ProgTest ► BIE-PA1 (11/12 ZS) ► Homework 02 ► Meeting schedule Logout

Meeting schedule

Submission deadline:	**2011-10-30 23:59:59**	1078644.029 sec
Evaluation:	**0.0000**	
Max. assessment:	**3.0000** (Without bonus points)	
Submissions:	0 / 10 Free retries + 20 Penalized retries (-2 % penalty each retry)	
Advices:	0 / 2 Advices for free + 2 Advices with a penalty (-10 % penalty each advice)	

Managers have to solve problems of many kinds. The most difficult problem of all managers, however, is their own time schedule. And time schedule gets even more complicated when organizing manager meetings. Your task is to develop a simple program to simplify meeting schedules.

We assume three managers have to agree on a meeting. In the ideal case, all three managers would be present. It is however acceptable if two of the three managers are present. For the sake of simplicity, we assume each manager provides one time interval when he/she is available for the meeting. The program shall read the three time intervals, analyze them, and propose possible time interval (or intervals) for the meeting.

The input of the program are three time intervals. Each interval is given by start time (hour and minute) and end time (hour and minute). Hours and minutes are separated using colon (:), start and end time is separated by a dash (-). The format is shown below.

The output is the analysis of possible meeting schedules. If there exists a time interval where all three managers could be present, the program preferably displays this time interval. Else, if only two managers could be present, the program displays a list of all such intervals (caution - it could be more than one interval). Finally, if the input intervals do not overlap at all, the program displays that the meeting cannot be organized. The exact output format is shown below.

The program must detect erroneous input. If the input is erroneous, the program detects it, displays an error message, and terminates. The program must detect invalid input immediately when the input is read (i.e. do not postpone input validation after whole input is read). The following is considered an error:

- non-numerical value,
- invalid (out-of-range) value (note: valid hours are 0-23),
- missing colon or dash in the interval description,
- zero length interval, or
- interval start time is not before interval end time.

The output of your program must exactly match that of the reference. Again, use the enclosed archive and test your program with the provided input/expected output test data (see FAQ). Do not forget newlines, especially after the last line of the output.

Your program will be tested in a restricted environment. The testing environment limits running time and available memory. The exact time and memory limits are shown in the reference solution testing log. However, neither time nor memory limit could cause a problem in this simple program.

Sample program output:

```
Manager A:
12:00 - 16:00
Manager B:
10:00 - 15:00
Manager C:
11:30 - 14:45
All three managers: 12:00 - 14:45
```

```
Manager A:
9:00 - 12:02
Manager B:
```

```
11:30 - 14:30
Manager C:
13:15 - 18:00
Two managers: 11:30 - 12:02, 13:15 - 14:30
```

```
Manager A:
10:00 - 12:00
Manager B:
10:00 - 13:15
Manager C:
16:00 - 20:00
Two managers: 10:00 - 12:00
```

```
Manager A:
9:00 - 11:00
Manager B:
12:30 - 14:20
Manager C:
14:21 - 17:32
Meeting is not possible.
```

```
Manager A:
2:13 - 4:15
Manager B:
4:15 - 6:32
Manager C:
2:13 - 6:32
Two managers: 2:13 - 6:32
```

```
Manager A:
14:15 - 16:78
Invalid input.
```

```
Manager A:
13:30 - 13:15
Invalid input.
```

```
Manager A:
13:15   13:30
Invalid input.
```

```
Manager A:
12:45 - 16:abcd
Invalid input.
```

Help

- Time expressed as a pair hour:minute is inconvenient for any comparison or other manipulation. Thus, it is recommended to convert the time into some linear representation (e.g. the total number of minutes from the midnight).
- This problem requires many conditions and logical operators (&& a ||). Try to simplify conditions as much as possible. Two comparisons are enough to test intersection/overlapping/... of two intervals.
- It is recommended, through not required, to use functions in your implementation.

Sample data:		**Download**
Submit:	[redacted]	**Submit**

□ **Reference**