

# Fair billing

You work for a hosted application provider which charges for the use of its application by the duration of sessions. There is a charge per second of usage. The usage data comes from a log file that lists the time at which a session starts or stops (in the format HH:MM:SS), the name of the user (which is a single alphanumeric string, of arbitrary length) and whether this is the start or end of the session, like this:

```
14:02:03 ALICE99 Start
14:02:05 CHARLIE End
14:02:34 ALICE99 End
14:02:58 ALICE99 Start
14:03:02 CHARLIE Start
14:03:33 ALICE99 Start
14:03:35 ALICE99 End
14:03:37 CHARLIE End
14:04:05 ALICE99 End
14:04:23 ALICE99 End
14:04:41 CHARLIE Start
```

Unfortunately, the developer of the application omitted some vital information from the log file. There is no indicator which start and end lines are paired together. Even more unfortunately, the log files are re-written on a regular basis, so sessions may overlap the time boundaries of the log file. In other words, there may be “End” entries for sessions that were already in progress when the log file started, which will have no preceding “Start”. Similarly, when the log files are retrieved, there may be sessions still in progress that have a “Start” but no “End”.

Your task is to take the log file and to print out a report of the users, the number of sessions, and the minimum possible total duration of their sessions in seconds that is consistent with the data in the file. As you can see in the example above, a user can also have more than one session active concurrently. Where there is an “End” with no possible matching start, the start time should be assumed to be the earliest time of any record in the file. Where there is a “Start” with no possible matching “End”, the end time should be assumed to be the latest time of any record in the file. So, for a file containing only these records:

```
14:02:03 ALICE99 Start
14:02:05 CHARLIE End
14:02:34 ALICE99 End
```

the start time for CHARLIE's record should be assumed to be the earliest time in the file, i.e. 14:02:03. Similarly for the first example above:

```
...
14:04:05 ALICE99 End
14:04:23 ALICE99 End
14:04:41 CHARLIE Start
```

the last record is a “Start” and there are no later records at all so CHARLIE's last session will be considered to have finished at 14:04:41, i.e. it will be 0 seconds in duration.

Putting this all together, the results for the original data shown above would be as follows (name, sessions and total time):

```
ALICE99 4 240
CHARLIE 3 37
```

Your program should take a single command line parameter, which will be the path to the data file to read. You can assume that the data in the input will be correctly ordered chronologically, and that all records in the file will be from within a single day (i.e. they will not span midnight).

Finally, you should note that, as with most log files, there may be other invalid or irrelevant data within the file. Therefore, any lines that do not contain a valid time-stamp, username and a Start or End marker should be silently ignored and not included in any calculations.