Interactive Primes Guessing

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 seconds
Memory limit:	256 megabytes

The jury has chosen the numbers X_0 and P ($1 \le X_0 < P \le N$, P is a prime number). You are given the number N: the upper limit for P.

Your task is to guess the number P.

In order to do that, your program will ask the questions in the form of A_i ($2 \le A_i \le N$, A_i is a prime number) where $i = 1, 2, \ldots$ is the index of the question.

After each question, the answer comes. In case the i-th question does not guess the number P, the jury calculates $X_i = (X_{i-1} \cdot A_i) \mod P$ and returns your the result of comparison of X_i and X_{i-1} ("`>`", "`<`" or "`=`").

Your program must guess the number P by asking not more than 42 questions.

Input

The first line of input contains a single integer N ($2 \le N \le 3 \cdot 10^5$). The answers for the questions of your program are given on separate lines.

If the i-th question is the number P, then the string "`OK`" is given as an answer. In other cases, the result of comparison between X_i and X_{i-1} is returned. If $X_i > X_{i-1}$ then the answer is "`>`", if $X_i < X_{i-1}$ then the answer is "`<`", otherwise the answer is "`=`".

After receiving the answer "`OK`", your program must stop sending questions.

Output

Your program must output a single question on a separate line. Each question must consist of a single prime number A_i ($2 \le A_i \le N$).

Examples

stdin	stdout
100 > > > > > OK	29 43 23 17 29 17 73

Note

Among all possible pairs (X_0, P) where $P \le 100$, only the pair $(41, 73)$ satisfies the following equations for the questions given in the first example test: $X_1 < X_0$, $X_2 > X_1$, $X_3 > X_2$, $X_4 > X_3$, $X_5 > X_4$, $X_6 > X_5$ ($X_0 = 41$, $X_1 = 21$, $X_2 = 27$, $X_3 = 37$, $X_4 = 45$, $X_5 = 64$, $X_6 = 66$).

The pipe from your program to the interactor program and the pipe back have limited size. Your program must read from the standard input to avoid deadlock. Deadlock condition is reported as "`Time Limit Exceeded`".

To flush the standard output stream, use the following statements:

In C, use `fflush(stdout);`

In C++, use `cout.flush();`

In Java, use `System.out.flush();`

In Python, use `sys.stdout.flush()`.

If your program receives an EOF (end-of-file) condition on the standard input, it MUST exit immediately with exit code 0. Failure to comply with this requirement may result in "`Time Limit Exceeded`" error.