Ali works in a small software company in which the waging method is based on the useful working hours of people (not just their presence). This information is provided by the employees themselves based on a mutual trust. Ali is a rigorous person on such issues. So, during the day, he records all the information needed to calculate the total useful working hours of that day. During months of his work, he has evolved a special format for his recordings. The recording consists of several terms. Each term is a positive or negative time (with separate hours and minutes parts), e.g. `-8:30` or `+09:20`. Ali found that he can express all he needs to record using such terms. For example:

- To show a 90 minute working chunk, he can add either of the terms `+0:90` or `+1:30`.
- If he started a work at 9:00 AM and finished it at 1:30 PM, he can write it with terms `-9:00` and `+13:30`.
- To show a 15 minute break during his job, he can just add a `-0:15` term.
- If he has paused his work at 13:15 and resumed it at 14:05, he can write the break with terms `+13:15` and `-14:05`.

Obviously, the order of the terms does not affect the final result. As a programmer, Ali knows summing up the terms is a boring and error-prone task — something which could be done by computers. Your task is to write a program that does all these summations for Ali. But, entering the recorded terms in the above format was still annoying for Ali. So he tweaked the format a little bit in order to speed up the data entry process:

1. The hour and minute parts are non-negative, but there is no constraint on their maximum value. So, all of the terms `+25:30`, `+24:90`, and `+23:150` are valid and have the same meaning.
2. Typing leading zeros are not necessary. So, `+8:5` is the same as `+08:05`.
3. Typing zero parts are not necessary. So, `+8:  = +8:0` and `-:5 = -0:05`.
4. Typing the '`:`' character in each term halves the speed of data entry, because it is the only character which is not available in the number-pad (on the right side of the keyboard) and it also needs the shift key! Thus, from now on, the '`.`' character has the same meaning as the '`:`' character. So, the term `+8.5` means `+8:05`, not `+8:30`!

Input

There are several test cases in the input. Each test case, consists of several lines (at most 1000 lines) each containing a single term. Test cases are separated with a line containing '`$$$`', and a line containing '`###`' follows the last test case.

Output

For each test case, output a line containing the sum of the terms. It is guaranteed that the answer is positive. The answer should be written in the '$H:MM$' format. So, the minutes part (MM) is always a two digit number in range [00..59], and the hours part (H) is a non-negative integer with no leading zeros.

Sample Input

```
-11.45
-:5
-1:10
-.30
-.5
+21.55
$$$
-8.
-1.
-:3
+12.
+.10
###
```

Sample Output

```
8:20
3:07
```