Problem A
Equivalence

Input File: A.in
Output File: standard output
Time Limit: 0.2 seconds (C/C++)
Memory Limit: 256 megabytes

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

<formula> ::= *<variable>* **|** ~*<formula>* **|** (*<formula>*) **|** *<formula>* *<operator>* *<formula>*
<operator> ::= `^` **|** `V`
<variable> ::= `a-zA-Z` (except character `V`)

where '`^`' encodes boolean **AND**, '`V`' encodes boolean **OR**.

An ***interpretation*** is a truth-assignment for all variables ocurring 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.

The input file 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.

The output must be **1** if the two formulae are equivalent, and **0** otherwise.

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

x ^ y V z

will be presented as either

(x ^ y) V z or

x ^ (y V z).

Sample input	Sample output
x ^ y y ^ x	1
A ^ (~y V z) (A^~y) V (A ^ z)	1
a V (b ^ ~b) V (c ^ ~c) a	1
~x ^ ~y ~(x V y)	1
a ^ b ^ c ^ d a V b V c V d	0