Cron

This coding task is all about seeing how well you can take a spec and produce a functional solution.

It is meant to be challenging enough to show your skill but should be achievable within 2 hours. Take as long as you like but If you find yourself running over, it is more important that the code is functional than that it is beautiful. You're welcome to leave notes on what you would have done given more time.

On receiving your submission, we will run it against some test cases and take a look at the code itself. The testing is automated, so please take special care to ensure that the submission inputs and outputs are to spec.

Specification

We have a set of tasks, each running at least daily, which are scheduled using some simple values in a text file. You might recognise this from writing a crontab configuration in the past.

Examples of the scheduler config:

```
30 1 /bin/run_me_daily
45 * /bin/run_me_hourly
* * /bin/run_me_every_minute
* 19 /bin/run_me_sixty_times
```

The first field is the minute past the hour, the second field is the hour of the day and the third is the command to run. For both cases * means that it should run for all values of that field. In the above example run_me_daily has been set to run at 1:30am every day and run_me_hourly at 45 minutes past the hour every hour. The fields are whitespace separated and each entry is on a separate line.

We want you to write a command line program that takes a single argument. This argument is the simulated 'current time' in the format HH:MM. The program should accept config lines in the form above to **STDIN** and output the soonest time at which each of the commands will fire and whether it is today or tomorrow. In the case when the task should fire at the simulated 'current time' then that is the time you should output, not the next one.

For example given the above examples as input and the simulated 'current time' command-line argument 16:10 the output should be:

```
1:30 tomorrow - /bin/run_me_daily
16:45 today - /bin/run_me_hourly
16:10 today - /bin/run_me_every_minute
19:00 today - /bin/run_me_sixty_times
```

You can submit your solution in **any language** you like. Python, PHP, C, JavaScript, Go, Rust, Swift, Awk, Prolog, using the standard libraries that are part of that language (i.e. no 3rd party libraries for this task please!). The choice is yours as long as we can run it as described below.

We will want to run your code, so please supply instructions for building and running it. Additionally, it must work on at least one of OS X or Linux.

We want to run your code on the command line using an input like

```
./<your app> HH:MM < config
```

For example: `./application.py 16:10 < config`

Where ‘config’ is a file containing various cron style inputs like we described above.

Your code must be able to be run in this way.

Submitting your code

Please send us your code as an attachment in an email. Preferably as a zip or tar file.

If you’ve uploaded your code to GitHub or some other platform that’s fine, but please still send us the code as an attachment too.

We aim to make sure all code tests are checked anonymously to reduce implicit bias in our interview process. To help us with this we’d appreciate it if you try and avoid putting any identifying content inside the file you send us.