

CSCI 1120 (Low-Level Computing), Fall 2016

Homework 3

Credit: 10 points.

1 Reading

Be sure you have read the assigned readings for classes through 9/7.

2 Honor Code Statement

Please include with each part of the assignment the Honor Code pledge or just the word “pledged”, plus one or more of the following about collaboration and help (as many as apply).¹ Text *in italics* is explanatory or something for you to fill in. For written assignments, it should go right after your name and the assignment number; for programming assignments, it should go in comments at the start of your program.

- This assignment is entirely my own work.
- This assignment is entirely my own work, except for portions I got from the assignment itself (*some programming assignments include “starter code”*) or sample programs for the course (*from which you can borrow freely — that’s what they’re for*).
- I worked with *names of other students* on this assignment.
- I got help with this assignment from *source of help — ACM tutoring, another student in the course, the instructor, etc.*
- I got significant help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc.. (“Significant” here means more than just a little assistance with tools — you don’t need to tell me that you looked up an error message on the Web, but if you found an algorithm or a code sketch, tell me about that.)*
- I provided significant help to *names of students* on this assignment. (*“Significant” here means more than just a little assistance with tools — you don’t need to tell me about helping other students decipher compiler error messages, but beyond that, do tell me.*)

3 Programming Problems

Do the following programming problems. You will end up with at least one code file per problem. Submit your program source (and any other needed files) by sending mail to `bmassing@cs.trinity.edu` with each file as an attachment. Please use a subject line that mentions the course and the assignment (e.g., “csci 1120 hw 3” or “LL hw 3”). You can develop your programs on any system that provides the needed functionality, but I will test them on one of the department’s Linux machines, so you should probably make sure they work in that environment before turning them in.

¹Credit where credit is due: I based the wording of this list on a posting to a SIGCSE mailing list. SIGCSE is the ACM’s Special Interest Group on CS Education.

1. (5 points) Write a C program that asks the user for three integers and prints them in order from smallest to largest, or an error message if what was entered is something other than three integers. (You may recognize this problem as a special case of one you probably spent some time on in CS1. Don't try to solve that problem in general for this assignment; just solve the "three numbers" problem using what we've discussed in class so far.) *Clarification:* Notice that the numbers do not have to be distinct — for example, the user could enter three values of 0.
2. (5 points) Write a C program that asks the user for two non-negative integers, call them a and b , not both zero, and computes and prints $\text{gcd}(a, b)$, the greatest common divisor of a and b , using a recursive version of Euclid's algorithm. Print an error message if what was entered is not two integers, or either input is negative, or both are zero.

Euclid's algorithm can be described recursively thus: For non-negative integers a and b , not both zero, with $a \geq b$,

$$\text{gcd}(a, b) = \begin{cases} a & \text{if } b = 0 \\ \text{gcd}(b, a \bmod b) & \text{otherwise} \end{cases}$$

where $a \bmod b$ is the remainder when a is divided by b .