**Introduction to Functions**

**LAB #3**

**SECTION 5**

**SUBMITTED BY:**

**Jackson Collalti**

**SUBMISSION DATE:**

**09-19-2022**

**Problem**

With the goal of learning how to implement and use functions, this lab will show how to create functions, format output practice, and practice writing mod and integer division expressions. Problem 1, using a Dualshock 4 controller, change the format's acceleration and time. Problem 2, learn how to use math functions such as square roots and functions to display the magnitude of accelerations. Problem 3, Change milliseconds into minutes:seconds: milliseconds. Problem 4, Create a program to detect how many buttons are being pressed without using print and scan in my function.

**Analysis**

Problem 1, with the source code given, change the outputs so that the program outputs seconds, all on one line, separated by commas. Change the output of milliseconds to seconds and display eight characters with three digits of precision. Finally, change the acceleration output to have seven total characters, 4 of which are after the decimal point. No change in inputs.

Problem 2, create a function, *mag,* that calculates the magnitude of acceleration. The function's inputs would be the doubles of acceleration in the x, y, and z directions. An equation that would be used is the square root of (x^2+y^2+z^2). The output should be the result of the square root.

Problem 3, using three functions, produce minutes, seconds, and milliseconds. The function for minutes will be called minutes, seconds will be called seconds, and milliseconds will be called millis. All three will have the same input of the integer for time. The outputs will each go with the name of the function. To complete this, I’ll need to use both integer division and remainder division.

Problem 4, create a program that outputs the number of buttons being pressed, with the constraints of not being able to use scan and print in the function itself. Since print will be in an infinite while loop, fflush(stdout); will be needed. Inputs will be the four buttons circle, triangle, square, and x.

**Design**

Problem 1, the source code already has commas that separate the time and acceleration, so that problem is already solved. To get seconds displayed as a real number with eight characters and three decimal places, I’ll need to create a new variable. I’ll call the new variable secs since it needs to be a real number secs will be a double. To convert milliseconds to seconds, I’ll need to divide milliseconds by 1000.000. For acceleration and time outputs, the format will be % number of total characters. Number of decimal places lf (%8.3lf for seconds, %7.4lf for accelerations).

Problem 2, To start, I need to create a function called mag which will produce a double. The function mag will have three inputs, ax, ay, and az, all of which are doubles. I will need to find out how to enter a square root function into my program. Instead of ax^2+ay^2+az^2, I plan on writing ax\*ax+ay\*+ay+az\*az since I feel this is easier than learning how to use a power function.

Problem 3, I feel it’s easiest to start with the minutes function since that is only one math equation, t(in milliseconds) / 60000. I want integer division since I want 2.5 minutes to display as 2 minutes. I need to do remainder division for the seconds to get the milliseconds not already in the minutes function. Then I need to divide the remainder of milliseconds by 1000. I need to find the remainder of what's left after the seconds function and display that number for the milliseconds' function.

Example input 150001 milliseconds.

(150001/60000=2 minutes)

(150001%60000=30001, 30001/1000=30 seconds)

(150001%60000=30001, 30001%1000=1 milliseconds)

Problem 4, I need to create a function for the number of buttons being pressed. This would be the sum of the variables assigned to the buttons. If a button is pressed, the variable should be 1 and 0 if not pressed. Since print and scan can’t be in the function, they need to be in main. I’ll need the print statement that calls to and prints the function using the variables from the scan function in main. After the print statement, I’ll need to add a fflush(stdout); because problems will happen with a print statement in a while loop.

**Testing**

Problem 1, before making changes, the instructions said to run the program and view the results. From there, I modified the code to solve the problem I was tasked with. I tested my program and compiled and let it run for a couple of seconds, ensuring that the values were displayed correctly. I watched until the seconds displayed 10.

Problem 2, After I created the mag function, I ran and compiled using -lm to link math libary. While running the code, I moved around the controller to get different values and to ensure the outputs made sense. With the function I had created, it should be impossible to have negative numbers.

Problem 3, I wrote the code for my three functions and ran the program. I waited and watched as numbers rolled over. I watched until a bit past the one-minute mark to ensure all three functions worked properly.

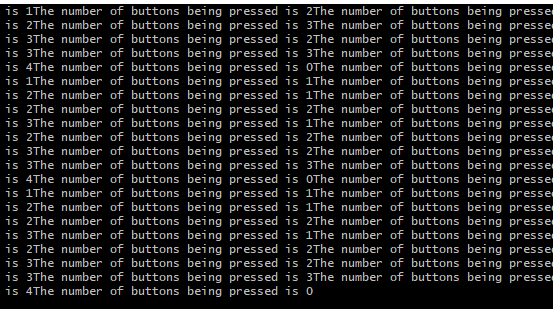
Problem 4, After setting up and writing my program, I tested it out by pushing different buttons. Since I only had four buttons assigned as variables, the max value I could have is four. Anything more would mean something is wrong. This also means that negative numbers should be impossible. During testing, the program worked as intended.

**Comments**

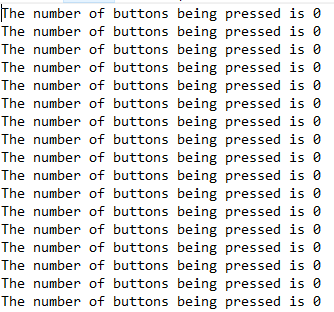
Problem 1, I got a refresher on how to display numbers in certain formats. I also changed an input variable from an integer to a double. The task was simple, and the only mistake I made was that I originally forgot to change my new double for time to %lf.

Problem 2, I had to google how to get a square root into my program, and learned it’s sqrt();. I ran into problems but I reread directions and added -lm to my command when about to run my program.

Problem 3, it helped that I already had a good idea of what the equations for functions would look like. This problem is helpful for future labs or anything where I need to write code to convert units.

Problem 4, this is where I ran into most of my problems. They were all relatively small and not super big issues.I forgot to add the fflush(stdout); and learned what happens if a print statement is used freely in a while loop. This was a simple solution of adding the fflush statement.

The second problem I had was the text file I was outputting would only show 0 as the output. When I’d run my code in cygwin the program would output correct values. I forget how I fixed it but it might have been adding -lm to the command statement.



As far as coding I wouldn’t change anything. I would change my time management of the lab, I left the my lab on Tuesday a bit early to prepare for another class. I planned to finish it over the weekend, not knowing that Coover is closed and the doors are locked on weekends. As a result, I had to go in early Monday right after the building openned to finish the lab before my first class started.

**Implementation**

**Source Code given along with a file Ds4rd.exe**

/\* 185 Lab 3 Template \*/

#include <stdio.h>

#include <math.h>

/\* Put your function prototypes here \*/

int main(void) {

/\* DO NOT MODIFY THESE VARIABLE DECLARATIONS \*/

int t;

double ax, ay, az;

/\* This while loop makes your code repeat. Don't get rid of it. \*/

while (1) {

scanf("%d,%lf,%lf,%lf", &t, &ax, &ay, &az);

/\* CODE SECTION 0 \*/

printf("Echoing output: %d, %lf, %lf, %lf\n", t, ax, ay, az);

/\* CODE SECTION 1 \*/

/\*printf("At %d ms, the acceleration's magnitude was: %lf\n",

t, mag(ax, ay, az)); \*/

/\* CODE SECTION 2 \*/

/\*printf("At %d minutes, %d seconds, and %d milliseconds it was: %lf\n",

minutes(t), seconds(t), millis(t), mag(ax,ay,az)); \*/

}

return 0;

}

/\* Put your functions here \*/

**Code for problems 1-3**

/\* 185 Lab 3 Template \*/

#include <stdio.h>

#include <math.h>

/\* Put your function prototypes here \*/

double mag(double ax, double ay, double az){

double Mag;

Mag = sqrt(ax \* ax + ay \* ay + az \* az);

}

int minutes(int t){

t = t / 60000;

}

int seconds(int t){

t= t % 60000;

t= t / 1000;

}

int millis(int t){

t = t %60000;

t = t %1000;

}

int main(void) {

/\* DO NOT MODIFY THESE VARIABLE DECLARATIONS \*/

int t;

double ax, ay, az;

/\* This while loop makes your code repeat. Don't get rid of it. \*/

while (1) {

scanf("%d,%lf,%lf,%lf", &t, &ax, &ay, &az);

/\* CODE SECTION 0 \*/

/\* double secs;

secs = t / 1000.000;

printf("Echoing output: %8.3lf, %7.4lf, %7.4lf, %7.4lf\n", secs, ax, ay, az);

\*/

/\* CODE SECTION 1 \*/

/\* printf("At %d ms, the acceleration's magnitude was: %lf\n",

t, mag(ax, ay, az));

\*/

/\* CODE SECTION 2 \*/

printf("At %d minutes, %d seconds, and %d milliseconds it was: %lf\n",

minutes(t), seconds(t), millis(t), mag(ax,ay,az));

}

return 0;

}

/\* Put your functions here \*/

/\*

double mag(double ax, double ay, double az){

double Mag;

Mag = sqrt(ax \* ax + ay \* ay + az \* az);

}

int minutes(int t){

t = t / 60000;

}

int seconds(int t){

t= t % 60000;

t= t / 1000;

}

int millis(int t){

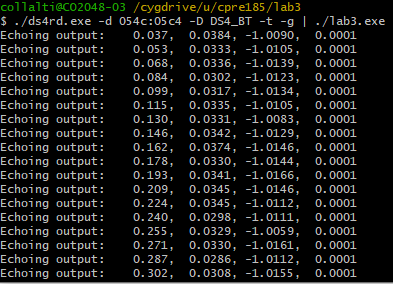
t = t %60000;

t = t %1000;

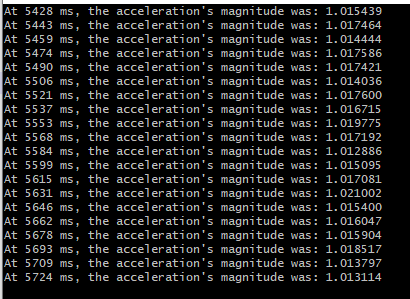
}

\*/

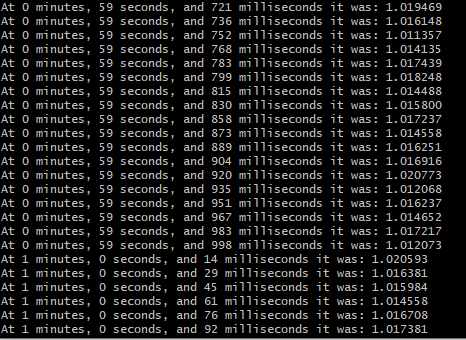
**Problem 1 Output**



**Problem 2 Output**



**Problem 3 Output**



**Problem 4**

/\* 185 Lab 3 Template \*/

#include <stdio.h>

#include <math.h>

/\* Put your function prototypes here \*/

int numButtons(int bx,int bc,int bs,int bt) {

int pressedButtons;

pressedButtons = bx + bc + bs + bt;

}

int main(void) {

int bx;//button x

int bc;//button circle

int bs;//button square

int bt;//button triangle

while(1)

{

scanf("%d, %d, %d, %d",&bt,&bc,&bx,&bs);

{

printf("The number of buttons being pressed is %d\n",numButtons(bx,bc,bs,bt));

fflush(stdout);

}

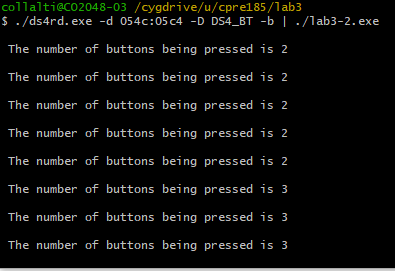
}

return 0;

}

/\* Put your functions here \*/

**Problem 4 Output**



**Problem 4 Output Text File**

