EE1301: Introduction to Computing Systems
Fall, 2018
Department of Electrical and Computer Engineering
University of Minnesota

Homework #1

Instructions (follow these instructions for every homework assignment):

1. This is an individual assignment. You may discuss the problems with other students, with the TAs, and with the instructor. However, you must write you own solutions, including writing and debugging all of the programs yourself. What you turn in for grading must be the result of your own individual efforts. Submitting solutions that are identical or nearly identical to someone else is considered academic dishonesty and will result in a grade of F. You are encouraged to bring questions about the homework to your lab sessions.

2. Put each problem into a separate source code file named, "problemX.cpp", where X is the problem number, such as "problem1.cpp". Submit the source code for each problem on moodle.

3. You must ensure that your programs compile and execute on the CSE Labs computers running Linux. You can develop your programs on any computer you choose, but they must execute correctly on the CSE Labs computers to receive a grade.

4. You must follow the coding style guide posted on the class web site so that your source code is easy to read and well documented. Good coding style will ensure that the graders can understand your code, and it will give you important practice in documenting your code. You will be graded on your use of proper coding style.

5. Use the debugger, gdb, to help debug your code. You will be glad you did.

Problem 1.

Write a C/C++ program that prompts the user to input a floating-point value x, where x is a positive distance in miles. Your program should convert this distance to kilometers, meters, furlongs, feet, and inches, and then print each value.

Problem 2.

Write a program to compute sales tax on a purchase. Your program should prompt the user to enter a value in dollars and cents and the sales tax in percent. Print the amount of the tax and the total amount paid. Here is an example interaction with your program. The underlined items are the user's input. Note: you can print more than two digits after the decimal point.

```
Enter the amount of the purchase:  96.83
Enter the tax rate:  8.625
The amount of tax you paid was:  $8.35
The total amount you paid was:  $105.18
```

Problem 3.

Write a program that prompts the user to enter the length of two sides of a right triangle. Then print the length of the third size using the Pythagorean Theorem. You can use only the basic operators, +, -, *, and /, and the library function `sqrt()`. Note that you must include the line `#include <math.h>` at the top of your source file to be able to use this function. All of your variables should be declared as `double`.

Problem 4.

Write a C program that repeatedly prompts the user to input an integer value n, where n >= 1. Then compute and print the value of the following function. Hint: use a for loop to count through the sequence of values. Then enclose that loop within the body of a while loop that repeatedly prompts the user to enter n, computes the value of f(n), and prints the result.

$$f(n) = n! = 1 * 2 * 3 * ... * n.$$

Problem 5.

Write a program that prompts the user to enter a positive integer. Continue requesting values until the user enters -1. You must check that each input is valid (i.e., it is a positive integer). If it is not, ask the user to re-enter the value. Then print the number of values entered, the average of all of the values, and the minimum and maximum values. Note that you do not need to store every value entered by the user. You need to save only a running sum of the values, the number of values, and the running min and max values.