

7173 Equivalence

A **propositional formula** is generated by the following grammar:

$$\begin{aligned} \langle formula \rangle &::= \langle variable \rangle \mid \sim \langle formula \rangle \mid (\langle formula \rangle) \mid \langle formula \rangle \langle operator \rangle \langle formula \rangle \\ \langle operator \rangle &::= \wedge \mid \vee \\ \langle variable \rangle &::= a-z, A-Z \text{ (except character V)} \end{aligned}$$

where ‘ \wedge ’ encodes boolean **AND**, ‘ \vee ’ encodes boolean **OR**.

An **interpretation** is a truth-assignment for all variables occurring in a formula. The truth-value of a formula with respect to an interpretation can be determined by applying boolean operations on the values of variables, in the standard way.

Two propositional formulae are **equivalent** if they produce the same truth-value for all possible interpretations.

Input

The input file contains several test cases, each of them as described below.

The input will contain two formulae generated by the above grammar. The formulae are separated by the newline character. Variables are encoded by alphabetic characters, except the character ‘V’ which is reserved for encoding **OR**. Each formula will have at most 51 variables. Whitespaces can occur freely anywhere in the input.

Output

For each test case, the output must be ‘1’ if the two formulae are equivalent, and ‘0’ otherwise.

Note: In your implementation, you do not need to take into account operator precedence (priority). For instance, a formula such as:

$$x \wedge y \vee z$$

will be presented as either

$$(x \wedge y) \vee z \text{ or}$$
$$x \wedge (y \vee z)$$

Sample Input

```
x ^ y
y ^ x
A ^ (~y V z)
(A ^ ~y) V (A ^ z)
a V (b ^ ~b) V (c ^ ~c)
a
~x ^ ~y
~(x V y)
a ^ b ^ c ^ d
a V b V c V d
```

Sample Output

```
1
1
1
1
0
```