JETBRAINS icpc global sponsor programming tools
HUAWEI icpc diamond multi-regional sponsor
Jane Street icpc platinum multi-regional sponsor
Jump TRADING icpc gold sponsor

Problem E
Minus Operator
Time limit: 2 seconds

The *minus operator* on binary values a and b is defined by $(a - b) = 1$ if $a = 1$ and $b = 0$; otherwise, $(a - b) = 0$. Also, the syntax of an expression is defined as follows. Here, `x`, parentheses, and the minus are terminal symbols, and E is the start symbol.

$$E ::= \texttt{x} \mid (E - E)$$

The *judge program* possesses an expression adhering to E. The expression is hidden from you. At the start, you are only provided with the number of terminal symbols `x` in the expression, denoted by n.

Your task is to guess the expression by making a limited number of *queries*.

In a single query, you specify a binary string S of length n. The judge program then temporarily replaces each occurrence of the i-th terminal symbol `x` from the left with S_i, for $i = 1, \ldots, n$, and evaluates the replaced expression based on the definition of the minus operator. After that, the judge program returns the evaluated value to you, which is either 0 or 1.

Interaction

The first line of input contains an integer n ($3 \leq n \leq 200$). After reading the integer, your program can start making queries. For each query, your program should write a line of the form "`query` S". Here, S is a binary string of length n. Each character of S must be either `0` or `1`. In response, an input line containing the evaluated value becomes available.

Your program can make up to **500 queries.** If your program makes more than 500 queries, it will be judged as "Wrong Answer."

When your program identifies the expression that the judge program possesses, it should write a line of the form "`answer` T", where T is the expression. After that, the interaction stops and your program should terminate.

Under the constraints above, it can be shown that the expression can be uniquely identified.

Notes on interactive judging:

- The evaluation is non-adversarial, meaning that the expression is chosen in advance rather than in response to your queries.
- Do not forget to flush output buffers after writing. See the "Judging Details" document for details.
- You are provided with a command-line tool for local testing, together with input files corresponding to the sample interactions. You can download these files from DOMjudge. The tool has comments at the top to explain its use.

Read	Sample Interaction #1	Write
3		
	query 111	
0		
	answer ((x-x)-x)	

Explanation for the sample interaction #1

When $n = 3$, there are only two possible expressions: `((x-x)-x)` or `(x-(x-x))`. For a query $s = 111$, the evaluated values for these expressions are,

- $((1 - 1) - 1) = (0 - 1) = 0$, and
- $(1 - (1 - 1)) = (1 - 0) = 1$.

In this example, the judge program returns 0, indicating that the first expression is the correct one.

Read	Sample Interaction #2	Write
4		
	query 0000	
0		
	query 1001	
1		
	answer ((x-x)-(x-x))	