In this technical exercise, you are asked to write two simple functions for validating and modifying telemetry data.
Telemetry data is energy consumption data in 15 minute-intervals.
One of the purposes of the Energyworx platform is to perform validation, estimation and editing of incoming data (also known as VEE).
As a technical consultant, one of your jobs is to write the logic that performs these processes.

Deliverables
- Source code;
- Instructions on how to setup a virtual environment with the necessary Python libraries for running the code;
- Instructions on how to use the code;
- Instructions on how to run the unit test(s).

Requirements:
- Written in Python;
- Provides functions for the following two processes:
    - Checking for zero reads:
        Sometimes, meters can return consumption values that are equal to zero.
        As devices always have a standby consumption, a zero read is usually an indication that this is an invalid read and thus should be flagged as such.
        -> Write a function that reads in the telemetry data for a single month given in 'zero_reads_data.csv'; replaces all zero reads with a None/NaN; and writes the data back out to a CSV file.
        Note: You can assume that there is no redelivery through solar panels or other systems on this meter.
    - Temporal aggregation:
        P4 data always comes as 15-minute interval data.
        However, it can be beneficial to aggregate this data to, for example, hourly intervals to be able to compare to other data with lower frequencies, or to be able to see more global trends.
        -> Write a function that reads in the telemetry data for a single month given in 'aggregate_data.csv'; aggregates the data to hourly intervals; and writes the data back out to a CSV file.
        Note: You can assume that this data has already been validated.
- Everything must be executable on Linux and MacOS-X;
- Unit test(s) for the implemented functions.

Tips:
- Keep the code clean and simple;
- Make sure your unit test(s) is/are only testing your own code;
- You can use any existing open-source Python library, but keep in mind that we are mostly interested in seeing how you structure and implement the above requested algorithms.