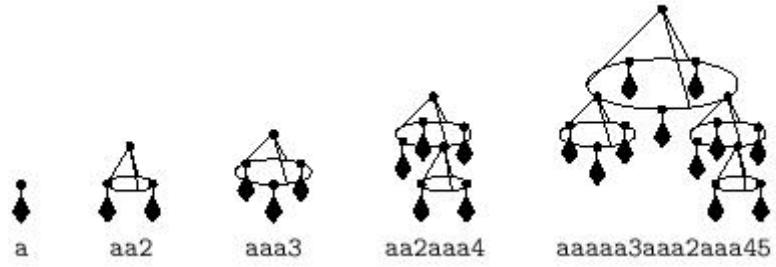
3215 Chandelier

Lamps-O-Matic company assembles very large chandeliers. A chandelier consists of multiple levels. On the first level crystal pendants are attached to the rings. Assembled rings and new pendants are attached to the rings of the next level, and so on. At the end there is a single large ring — the complete chandelier with multiple smaller rings and pendants hanging from it.

A special-purpose robot assembles chandeliers. It has a supply of crystal pendants and empty rings, and a stack to store elements of a chandelier during assembly. Initially the stack is empty. Robot executes a list of commands to assemble a chandelier.



On command "`a`" robot takes a new crystal pendant and places it on the top of the stack. On command "`1`" to "`9`" robot takes the corresponding number of items from the top of the stack and consecutively attaches them to the new ring. The newly assembled ring is then placed on the top of the stack. At the end of the program there is a single item on the stack — the complete chandelier.

Unfortunately, for some programs it turns out that the stack during their execution needs to store too many items at some moments. Your task is to optimize the given program, so that the overall design of the respective chandelier remains the same, but the maximal number of items on the stack during the execution is minimal. A pendant or any complex multi-level assembled ring count as a single item of the stack.

The design of a chandelier is considered to be the same if each ring contains the same items in the same order. Since rings are circular it does not matter what item is on the top of the stack when the robot receives a command to assemble a new ring, but the relative order of the items on the stack is important. For example, if the robot receives command "`4`" when items $\langle i_1, i_2, i_3, i_4 \rangle$ are on the top of the stack in this order (i_1 being the topmost), then the same ring is also assembled if these items are arranged on the stack in the following ways: $\langle i_2, i_3, i_4, i_1 \rangle$, or $\langle i_3, i_4, i_1, i_2 \rangle$, or $\langle i_4, i_1, i_2, i_3 \rangle$.

Input

Input file contains several test cases. Each of them consists of a single line with a valid program for the robot. The program consists of at most 10 000 characters.

Output

For each test case, print two output lines. On the first line write the minimal required stack capacity (number of items it can hold) to assemble the chandelier. On the second line write some program for the assembly robot that uses stack of this capacity and results in the same chandelier.

Sample Input

```
aaaaa3aaa2aaa45
```

Sample Output

```
6
aaa3aaa2aaa4aa5
```