Markov Algorithm

Markov schemes are well known field of study in theory of algorithms. Let's define some slightly simplified schemes. We have some character set (alphabet), and a string composed of these symbols. Besides, we have rules of replacement, and each rule specifies the substring to be replaced and the replacement string. The replacement string can be empty. The order of rules is fixed. Everything functions as follows. Rules are looked through in input order. The first rule which can be applied, is applied exactly once (the given replacement which is described by the rule is carried out), and then the operation cycle repeats (the list is looked through anew). If there are many substrings in the line which satisfy a given rule then only the first (i.e., the leftmost) will be replaced. The process comes to an end if after the last rule application the line has not changed. During the process of replacements the length of the initial string can change, increasing or decreasing.

Example:
Let's consider the string *ababaab*
and the set of rules
ab->c
cc->ab
As a result of the application of rules the initial string will be transformed into "cac". The following intermediate results will appear: *ababaab, cabaab, ccaab, ccac, abac, cac*.

That's all. And now some tasks.

Task 1: The initial non-empty string was obtained from an ordinary arithmetic formula by the deletion of every symbol except brackets. You are to write the set of rules which transforms this string into the string "RIGHT" or "WRONG" depending on the correctness of brackets positions, following rules of bracket use in arithmetical expressions. For example, the string ()((()))() must be replaced by the string RIGHT, while the string (() must be replaced by the string WRONG with the same replacement rules.

Task 2: The initial string represents an arbitrary string of the following type [integer1]+[integer2]=?, where [integer1] and [integer2] are decimal representation of some positive integers. You are to write a set of rules which translates the initial string into a string of the following type: [integer1]+[integer2]=[sum], where [sum] is the decimal representation of the sum of two numbers integer1 and integer2. For example, the string 2+2=? must be replaced by 2+2=4, and the string 25+76=? must be replaced by 25+76=101. The integers can be up to 100 digits in decimal notation.

Task 3: You are given a string which consists of uppercase letters (A-Z), and ends with the "?" sign. You are to output this string in reverse order without the "?" sign. For example, the string ABBCD? must be transformed into DCBBA

Task 4: You are given a binary integer which consists of 0s and 1s. You are to write it as a string of letters "z", and the total number of such letters must be equal to the given binary number. For example, 110 must be translated by the algorithm into "zzzzzz"

Task 5: You are given a string which consists of uppercase letters (A-Z), and ends with the "?" sign. You are to output this string in non-descending order without the “?” sign. After applying the rules to DFAAS?, the string should look like this: AADFS

Task 6: You are given two decimal positive integers separated by the "_" symbol and followed by "=?", for example 30_42=?. You are to find and output the value of the greatest common divisor for this pair. For example for 30_42=? it’ll be 6.

Note: The limit on the increase of the string during the work of the algorithm is 100000 symbols.

Input

There is no input data for this problem.

Output

You should output your solutions to every task one by one. If you don’t want to solve some particular task just output 0, otherwise output the length of your solution N. Then output exactly N lines *abc->def*, where *abc* is the substring for replacement, and *def* is the substring by which it will be replaced.

Score

For each correctly solved task you gain 1 point and additionally 1/N bonus points, where N is the length of the proposed solution.

Example

Output:

```
0
0
0
5
a->b
b->c
d->v
g->l
l->a
0
0
```

Score:

In this case (if the set of rules leads to a correct answer)
you’ll get 1 + 0.2 = 1.2 points for the 4-th task.