COCI '18 Contest 5 #1 Titlovi

Time limit: 1.0s **Memory limit:** 64M

In order to understand speech in foreign films, the translation of the spoken text is displayed at the bottom of the screen, colloquially called subtitles. In the digital world, subtitles are often written in .SRT files that consist of blocks of the following format:

```
[Ordinal number of the block]
[Beginning time of the subtitles --> Ending time of the subtitles]
[Subtitle text in one or more lines]
```

These blocks are separated by one blank line. The time is in the form of `HH:MM:SS,TTT` (note the colon and the comma), meaning `hour:minute:second,millisecond`. One second contains $1\,000$ milliseconds. Here's an example of a .SRT file:

```
1
00:00:01,600 --> 00:00:04,200
Good day!

2
00:00:05,900 --> 00:00:07,999
Good day to you too!
Here you go!

3
00:00:10,000 --> 00:00:14,000
May I please have ten garlic sausages?
```

Sometimes subtitles are not fully synchronized with the associated movie, and appear for example two seconds too soon or too late. Because of that, we want all the times in the appropriate .SRT file, or in some part of that file, to shift forward or backward for a specified number of (milli)seconds. Write a program which does that.

Input

Input data contains up to 30 lines that represent part of some .SRT file.

The first line begins with a block with the ordinal number X, followed by blocks with ordinal numbers $X+1, X+2, \ldots, X+K$, in the form described in the task statement.

Blocks are separated by a single blank line. The subtitles text within the block contains one or more non-empty lines containing the letters of the English alphabet and the punctuation marks `',.?!`.

The line after the last block contains only the `#` sign that indicates the end of the observed part of the file and that character will not appear elsewhere.

The following (last) line contains the time in milliseconds $(-10\,000 \leq T \leq 10\,000)$, which needs to be added to all subtitle blocks.

Output

Print part of the .SRT file given in the input data, whereby the timestamps (beginning and ending time of each block) should be increased by T milliseconds. Test samples will ensure that the required timestamps are not negative.

Sample Input 1

```
8
00:00:01,600 --> 00:00:04,200
We thought you was...

9
00:00:05,900 --> 00:00:07,999
a toad.
#
300
```

Sample Output 1

```
8
00:00:01,900 --> 00:00:04,500
We thought you was...

9
00:00:06,200 --> 00:00:08,299
a toad.
#
```

Sample Input 2

```
624
00:43:30,566 --> 00:43:32,108
Howdy do, ladies?

625
00:43:32,276 --> 00:43:33,943
Name of Pete.

626
00:43:47,124 --> 00:43:48,082
Ain't you gonna
introduce us, Pete?
#
-500
```

Sample Output 2

```
624
00:43:30,066 --> 00:43:31,608
Howdy do, ladies?

625
00:43:31,776 --> 00:43:33,443
Name of Pete.

626
00:43:46,624 --> 00:43:47,582
Ain't you gonna
introduce us, Pete?
#
```