# Xperimental Baking (x)

Memory limit: 1024 MB    Time limit: 1.00 s

This task is interactive. After printing each line, you should flush the output buffer. You can use `cout << flush` in C++, `System.out.flush()` in Java, and `sys.stdout.flush()` in Python. You must strictly follow the instructions in the *Interaction* section; otherwise **you may receive verdicts like *wrong answer*, *time limit exceeded*, or others**.

---

Dwarf the Cook is planning to bake a cake today. Unfortunately, he has forgotten the most important part of the recipe: the amount of flour he needs to add to the dough! Now there is only one way to find it: he has to determine it experimentally!

He remembers that the number of grams of flour needed for the cake is an integer between 1 and $N$. He also knows that if he adds too much flour, the cake will turn out dry and hard, and if he adds too little, it will be wet and doughy. He has calculated that his remaining ingredients allow him to make **at most** 30 baking attempts.

Can you help him figure out how much flour to use in each attempt?

## Interaction

First, read a single integer $N$.
Then, to try baking the cake with $k$ grams of flour (where $1 \leq k \leq N$), print this value in a single line and read a single character from the input. If the character is `>`, the cake turned out too wet; if it is `<`, the cake turned out too dry. If the character is `=`, the cake is perfect and your program should terminate.
You cannot make more than 30 baking attempts.

## Limits

$1 \leq N \leq 10^9$.

## Example interaction

In this example, $N = 10$ and the desired amount of flour is 7.

| Input | Output |
|---|---|
| `10` | |
| | `5` |
| `>` | |
| | `8` |
| `<` | |
| | `7` |
| `=` | |

## Local testing

In the section *Files* you can find **X.zip** containing sample tests and grader. To test your solution, compile it, then pass the test name and your executable to `./grader`:

```
./grader [test] [executable]
```

For example: `./grader 0a.in ./abc`

Before the first run, you may need to make `grader` executable. This can be done using the command:

```
chmod +x grader
```

The sample grader is **not guaranteed** to behave identically to the official one. However, neither is adaptive.