UVa Online Judge

969 AlienAlgegra

In the planet SillyMath, the algebraist Bongo invented the Bongo Algebra. The Bongo Algebra is a calculation system for a language that only contains three operators (`k`,`i`,`t`) and a constant (`0`). The language of Bongo expressions is defined as follows:

1. `0` is a Bongo Expression.
2. If $e1$ and $e2$ are Bongo expressions, then $\texttt{i}(e1)$, $\texttt{k}(e1, e2)$ and $\texttt{t}(e1, e2)$ are also Bongo expressions.
3. There are no more Bongo expressions, except those defined by 1. and 2. above.

The motivation for Bongo Algebra is connected to Bimbo Theory; unfortunately, we cannot explain the connections in detail for lack of space. Anyway, the purpose of Bongo Algebra is to verify if equations between Bongo expressions are valid. This has been an open problem for quite a while, since Tango has raised it at the Interplanetary Algebra Congress for the first time. What has bugged researchers for so long is that nothing (really!) is known about the operators `k`, `i`, and `t`, except that they obey the two following fundamental laws:

(Law1)	$\texttt{i}(x) = x$	for any x.
(Law2)	$\texttt{t}(y, \texttt{t}(x, 0)) = \texttt{k}(x, y)$	for any x,y.

These laws have been named the Bongo Laws even if other researchers have been claiming credit for similar (unfortunately incorrect) solutions. The interest of these laws is clear: using them, one can easily determine if two Bongo expressions are equal. For example, we can check that the equation

$$\texttt{i}(\texttt{i}(\texttt{k}(\texttt{t}(0, 0), \texttt{i}(0))) = \texttt{t}(0, \texttt{t}(\texttt{t}(0, 0), 0))$$

is valid in the Bongo Algebra. Indeed, we have

i(i(k(t(0,0),i(0))) = k(t(0, 0), i(0))	applying Law1, twice
k(t(0,0),i(0)) = k(t(0,0),0)	applying Law1
k(t(0,0),0) = t(0,t(t(0,0),0))	applying Law2

Write a program that reads a sequence of pairs of Bongo expressions and determines whether each pair consists of expressions that can be proved equal by means of the Bongo Laws.

Input

The first line contains a positive integer N, indicating the number of pairs of expressions to process. The following lines include $N*2$ Bongo expressions, represented as strings built using just the characters '`t`', '`k`', '`i`', '`(`', '`)`', '`,`', and '`0`'. Starting from the first string, each consecutive pair of strings represents a pair of expressions to be compared.

Output

If the first line of the input contained the number N, then the output must contain exactly N lines, each one containing either '`true`' or '`false`', depending on whether the corresponding equation in the input is valid or invalid.

Sample Input

```
2
t(0,t(t(0,0),0))
i(i(k(t(0,0),i(0))))
i(t(0,t(0,0)))
k(0,t(0,0))
```

Sample Output

```
true
false
```