Lab 3: Introduction to Functions

Objectives:

- Introduction to functions
- Practice writing mod and integer division expressions
- Formatting output practice

Starting Point:

- *lab3.c*

Process:

Creating a New Folder

Create a new folder named *lab3* on your U: drive. **You will want to copy over ds4rd.exe to the *lab3* folder.**

Part 1

1. Download lab3.c. You will be modifying this file for this lab.
2. Find the DualShock 4 associated with your PC as in past labs.
3. Open the cygwin shell and change to your new *lab3* folder.
4. Try:
   ```
   ./ds4rd.exe -d 054c:05c4 -D DS4_BT -t -g
   ```
   You should see 4 columns of numbers. The first is time in **milliseconds** (since the ds4rd.exe program was started), the next three are accelerations in the x, y, and z axes, respectively. They should be separated by commas.
5. Hit Control-Break or Control-C to stop the program.
6. Compile lab3.c to lab3.exe
   ```
   gcc –o lab3.exe lab3.c
   ```
7. Try:
   ```
   ./ds4rd.exe -d 054c:05c4 -D DS4_BT -t -g | ./lab3.exe
   ```
8. Modify CODE SECTION 0 to meet the following criteria. You may create local variables to help you.
   - Your program outputs seconds, ax, ay, and az all on one line separated by commas.
   - Seconds (not milliseconds) are displayed as a real number in an 8 character area with 3 decimal digits of precision.
   - Each acceleration is shown in a 7 character area with 4 digits of precision.

9. Capture the output using a Snipping Tool or a screenshot for your report.

Part 2

1. Comment out CODE SECTION 0 and remove the comments around CODE SECTION 1. Note that section 1 uses a function named *mag*.
2. Create the *mag* function at the bottom of the program. Remember to put the prototype at the top.
    a. *mag* will take 3 real arguments and return a real value. What datatypes should you use?
    b. You can make use of the math library for the sqrt and pow functions, but you will have to look up how to use them. Ask Professor Google.
    c. When you compile, you have to include a -lm at the end of the command. The “-l” means link the “m” means math. Using “-lm” and having `#include <math.h>` tell the compiler to link the math library with your program.

   mag is the magnitude of the acceleration, which is the square root of the sum of the squared values of the accelerations on that line (i.e. $\sqrt{X^2 + Y^2 + Z^2}$).

3. Discuss with your neighbors what the output of *mag* should look like.
4. Test your code to confirm your implementation of the *mag* function works.
5. Capture the output for your report.

Part 3

1. Comment out CODE SECTION 1 and uncomment CODE SECTION 2. Your job is to make this line return time in a more readable format. For instance, instead of 129313 ms, it should show 2 minutes, 9 seconds, and 313 ms.
2. Write the 3 additional functions to make the line operate.
3. **Show your functions and output in your lab report.**
4. Test your code to confirm that it works.

Part 4

In this section, you will create a program that outputs the number of buttons pressed each time the DualShock 4 outputs a line. The following will show you the shape buttons being pushed (TRIANGLE, CIRCLE, X, SQUARE).

```
./ds4rd.exe -d 054c:05c4 -D DS4_BT -b
```

There are more buttons on the ds4 but we will limit it to only the shape buttons that are provided when you use the -b flag.

1. Copy your program from above (lab3.c) and write a new program called lab3-2.c
2. Eliminate the functions from before and all the code inside the "while loop" brackets.
3. Write a function that returns the number of buttons currently pressed. The function is not allowed to use printf() or scanf(). Think, what are other ways you can get data to this function?
4. Print the number returned from the function in (3) within the while loop.
5. You will need a line that says `fflush(stdout);` after your `printf` inside the infinite loop, ask your TA why.
6. Capture the output using the '>' trick.

   ```
   ./ds4rd.exe -d 054c:05c4 -D DS4_BT -b | ./lab3-2.exe > output.txt
   ```

   Include a copy of your program source code and a single page of output from it in your lab report.

Turn-In:

You must use the lab report format for this lab report.

In the lab report, you should have subsections for each of Part 1, 2, 3, and 4 from above. These are small problems, so each subsection can be fairly short. If a section above says to include something in the report, be sure to include it. The Turn-In section is **not** an exhaustive list of what needs to be include.

These are some common gotchas though:

1. Lab report detailing your experiences (follow the lab report format!).
2. Source code from all sections.
3. Output from all sections.