Introduction to Programming

Assignment: Task 2

The Problem

A kindly cat owner has installed a cat shelter in his garden. This acts as a refuge for his feline friend should the creature be caught outside in inclement weather when its human servant is unavailable to open the back door.

The cat shelter contains a small camera, along with a motion detector and an RFID reader that detects a chip in a tag attached to the cat's collar. This is connected via WiFi to a laptop in the house, where a program creates a log of when the cat enters and leaves the shelter. If the chip is not detected, but motion is, it can be assumed that the neighbour's cat has trespassed into the shelter. In this case, a loud siren plays automatically, and jets of ice cold water are directed at the intruder. The ferocity of the attack is such that the cat is assumed to have stayed for one minute.

There has been a significant investment in this project, and the cat owner is keen to find out how much the shelter is being used. A program to analyse a daily log file is required.

The Task

The laptop stores a data stream in a file that can be analysed. It records when a cat entered, when it left, and whether the cat was an intruder. One line is written for every cat activity, so a single line contains both the entry and departure times.

Times are stored as minutes, with a start of midnight. So midnight would be represented as zero, 1am is 60, and 9:30am would be 570 (9 x 60 + 30). There is one file created each day.

The cat shelter is small, so there can only be one cat in it at a time. It is often empty as the cat likes to bask on the roof.

The format of the file is shown by the following short sample.

```
OURS,600,630
THEIRS,700,701
OURS,842,900
THEIRS,1000,1001
THEIRS,1010,1011
END
```

So we see that the correct cat (`OURS`) entered the shelter at 10:00am (time 600), and stayed for 30 minutes (630 - 600). A different cat entered at time 700 … Hold on:

```
>>> 700 // 60
11
```

```
>>> 700 % 60
40
```

which is 11:40, and swiftly left, having been attacked with sound and water. And so on.

The data stream ends `END`, which is written automatically before the file closes at 23:59. The shelter is never occupied at this time.

It is safe to assume that the file is always in this format.

The required output from the analysis program that you will create is:

- The total number of times the correct cat has entered the house.
- The number of times intruding cats have been doused with water.
- The total time spent in the house by the correct cat.
- For the correct cat only, the average duration of each visit, the duration of the longest visit, and the duration of the shortest visit.

The name of the data file should be provided as a command-line argument. The program should therefore run from the command-line. It should handle the common errors associated with files on the command-line, such as missing arguments, files that cannot be opened, and so on.

Examples

The following illustrate what should happen when the program executes in a variety of scenarios. The examples are running from the command-line on Linux.

These examples show that command-line argument errors should be handled.

```
$ ./cat_shelter.py
Missing command line argument!

$ ./cat_shelter.py file_that_does_not_exist.log
Cannot open "file_that_does_not_exist.log"!
```

These examples use the files also included in this folder. Note that these files are all generated randomly, by a program that makes certain assumptions about feline behaviour. This is probably why the output numbers are quite similar for each file.

```
$ ./cat_shelter.py shelter_2023-08-25.log

Log File Analysis
=================

Cat Visits: 12
Other Cats: 58

Total Time in House: 4 Hours, 5 Minutes

Average Visit Length: 20 Minutes
```

```
Longest Visit:        45 Minutes
Shortest Visit:       5 Minutes
```

Note how times longer than one hour should be formatted.

```
$ ./cat_shelter.py shelter_2023-08-26.log

Log File Analysis
==================

Cat Visits: 11
Other Cats: 45

Total Time in House: 5 Hours, 0 Minutes

Average Visit Length: 27 Minutes
Longest Visit:        45 Minutes
Shortest Visit:       10 Minutes
```

```
$ ./cat_shelter.py shelter_2023-08-27.log

Log File Analysis
==================

Cat Visits: 13
Other Cats: 53

Total Time in House: 5 Hours, 50 Minutes

Average Visit Length: 26 Minutes
Longest Visit:        50 Minutes
Shortest Visit:       5 Minutes
```