INTRODUCTION TO FUNCTIONS LAB REPORT

LAB #3

SECTION M

SUBMITTED BY:

SHOUNAK LAHIRI

SUBMISSION DATE:

9/13/2018

Lab Problem

The purpose of this lab is to modify an existing program which contains input and output statements and displays the gyroscopic measurements of the PlayStation 4 controller. The program must be modified multiple times with different goals for each modification. The goals of the modifications include changing the unit of time from milliseconds to seconds, calculating the magnitude and acceleration, changing how the time is displayed (outputted), and to show the number of buttons being pressed on the controller at the given time.

Analysis

For part 1 of the lab, the problem states that the output must be formatted to a fit on one line with a specified amount of characters and the time be displayed in seconds rather than milliseconds. Part 2 of the lab the problem states that the magnitude of the acceleration must be calculated using a function called mag. The magnitude of the acceleration can be calculated using by taking the square root of the squares of each the x, y, and z measurements. For part 3 of the lab, the problem states that the time must be put in a more readable format, minutes, seconds, and milliseconds instead of just milliseconds. It is important to know that the conversion, 1 minute is equal to 60 seconds and 60000 milliseconds. For part 4 of the lab, the problem states that the number of buttons pressed on the controller must be outputted using a function and no printf or scanf statement may be used.

Design

Part 1:

Our problem was to make the program output on one line, change the time from milliseconds to seconds, and format the output to a certain number of characters. Working in the section of code labeled section 0, we were able to alter the printf statements that controlled the output of the program.

Part 2:

Our problem was to create a mag function that took in three values and outputted the magnitude of the acceleration of the controller. The first step was to comment out the code form the previous part and uncomment section of code labeled section 1. Then, we created the function and the function protype and placed them in the specified regions of the code. The mag function required the input of three integer values, it then made use of the square root and power functions to calculate the magnitude of the acceleration, and returned it to the mag function call.

Part 3:

Our problem was to convert the outputted time from only milliseconds to a more readable format like minutes, seconds, and milliseconds using only three functions. We broke the problem into smaller sub steps:

1. Calculate the minutes
2. Calculate the seconds
3. Calculate the milliseconds

To calculate minutes we used the conversion factor of 1 minute = 60000 milliseconds. Dividing the number of milliseconds by 60000 gave the minutes. To calculate the seconds we used the idea that remainder from the previous calculation was the number of seconds in milliseconds, so we divided the remainder by 1000 to get the seconds. The remainder of the previous calculation would give the number of milliseconds.

Part 4:

Our problem was to create a function that calculated the number of buttons pressed on the controller at the current time without using the scanf and printf commands inside of the function. First, we created a function and the function prototype, the function needed four integers, each one representing a button on the controller. The function then added the incoming integers together to get the number of buttons pressed, this method works because if the button is pressed the value of the variable is 1, if it is not pressed then the value of the variable is 0.

Testing

Part 1:

To test that the program was working correctly, we ran the program and moved the controller around for a few seconds. Then, we looked at the output data and noticed that the output was looked formatted, to verify the number of spaces we counted the number of spaces.

Part 2:

There was no real way to verify the magnitude of the acceleration because the input values were not being outputted. The only way that we could see if the results were making sense was to compare our results with the results from the people seated nearby.

Part 3:

To test the results, we used the stopwatch on our phone and started the time at the same time that we started running the program. We stopped the stopwatch at the same time that the program was stopped. Then, we compared the last entry to see that the time was very close to the time on the stopwatch.

Part 4:

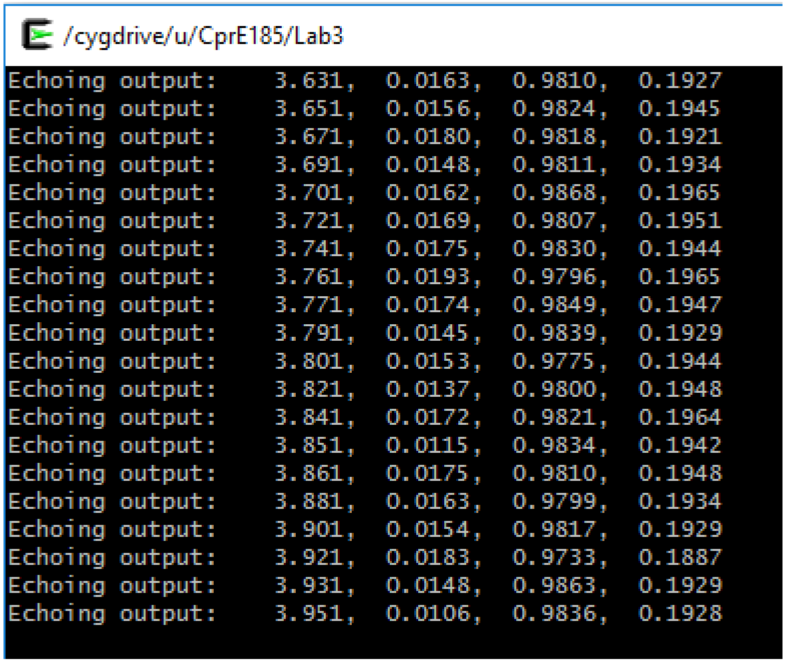
To test the results we pressed a certain number of buttons repeatedly. For a certain amount of time only one button would be pressed, then two, three… When looking at the output the number of buttons that were pressed match the order in which they were pressed, which confirmed that the program was working correctly.

Comments

In doing this lab I learned that checking multiple outputs is very important because using only one value to test the program is not good enough, as there can be loopholes that make it seem like the program is working correctly, when in fact it is not.

Implementation

Part 1:

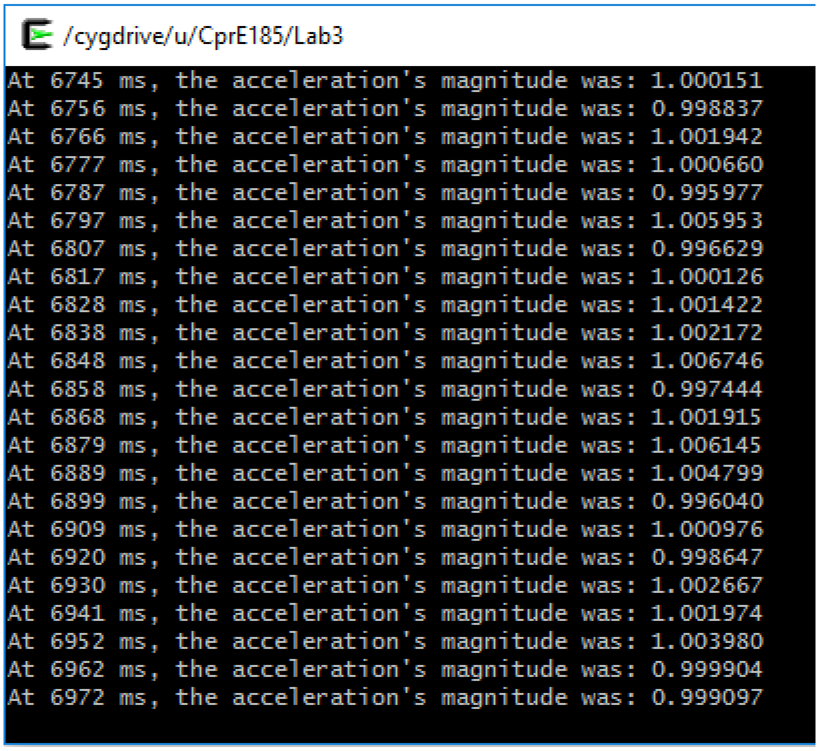


```
Echoing output:    3.631,  0.0163,  0.9810,  0.1927
Echoing output:    3.651,  0.0156,  0.9824,  0.1945
Echoing output:    3.671,  0.0180,  0.9818,  0.1921
Echoing output:    3.691,  0.0148,  0.9811,  0.1934
Echoing output:    3.701,  0.0162,  0.9868,  0.1965
Echoing output:    3.721,  0.0169,  0.9807,  0.1951
Echoing output:    3.741,  0.0175,  0.9830,  0.1944
Echoing output:    3.761,  0.0193,  0.9796,  0.1965
Echoing output:    3.771,  0.0174,  0.9849,  0.1947
Echoing output:    3.791,  0.0145,  0.9839,  0.1929
Echoing output:    3.801,  0.0153,  0.9775,  0.1944
Echoing output:    3.821,  0.0137,  0.9800,  0.1948
Echoing output:    3.841,  0.0172,  0.9821,  0.1964
Echoing output:    3.851,  0.0115,  0.9834,  0.1942
Echoing output:    3.861,  0.0175,  0.9810,  0.1948
Echoing output:    3.881,  0.0163,  0.9799,  0.1934
Echoing output:    3.901,  0.0154,  0.9817,  0.1929
Echoing output:    3.921,  0.0183,  0.9733,  0.1887
Echoing output:    3.931,  0.0148,  0.9863,  0.1929
Echoing output:    3.951,  0.0106,  0.9836,  0.1928
```

```
/* 185 Lab 3 Template */

#include <stdio.h>
#include <math.h>

/* Put your function prototypes here */

double mag(double x, double y, double z);
int minutes(int t);
int seconds(int t);
int millis(int t);

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 timeSec = (t / (1000.0));
        printf("Echoing output: %8.3lf, %7.4lf, %7.4lf, %7.4lf\n", timeSec,
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 x, double y, double z)
{
      double value= pow(x,2) + pow(y,2) + pow(z,2);
      return sqrt(value);

}

int minutes (int time)
{
      return time/60000;
}

int seconds(int time)
{
      return (time % 60000) /1000;
}
```

```
int millis(int time)
{
	return ((time % 60000) % 1000);
}
```

Part 2:

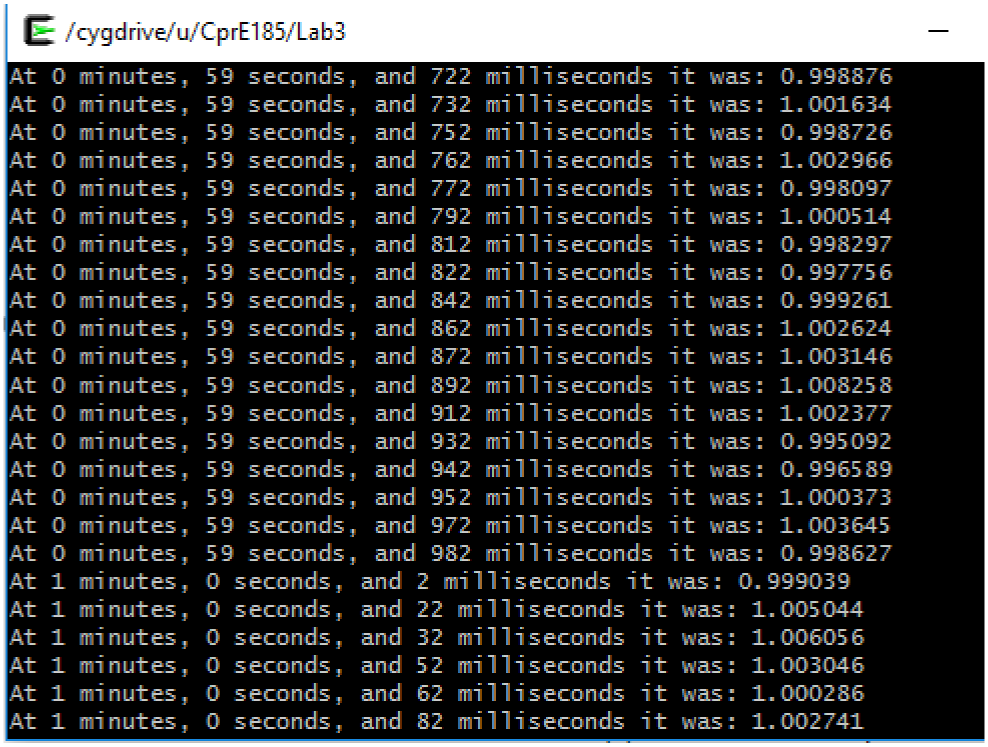


```
At 6745 ms, the acceleration's magnitude was: 1.000151
At 6756 ms, the acceleration's magnitude was: 0.998837
At 6766 ms, the acceleration's magnitude was: 1.001942
At 6777 ms, the acceleration's magnitude was: 1.000660
At 6787 ms, the acceleration's magnitude was: 0.995977
At 6797 ms, the acceleration's magnitude was: 1.005953
At 6807 ms, the acceleration's magnitude was: 0.996629
At 6817 ms, the acceleration's magnitude was: 1.000126
At 6828 ms, the acceleration's magnitude was: 1.001422
At 6838 ms, the acceleration's magnitude was: 1.002172
At 6848 ms, the acceleration's magnitude was: 1.006746
At 6858 ms, the acceleration's magnitude was: 0.997444
At 6868 ms, the acceleration's magnitude was: 1.001915
At 6879 ms, the acceleration's magnitude was: 1.006145
At 6889 ms, the acceleration's magnitude was: 1.004799
At 6899 ms, the acceleration's magnitude was: 0.996040
At 6909 ms, the acceleration's magnitude was: 1.000976
At 6920 ms, the acceleration's magnitude was: 0.998647
At 6930 ms, the acceleration's magnitude was: 1.002667
At 6941 ms, the acceleration's magnitude was: 1.001974
At 6952 ms, the acceleration's magnitude was: 1.003980
At 6962 ms, the acceleration's magnitude was: 0.999904
At 6972 ms, the acceleration's magnitude was: 0.999097
```

```
/* 185 Lab 3 Template */

#include <stdio.h>
#include <math.h>

/* Put your function prototypes here */

double mag(double x, double y, double z);
int minutes(int t);
int seconds(int t);
int millis(int t);
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 timeSec = (t / (1000.0));
        printf("Echoing output: %8.3lf, %7.4lf, %7.4lf, %7.4lf\n", timeSec,
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 x, double y, double z)
{
      double value= pow(x,2) + pow(y,2) + pow(z,2);
      return sqrt(value);

}
```

```
int minutes (int time)
{
      return time/60000;
}

int seconds(int time)
{
      return (time % 60000) /1000;
}

int millis(int time)
{
      return ((time % 60000) % 1000);
}
```

Part 3:



```
At 0 minutes, 59 seconds, and 722 milliseconds it was: 0.998876
At 0 minutes, 59 seconds, and 732 milliseconds it was: 1.001634
At 0 minutes, 59 seconds, and 752 milliseconds it was: 0.998726
At 0 minutes, 59 seconds, and 762 milliseconds it was: 1.002966
At 0 minutes, 59 seconds, and 772 milliseconds it was: 0.998097
At 0 minutes, 59 seconds, and 792 milliseconds it was: 1.000514
At 0 minutes, 59 seconds, and 812 milliseconds it was: 0.998297
At 0 minutes, 59 seconds, and 822 milliseconds it was: 0.997756
At 0 minutes, 59 seconds, and 842 milliseconds it was: 0.999261
At 0 minutes, 59 seconds, and 862 milliseconds it was: 1.002624
At 0 minutes, 59 seconds, and 872 milliseconds it was: 1.003146
At 0 minutes, 59 seconds, and 892 milliseconds it was: 1.008258
At 0 minutes, 59 seconds, and 912 milliseconds it was: 1.002377
At 0 minutes, 59 seconds, and 932 milliseconds it was: 0.995092
At 0 minutes, 59 seconds, and 942 milliseconds it was: 0.996589
At 0 minutes, 59 seconds, and 952 milliseconds it was: 1.000373
At 0 minutes, 59 seconds, and 972 milliseconds it was: 1.003645
At 0 minutes, 59 seconds, and 982 milliseconds it was: 0.998627
At 1 minutes, 0 seconds, and 2 milliseconds it was: 0.999039
At 1 minutes, 0 seconds, and 22 milliseconds it was: 1.005044
At 1 minutes, 0 seconds, and 32 milliseconds it was: 1.006056
At 1 minutes, 0 seconds, and 52 milliseconds it was: 1.003046
At 1 minutes, 0 seconds, and 62 milliseconds it was: 1.000286
At 1 minutes, 0 seconds, and 82 milliseconds it was: 1.002741
```

```c
/* 185 Lab 3 Template */

#include <stdio.h>
#include <math.h>

/* Put your function prototypes here */

double mag(double x, double y, double z);
int minutes(int t);
int seconds(int t);
int millis(int t);

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 timeSec = (t / (1000.0));
        printf("Echoing output: %8.3lf, %7.4lf, %7.4lf, %7.4lf\n", timeSec,
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 x, double y, double z)
{
      double value= pow(x,2) + pow(y,2) + pow(z,2);
      return sqrt(value);

}

int minutes (int time)
{
      return time/60000;
}

int seconds(int time)
{
      return (time % 60000) /1000;
}
```

```
int millis(int time)
{
	return ((time % 60000) % 1000);
}
```

Part 4:

Output:

```
1 Buttons Are Pressed Currently
1 Buttons Are Pressed Currently
1 Buttons Are Pressed Currently
1 Buttons Are Pressed Currently
1 Buttons Are Pressed Currently
2 Buttons Are Pressed Currently
2 Buttons Are Pressed Currently
2 Buttons Are Pressed Currently
2 Buttons Are Pressed Currently
2 Buttons Are Pressed Currently
3 Buttons Are Pressed Currently
3 Buttons Are Pressed Currently
3 Buttons Are Pressed Currently
3 Buttons Are Pressed Currently
3 Buttons Are Pressed Currently
4 Buttons Are Pressed Currently
4 Buttons Are Pressed Currently
4 Buttons Are Pressed Currently
4 Buttons Are Pressed Currently
4 Buttons Are Pressed Currently
```

```
2 Buttons Are Pressed Currently
2 Buttons Are Pressed Currently
2 Buttons Are Pressed Currently
2 Buttons Are Pressed Currently
2 Buttons Are Pressed Currently
2 Buttons Are Pressed Currently
2 Buttons Are Pressed Currently
2 Buttons Are Pressed Currently
2 Buttons Are Pressed Currently
2 Buttons Are Pressed Currently
3 Buttons Are Pressed Currently
3 Buttons Are Pressed Currently
3 Buttons Are Pressed Currently
3 Buttons Are Pressed Currently
3 Buttons Are Pressed Currently
1 Buttons Are Pressed Currently
1 Buttons Are Pressed Currently
1 Buttons Are Pressed Currently
1 Buttons Are Pressed Currently
1 Buttons Are Pressed Currently
```

```c
      /* 185 Lab 3 Template */

#include <stdio.h>
#include <math.h>

/* Put your function prototypes here */
int numPressed(int tri, int circ, int x, int sqr);

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) {

                        int triangle, circle, x, square;
                        scanf(" %d , %d , %d , %d " ,&triangle , &circle,
&x , &square);
                        printf("%d Buttons Are Pressed Currently\n",
numPressed(triangle, circle, x, square));
                        fflush(stdout);

    }

return 0;
}

/* Put your functions here */


int numPressed(int tri, int circ, int x, int sqr)
{
      int num =tri + circ + x + sqr;
      return num;
}
```