

37Widgets makes inexpensive thermometers and humidity sensors. In order to spot check the manufacturing process, some units are put in a test environment (for an unspecified amount of time) and their readings are logged. The test environment has a static, known temperature and relative humidity, but the sensors are expected to fluctuate a bit.

As a developer, your task is to process the logs and automate the quality control evaluation. The evaluation criteria are as follows:

- 1) For a thermometer, it is branded “ultra precise” if the mean of the readings is within 0.5 degrees of the known temperature, and the standard deviation is less than 3. It is branded “very precise” if the mean is within 0.5 degrees of the room, and the standard deviation is under 5. Otherwise, it’s sold as “precise”.
- 2) For a humidity sensor, it must be discarded unless it is within 1% of the reference value for all readings.

An example log looks like the following. The first line means that the room was held at a constant 70 degrees, 45% relative humidity. Subsequent lines either identify a sensor (<type> <name>) or give a reading (<time> <value>).

```
reference 70.0 45.0
thermometer temp-1
2007-04-05T22:00 72.4
2007-04-05T22:01 76.0
2007-04-05T22:02 79.1
2007-04-05T22:03 75.6
2007-04-05T22:04 71.2
2007-04-05T22:05 71.4
2007-04-05T22:06 69.2
2007-04-05T22:07 65.2
2007-04-05T22:08 62.8
2007-04-05T22:09 61.4
2007-04-05T22:10 64.0
2007-04-05T22:11 67.5
2007-04-05T22:12 69.4
thermometer temp-2
2007-04-05T22:01 69.5
2007-04-05T22:02 70.1
2007-04-05T22:03 71.3
2007-04-05T22:04 71.5
2007-04-05T22:05 69.8
humidity hum-1
2007-04-05T22:04 45.2
2007-04-05T22:05 45.3
2007-04-05T22:06 45.1
humidity hum-2
2007-04-05T22:04 44.4
2007-04-05T22:05 43.9
2007-04-05T22:06 44.9
2007-04-05T22:07 43.8
2007-04-05T22:08 42.1
```

Sample Output

```
{
  "temp-1": "precise",
  "temp-2": "ultra precise",
  "hum-1": "keep",
  "hum-2": "discard"
}
```

Create a JavaScript module that exports the following function: **evaluateLogFile(fileContents)** { } The contents of the log file will be passed as a string on the parameter **fileContents** to the function **evaluateLogFile()**. The function should return a JavaScript object like the sample output above. In the end, you will own this process, so you should solve the problem as described, but feel free to advocate for any changes or demonstrate any practices you think would make sense to improve the process (split into multiple files, change log format, etc). For example, one of the future plans that 37Widgets has is to make more IoT devices.