

CS 211 Lab 1 – Summer 2023

Logistics

Before starting this lab, review the Quick Start Guide provided on the course website to get familiar with the remote server *bertvm* and *SecureCRT* and *SecureFX*. You should be able to compile and run a program on *bertvm*. This lab guide assumes you are familiar with the edit/compile/run process and can use a Linux environment.

Special Notice Concerning Summer 2023 Lab Work: During an ordinary term, attendance at lab sessions is required, and all lab exercises must be completed during lab time. That is still the preferred model, and students are expected to do their lab work during the regular lab time if at all possible.

However that may not always be possible for all students in all lab sessions of Summer 2023. Therefore as an alternative, students may do the lab work on their own, up until midnight on the day on which the lab was assigned. If you must use this option, you are on your honor not to spend more than 90 to 120 minutes on the lab work. (In other words, this alternative provides a shifted time, but not additional time.) Note that this is expected to be a **RARE** occurrence, not a normal alternative to regular lab attendance.

All lab work for CS 211 Summer 2023 is to be submitted via Gradescope. (<http://www.gradescope.com>) Each lab assignment is worth 2 points towards a student's overall semester score, regardless of the scale on which it is graded. Lab grades are based primarily on participation and effort. Perfect results are not needed.

Important Notice: All assignments will be tested and graded on *bertvm*. If a program does not compile or run on *bertvm*, then points will be taken off. Make sure all assignments are completed (or at least tested) on *bertvm* before submission. This holds for both lab and project assignments.

Overview

This lab is an introduction/review on the basics of C language and the use of pointers for accessing simple variables and passing values to and from functions. All related materials are posted on the course website, under Lab Assignments → Lab 1.

To complete this lab, you should have access to the following:

- lab1a.c – Compile and run, and answer questions on the lab exercise.
- Lab 1 Exercise – Answer questions directly online via Gradescope.
- lab1b.c – Edit this program to make it functional and clean.

To receive full credit for this lab you must submit the following on Gradescope:

- Lab 1 Exercise - Answer questions directly online via Gradescope.
- lab1b.c edited to be functional, with good programming style.

Part 1

Download `lab1a.c` from the course web site to your local computer, and then use SecureFX to transfer the file to `bertvm`, in a directory of your choosing. Compile it using `gcc`, run the program, and then answer the questions in Lab 1 Exercise on Gradescope.

Part 2

Download the program `lab1b.c` from the Lab Exercises page and transfer it to `bertvm`. This program is supposed to read in a list of integers until the value of `-999` is input, and then print out the number of values read, the total of these values and the average of these values. Your task is to make it function properly, and to clean it up as necessary with suitable comments, etc.

1. Declare the proper variables for number of values (`count`), total, and average.
2. In the while loop, update your variables after reading a new value.
3. Calculate the average.
4. Display average, count, and total. The `printf` statements are already given in the base code.

To make sure your program is working correctly, check your program with the values given below. If your program output is not printing the expected output, then find and address the cause of it.

- 2 4 6 8 -999
- 2 3 5 -999
- -7 -10 -5 -9999 -999
- 0 0 0 -999
- 0 0 1 -999
- -999

You are to submit the completed `lab1b.c` for this lab via Gradescope. To help the TA, name your file with your net-id and the lab assignment, such as:

`netid-lab1b.c` → `jbelle-lab1b.c`

You should also modify file comments to reflect the new name.