

# V BXComp

5<sup>th</sup> Programming Championship for System's Information freshmen of 2015

## 7<sup>th</sup> Stage – Challenge 1

### Identifying expressions

Recently, archeologists have found man paintings in a cave located in Lascaux, France. Beyond the conventional animal's paintings, they have found something interesting: numbers in decimal base were painted on the cave's walls. After a long investigation, the project's scientists have concluded that the symbols might be part of mathematical equations. If this were true, it would indicate that the ancient residents were more sophisticated than previously believed.

However, the problem is that the only symbols found on the walls are digits. Therefore, Roberto Langdon, a famous historian, was called to help with the interpretation of these lost symbols. Based on vague indications from other studies, he is working with the hypothesis that the people that lived in Lascaux's cave knew only three arithmetic operations – addition, subtraction and multiplication. His next step is to test all the possible combinations of the symbols '+', '-', '\*', repeated as many times as necessary, with only one '=' to find out if it would result in valid equations. For example, in one of the walls the writings "5 4 3 6" was discovered. Here, a possible solution is  $5+4=3+6$ .

Nevertheless, he is not doing this alone: he has called you to be part of his journey.

### Task

Your task is to develop a program in which, once it is given a string with several numbers, all possible combinations of the symbols '+', '-', '\*' and '=' are tested to figure out if it results in a valid equation. Afterwards, you must print the final decision: whether the combination is possible, showing the equation, or not, showing the message "Impossible".

For this, consider the following points:

- There are only binary operators, which means that the symbol '-' never means unary negation, for example, "-41";
- The operations must be solved from left to right. For example, in case of "3 3 5 30", the given solution must be " $3+3*5=30$ ", since the addition will be done first;

- The priority of operations is, from high to low: addition, subtraction and multiplication. For example, in case of “4 2 2”, the given solution must be “4=2+2”, even if “4=2\*2” is also possible;
- The final equation must consider as less digits and symbols on the left of the “=” as possible. For example, in case of “3 2 1”, the given solution must be “3=2+1”, even if “3–2=1” is also possible.

## Input

The input of your program will consist of a sequence of test cases. In each test case, there will be only a line with a sequence of  $n$  non-zero positive numbers ( $2 \leq n \leq 12$ ), whose product will be less than  $2^{31}$ . Each number will be separated with a single white space.

## Output

For each test case, there must be one output line containing the solution to that case, with the numbers of the input and the respective signs ‘+’, ‘-’, ‘\*’ and ‘=’ inserted in the correct spaces, forming a valid equation. Do not print any white space in the equation.

If there is no way to insert operators to make a valid equation, then output the message “Impossible” (without quotation marks).

## Input Example

```
3 3 5 30
18 3 3 5
5 3 3
4 2 2
```

## Output Example

```
3+3*5=30
18-3=3*5
Impossible
4=2+2
```