EE 285 Lab 1

Learning to use your IDE and some C program basics

For each of the items a through e, show your completed work to one of your lab instructors. Each section is worth 10 points. The lab is worth a total of 50 points. The instructor will write your score for each completed section on this sheet. Turn in the completed sheet to the instructor before you leave the lab room.

First, install either xCode or Microsoft Visual Studio on your laptop, depending on which operating system you plan to use. It is best to do this before coming to lab, since downloading and installing may take an hour or more. Then, using your IDE, do the following exercises

a. Download the text file "lab_1a.c" from GT's GitHub.

   https://github.com/gtuttle/ee285_lab/blob/master/lab_1a.c

   (Copying and pasting into a blank .c text file in your IDE is fine.)

   There are a number of problems with this program. Some errors will prevent it from compiling properly. Other issues may not prevent the program from compiling, but they still deviate from standard practice in writing C programs or they may make the output look bad. Fix all of problems that you can, run the program, and demonstrate the output to one of the TAs.

b. Starting with a clean C text file, write program that outputs the text "Oh, we will fight, fight, fight for Iowa State…" using the printf() function.

   Comment: I realize that, when writing a new program, it is common to start with a program file that is known to work, and then simply edit it or paste in new pieces to make a new one. However, when learning a new programming language, I think that it can be useful to start from scratch and work from a new, clean text file. This helps develop some feeling for the structure of the program and what pieces are important. We won't always want to start from scratch, but for the first few programs, where the programs are short and simple, doing a little extra typing may be beneficial.

c. Starting from scratch, write a program that declares an integer variable named `barakObama`. Set the value of the variable to 44. Then use printf() to print out a statement that says "Barak Obama was the 44th president of the United States." where the number that is printed is the value of the integer variable that you defined. Check the operation by changing the value and running the program, again. (Feel free to use another president, if you prefer, but make sure to get the right number that goes with your choice of president.)

d. Starting from scratch, write a program that declares four integer variables, g, x, y, and z. Set the value of x to 7, the value of y to –4, and the value of z to 17. Then write an assignment expression to set g = x * y – z. Use printf to print the values of x, z, and z and the calculated value of g. Make the output more readable by including "x =" , "y =", etc in the output statement.

Run the program again, but add a second assignment line to set g = x * (y – z), and another printf() line to print this new result along with the initial result.

e. Starting from scratch, write a program that declares two integers, i and j. Set the initial values to i = 0 and j = 10. Use the printf() command to write the values to the console. Then increment i using i = i + 1 and decrement j using j = j – 1 and write the values using printf(). Repeat this pair of lines 9 more times in your program. (OK, now you can use copy and paste.) Run the program. The output should be a progression of printed lines with i counting up from 0 to 10 and j counting down from 10 to 0.