5953 Fast Times at Ridgemont's Planet

NASA has established communications with a new planet discovered by Dr. Ridgemont. This planet, coincidentally, has a calendar very similar to the calendar we use here on Earth. But the minutes, hours, days, and so forth are all of a different duration. For example, on Ridgemont's Planet, a minute might last 54 seconds, there might be 61 minutes per hour, 23 hours per day, and 340 days per year. The year is split into months, and as on Earth, each month may have a different number of days. The inhabitants have set the calendar this way so that there is no need to handle leap years or daylight savings time, but other than that everything is different.

NASA software engineers want to practice good code reuse, so naturally they want to port their Linux source code into the communications system they use to talk to the inhabitants on this distant world. Much of their software utilizes the function "difftime", which takes two specifications of a date and time and returns the number of seconds between them. For example, on Earth, if I ask for the difference between 12:35:00 on one day and 12:35:15 on the next day, the answer is 86,415. This represents one day's worth of seconds (24 hours times 60 minutes times 60 seconds is 86,400) plus an additional 15 seconds. All date/time specifications are expressed as a string with the format "$MM/DD/YY\text{-}HH\text{:}MM\text{:}SS$".

Intuitively, the procedure used by difftime is to determine how many complete days have elapsed between the input specifications and add in the correct number of seconds. Then determine how many more hours are needed, then how many more minutes are needed, and so forth.

Example: Suppose that the planet has 13 months and the number of days in each month as shown in the table below. There are a total of 399 days in a year.

Month	1	2	3	4	5	6	7	8	9	10	11	12	13
Days	30	30	31	31	31	31	31	31	31	31	31	30	30

The planet has 23 hours per day, each hour has 53 minutes, and each minute has 61 seconds.

A NASA engineer wants to know the difference between 02/29/11-22:30:05 and 03/01/1110:00:00. The calculation performed by difftime might proceed as follows:

Description	Seconds on the Planet
02/29/11-22:30:05 to 02/29/11-22:31:00	56
02/29/11-22:31:00 to "midnight" 02/30/11-00:00:00	1342
02/30/11-00:00:00 to "midnight" 03/01/11-00:00:00	74359
03/01/11-00:00:00 to 03/01/11-10:00:00	32330
Total number of seconds	108087

And in this case the function returns 108087 "planet seconds".

Write a program that reads the characteristics of time on a distant planet and two date/time specifications, and then prints the difference between the two times. You are guaranteed that the first date/time specification is not after the second time.

Input

There will be an arbitrary number of cases to consider. The input for each case will begin on a new line and consist of a series of values that span one or more lines. Each series will contain the following data items in the order specified, separated by one or more spaces and/or newlines. The allowable range of values for each item is given in parentheses following the item.

- The number of seconds in a minute (1..100)

- The number of minutes in an hour (1..100)

- The number of hours in a day (1..100)

- The number of months in a year (1..99), followed by that many integers, each of which represents the number of days in the corresponding month (1..99)

- A start date/time specification in the format $MM/DD/YY\texttt{-}HH\texttt{:}MM\texttt{:}SS$, without embedded whitespace.

- An end date/time specification in the format $MM/DD/YY\texttt{-}HH\texttt{:}MM\texttt{:}SS$, without embedded whitespace.

Like on Earth, months and days are numbered starting with one, and years, hours, minutes, and seconds are numbered starting with zero.

The series following the input for the last case will contain a single zero.

Output

For each case, your program is to print the difference between the start and end date/time specifications in seconds. Each result is to be printed on a line by itself, starting in the first column, without any trailing whitespace.

Sample Input

```
61 53 23 13 30 30 31 31 31 31 31 31 31 31 31 30 30 02/29/11-22:30:05 03/01/11-10:00:00
60 60 24 12 31 28 31 30 31 30 31 31 30 31 30 31 11/12/11-13:30:00 11/12/11-18:30:00
0
```

Sample Output

```
108087
18000
```