

Equivalent Expression

Time Limit: 2 Seconds

Memory Limit: 65536 KB

In computer program, there is a kind of expression, called boolean expressions. In this kind of expressions, the variables have only two values, T (TRUE) or F (FALSE) while the result can only be T or F too. All the variables in this kind of expressions are connected with boolean operation (from the highest priority to the lowest priority): ! (not), & (and), | (or) and ^ (xor). The meaning of them are:

1. ! (not). This operation has one variable. !A means to get the logic negative value of A. If A is TRUE, result is FALSE.
2. & (and). This operation has two variables. The result of A&B is TRUE if and only if both A and B are TRUE.
3. | (or). This operation has two variables. The result of A|B is FALSE if and only if both A and B are FALSE.
4. ^ (xor). This operation has two variables. The result of A^B is TRUE if A and B are not equals. If A and B are equals, the result is FALSE.

In the expressions, '(' and ')' are also used to get the higher priority inside the brackets.

One boolean expression can have many equivalent forms. Two expressions are equivalent only if the results are the same no matter what value the variables are. Now your task is to write a program to check if two boolean expressions are equivalent.

Input

The expressions are in groups in the input. Each group has two lines and one expression in each line. There are at most 200 characters in each line, at least 2 variables in each expression and no variable named as "TRUE" or "FALSE". There are at most 15 variables for each group of expressions, and each variable is a case sensitive string with at most 10 letters. Please note space can be used at proper place in the expression. An empty line means the end of input.

Output

For each group, you only need to output TRUE or FALSE in one line indicates if two expressions are equivalent.

Sample Input

```
A&(B|C)
(B|C)&A
A & B
B | A
```

Sample Output

```
TRUE
FALSE
```

Author: ZHOU, Ran