**Summer Internship Programming Exercise**

This exercise is designed for us to get a little insight into you as a programmer. As such, what we want to see is nice clean code with lots of comments to help us understand your thought process. You should try to aim to complete the task within 4 hours. If you are not able to fully complete the exercise, please still send it in. More than anything this task is designed to be a little bit of fun and for you to show off your super coding skills!

**Outline**

In the energy industry, a lot of messages are sent between different participants in the form of “flows”. Flows are texted base files, and the information included in the flow is dependent on a schema. We want you to develop a command-line utility to read a flow containing meter readings (provided in a separate file).

The utility should use `argparse` which is part of the python standard library, and perform some simple data processing on it and output the results to the console.

**What we expect to see within your code:**

* We expect the whole file to be parsed into some sort of python data structure (maybe a dictionary), with the string values in the file transformed into their equivalent python data type (e.g. 20210401 would be parsed to a python datetime.date object)
* A few simple unit tests utilising the `unittest` library that validate your solution
* Clear readable code is key, with appropriate comments as you see fit

**Console** **outputs from the utility:**

* The count of meters in the file
* The total sum of valid meter readings within the file
* The total sum of invalid meter readings within the file
* The highest and lowest valid meter reading within the file
* The most recent and oldest meter reading within the file

**File Schema:**

HEADER |

METER | METER\_ID |

READING | READING\_ID | VALUE | DATE(YYYYMMDD) | STATUS (V = valid, F = Invalid) |

FOOTER |

The file consists of a HEADER row followed by multiple pairs of meter and meter reading lines. The meter is on the first line and the meter reading is on the second. The file is terminated with a FOOTER row.

**Notes:**

* You should only use the Python standard library i.e. when we test your solution, we should not need to install any third-party packages
* We will test your solution using a different file from the one that we supplied you
* You can assume that the file we will test your solution against will be valid i.e. conforms to the schema supplied, so you don’t need to worry about error handling
* If you think that it is appropriate, include some instructions detailing how to run your programme