CSCI 1120 (Low-Level Computing), Spring 2020
Homework 2

Credit: 5 points.

1 Reading

Be sure you have read, or at least skimmed, the assigned readings for classes through 1/22, including the video lectures.

2 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 my TMail address (or you can use `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 2" or "LL hw 2"). 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.

1. (5 points) Write a C program to convert seconds into years, days, hours, minutes, and seconds. Your program should prompt the user for a number of seconds, get the number entered, and print the equivalent number of years, days, etc. (e.g., 100 seconds is 0 years, 0 days, 0 hours, 1 minute, and 40 seconds, while 100000000 seconds is 3 years, 62 days, 9 hours, 46 minutes, and 40 seconds). Assume 365 days in a year (not quite right but makes the calculations simpler). *For this assignment only*, you do not need to do any kind of checking that what the user enters is actually an integer and non-negative, since we haven't yet talked about conditional execution. Just assume it is and do the required calculations.

   Hints:

   - Probably the best way to do the required calculations is with the integer-division (`/`) and remainder (`%`) operators.
   - Be advised that a C-idiomatic way to define constants is with `#define`, e.g.,

     ```
     #define SECONDS_PER_MINUTE 60
     ```

     usually before the first function that uses them. This can help make code more human-readable.

3 Honor Code Statement

Include the Honor Code pledge or just the word "pledged", plus *at least one 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 programming assignments, this should go in the body of the e-mail or in a plain-text file `honor-code.txt` (no word-processor files please).

[1] Credit 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.

- This assignment is entirely my own work. *(Here, "entirely my own work" means that it's your own work except for anything you got from the assignment itself — some programming assignments include "starter code", for example — or from the course Web site. In particular, for programming assignments you can copy freely from anything on the "sample programs page".)*
- 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. (Here, "help" means significant help, beyond a little assistance with tools or compiler errors.)*
- I got help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc.. (Here too, you only need to mention significant help — 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 help to *names of students* on this assignment. *(And here too, you only need to tell me about significant help.)*

4 Essay

Include a brief essay (a sentence or two is fine, though you can write as much as you like) telling me what about the assignment you found interesting, difficult, or otherwise noteworthy. For programming assignments, it should go in the body of the e-mail or in a plain-text file `essay.txt` (no word-processor files please).