

Interview Question - Parse the logs

For this question, the main objective is to write a program that parses a mock log file, and returns a result summary. This programming assignment is nice, because it is practical, and could indeed be something that happens in the workplace. One would never actually write a string reversal algorithm in the workplace.

The requirements are listed below:

Output simple text diagram with the following information. For each host, list errors, and how many times they happened. Sort by host with most errors, and within each host, sort by most common error type.

Display statistic on how many records searched, how many were errors or warnings, and percentage of errors/warnings

Input guarantees: All servers will be listed with either **Consumer <id>** or **Producer <id>**, somewhere in the log message. All message entries will follow the format **<timestamp> [<error type>] <message>**. (mock_log1.log has less variety, and does not contain entries where the server identifier is surrounded in parenthesis).

Depending on the speed of the candidate, and the time allotted, only some of the requirements may be met in the timeframe. In general, the requirements should be solved in this order:

1. Reading in the file, and parsing it into an internal model
2. Sorting the error messages by server
3. Sorting the error messages into buckets of similar error messages (within the server id)
4. Getting statistical information for each server
5. Getting statistical information for the log as a whole
6. (Optional) Formatting the output in a nice html page

The candidate should be able to prioritize these tasks themselves, but if they are struggling with doing that, you should help them, so they keep on task.

Example input:

Sample Input

```
p.pl {margin: 0.0px 0.0px 0.0px 0.0px; font: 10.0px Monaco; color: #f4f4f4;
background-color: #000000; background-color: rgba(0, 0, 0, 0.85)}
span.s1 {font-variant-ligatures: no-common-ligatures}
```

```
2017-02-09T02:37:44 [ERROR] Consumer iwjef99 could not be contacted
2017-02-09T02:37:46 [INFO] Message received from Producer w89fj93
2017-02-09T02:37:51 [ERROR] Consumer 7sjeuf returned 504
2017-02-09T02:37:53 [INFO] Message received from Producer a99jef9
2017-02-09T02:37:59 [INFO] Message sent to Consumer a99jef9
2017-02-09T02:38:02 [INFO] Message received from Producer a99jef9
2017-02-09T02:38:08 [INFO] Message received from Producer 7sjeuf
2017-02-09T02:38:10 [INFO] Message sent to Consumer 98w9jfi9
2017-02-09T02:38:10 [INFO] Message sent to Consumer a9i5f
2017-02-09T02:38:12 [INFO] Message received from Producer 98w9jfi9
2017-02-09T02:38:15 [INFO] Message received from Producer w89fj93
2017-02-09T02:38:17 [INFO] Message sent to Consumer i9wjf99
2017-02-09T02:38:21 [INFO] Message sent to Consumer 7sjeuf
2017-02-09T02:38:27 [INFO] Message received from Producer iwjef99
2017-02-09T02:38:29 [INFO] Message received from Producer i9wjf99
2017-02-09T02:38:31 [INFO] Message received from Producer i9wjf99
2017-02-09T02:38:35 [INFO] Message received from Producer 98w9jfi9
2017-02-09T02:38:40 [INFO] Message received from Producer 7sjeuf
2017-02-09T02:38:40 [INFO] Message sent to Consumer w89fj93
2017-02-09T02:38:44 [INFO] Message received from Producer i9wjf99
2017-02-09T02:38:49 [INFO] Message received from Producer 98w9jfi9
2017-02-09T02:38:55 [ERROR] Consumer a99jef9 disconnected unexpectedly
2017-02-09T02:38:55 [ERROR] Consumer i9wjf99 returned 503
2017-02-09T02:38:58 [INFO] Message received from Producer w89fj93
2017-02-09T02:39:02 [INFO] Message received from Producer iwjef99
2017-02-09T02:39:04 [INFO] Message sent to Consumer a99jef9
2017-02-09T02:39:07 [INFO] Message sent to Consumer 98w9jfi9
2017-02-09T02:39:11 [INFO] Message received from Producer 63di9
2017-02-09T02:39:15 [INFO] Message sent to Consumer 8efkwi
2017-02-09T02:39:20 [INFO] Message sent to Consumer a99jef9
```

Example output:

Sample Output

```
Host gfedcba
=====
5 - [ERROR] Consumer gfedcba returned 503
4 - [ERROR] Producer gfedcba returned invalid data
2 - [WARNING] Consumer gfedcba returned 410

11% Error/Warning

Host abcdefg
=====
5 - [ERROR] Consumer abcdefg returned 503
3 - [ERROR] Consumer abcdefg could not be contacted
1 - [WARNING] Consumer abcdefg returned 410

9% Error/Warning

Total: 14% Error/Warning - 200 Messages sent/received
```

I have attached 2 sample log files, each with 2000 entries each. The program should work with both.



mock_log1.log



mock_log2.log