# ­

Scientific Programming in Python – Lab Assessment

**Instructions:**

1. Write a Python program to solve each of the tasks described below, the program should have a main function for Q1 and a separate main function for Q2 (call them whatever you wish, eg. main1 and main2).
2. Upload your solution python file (.py file) to Canvas before the end of the lab.
3. Once you have submitted your file **you should verify that you have correctly uploaded it**. It is your responsibility to make sure you upload the correct file.
4. Please make sure you **fully comment** your code.
5. Use comments to clearly indicate the question that each code segment is related to.
6. Please put your **student name** and **number** as comments at the top of your python file.
7. Save your file in format **Surname\_Idnumber\_Lab8060.py**
8. You **must also email your instructor**, via Canvas, this word document with your **name** and **date** filled in at the end of the academic integrity section (final page)

**Q1: Basic Python: Data Structures, Iteration, IO, etc**

**For this question, you must not use NumPy or Pandas, so standard file reading as covered in week 7**.

Write a function called *Q1ParseFile* that:

1. Takes as **input** the name of a file
2. **Opens that file**, which will be a comma separated file. You will need to ***rstrip(“\n”)*** the line to remove “\n” and then **split** the line based on ‘,’.
3. **Creates a dictionary** as follow: for all lines in the file with **at least two entries**, the **key** is the **second** item of the line and the **value** is the **first** item (you do not need to worry about duplicate keys, there aren’t any).
4. **Returns** the created dictionary

**[30 Marks]**

You should only consider lines with at least two words (a number of lines only have one word).

In the main function, you should call this function for file “locations.txt” (found in the lab assessment folder). After the call to this function, in the main function the user must then be asked for a location and the country returned if that location is a key in the dictionary, otherwise the user should be informed that it is not in the dictionary. For example (where the user entered *Phoenix* when prompted):



**[10 Marks]**

**Q2: Pandas**

In the lab assessment folder you will find a dataset called importsAuto.csv containing information about second hand cars. It contains in total 24 features (columns) describing each car such as the make, number of doors, engine-size, price, etc. The following is a full list of the features (please note that the questions that follow will focus on a subset of these features).

1. **make**: alfa-romero, audi, bmw, chevrolet, dodge, honda,

isuzu, jaguar, mazda, mercedes-benz, mercury,

mitsubishi, nissan, peugot, plymouth, porsche,

renault, saab, subaru, toyota, volkswagen, volvo

2. **fuel-type**: diesel, gas.

3. **aspiration**: std, turbo.

4. **num-of-doors**: four, two.

5. **body-style**: hardtop, wagon, sedan, hatchback, convertible.

6. **drive-wheels**: 4wd, fwd, rwd.

7. **engine-location**: front, rear.

8. **wheel-base**: continuous from 86.6 120.9.

9. **length**: continuous from 141.1 to 208.1.

10. **width**: continuous from 60.3 to 72.3.

11. **height**: continuous from 47.8 to 59.8.

12. **curb-weight**: continuous from 1488 to 4066.

13. **engine-type**: dohc, dohcv, l, ohc, ohcf, ohcv, rotor.

14. **num-of-cylinders**: eight, five, four, six, three, twelve, two.

15. **engine-size**: continuous from 61 to 326.

16. **fuel-system**: 1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi.

17. **bore**: continuous from 2.54 to 3.94.

18. **stroke**: continuous from 2.07 to 4.17.

19. **compression-ratio**: continuous from 7 to 23.

20. **horsepower**: continuous from 48 to 288.

21. **peak-rpm**: continuous from 4150 to 6600.

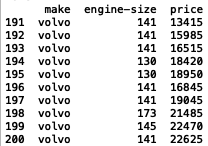
22. **city-mpg**: continuous from 13 to 49.

23. **highway-mpg**: continuous from 16 to 54.

24. **price**: continuous from 5118 to 45400.

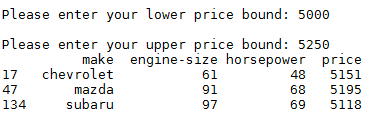
Open the file importsAuto.csv file using Pandas read\_csv function and write Python code to complete the following tasks. Please write a separate function for each of the tasks below. **Each function should take in a DataFrame (containing the car data) as an argument**.

1. Write a function that prints out the last 10 rows for three columns of data: the **make**, **engine-size** and **price** of each car in the dataset. Sample output is shown below



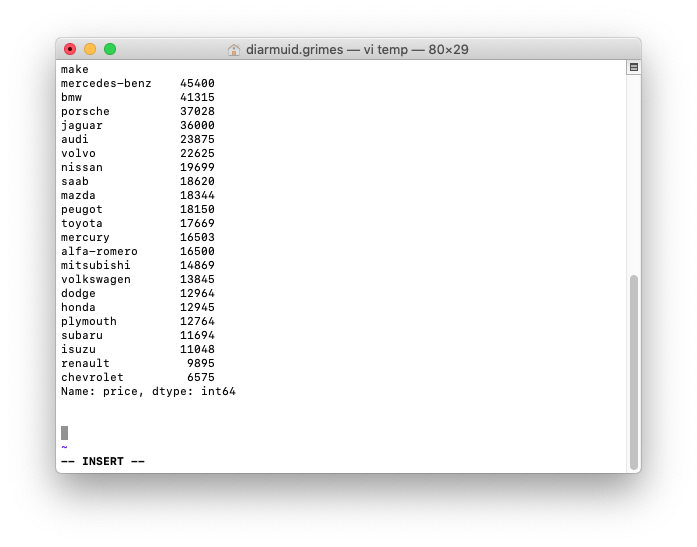
**[15 Marks]**

1. Write a function that will ask the user to specify an upper and lower price value and it should print out the **make, engine-size, horsepower and price** of all cars that meet that specification. In the example below the user specifies a lower price bound of 5000 and an upper price of 5250. The program then outputs the relevant details of all cars that satisfy these constraints.



**[20 Marks]**

1. Write a function that will calculate the **maximum** selling price for each car **make**. Your code should print out a line for each make with its name and the maximum price over all cars of that make. The output should be **sorted** in **descending order** according to the maximum price. The following is the output that your code should produce:



**[25 Marks]**

**ACADEMIC INTEGRITY:** Student Examination/Assessment Declaration

I declare that the work submitted is my own. I declare that I have not obtained unfair assistance via use of the internet or a third party in the completion of this examination/assessment. I acknowledge that the Academic Department reserves the right to request me to present for oral examination as part of the assessment regime for this module. I confirm that I have read and understood the policy and procedures concerning academic honesty, plagiarism and infringements. I understand that where breaches of this declaration are detected, these will be reviewed under CIT’s policy and procedures concerning academic honesty, plagiarism and infringements, as well as any other Institute regulations and policies which may apply to the case. I also understand that any breach of academic honesty is a serious issue and may incur penalties.

**NAME**:    Bernard McNamee

**DATE**:      7/12/’20