Data and files provided

There are many different ways of calculating an index of air quality, used in different countries around the world (see, for example, [the wikipedia page “Air quality index”](https://en.wikipedia.org/wiki/Air_quality_index)). The Australian system considers seven different pollutants: carbon monoxide, nitrogen dioxide, ozone, sulphur dioxide, lead, and two sizes of dust particles, PM10 and PM2.5. The measured concentration of each pollutant is linearly scaled into an index, based on what is considered an acceptable standard, so that the indices for all pollutants use the same scale. An index value of 0 means the air is clear of the pollutant, while an index value of 100 means the standard level has been reached; therefore, an index value of 100 or above is considered “poor” air quality, and an index value of 200 (twice the acceptable standard) or above is considered “hazardous”. The reported index is based on the average scaled measurement over an interval of time, which is different for different pollutants. For dust particles (PM10 and PM2.5) it is over the last 24 hours.

The data files provided by ACT Health do not record all seven pollutants. The data files are in comma-separated value (CSV) format. To simplify the assignment, we have split them into one file for each monitoring station:

* [aqi\_data\_civic.csv](https://cs.anu.edu.au/courses/comp1730/assessment/project/aqi_data_civic.csv)
* [aqi\_data\_florey.csv](https://cs.anu.edu.au/courses/comp1730/assessment/project/aqi_data_florey.csv)
* [aqi\_data\_monash.csv](https://cs.anu.edu.au/courses/comp1730/assessment/project/aqi_data_monash.csv)

Each file follows the same format. The first line of the file is a header, which gives the names of the columns. Each following line is a data entry, and contains the index values recorded at a particular date and time. The columns are:

* Name: The name of the monitoring station. This will be the same for all entries in each file, since we have split the data up by station.
* GPS: The location of the monitoring station. (The coordinates appear to be occasionally wrong, but this does not affect the assignment since we will not be using them.)
* DateTime: The date and time of the entry, in the format DD/MM/YYYY HH:MM:SS AM/PM. All entries should be on whole hours, meaning the minutes and seconds are zero.
* NO2: The nitrogen dioxide measurement.
* O3\_1hr: The ozone measurement (last hour).
* O3\_4hr: 4-hour rolling average of the ozone measurement.
* CO: The carbon monoxide measurement (8-hour rolling average).
* PM10: The PM10 particles measurement (24-hour rolling average).
* PM2.5: The PM2.5 particles measurement (24-hour rolling average).
* AQI\_CO, AQI\_NO2, AQI\_O3\_1hr, AQI\_O3\_4hr, AQI\_PM10 and AQI\_PM2.5: The air quality index values corresponding to the measurements.
* AQI\_Site: The combined air quality index for the site at the time. This should equal the highest of the index values for the measured pollutants.
* Date: The date of the entry, in the format day month year (the name of the month, rather than the number).
* Time: The time of the entry, in 24-hour format (i.e., HH:MM).

Two important facts to note:

* *The data is not complete*. There are dates/hours for which there is no entry, and even when there is an entry, some of the pollutant measurements or corresponding index values may be missing. Missing values are indicated by empty fields in the CSV file.
* *Entries in the CSV file are****not****ordered by date and time*. In fact, they appear in no particular order.

For more advice on how to read and process CSV files, see [Lab 6](https://cs.anu.edu.au/courses/comp1730/labs/lab6/).

Questions for analysis (code)

In this assignment, we will consider only the air quality index value for PM2.5 particles (column AQI\_PM2.5).

A template file for the assignment code is provided here:

* [assignment.py](https://cs.anu.edu.au/courses/comp1730/assessment/project/assignment.py)

In this file, there is only one function that you must implement: analyse(path\_to\_file). The function takes a single argument, which is the complete path (file name and optionally a directory) to the data file that it should read and analyse. You can assume that the path will be a string. The function should print out the results of the analysis. It does not have to return any value. The specific questions that your analysis should answer are described below. The following are some general requirements and things to keep in mind:

* You do not have to solve all the questions, but you can only gain marks for the ones that you have attempted (see [Marking criteria](https://cs.anu.edu.au/courses/comp1730/assessment/project/#marking-criteria) below for details on how we will mark your submission).
* Although we do not specify the exact format in which you should print the results of the analysis, you should make it easy for the user (and marker) to see what is being shown. *Ease of reading the output of your program is part of the marking criteria*.
* Although there is only one function in the assignment template that you must implement, you can define other functions and use them in your solution. Indeed, *good code organisation, including appropriate use of functional decomposition, is part of the marking criteria*.

Question 1

An air quality index of 100 or above is considered poor. For each year that is present in the data file, count the number of days that have at least one entry with an AQI PM2.5 of 100 or above. Print the results with one line per year, for example like this:

```

analysing data from file aqi\_data\_civic.csv

Question 1:

2014 had 0 days with an AQI PM2.5 of 100 or above

2015 had 0 days with an AQI PM2.5 of 100 or above

2016 had 1 days with an AQI PM2.5 of 100 or above

2017 had 2 days with an AQI PM2.5 of 100 or above

2018 had 2 days with an AQI PM2.5 of 100 or above

2019 had 36 days with an AQI PM2.5 of 100 or above

2020 had 25 days with an AQI PM2.5 of 100 or above

```

Question 2

In 2020, so far, January has been by far the worst month for air quality, because of the bushfires. But is that normally the case? We want to find out which month of the year is most frequently the worst.

To answer this question, we first have to define what it means for a month to be “worst”. We will use the same approach as in Question 1, and count the number of days in each month that have at least one reading equal to or above a threshold value; the month that has the highest number of such days is the one we consider the worst. For this question, we will use the threshold value 33, because an index below 33 is considered “good” air quality.

For each year that is present in the data file, determine which month has the most days that have at least one reading of 33 or above; then for each month, determine how many years it was the worst. Print the results for each month, excluding those that are never the worst. For example, like this:

```

analysing data from file aqi\_data\_civic.csv

Question 1:

...

Question 2:

December was the worst month in 2 years (2018, 2019)

January was the worst month in 1 years (2020)

April was the worst month in 1 years (2016)

May was the worst month in 1 years (2017)

June was the worst month in 1 years (2017)

July was the worst month in 1 years (2015)

```

Note that you will have to decide how to handle ties (if two or more months in one year had the same number of days with a reading of 33 or more). However, a month that has zero days with a reading of 33 or higher should not be considered “worst”. Remember that you should document your handling of ties (and any other interpretations or design decisions that you make) in your code, using comments and/or docstrings as appropriate.

Question 3

The highest AQI PM2.5 value in the ACT data files, 5185, was recorded at the Monash monitoring station on 2020-01-01 at 8pm. However, this is actually the average value over the last 24 hours (i.e., from 8pm the previous day). We want to calculate the actual value at that point in time. The PM2.5 index is a rolling average, calculated over the last 24 hours. This means that if I[T] is the actual index value at time T, then the index value recorded in the data file is

X[T] = (I[T] + I[T-1] + ... + I[T-23]) / 24

This implies that if the index changes from X[T] at time T to X[T+1] at time T+1, then 24 \* (X[T+1] - X[T]) = I[T+1] - I[T-23]. Also note that an AQI value (point value or average) cannot be negative, since a value of zero means the pollutant is absent from the air.

It may not be possible to calculate the maximum point value, because the record is not complete and because in general averaging is not a reversible operation. However, even if we cannot obtain an exact answer, we would like to obtain some approximation, for example in the form of an interval in which we think the true highest reading lies.

Print the highest AQI value found in the data file, the date and time at which this value was recorded, and your estimate of the actual index value at that time. For example:

```

analysing data from file aqi\_data\_civic.csv

Question 1:

...

Question 2:

...

Question 3:

max AQI PM2.5 of 3839 found at 2020-01-01 19:00:00

estimated actual reading at this time: min = 0, max = 92136

```

(Note that the estimate range in this example is a very poor one.)

For this question, it is important that you document how you calculate your estimate, and any assumptions that you have made. Use comments and docstrings (as appropriate) in your assignment code.

Restrictions on code

There are certain restrictions on your code:

* You should NOT use any global variables or code outside of functions, except for the test cases in the if \_\_name\_\_ == '\_\_main\_\_' section. You can add other test cases in the main section if you wish.
* You can import modules that you find useful. However, we will test your code using the Anaconda distribution of python, so only modules available in Anaconda can be used. If in doubt about whether a module can be used, post a question to the [wattle discussion forum](https://wattlecourses.anu.edu.au/mod/forum/view.php?id=1959192) before you decide to use it.
* The argument to the analyse function is the path to the file to analyse. You must only read this file. You must NOT write any file, or use any of the functions of the os module (such as changing the working directory).