

Problem I

Entering the Time


Problem ID: enteringthe

CPU Time limit: 1 second

Memory limit: 1024 MB

Author: Johan Sannem

Source: LTH Challenge 2

License:  CC BY-SA

Arghs! Yet again, all the clocks in Petra's home show the wrong time due to a power outage that occurred while she was sleeping. This means she has to spend her day resetting all the clocks to the correct time. Now, you might not consider this a big deal. After all, how many clocks does any single household really have? Well, Petra just so happens to be a collector of clocks. She literally has hundreds of clocks – and that is just in her bedroom! Thus, it is understandable that she does not look forward to all this resetting.



A digital clock changing numbers. License: CC BY-SA 3.0. Author: Beyond silence.

You, as her friend, would prefer if she did not need to spend her entire day resetting all her clocks. Especially since this means she would not have any time to play with you! If only you could construct some automated way of entering the correct time into all the clocks, perhaps through some kind of computer code...

A-ha, of course! You can write a program to determine how to set the correct time as quickly as possible!

Each clock has a display of 4 digits: two are used to display the hour (between 00 and 23), and two are used to display the minute (between 00 and 59). The time can be changed by selecting a digit and either decreasing or increasing by 1. Decreasing a digit that is 0, it is turned into 9, and increasing a digit 9 turns it into 0. However, the clock can not display invalid times. This means that at any given time, the hour must be between 00 and 23 and the minute between 00 and 59.

Write a program that, given the original time of a clock and the current time, determines how to set the clock correctly.

Input

The input consists:

- one line with the time that the clock is currently set to.
- one line with the current time.

Each time has the format `hh:mm`, where `hh` is a two-digit number between 00 and 23, and `mm` is a two-digit number between 00 and 59.

Output

The first line contains the number of different times seen on the clock when setting it correctly. Then for each time output it on the same format as above `hh:mm` on a separate line. Include both the original time and the final time.

Sample Input 1

```
00:00
01:01
```

Sample Output 1

```
3
00:00
01:00
01:01
```

Sample Input 2

```
00:08
00:00
```

Sample Output 2

```
3
00:08
00:09
00:00
```

Sample Input 3

```
09:09
20:10
```

Sample Output 3

```
6
09:09
09:00
09:10
00:10
10:10
20:10
```

