2001/2002 ACM International Collegiate Programming Contest
University of Ulm Local Contest

Problem H: Hard to Believe, but True!

Source file: hard.(c|C|hs|java|pas)
Input file: hard.in

The fight goes on, whether to store numbers starting with their most significant digit or their least significant digit. Sometimes this is also called the "Endian War". The battleground dates far back into the early days of computer science. Joe Stoy, in his (by the way excellent) book "Denotational Semantics", tells following story:

> "The decision which way round the digits run is, of course, mathematically trivial. Indeed, one early British computer had numbers running from right to left (because the spot on an oscilloscope tube runs from left to right, but in serial logic the least significant digits are dealt with first). Turing used to mystify audiences at public lectures when, quite by accident, he would slip into this mode even for decimal arithmetic, and write things like `73+42=16`. The next version of the machine was made more conventional simply by crossing the *x*-deflection wires: this, however, worried the engineers, whose waveforms were all backwards. That problem was in turn solved by providing a little window so that the engineers (who tended to be behind the computer anyway) could view the oscilloscope screen from the back.
> [C. Strachey - private communication.]"

You will play the role of the audience and judge on the truth value of Turing's equations.

Input Specification

The input contains several test cases. Each specifies on a single line a Turing equation. A Turing equation has the form "*a*+*b*=*c*", where *a, b, c* are numbers made up of the digits *0,...,9*. Each number will consist of at most *7* digits. This includes possible leading or trailing zeros. The equation `"0+0=0"` will finish the input and has to be processed, too. The equations will not contain any spaces.

Output Specification

For each test case generate a line containing the word `"True"` or the word `"False"`, if the equation is true or false, respectively, in Turing's interpretation, i.e. the numbers being read backwards.

Sample Input

```
73+42=16
5+8=13
10+20=30
0001000+000200=00030
1234+5=1239
1+0=0
7000+8000=51
0+0=0
```

Sample Output

```
True
False
True
True
False
False
True
```

```
True
```