

## CIS 313 Lab 0

**Due: Oct 4th, 2018 at 4:55pm**

**This lab is an introduction into lab procedure and acts as a review of Java**

### Overview

For this assignment, you are to write a program which will read a series of pairs of integers  $X$  and  $Y$  and print pairs  $\text{GCD}(X, Y)$  and  $\text{LCM}(X, Y)$ , where

- $\text{GCD}(X, Y)$ : the greatest common divisor of  $X$  and  $Y$ ,
- $\text{LCM}(X, Y)$ : the least common multiple of  $X$  and  $Y$ .

The purpose of this assignment is two-fold

- Practice using standard input and output.
- Learn how to submit your work.

You should write your program in Java only.

### Input Description

The input will be a text file, for example *inSample.txt* below will be provided. The first line will contain an integer  $N$ , which is the number of lines to follow. Each of the  $N$  lines contains two integers  $X$  and  $Y$ , separated by a space.

```
5
4 2
8 9
143 247
1 1
10 10
```

**Note:** When using an editor, you may also manually type in input to the command window. However, you will be tested with a file similar to *inSample.txt*.

## Output Description

For each of the pairs  $X\ Y$  output  $\text{GCD}(X,Y)$  and  $\text{LCM}(X,Y)$  on a line, separated by a single space. For example, using the sample input above, your program should output:

```
2 4
1 72
13 2717
1 1
10 10
```

## Testing Protocol

We will test your program by running your program at the command line. You will need to use **STANDARD INPUT**. Do **not** pass in the name of the file as an argument - do **not** encode the name of your input file in your program. We will run your program on several different test files.

Examples of the commands we will use to test your program look like the following. Here *lab0.java* is a name of a turned-in program and *inSample.txt* is the sample test file. At the command line we might say something like

```
java lab0 < inSample.txt
```

We could also have used the unix *pipe* command:

```
cat inSample.txt | java lab0
```

## Testing Scripts

You will also be provided with execute only testing scripts (i.e. test.sh, compile.sh, etc..). You will not have permissions to view these files, but you are encouraged to use them. They are **very** similar to how you will be graded, so it is your responsibility to understand why your program fails the test if it does so. In other words, your grade will be **very** similar to the grade the test script outputs for you.

To use the test scripts, you need to gain access to the @ix-dev.cs.uoregon.edu server.

1. Set up your ix-dev account with Cheri in the front office of the CIS building.
2. Once you have access, you need to log into ix-dev. From the command line run:  
`ssh yourUserName@ix-dev.cs.uoregon.edu`
3. Navigate to your workspace: (if you do not have these folders, use command mkdir to create them)  
`cd Documents/workspace`

4. Create a cs313 folder by running:  
`mkdir cs313`
5. Navigate into cs313 and create another folder lab0  
**note:** to find out the path to this folder navigate into the newly created lab0 folder and run:  
`pwd`
6. exit out of ssh by running the command "exit"
7. Now we need to copy over your .java files to your ix-dev account. While in the folder containing your .java files for this project on your personal machine run:  
`scp *.java yourUserName@ix-dev.cs.uoregon.edu:~/Documents/workspace/cs313/lab0`
8. ssh back into yourUserName@ix-dev.cs.uoregon.edu
9. Run the test script for lab0:  
`/home/users/smergend/public/cs313/lab0/test.sh path/to/your/lab0/files`

## Submission

You are provided with skeleton code (lab0.java) that you may use, which should streamline the assignment. Submit **only** your .java file(s) via Canvas. Your main method must be in a java file named lab0.java. Remember to remove any package statements (note that this is different than import statements).

## Grading

Typically, half of your grade will be determined by attempting to implement the correct data structure, and half will be on correctness. For the assignment, there is only the correctness component. There will be 50 total points possible for this assignment. Two points are assigned for each of the 25 test cases for this assignment. 5 of the test cases are given to you above, and 20 are hidden.