CIS 313, Intermediate Data Structures
Fall 2019

CIS 313 Lab 0

Due: October, 18th, 2019 at 11:55pm

This lab is an introduction into lab procedure and a review of Python

Overview

Write a program that reads a series of pairs of integers (X, Y) and prints pairs GCD(X, Y) and
LCM(X, Y) where:

· GCD(X, Y) is the Greatest Common Divisor of X and Y

· LCM(X, Y) is the Least Common Multiple of X and Y

The purpose of this assignment is:

- An overview of using unittest module in python to test your code.
- Learn how to submit your work using Gradescope.
- Learn how to write test cases in python.
- Learn how to use exceptions in python.

You should write your program in Python 3 only. You may not import any math libraries to compute
LCM or GCD for you.

Input Description

You need to implement the LCM and GCD methods in the class **mathOps**. You need to write as many test
cases as required to test your code. For example, listed below are some possible inputs to be used in the test
cases. A few test cases are given to you in the file **test_lab0.py**. Please make sure you test your program
runs properly for all the possible edge cases. Eg: testing your code with malicious inputs like two strings or
floating point numbers instead of integers, using extremely large numbers like 2^{100} and 2^{99} etc. You should
make sure the program ends gracefully in these situations throwing proper exceptions and does not crash.

You also need to consider the time complexity while implementing the methods.
Running time of Euclidean algorithm for Gcd is O(log(n)).
Running time of LCM is O (1) – constant time.

Possible set of input values:
```
29 31
0 45
45 0
inf 0
```

Testing:

Listed below are output values for the above mentioned input values. These test cases are just a sample and it is highly recommended that you write as many additional test cases as possible to make sure that your code is robust. You methods pass all the test cases given in Gradescope to get full credit.

GCD OUTPUT:
1
45
45
should raise an exception: one of the values is infinity

LCM OUTPUT:
899
should raise an exception: lcm of 0 and 45 is undefined
should raise an exception: lcm of 45 and 0 is undefined
should raise an exception: one of the values is infinity

Files to submit:

You should submit the **lab0.py** with the filled in methods. You should also submit the **test_lab0.py** file with all the test cases that you had written to test your code. These two files should be compressed and uploaded in .zip format under the filename **"submission.zip"**. Please make sure you select the **lab0.py** and **test_lab0.py** files and compress them. Do not compress the folder named submission directly which will result in another subfolder named submission being created after converting to zip format. The screenshot is an example link on how to do achieve this. Once you submit your files to Gradescope, you will get the results and the score for your assignment. If any of the test cases fail, you need to modify your code and resubmit until all the test cases pass in order to get full credit.
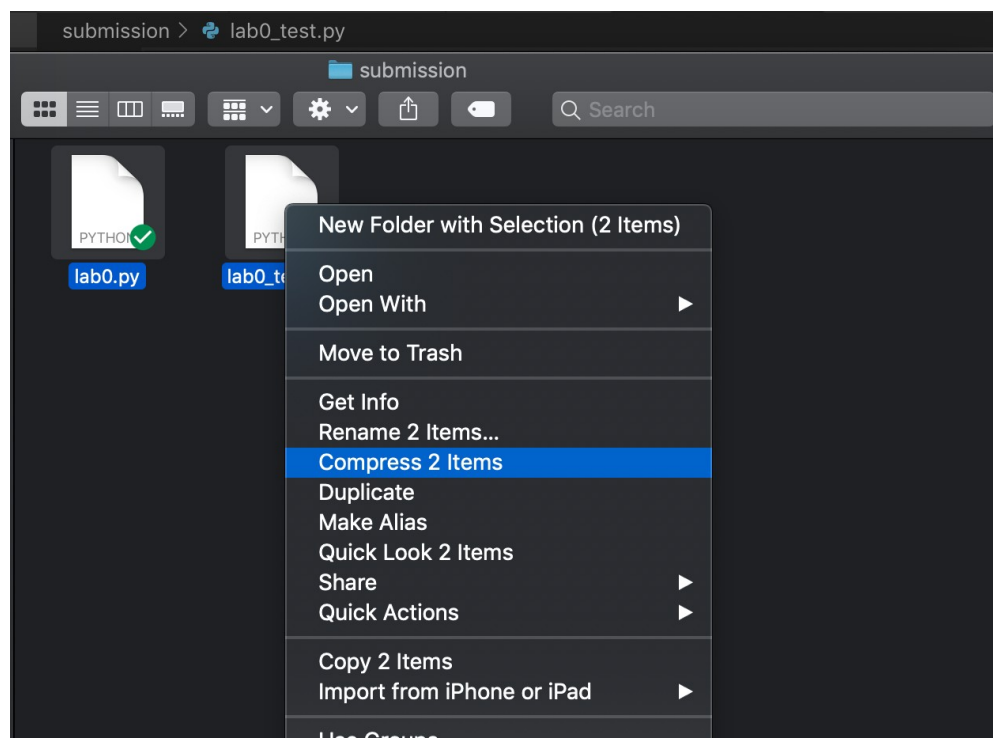
Grading

Typically, half of your grade will be determined by attempting to implement the correct data structure, and half will be on correctness and style (style will be discussed in labs). This assignment only has the correctness component. There will be 20 points possible for this assignment. Since this is the first assignment, I will try to provide as much assistance as possible during the labs and office hours to make sure you get full credit and all your doubts are clarified regarding submission and writing test cases.

Very important things to keep in mind:
Facts about LCM and GCD and things that I have assumed:
- Gcd of 0 and 0 is 0
- Gcd of 0 and any other number (say 45) is the number (45).
- The absolute value for negative numbers (|-6| = 6) should be used for calculating Lcm or Gcd.
- Lcm of 0 and any other number is undefined (should raise an exception)
- Lcm and Gcd of infinity and any other number is undefined (should raise an exception)
- If floating point numbers are used, the ceiling value is used. (eg: 39.7 is rounded to 40)

Your code will be tested against the following invalid values:
- strings
- infinity
- very large numbers
- zero and negative numbers

So it is of paramount importance that you are aware of raising exceptions and using the try and the except block in python. You **should be able to raise some inbuilt exceptions like OverflowError, TypeError, ValueError etc in python**. You **need not** write your own exceptions for this assignment. This might seem challenging initially but we are there to provide as much help as possible. And we will do our best to ensure that you score full 20 points in this sample assignment. Please get as much assistance during the office hours as possible. Good luck!!

Note:
Passing the test cases carries some points. And some points will be allocated for style. If you use proper variable names and your code is comfortable to read, you will probably get full points for style. Only in rare circumstances will you lose points for style.