**Python Training Task**

**General Instruction:**

* (Time: 2 days)
* Use Proper Variables, Methods, Files and Flags name
* Use Proper Print statement
* Try to Handle all corner cases
* Create a GIT repository for these tasks and follow good GIT practices

**1:** Write a python script that finds numbers in range (2000-3600) that are divisible by 7 and not divisible by 5 and print a list of numbers on a single line in comma separated format.

**2:** Write a python script that display the following parameters:

* Byte order
* Core
* Model name
* CPU max Frequency
* CPU min frequency
* Virtualization
* L1i cache size
* L1d cache size
* L2 cache size
* L3 cache size
* Thread(s) per core
* Distributor ID
* Distributor Description
* Distributor codename

**3:** Write a Python script that reads from a Json File ([json file](https://drive.google.com/file/d/1lqPAf0WeKP8KYkALMUm7NTwqCOZisOm7/view?usp=share_link)) containing data about a group of people. The JSON file should contain a list of dictionaries, with each dictionary representing a person and their attributes (e.g. name, age, occupation). The script should then create a new dictionary that groups the people by their occupation. The keys of the new dictionary should be the occupation names, and the values should be lists of dictionaries representing the people in that occupation. Finally, the script should write the new dictionary to a new JSON file.

**Example Input:** [ { "name": "Alice", "age": 25, "occupation": "teacher" }, { "name": "Bob", "age": 30, "occupation": "programmer" }, { "name": "Charlie", "age": 35, "occupation": "teacher" }]

**Example OUTPUT:**

{ "teacher": [ { "name": "Alice", "age": 25, "occupation": "teacher" }, { "name": "Charlie", "age": 35, "occupation": "teacher" } ], "programmer": [ { "name": "Bob", "age": 30, "occupation": "programmer" } ]}

**4:** Write a python script that takes two coordinates (x1,y1) , (x2, y2) as command line arguments and calculates distance between these points. (use python module math, argparser)

**5:** Write a python script that perform following steps:

* Take Path to a file (***/home/user/Desktop/my\_dummy\_file.tx*t**) and make sure this file exists in the directory if not create it.
* Add some text in file and run this bash command from python script `***chmod a=- my\_dummy\_file.txt***`
* Try to write/read to that file, identify the issues in read/write and resolve these issue using another python script.

**6:** Create a dummy text file and add the following line to it: ***1,2,3,4,5,6,7,8,9,10.*** write a python script that perform the following task:

* Use **Virtual Environment** for this task
* Read the file and convert the comma separated text into a python list
* Create another python file with two method/function that calculate the square and cube of that list and return square and cube list
* Import that python module in first script and call these function and plot graph between square and cube lists as x and y axis respectively using **matplotlib**

**7:** write a python script that perform the following tasks:

* Write a method name **find\_divisibles** that finds the total divisible number in range from **1-range** by a **divisor**. Example: (20,3) find numbers divisible by 3 in range 1-20 which is [3, 6, 9, 12, 15, 18].
* Write a python script that call this method **synchronously** and run for all these ranges
  + (50800000, 34113)
  + (100052,3210)
  + (20000,5)
* Write another python script that call this method **asynchronously** and create three different task for each pair: ( There should be context switching after each division)
  + (50800000, 34113)
  + (100052,3210)
  + (20000,5)

Also do logging and calculate total time take by the script to complete its task

**8:** write a python script that takes path to a file ([text file](https://drive.google.com/file/d/1sKsyNlR29Cn9N8P1uaAADAttpoDgs3D7/view?usp=sharing)) as a command line argument and and calculate the following parameters using **numpy** module:

* Mean
* Median
* Max
* Min
* Standard Deviation
* 99th Percentile
* 99.9th Percentile
* 99.99th Percentile
* 99.999th Percentile

**9:** Write a python based client server application using websockets with following features:

* **Client side**: Read a json data file ([json file](https://drive.google.com/file/d/1kW9CaQ0dZHJQR_QFuO0WmaQ7xG7ezyOF/view?usp=sharing)), send messages from client to server one by one using json data format. Make sure you connect with the server using websocket before sending a message.
* **Server side:** Server will open a websocket connection and wait to receive the query/message sent by client and append current timestamp in the received message and send response message back to client.
* Client will also Display the response received and time taken by websocket from send message to response receive. Also store all the messages received in a file. (Do logging on both side)