

**Lab 3**

**LAB 3**

**SECTION C**

**Akash Patel**

**SUBMISSION DATE: 2/13/19**

**2/13/19**

## Problem

The purpose of this lab was to see how the output of the program changes depending on what you do with the controller and how to change the output by altering the code.

## Analysis

The first program required me to change the orientation of the controller, get the output and create a graph in excel. The second program I had to convert units and change values to show proper decimal places and required unit of time. The third program had to count the numbers being pressed and output that number.

## Design

Program one: There was no change in code for this. It was a simple run the program, look at the output and turn it into a scatter plot graph

Program two: I declared seconds, minutes and milliseconds as variables and converted them to their respective values. I also changed the values in scanf() to return double type numbers. The sqrt function included three variables which return the mag function as a double type.

Program three: This program counted the number of buttons being pressed and outputted that value. I created a variable for each button and declared "tot" as a variable. I then created an equation which adds the variables and outputs the total as an integer.

## Testing

The first program was simple, I did it a few times to be sure that I was doing it right. The second program took some time to get right. I was getting the correct decimal places but not characters so I had to change the .lf to "x.xlf" in the output. That took a few tests to get correct as well as getting the mag function. The third program also took a good amount of testing. I misunderstood it as counting the buttons pressed and outputting the total so I was using a counter then I realized that it asked for how many buttons are being pressed and changed

my code to represent the four buttons and total them. The program then started outputting the amount of buttons being pressed at a time as it was running.

## Comments

I learned to double check my work and make sure I'm using the syntax properly. I also learned that I should reread the instructions. This lab wasn't so difficult sine I have done this before.

## Source Code

```
/*-----  
-                               SE 185 Lab 03  
-       Developed for 185-Rursch by T.Tran and K.Wang  
-       Name:   
-       Section:   
-       NetID:   
-       Date:   
-----*/  
  
/*-----  
-                               Includes  
-----*/  
#include <stdio.h>  
#include <math.h>  
  
/*-----  
-                               Defines  
-----*/  
#define TRUE 1  
  
/*-----  
-                               Prototypes  
-----*/  
/* 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);  
  
/*-----  
-                               Implementation  
-----*/  
int main(void) {  
    /* DO NOT MODIFY THESE VARIABLE DECLARATIONS */  
    int t;  
    double ax, ay, az;  
  
    while (TRUE) {  
        scanf("%d,%lf,%lf,%lf", &t, &ax, &ay, &az);  
  
        /* CODE SECTION 0 */  
        printf("Echoing output: %8.3lf, %7.4lf, %7.4lf, %7.4lf\n", t/1000.0, 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) {
    //Step 8, uncomment and modify the next line
    return sqrt((pow(x,2) + pow(y,2) + pow(z,2)) );
}
```

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

}

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

3

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

3

```

/*-----*
SE 185 Lab 03
Developed for 185-Rursch by T.Tran and K.Wang
Name:
Section:
NetID:
Date:
-----*/

```

```
/*-----  
- Includes -----  
-----*/  
#include <stdio.h>  
#include <math.h>
```

```
/*-----  
- Defines -----*/  
#define TRUE 1
```

```
/*-----  
- Prototypes -----*/
```

```
/*-----  
- Implementation -----*/
```

```
int main(void) {  
    int s;  
    int t;  
    int c;  
    int x;  
    int tot;
```

```
    while (TRUE) {
```

```
        scanf("%d,%d,%d,%d",&s,&t,&c,&x);
```

```
        tot = s + t + c + x;
```

```
        fflush(stdout);
```

```
        printf("Buttons being pressed: %d\n",tot);
```

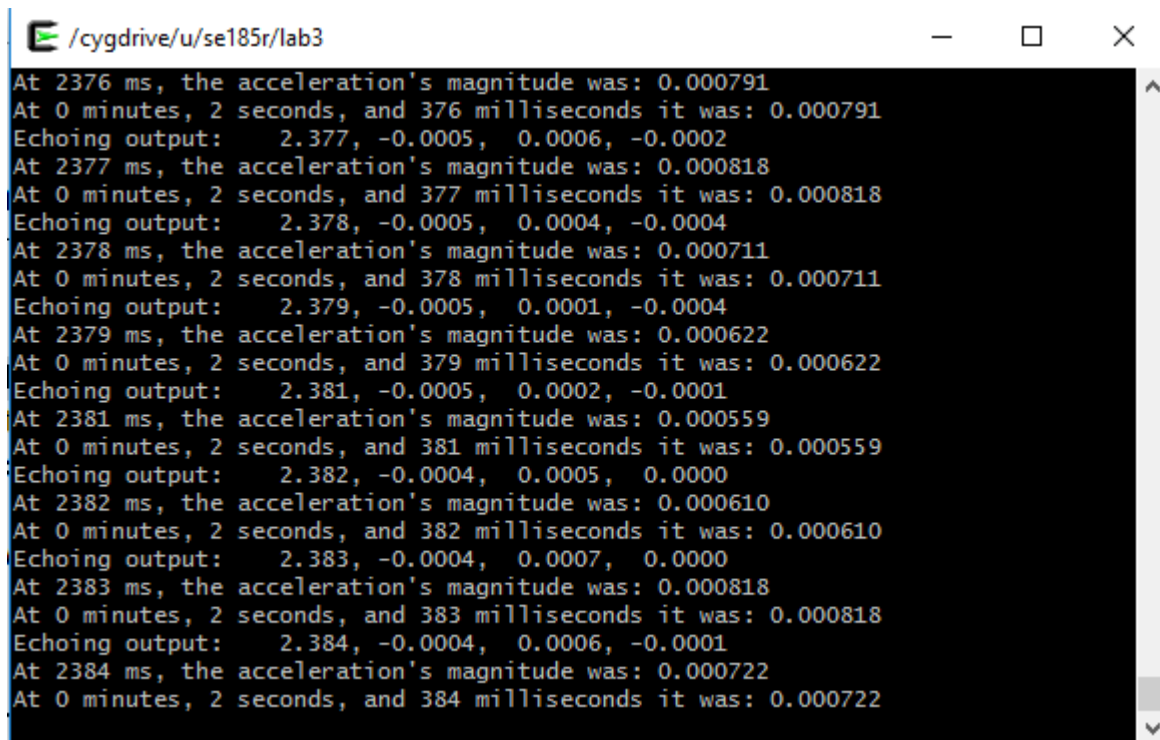
```
    }
```

```
    return 0;
```

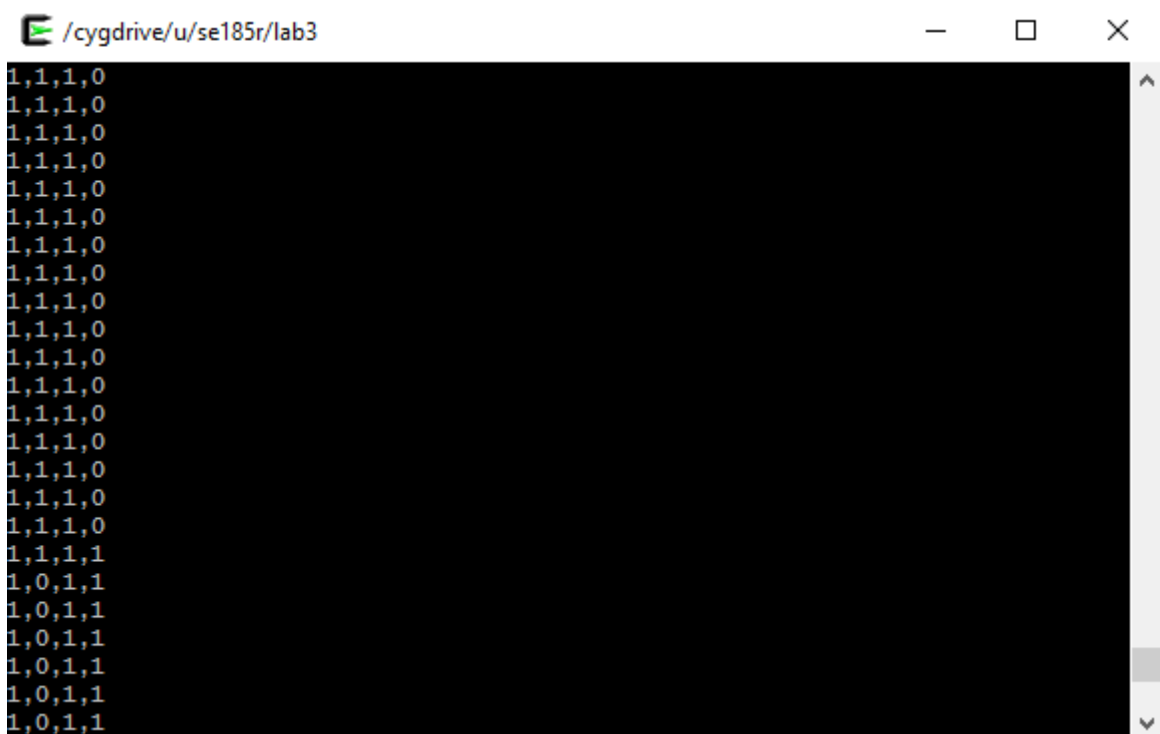
```
}
```

```
/* Put your functions here */
```

## Screen Shots



```
/cygdrive/u/se185r/lab3
At 2376 ms, the acceleration's magnitude was: 0.000791
At 0 minutes, 2 seconds, and 376 milliseconds it was: 0.000791
Echoing output: 2.377, -0.0005, 0.0006, -0.0002
At 2377 ms, the acceleration's magnitude was: 0.000818
At 0 minutes, 2 seconds, and 377 milliseconds it was: 0.000818
Echoing output: 2.378, -0.0005, 0.0004, -0.0004
At 2378 ms, the acceleration's magnitude was: 0.000711
At 0 minutes, 2 seconds, and 378 milliseconds it was: 0.000711
Echoing output: 2.379, -0.0005, 0.0001, -0.0004
At 2379 ms, the acceleration's magnitude was: 0.000622
At 0 minutes, 2 seconds, and 379 milliseconds it was: 0.000622
Echoing output: 2.381, -0.0005, 0.0002, -0.0001
At 2381 ms, the acceleration's magnitude was: 0.000559
At 0 minutes, 2 seconds, and 381 milliseconds it was: 0.000559
Echoing output: 2.382, -0.0004, 0.0005, 0.0000
At 2382 ms, the acceleration's magnitude was: 0.000610
At 0 minutes, 2 seconds, and 382 milliseconds it was: 0.000610
Echoing output: 2.383, -0.0004, 0.0007, 0.0000
At 2383 ms, the acceleration's magnitude was: 0.000818
At 0 minutes, 2 seconds, and 383 milliseconds it was: 0.000818
Echoing output: 2.384, -0.0004, 0.0006, -0.0001
At 2384 ms, the acceleration's magnitude was: 0.000722
At 0 minutes, 2 seconds, and 384 milliseconds it was: 0.000722
```



```
/cygdrive/u/se185r/lab3
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,0
1,1,1,1
1,0,1,1
1,0,1,1
1,0,1,1
1,0,1,1
1,0,1,1
1,0,1,1
1,0,1,1
```

```
Buttons being pressed: 2
Buttons being pressed: 2
Buttons being pressed: 2
Buttons being pressed: 2
Buttons being pressed: 2
Buttons being pressed: 2
Buttons being pressed: 2
Buttons being pressed: 2
Buttons being pressed: 2
Buttons being pressed: 3
Buttons being pressed: 3
Buttons being pressed: 3
Buttons being pressed: 3
Buttons being pressed: 3
Buttons being pressed: 3
Buttons being pressed: 3
Buttons being pressed: 3
Buttons being pressed: 3
Buttons being pressed: 3
Buttons being pressed: 3
Buttons being pressed: 3
Buttons being pressed: 3
Buttons being pressed: 3
```

FRONT

