	