Aurora Python Interview.

Please Answer 2 Questions:

1. Output Validator

a. Create Python class.

b. Use the provided main.c file

c. Class should implement the following APIs:

i. **Replace** __VALUE__ in the provided C file with some value.

ii. Compile the **modified file** with gcc and create an executable.

iii. **Run the executable** and get the returned value of the executable.

iv. Check that the return value is the same as you provided

d. Each API should handle errors (wrong input and etc..)

e. Add python tests to verify the Class APIs work as expected

f. Add main and explain how to run it

Example: replace __VALUE with 123. The return value of the executable should be 123.

Notes:

- You can Use Linux or Windows to solve the Question
  - Use gcc on Windows you have 2 options:
    - Use VM with Linux for this question
    - Use mingw or cygwin -> https://gcc.gnu.org/install/binaries.html
- The replace __ VALUE __ should come from the user – from the python script

2. Top process parser

a. Use the provided top_linux.csv file

b. **Create Python class** with the following **SW APIs**.

i. Dump file to json

ii. Print all users

iii. Print all commands of user (provide username)

iv. Get command name by pid

v. Print top 5 commands sorted by MEM usage

c. **For each API provides the complexity**

d. Each API should handle errors (wrong input and etc..)

e. Add 5 python tests to verify the Class APIs work as expected

f. Add main and explain how to run it

Notes:

- You can use the csv or the json for the rest of the question
- The top_linux.csv is just an example, the values in the csv can be different – the APIs should not look in the data
- YOU SHOULD NOT RAISE A WEB SERVER!
- Please sort the MEM section regardless of the values inside – the given values are just an example.
- Each API should do the action is needed, regardless of the input.

Tips and hits:

1. We love simple and clean code.
2. We love efficient code, please keep on code reuse, and object-oriented code style.
3. Part of the exam is to understand the questions, you can use google or any resource you want
4. You can use any package you want (pandas / Pytest / doctest …)
5. All the function should be in class
6. **Please ZIP the files and send them back to us**
7. in the zip file you should provide **only 2 python files**:
    a. Output Validator.py
    b. Top process parser.py