

The executive officers of the company where you work want to send each other encrypted messages. Rather than use off-the-shelf encryption software such as PGP, they have tasked the IT staff with handling the encryption problem. The IT staff decided on a solution that requires “public” and “private” integer keys. The idea is that everyone can see your public key, but only you know your private key.

Your best friend in the company is a wonderful person but a not-so-wonderful programmer. He has created a public/private key scheme as follows. A public key can be any positive integer. The corresponding private key is the smallest bipartite number that is greater than and a multiple of the public key.

A bipartite number is any positive integer that contains exactly 2 distinct decimal digits  $s$  and  $t$  such that  $s$  is not 0 and all occurrences of  $s$  precede all occurrences of  $t$ . For example 44444411 is bipartite ( $s$  is 4 and  $t$  is 1), So are 41, 10000000, and 5555556. However, neither 4444114 nor 44444 are bipartite.

Notice that the large bipartite number 8888888888800000 can be nicely described as 12 8’s followed by 5 0’s. You can express any bipartite number using four numbers:  $m s n t$ . The numbers  $s$  and  $t$  are the leading and trailing digits as described above,  $m$  is the number of times the digit  $s$  appears in the bipartite number, and  $n$  is the number of times the digit  $t$  appears.

The trouble with your friend’s scheme is that it is not too difficult to compute a private key if you know the public key. You need to convince your friend that his public-private key scheme is inadequate before he loses his job over his bad decision! You must write a program that takes public keys as input and displays the corresponding private keys.

## Input

The input consists of several test cases. Each test case is on a separate line, and it consists of a single public key in the range  $1 \dots 99999$ .

The last case is followed by a line containing the integer zero.

## Output

For each test case, display a line consisting of the public key, a colon, then 4 integers ‘ $m s n t$ ’ where  $m$ ,  $n$ ,  $s$ , and  $t$  are as described above.

## Sample Input

```
125
17502
2005
0
```

## Sample Output

```
125: 1 5 2 0
17502: 4 7 4 8
2005: 3 2 3 5
```