Problem SAMEEXPR: Same expression

Jakob is teaching compiler construction again. This time, he has an exercise for his students, where they have to simplify expressions with constants. Sadly, the simplification code his students have written is so complicated that he doesn't understand it. Instead, he resorts to testing it. He creates several expressions and passes them through the student's simplification algorithm. Now he just has to check whether the resulting expression and the original expression are equivalent.
Expressions consist of integers, variables, additions and multiplications, nested in an arbitrary depth with braces.
Grammar:

```
EXPR ::= FACTOR | FACTOR '+' EXPR
FACTOR ::= ATOM | ATOM '*' FACTOR
ATOM ::= INT | 'x' | '(' EXPR ')'
```

An "`INT`" will match the regular expression `'[1-9][0-9]*'`, so there will be only positive integer literals. To further simplify things, there will be only one variable (denoted "x"), which will appear at most 12 times in the whole expression.

Input

Input contains two lines, with the original expression in the first line, and the modified version in the second line. Every integer in the expression will be smaller than 2^{10}, and the expression will be no longer than 127 characters. To make your task simpler, you may assume that there is no integer overflow in either one of the expressions, that is, for every x in the range $\{1, \ldots, 31\}$, the expression evaluates to a result less than 2^{40}.

Output

Print one line, containing either `EQUIVALENT` if the two expressions describe the same mathematical function or `NOT EQUIVALENT` if they don't.

Sample Input 1

```
1+2+3
2
```

Sample Output 1

```
NOT EQUIVALENT
```

Sample Input 2

```
1+x+3
4+x
```

Sample Output 2

```
EQUIVALENT
```

Sample Input 3

```
(x+2)*(x+2)*(2+x)
x*x*x+3*x*x*2+3*x*2*2+2*2*2
```

Sample Output 3

```
EQUIVALENT
```