**prompt>**  ./estimatecron  month  crontab\_file  estimates\_file

1. Your project should check for data errors in both the *crontab-file* and the *estimates-file*. If any errors are found, only the *first* error should be reported and the program should terminate.

which specifies the month (of the current year) to be analysed (either an integer 0..11, or a 3-character name), and the name of the two required text files (you may choose *any* names for the files).

Your submitted project may produce *any* output you wish, typically debug output. It does not need to describe what it is during. Only the *very last line* will be considered during (automated) marking. It must provide 3 words separated by one-or-more whitespace characters:

* + the name of the most frequently executed command (a single word),
  + the total number of commands invoked (a non-negative integer), and
  + the maximum number of commands running at any time (a non-negative integer).
  + E.g. daily-backup 36 1

Crontab files

the minute (0..59), the hour (0..23), the day of the month (1..31), the month (0..11), and the day of the week (0=sun...6=sat)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| min | Hour | Day | Month | Day of week | command to execute | Description |
| 0 | 3 | \* | \* | \* | daily-backup | # execute a command at 3AM every day |
| 15 | 4 | \* | \* | sun | weekly-backup | # execute a command at 4:15AM every Sunday |
| 0 | 10 | 22 | 7 | mon | deep-thought | # start thinking about the project.... |
| 0 | 2 | 1 | \* | \* | send-monthly-newsletter | # mail out a monthly newsletter |
| 59 | 16 | 16 | 8 | \* | submit-project | # submit my project automatically, just in case I forget! |

Estimate files

# format: command-name minutes

weekly-backup 6

daily-backup 2

send-monthly-newsletter

submit-project 1

deep-thought 600