Problem E. Expression Correction

Time limit: 3 seconds
Memory limit: 1024 megabytes

Eve is studying mathematics in school. They've already learned how to perform addition and subtraction of decimal numbers and are practicing it by solving fun puzzles. The specific type of the puzzle they are solving is described below. They are given an equality with addition and subtraction which may or may not be a correct one. They have to verify the equality, and if it is not a correct one, then they have to tell if it is possible to turn it into a correct one by moving one digit to a different place in the equality.

Let us formally define the equality in this puzzle:

- *Number* is a string of **at least one and at most 10 decimal digits** ('0' to '9') that has no extra leading zeroes (the only number that is allowed to start with the zero digit is "0").
- *Expression* is a string composed of one or more numbers, as defined above, that are separated with addition ('+') or subtraction ('-') operators.
- *Equality* is a string composed of an expression, as defined above, followed by an equals sign ('='), followed by another expression.
- *Correct equality* is an equality where both expressions on the left and right hand sides of the equals sign evaluate to the same decimal number according to the standard arithmetic. Note that while all the numbers in the expression are positive, the evaluated number can be negative. Also, the evaluated number can be longer than 10 digits.
- *Moving a digit* in an equality means removing a digit from any position in the string and inserting it into another position so that the resulting string is again an equality.

The puzzle is pretty straightforward once you know how to add and subtract decimal numbers, but it is tenuous. It is easy to get distracted and make a mistake while performing computation. Your task is to write a program that solves the expression correction puzzle to help Eve.

Input

The input file consists of a single line — an equality as defined in the problem statement. The total length of the input string does not exceed 100 characters.

Output

Write a single line to the output. If the input contains a correct equality, output a single word "`Correct`". Otherwise, if the input equality can be turned into a correct one by moving one digit, output the resulting correct equality. If there are multiple possible correct equalities after moving one digit, you may output any one of them. Otherwise, output a single word "`Impossible`".

Examples

standard input	standard output
2+2=4	Correct
123456789+9876543210=111111110+11-1	123456789+987654321=1111111100+11-1
10+9=10	Impossible
24=55-13	42=55-13
1000000000-10=9999999999	Impossible