O. Birthday Cake

```
Time limit:   1 second
Memory limit:   65535 kBytes
```

Description

You're turning 20 on today's contest day. On this occasion, you would like to have a cake made with a formula on it, consisting of the digits of your date of birth, along with some arithmetic operators and parentheses. The value of the formula should equal your age. You've seen this happen several times in your circle of friends, and you've even taken photos of it (see below).

When you think about the task, you realize that such a formula is not so easy to produce. Therefore, you plan to create a program for yourself that will generate all possible formulas for you.

The first step is to turn the digits of your date of birth into meaningful numbers in every possible way.

When you have a sequence of numbers (in your case $2, 0, 0, 4, 1, 12, 3$ is one such sequence), all you have to do is write operators and parentheses between and around the numbers, so that the value of the formula is your age. For example: $(2 + 0 + 0 + 4 - 1) * 12/3 = 20$. If you manage to get this far, you can even have the formula written on your cake. However, it's not always possible to calculate your age from a sequence of numbers, no matter what operators and however many parentheses you try to use.

Your Task

Since generating formulas doesn't seem an easy task, you only need to generate the valid number sequences, but you have to generate them in all possible ways.

A sequence of numbers is considered valid if it contains the digits of your date of birth, and the digits are in the same order as they appear in your date of birth. No more, no less, and no other digit shall appear in a valid sequence of numbers. No number in a sequence can begin with 0, but 0 itself is considered a valid number.

Input

The input contains at most 5 test cases, one per line.

Each test case consists of a date in Hungarian format ($YYYY.MM.DD.$, where $1900 \leq YYYY \leq 2100$, $01 \leq MM \leq 12$, $01 \leq DD \leq 31$).

Output

For each date of birth read from the input, write a block of lines to the standard output. The first line of the block should specify the number of valid number sequences for that date of birth. The remaining lines of the block should contain the valid number sequences themselves, one per line, in lexicographically ascending order. The numbers in each sequence should be separated by exactly one space.

The blocks for two consecutive dates of birth should be separated from each other by exactly one blank line.

Example

Input	Output

1900.01.01.	30
	1 9 0 0 0 1 0 1
	1 9 0 0 0 10 1
	1 9 0 0 0 101
	1 90 0 0 1 0 1
	1 90 0 0 10 1
	1 90 0 0 101
	1 900 0 1 0 1
	1 900 0 10 1
	1 900 0 101
	1 9000 1 0 1
	1 9000 10 1
	1 9000 101
	1 90001 0 1
	1 900010 1
	1 9000101
	19 0 0 0 1 0 1
	19 0 0 0 10 1
	19 0 0 0 101
	190 0 0 1 0 1
	190 0 0 10 1
	190 0 0 101
	1900 0 1 0 1
	1900 0 10 1
	1900 0 101
	19000 1 0 1
	19000 10 1
	19000 101
	190001 0 1
	1900010 1
	19000101