Addicted to 10-adics

Difficulty: Medium

by Taib Izzat Samawi

Time Limit: **0.1 s**

Memory Limit: **64 KB**

You are a freshman in **SaIAST** (Sophotopia Advanced Institute of Arts, Science, and Technology), an esteemed campus in Sophotopia. One day, when you are walking along the campus, you noticed a group of people debating about something. Upon walking to them, you found out that turns out they are also freshmans. They have just learned about a type of number called the 10-adics, and are fiercely debating on this strange type of number.

10-adics are numbers which have an infinite amount of digits going to the left,

> *Example: …666666666666666667 (there are infinitely many repeating 6's to the left)*
>
> > *By looking at the number alone, it may look obvious that this number is infinitely big, because the number of digits simply do not end.*

but is somehow equivalent to a finite number.

> ***…666666666666666667 is suprisingly equivalent to 1/3!***
>
> *Why? Lets try to multiply …666666666666666667 by 3:*
>
> > *…666666666666666667*
> > *3 x*
> > *———————————*
> > *…000000000000000001*
>
> *If you tried to multiply **6667** with **3**, you get **20,001** (you will carry over 2 up until the front-most digit).*
>
> *But in the case of …666666666666666667, the '**2**' will essentially be carried over forever because there is no front-most digit in this number. In other words, There's no place to put the carried over '**2**'*
>
> *So the number will just be 1 with an infinite amount of zeroes to its left. As such, the result of the equation (…000000000000000001) is exactly the same as **1**.*
>
> *This means that …666666666666666667 is one-third of …000000000000000001 but because …000000000000000001 is equivalent to 1,*
>
> ***…666666666666666667 is equal to 1/3!***

One of them noticed that you are the most popular freshman programmer from the Informatics department of the campus, and subsequently asked on whether it is possible to use a computer to know if a 10-adic number is equivalent to a regular non-infinite number.

Now knowing that you are a programmer, the other freshmans in the group beg you to help them to solve this problem. Turns out, they are on their 2nd week of sleepless nights trying to learn and know more about 10-adics, and your contribution will surely help them save up time for calculations on 10-adic numbers.

Feeling bad for them, you rush to open your laptop and create a program to help them save time and hopefully get them off from being obsessed with 10-adics.

The task is:
You are given the first 18 digits of a 10-adic number `{tenad_num}`. Then, you are given a regular non-infinite number `{reg_num}` between -9 and -1 or 1 and 9.

If `{reg_num}` is below 0, your task is to determine whether `{tenad_num}` is equivalent to `{reg_num}`.

If `{tenad_num}` is equivalent to `{reg_num}`,
print "Look!, …{tenad_num} in 10-adics is equal to {reg_num}!"
but if it isn't,
print "Hmm, it seems that …{tenad_num} in 10-adics is sadly not equal to {reg_num}"

Else, if the `{reg_num}` is above 0, your task is to determine whether `{tenad_num}` is equivalent to `1 / {reg_num}`.

If `{tenad_num}` is equivalent to `1 / {reg_num}`,
print "yes, …{tenad_num} in 10-adics is equal to 1/{reg_num}, very nice indeed"
but if it isn't,
print "well, turns out that …{tenad_num} in 10-adics is unfortunately not equal to 1/{reg_num}"

Constraints:

111111111111111111 <= tenad_num <= 999999999999999999
-9 <= reg_num < 0, 0 < reg_num <= 9
It is guaranteed that {tenad_num} has consistent numerical repeats after digit 18
You may only use `<stdio.h>` without loopings, arrays, and or strings (only module 0-1)

Testcase 0
Input:
```
666666666666666667 3
```
Output:
```
yes, ...666666666666666667 in 10-adics is equal to 1/3, very nice indeed
```
Explanation: …666666666666666667 = **1/3**

Testcase 1
Input:
```
999999999999999991 -9
```
Output:
```
Look!, ...999999999999999991 in 10-adics is equal to -9!
```
Explanation: … 999999999999999991 = **-9**

Testcase 2
Input:
```
666666666666666668 3
```
Output:
```
well, turns out that ...666666666666666668 in 10-adics is unfortunately not
equal to 1/3
```
Explanation: …666666666666666668 = **4/3**

Testcase 3
Input:
```
999999999999999999 -5
```
Output:
```
Hmm, it seems that ...999999999999999999 in 10-adics is sadly not equal to -5
```
Explanation: … 999999999999999991 = **-1**