LAB REPORT

LAB 4
SECTION C

SUBMITTED BY:

SCOTT VLASIC

SUBMISSION DATE:

FEBRUARY 17, 2017

Problem:

There were four main problems that I had to tackle in this lab. In Part 1, I had to modify the code so that the numbers being printed were displayed with the proper number of decimals and placeholders. In Part 2, I had to write a function that would print the magnitude of the acceleration. In Part 3, I had to write a function that would convert a time in milliseconds to minutes, seconds, and milliseconds. Finally, in Part 4, I had to write a new program that would return the number of buttons pressed when I pressed the buttons on the esplora.

Analysis:

To begin this lab I had to open lab4.c in Notepad++ and begin solving for Part 1. For Part 1 I had to look at the printf function and decide how I could make it so that seconds were displayed in an 8 character area with 3 decimal digits of precision and so the acceleration was displayed in a 7 character area with 4 digits of precision.

For Part 2, I had to look at the commented code and create a mag function at the bottom of the page so that the code would run. I had to include a prototype at the top of the page as well. The mag function would need to print the magnitude of the acceleration which is given as the sqrt(x^2 + y^2 + z^2).

In Part 3, I had to take a given time in milliseconds and convert it to a time in minutes, seconds and milliseconds. This also required a prototype and I had to look at the conversions from certain times so that milliseconds would convert properly into the three other time segments I wanted.

In Part 4, I was given the code from lab4.c and I had to write an entirely new program that would return the number of buttons pressed when I was using the esplora. This required a new function at the bottom and I had to take into account certain factors of the esplora such as the slider and how that affected the button.

Design:

To begin solving Part 1, I had to look at the code I was given and how I could modify the printf statement so that seconds and acceleration were printed with the correct number of characters and decimal places. To do this I added 8.3 to the seconds which would have shown in the printf statement as %8.3 and I also added 7.4 for the acceleration which would have shown as %7.4 in the printf statement. With these additions, the seconds and accelerations had the correct number of characters and decimals as can be seen in the picture of the output at the end.

With Part 2, I had to write a whole new function at the bottom of the code for the magnitude of the acceleration. The magnitude equals the sqrt(x^2 + y^2 +z^2), which are all double values initialized in the main function. When I wrote my code for the function for mag, I wrote it as a double with the parameters of double ax, double ay, and double az. The return

was the sqrt of all these parameters squared, thus providing us with the magnitude of the acceleration. An output example can be seen at the end as well.

In Part 3, I had to convert time in milliseconds into time in minutes, seconds, and milliseconds. To do this I had to write three separate functions at the end of the code, one for time in minutes, one for time in seconds, and one for time in milliseconds. For my function of minutes I took the variable for time, t, and divided it by the number of milliseconds in a minutes which is 60000. Then, for seconds I took time%60000 and divided it by the number of milliseconds in a second, which is 1000. Finally, to find milliseconds, I took (time%60000)%1000 which in turn gave the number of milliseconds. The output for the function can be seen at the end.

Finally, for Part 4, I had to write code that would print the number of buttons being pressed when I was working on the esplora. Therefore, I had to initialize variables for all of the buttons as well as the slider on the esplora. However, in my while loop I had all the variables be scanned yet only the button variables be printed since the slider is not a button by definition. I also wrote a function below my main function that returned the button being pressed given the parameters of all the buttons. This was done by returning all the buttons added together and then printing only the number of ones that were being pressed. An example of the output is below. The 1 in the output shows that one button is being pressed while the two says that two buttons are being pressed.

```
At 196590 ms, the acceleration's magnitude was: 1.058778
At 3 minutes, 16 seconds, and 590 milliseconds it was: 1.058778
Echoing output:   196593,  0.0047,  0.0547,  1.0574
At 196593 ms, the acceleration's magnitude was: 1.058778
At 3 minutes, 16 seconds, and 593 milliseconds it was: 1.058778
Echoing output:   196595,  0.0108,  0.0483,  1.0512
At 196595 ms, the acceleration's magnitude was: 1.052393
At 3 minutes, 16 seconds, and 595 milliseconds it was: 1.052393
Echoing output:   196597,  0.0047,  0.0483,  1.0512
At 196597 ms, the acceleration's magnitude was: 1.052348
At 3 minutes, 16 seconds, and 597 milliseconds it was: 1.052348
Echoing output:   196600,  0.0047,  0.0547,  1.0574
At 196600 ms, the acceleration's magnitude was: 1.058778
At 3 minutes, 16 seconds, and 600 milliseconds it was: 1.058778
Echoing output:   196602,  0.0047,  0.0547,  1.0574
At 196602 ms, the acceleration's magnitude was: 1.058778
At 3 minutes, 16 seconds, and 602 milliseconds it was: 1.058778
Echoing output:   196604,  0.0047,  0.0547,  1.0635
At 196604 ms, the acceleration's magnitude was: 1.064897
At 3 minutes, 16 seconds, and 604 milliseconds it was: 1.064897
Echoing output:   196606, -0.0013,  0.0547,  1.0635
At 196606 ms, the acceleration's magnitude was: 1.064887
At 3 minutes, 16 seconds, and 606 milliseconds it was: 1.064887
Echoing output:   196610,  0.0047,  0.0547,  1.0635
At 196610 ms, the acceleration's magnitude was: 1.064897
At 3 minutes, 16 seconds, and 610 milliseconds it was: 1.064897
Echoing output:   196612,  0.0108,  0.0547,  1.0635
At 196612 ms, the acceleration's magnitude was: 1.064941
At 3 minutes, 16 seconds, and 612 milliseconds it was: 1.064941
Echoing output:   196614,  0.0047,  0.0547,  1.0635
```

Testing:

To test the values for Part 1, I simply looked at my code to see if the numbers being printed from the esplora were containing the character numbers and decimal places that were being asked for. If not I went back and corrected it until I finally had the correct output.

To get the magnitude for Part 2, I looked at the values for Part 1 and then calculated them myself and saw if the answer I received was the same as the one that the code was outputting. The equation I used to test this was the same one that I put in the code, sqrt(x^2 + y^2 + z^2).

To see if the millisecond conversion was correct in Part 3, I used the example that was given to us in the lab instructions. 129313ms should've printed as 2 minutes, 9 seconds, and 313 ms. Therefore, I used this as the testing value for my code and once my code displayed as the desired output, I knew my code had worked.

For Part 4, I had to ask my TA for help with testing since I was a tad confused about how the slider worked. Looking at the output I should be able to tell if it's working by looking at how the numbers changed as I pressed more or less buttons on the esplora. For example. When I was pressing 2 buttons, the output should've been 2.

Comments:

Looking at the code I wrote I feel more comfortable with while loops and how they work. Before this lab I was confused as to how the outputs of while loops worked but now that I have completed it I feel more solidified on the topic. One major issue I had regarding the lab was the slider and how it contributed into Part 4. I feel as though specifying that the slider needed

to be accounted for would've made the lab easier to understand for me. I just asked my TA for help and had my questions answered, which allowed me to complete the lab.

Implementation:

```
/* Lab 4 Wrapper Program */

#include <stdio.h>

#include <math.h>

/* Put your function prototypes here */

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


int main(void) {

        int t;

        double  ax, ay, az;

        while (1) {

                scanf("%d,%lf,%lf,%lf", &t, &ax, &ay, &az);
/* CODE SECTION 0 */

                printf("Echoing output: %8.3d, %7.4lf, %7.4lf, %7.4lf\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 */

double mag(double ax, double ay, double az){
        return sqrt(ax*ax+ay*ay+az*az);
}

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

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

int millis(int t){
        return ((t%60000)%1000);
}



/* Lab 4 Wrapper Program */

#include <stdio.h>

#include <math.h>



/* Put your function prototypes here */
```

```
int button(int a, int b, int c, int d, int e);

int main(void) {

	int t;

	int a,b,c,d,e,f;

	while (1) {
		scanf("%d,%d,%d,%d,%d,%d", &a, &b, &c, &d, &e, &f);
		printf("%d\n",button(a,b,c,d,e));
		fflush(stdout);
	}

	return 0;
}

int button(int a, int b, int c, int d, int e){
	return (a+b+c+d+e);
}
```