Problem 3—Those Wacky Clocks

Professor Dumbledore sure enjoys his collection of clocks at Hogwarts. Unfortunately, Muggles such as we are unable to make head or tail out of them. Some of his clocks look just like ordinary clocks, but they run at different speeds. It's been said that even a stopped clock is right twice a day, but clocks that are just a little slow or just a little fast are right very very infrequently. Given the initial times on two 12-hour analog clocks (meaning they do not register any difference between AM and PM), and their running speeds, you are to compute the first time starting from and including the present moment at which they coincide. Note: you do not have enough information to compute what the true time is; you are only to compute what time the clocks will read when they read the same time. If the first coinciding time occurs between minutes, round *down* to the nearest minute.

INPUT SPECIFICATION.

The file will consist of several lines. Each line (except the last) will contain six decimal integers and be terminated by **<EOLN>**. There will be no leading or trailing spaces on the line, but the six integers themselves may be separated by any number of spaces. The first two integers are the initial hour and minute of the first clock. The third integer is the rate of the first clock: it will fall inclusively between 50 and 200, and corresponds to the percentage ratio of its speed to the speed of a correct clock. For example, 50 indicates that the clock is running at half speed, 200 that the clock is running at double speed, and 100 indicates the clock is running at the correct speed. The fourth, fifth, and sixth integers are the initial hour, minute, and rate of the second clock, defined similarly. The final line will consist of 0**<EOLN>** and is not to be processed.

OUTPUT SPECIFICATION.

The output cases are to appear in the same order in which they appear in the input file. Each input case corresponds to one line in the output file. (Each line is to be terminated by **<EOLN>**.) If the clocks will never coincide, you should print out the sentence "The clocks will never converge." If they do converge, you should print out the sentence "The clocks will converge at *h*:*mm*." where *h*:*mm* is the time at which they coincide. This should look like a regular time: there will always be two digits in the minutes portion, between 00 and 59, and the hours portion will fall between 1 and 12, and will be either one or two digits depending on whether the hour is a one- or two-digit number. Remember: format counts. The output must be formatted exactly as specified.

SAMPLE INPUT.

```
4 30 50    5 30 100<EOLN>
12 29 100  12 30 100<EOLN>
0<EOLN>
<EOF>
```

SAMPLE OUTPUT.

```
The clocks will converge at 3:30.<EOLN>
The clocks will never converge.<EOLN>
<EOF>
```