CSCI 1120 (Low-Level Computing), Fall 2012

Homework 1

Credit: 10 points.

1   Reading

Be sure you have read the assigned readings for classes through 9/12.

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 `bmassing@cs.trinity.edu`, with each file as an attachment. Please use a subject line that mentions the course number and the assignment (e.g., "csci 1120 homework 1"). 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 first prompt the user for a number of seconds, and then do one of two things: If what was entered was a valid non-negative integer, it should print the equivalent number of years, days, etc. (e.g., 100 seconds is 0 years, 0 days, 0 hours, 1 minute, and 40 seconds). Otherwise it should print an error message and exit.

   Hint: Probably the best way to do the required calculations is with integer-division (`/`) and remainder (`%`) operators.

2. (5 points)   Write a C program to compute U.S. income tax given taxable income. The instructions for Form 1040 for 2011 are as follows (for single taxpayers):

   - If taxable income is not more than $100,000,
     tax is to be looked up in the tax table.

   - If taxable income is more than $100,000 but not more than $174,400,
     tax is 28% of taxable income, minus $6,383.

   - If taxable income is more than $174,400 but not more than $379,150,
     tax is 33% of taxable income, minus $15,103.

   - If taxable income is more than $379,150,
     tax is 35% of taxable income, minus $22686.

   Your program should first prompt the user for a taxable income (in whole dollars), and then do one of three things: If what was entered was not a valid positive integer, it should print an error message and exit. If the amount is no more than $100,000, it should tell the user to use the IRS's tax table; otherwise it should use the above formulas to compute and print the tax due, rounded to the nearest dollar (round 50 cents upward). For example, the tax

on \$110,000 is 28% of \$110,000 minus \$6,383; computing and rounding, this gives \$24,417. (Don't worry about printing amounts with commas; for this assignment it's fine to just print, for the example, \$24417.)

You could use floating-point numbers for the calculation, but the result will likely be more accurate if you instead do everything with integers, calculating tax in pennies and converting back to dollars at the end. (An easy way to round cents to the nearest dollar is to add 50 cents and then use integer division to convert to dollars.)