

# Heptadecimal Numbers

The Factory of Computer Enhanced Numbers (FCEN) has asked its Development Committee (DC) to come up with a way to handle numbers written in base 17. As everybody knows, base 17 is very important for many scientific applications, as well as for engineering and other practical uses. Numbers in base 17 can be tough, but are kind and soft if treated appropriately.

Numbers in base 17 are written by using a set of 17 characters: digits 0 to 9 with their usual values, and uppercase letters A to G that have values from 10 to 16, respectively. Base 17, probably because its basement on a prime number, does not require numbers to start with a non-zero digit, so each number has many representations. For instance, the decimal number 117 can be written as 6F, but also as 06F or even 0000006F. Because of this leading-zeroes thing, heptadecimal numbers are hard to compare.

As a member of the FCEN-DC, you were asked to write a program that helps in this difficult and challenging task.

## Input

The input contains several test cases. Each test case is described in a single line that contains two non-empty strings of at most  $10^5$  heptadecimal digits, separated by a single space. The last line of the input contains two asterisks separated by a single space and should not be processed as a test case.

## Output

For each test case output a single line with the sign "<" if the first heptadecimal number is smaller than the second one, the sign ">" if the first heptadecimal number is greater than the second one, or the sign "=" if both heptadecimal numbers are equal.

## Example

**Input:**  
006F B3B  
0000 0  
\* \*

**Output:**  
<  
=