CSCI 1120 (Low-Level Computing), Spring 2011

Homework 3

Credit: 20 points.

1 Reading

Be sure you have read the assigned readings for classes through 3/07.

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 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.

1. (10 points) The C standard library has several useful functions for operating on characters, for example `isalpha` to determine whether a character is alphabetic and `isdigit` to determine whether a character is a digit. Write a C program that uses these functions to examine a character string and report its total length and how many alphabetic characters and digits it contains. (For example, "hello 12" has five alphabetic characters, two digits, and eight characters in all.)

   As with previous programs, make this program self-contained — i.e., don't bother to try to get character-string input from the user (it's trickier than it might seem). Instead put the code to do the counting in a function and call it with several different inputs from the main program. (Sample program `string-length.c`[1], not shown in class, gives an example of doing something similar — defines a `string_length` function to compute the length of a character string, plus a `test_string_length` function to make it more convenient to demonstrate how the `string_length` function works.)

   To get full credit your program must do the actual counting and the printing of the results in different functions (this is to give you practice using pointer parameters).

2. (10 points) Write a C program that gets a sequence of integers from the user (ending with anything that's not an integer) and then prints:

   - The smallest number entered.
   - The largest number entered.
   - The average of all the numbers entered (sum of all numbers divided by how many there are).

[1] `http://www.cs.trinity.edu/~bmassing/Classes/CS1120_2011spring/SamplePrograms/Programs/string-length.c`

Here is a sample execution of the program (user input shown in italics):

```
enter some integers, anything non-numeric to end
20
40
-4
4
ruru

minimum = -4
maximum = 40
average = 15.000000
```

Your program should do something reasonable if no numbers are entered (e.g., print "no numbers entered"). It also should work for any number of inputs (so you probably should *not* try to use an array to store the input).

You may (or may not) find it helpful to use constants `INT_MIN` and `INT_MAX` (the smallest and largest `ints`), defined in `limits.h`.