13.8 Climate

Many of you will need to analyze data at some point. The final project is based on that idea.

In this assignment, you will write all the code to analyze some data and produce reports. The main program is provided and cannot be changed. You must write one module, **temperature.py**, to read the data and another, **report.py**, to analyze/report the data.

Grading

Grading is based on the 7 test cases defined below and is combined for a total of 100 points. You can only see the output from the first test case when you submit. You must compute the results from the input data in your program. You cannot simply print strings containing values you obtained by other means since we will be using a different file for testing.

10 points - Correctly output average data for the total period.

10 points - Correctly output average data for a given year.

10 points - Correctly output average data for a given month.

10 points - Correctly output average data for a given day.

20 points - Correctly output data table for the total period.

20 points - Correctly output data table for a given year.

20 points - Correctly output data table for a given month.

Assignment

What is the weather like in Fort Collins? Let's take a look at some historical data from a local weather station and generate reports regarding the temperature for different periods of time. The program will support two different types of reports, average and table:

average will output the average of the data points in a given period.

table will output a collection of data points in a given period; so if given a year, it will output the data points of every month in the specified year.

We would like to be able to report temperature trends for the following time periods:

* Total
* Year
* Month
* Day

And for the time period given, the report will display the following data columns:

* Time period (or lack thereof)
* Average temperature
* Maximum temperature
* Minimum temperature

Here are the specific program requirements:

* The program should allow the user to request a report and an optional period on a single line of input.
* The program should support two different types of reports, average and table.
* The report period may be specified as a year, month, or day. When no period is specified, the report should include all the data.
* The period specification may take the form of a year, month-year, or day-month-year in the same format as the database. Be careful, the period 1-Jan-2015 should not match 11-Jan-2015, 21-Jan-2015, and 31-Jan-2015 depending on the implementation.
* You should print all dates in the report with the following format dd-MMM-yyyy where dd is a 2-digit day of month, MMM is a 3-letter abbreviation for the month and yyyy is a four-digit year.
* The program should end when the type of report is stop

The following example shows average reports for different time periods.

Notice that the data in the following examples is obviously incorrect and for illustration purposes only.

file? Temperatures.csv

file: Temperatures.csv

report? average

report: average

Period Avg Max Min

49.9 99.9 -9.9

report? average 2015

report: average 2015

Period Avg Max Min

2015 49.9 99.9 -9.9

report? average Jan-2015

report: average Jan-2015

Period Avg Max Min

Jan-2015 49.9 99.9 -9.9

report? average 01-Jan-2015

report: average 01-Jan-2015

Period Avg Max Min

01-Jan-2015 49.9 99.9 -9.9

report?

Sometimes we wish to report the average, minimum, and maximum temperatures for all of the components of a period. We will use the table report for this purpose. The specified period determines the range of values to appear in the report. If a period is not specified, the table should contain a line for each year in the data. If only a year is specified, the table should contain a line for each month in that year. If a year and month are specified, the table should contain a line for each day in that month. These periods should always appear in the calendar order.

report? table

report: table

Period Avg Max Min

2008 49.9 99.9 -9.9

2009 49.9 99.9 -9.9

2010 49.9 99.9 -9.9

2011 49.9 99.9 -9.9

2012 49.9 99.9 -9.9

2013 49.9 99.9 -9.9

2014 49.9 99.9 -9.9

2015 49.9 99.9 -9.9

report? table 2014

report: table 2014

Period Avg Max Min

Jan-2014 49.9 99.9 -9.9

Feb-2014 49.9 99.9 -9.9

Mar-2014 49.9 99.9 -9.9

Apr-2014 49.9 99.9 -9.9

May-2014 49.9 99.9 -9.9

Jun-2014 49.9 99.9 -9.9

Jul-2014 49.9 99.9 -9.9

Aug-2014 49.9 99.9 -9.9

Sep-2014 49.9 99.9 -9.9

Oct-2014 49.9 99.9 -9.9

Nov-2014 49.9 99.9 -9.9

Dec-2014 49.9 99.9 -9.9

report? table Jan-2014

report: table Jan-2014

Period Avg Max Min

01-Jan-2014 49.9 99.9 -9.9

02-Jan-2014 49.9 99.9 -9.9

03-Jan-2014 49.9 99.9 -9.9

04-Jan-2014 49.9 99.9 -9.9

05-Jan-2014 49.9 99.9 -9.9

06-Jan-2014 49.9 99.9 -9.9

07-Jan-2014 49.9 99.9 -9.9

08-Jan-2014 49.9 99.9 -9.9

09-Jan-2014 49.9 99.9 -9.9

10-Jan-2014 49.9 99.9 -9.9

11-Jan-2014 49.9 99.9 -9.9

12-Jan-2014 49.9 99.9 -9.9

13-Jan-2014 49.9 99.9 -9.9

14-Jan-2014 49.9 99.9 -9.9

15-Jan-2014 49.9 99.9 -9.9

16-Jan-2014 49.9 99.9 -9.9

17-Jan-2014 49.9 99.9 -9.9

18-Jan-2014 49.9 99.9 -9.9

19-Jan-2014 49.9 99.9 -9.9

20-Jan-2014 49.9 99.9 -9.9

21-Jan-2014 49.9 99.9 -9.9

22-Jan-2014 49.9 99.9 -9.9

23-Jan-2014 49.9 99.9 -9.9

24-Jan-2014 49.9 99.9 -9.9

25-Jan-2014 49.9 99.9 -9.9

26-Jan-2014 49.9 99.9 -9.9

27-Jan-2014 49.9 99.9 -9.9

28-Jan-2014 49.9 99.9 -9.9

29-Jan-2014 49.9 99.9 -9.9

30-Jan-2014 49.9 99.9 -9.9

31-Jan-2014 49.9 99.9 -9.9

report?

You should note when an invalid report name is given and exit the program when stop is entered as the report name.

report? avg

Unknown report: avg

report? stop

Data file

The **Temperatures.csv** file contains hourly temperature and wind speed samples in Fort Collins from January 1, 2008 to December 31, 2015.

Date,Time,Temperature,Wind

1-Jan-2008,0:00,20.6,4.6

1-Jan-2008,1:00,19.8,3.4

1-Jan-2008,2:00,18.4,2.9

1-Jan-2008,3:00,12.2,0.5

...

31-Dec-2015,20:00,17.9,2.9

31-Dec-2015,21:00,14.7,0.8

31-Dec-2015,22:00,11.9,1

31-Dec-2015,23:00,10.9,0.8

The following columns are contained in the comma-separated-values file (the file is very large so you should only read it once):

* **Date** The date is given in day-month-year format, with capitalized three-letter abbreviations for the month.
* **Time** The time is given in hour:minute format, with hours ranging from 0 to 23 in lieu of AM/PM.
* **Temperature** The temperature is a decimal value.
* **Wind** The wind is a decimal value

Ignore temperature or wind values of \*\*\*.

Programming

The assignment has three files:

* **main.py** is a script containing the **main** program. It is provided for you and you may not change it. There are calls to functions in the **temperature** and **report** modules that you must implement.
* **temperature.py** is a module that includes the readData function to gather the data in the filename argument. The readData function may, in turn, call other functions that you define in the module.
* **report.py** is a module that includes the average and table functions that implement the corresponding reports above. These functions may call other functions that you define in this module or the **temperature** module (remember to import it).

You *must* implement this function in **temperature.py**:

1. ***readData()*** - This function should parse your CSV file and read all the lines, saving them for later computations. Lines read by this function should be reused by the rest of the functions. A dictionary might be useful. In this way, you avoid reading the file multiple times for each of the other functions.

You *must* implement these functions in **report.py**:

1. ***average()*** - Should print a report in the average format shown above.
2. ***table()*** - Should print a report in the table format shown above.

This program will require you to work with modules, functions, CSV files, strings, lists, and dictionaries. Everything you should need is in zyBooks.