I - Twenty Four, Again

Source file name: `twentyfour.c`, `twentyfour.cpp`, `twentyfour.java`, *or* `twentyfour.py`

Yes, we know ... we've used Challenge 24 before for contest problems. In case you've never heard of Challenge 24 (or have a very short memory) the object of the game is to take 4 given numbers (the *base values*) and determine if there is a way to produce the value 24 from them using the four basic arithmetic operations (and parentheses if needed). For example, given the four base values 3 5 5 2, you can produce 24 in many ways. Two of them are: 5*5-3+2 and (3+5)*(5-2). Recall that multiplication and division have precedence over addition and subtraction, and that equal-precedence operators are evaluated left-to-right.

This is all very familiar to most of you, but what you probably don't know is that you can *grade* the quality of the expressions used to produce 24. In fact, we're sure you don't know this since we've just made it up. Here's how it works: A perfect grade for an expression is 0. Each use of parentheses adds one point to the grade. Furthermore, each inversion (that is, a swap of two adjacent values) of the original ordering of the four base values adds two points. The first expression above has a grade of 4, since two inversions are used to move the 3 to the third position. The second expression has a better grade of 2 since it uses no inversions but two sets of parentheses. As a further example, the initial set of four base values 3 6 2 3 could produce an expression of grade 3 – namely (3+6+3)*2 – but it also has a perfect grade 0 expression – namely 3*6+2*3. Needless to say, the lower the grade the "better" the expression.

Two additional rules we'll use: 1) you cannot use unary minus in any expression, so you can't take the base values 3 5 5 2 and produce the expression -3+5*5+2, and 2) division can only be used if the result is an integer, so you can't take the base values 2 3 4 9 and produce the expression 2/3*4*9.

Given a sequence of base values, determine the lowest graded expression resulting in the value 24. And by the way, the initial set of base values 3 5 5 2 has a grade 1 expression – can you find it?

Input

Input consists of a single line containing 4 base values. All base values are between 1 and 100, inclusive.

The input must be read from standard input.

Output

Display the lowest grade possible using the sequence of base values. If it is not possible to produce 24, display `impossible`.

The output must be written to standard output.

Sample Input 1	Sample Output 1
`3 5 5 2`	`1`

Sample Input 2	**Sample Output 2**
`1 1 1 1`	`impossible`