INTRODUCTION TO FUNCTIONS

LAB 4

SECTION X

SUBMITTED BY:

NATHAN SHULL

SUBMISSION DATE:

SEPTEMBER 29, 2016

Problem

The problem for part 1 was to modify the code so that seconds was displayed in an 8 character area with 3 decimal digits of precision. Also, acceleration was to show in a 7 character area with 4 digits of precision. The problem for part 2 included creating a function for magnitude, but specifically after the main function. The purpose for part 3 was to change the time that the esplora outputted from milliseconds into minutes, seconds, and milliseconds. This was done to make it more readable. For part 4, the purpose was to write a code that took the input of buttons pressed on the esplora, and output how many buttons were being pressed at any given time.

Analysis

The problem in part 1 states that we must display seconds in an 8 character area with 3 decimals of precision. We must utilize digits in from of either the double or long floating that we have to call to accomplish this. In part 2, you have to put a prototype above main, so that it knows that there is a function coming later in the program. In part 3, we must make use of the operators we've learned to give us how many seconds remain after we take out minutes, and how many milliseconds remain after we've taken out seconds. Lastly, part 4 states that we must scan in the number of buttons being pressed on the esplora, and then print out that number.

Design

The design for part 1 was fairly simple. All I had to do to change the character area to 8 was insert an 8 after % and before lf. To go to 3 decimal places, I put an 8.3 instead of 8.

For part 1, I had to make a new function called "mag," that included three parameters (x, y, and z), and had to make sure to define the insert a prototype before main, as the function wasn't called until after main. The only requirement for the function mag was to square all three values, and then add them together. I just assigned this value to return when mag was called.

For part 3, I had to make use of the modulo operator in order to get remainders of minutes, seconds, and milliseconds. I made functions for each minutes, seconds, and milliseconds. I first divided the total milliseconds by 60,000 and assigned that vale to int minutes. I used 60,000 because that's how many milliseconds are in a minute. For seconds, I did t % 60,000, which would give us how many milliseconds are leftover, accounting for minutes being taken out. Then I divided that value by 1,000, which gives us the amount of seconds.

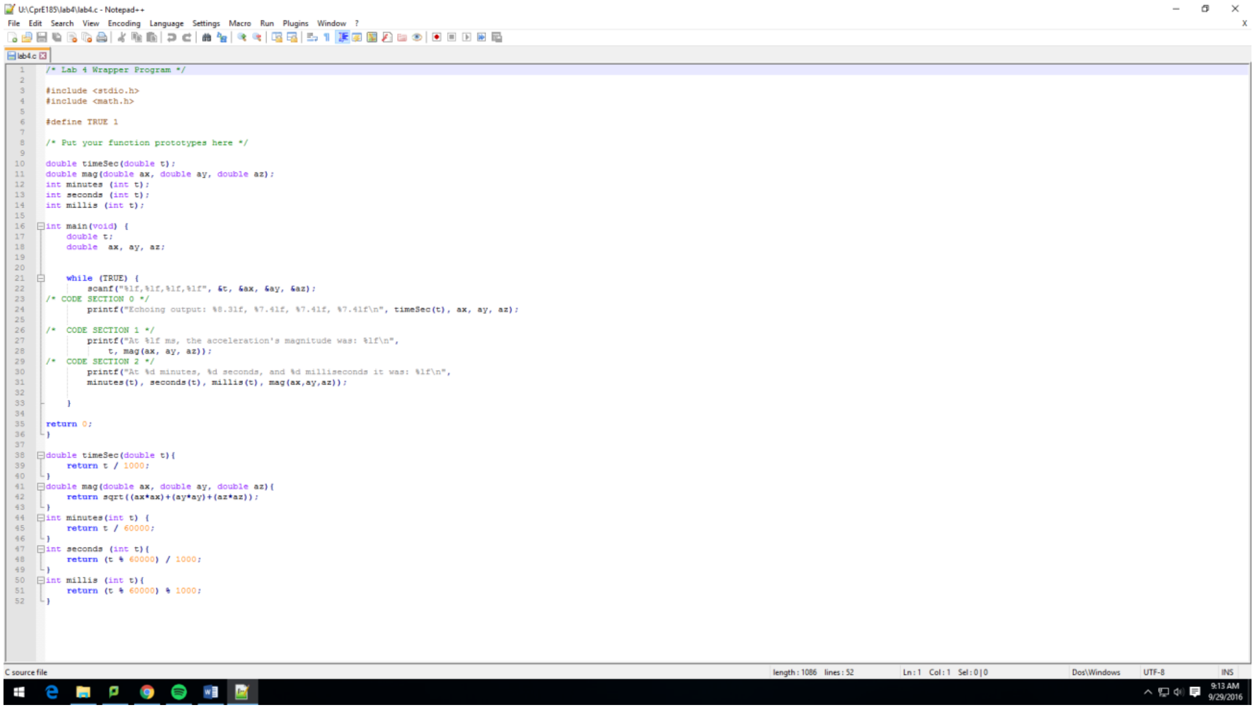
Finally, for milliseconds I again did t % 60,000 to give us how many seconds there were, and then with that value I used the modulo (%) 1000, which gives how many milliseconds are leftover. Problem four was a bit different, as I removed all previous code, and instead had to read in values of buttons being pressed. I first wrote a scanf( ) line that would take the input of the esplora and tell us how many buttons were being pressed. There are 6 buttons, so each button was assigned a variable a-f. I then ran a printf( ) line that added the number of buttons (1=pressed, 0=not pressed), and displayed it.

Testing

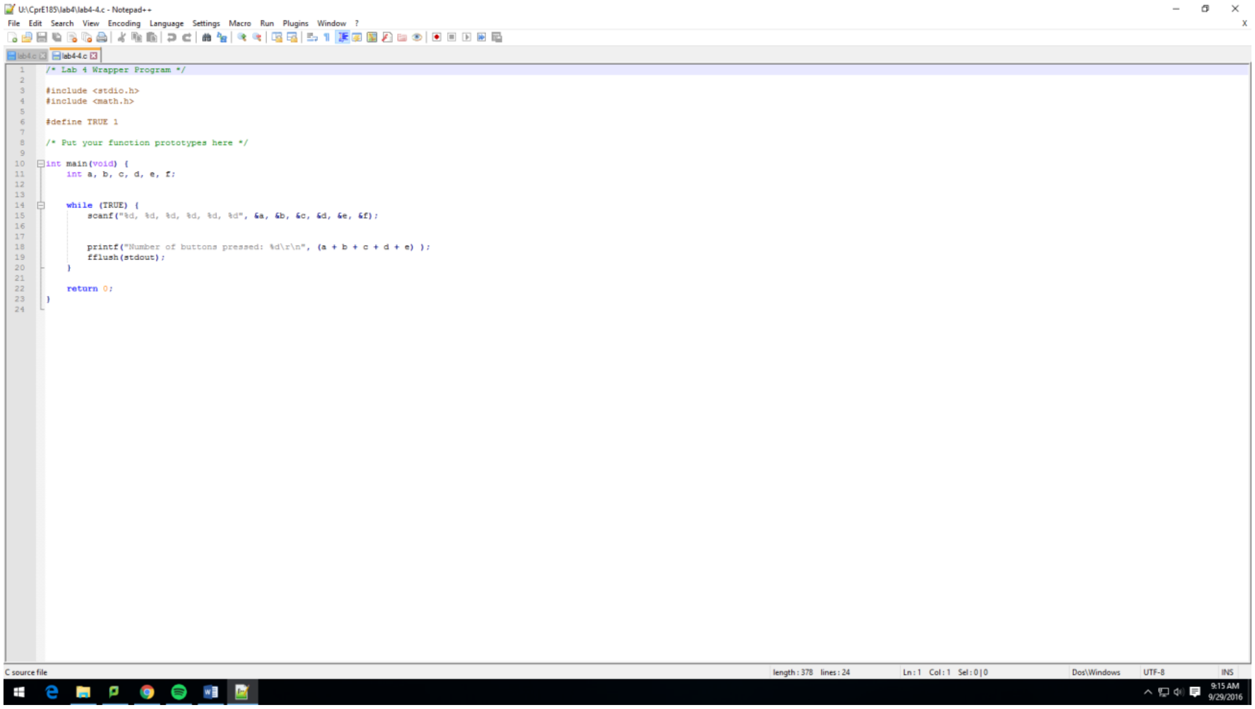
In order to verify that all my code was in face correct, I had to see its output in Cygwin. On my first run of the code, parts 1 and 2 were correct on the first try. Where I encountered problems was part 3. I couldn't correctly display the time in minutes, seconds, and milliseconds. I thought about it for a while, worked with classmates, and used the helps of the TA's. After playing with the code and seeing what the output was, I realized I wasn't using the modulo operator. Instead, I was using integer division, and not getting the remainders I needed. For part 4, I encountered one problem when testing. I was told only 5 button would return a value, so I only scanned in 5 variables. However, when I ran this code, number of buttons pressed always returned a value of 0. I added another variable to the input, and this quickly fixed the issue.

Comments

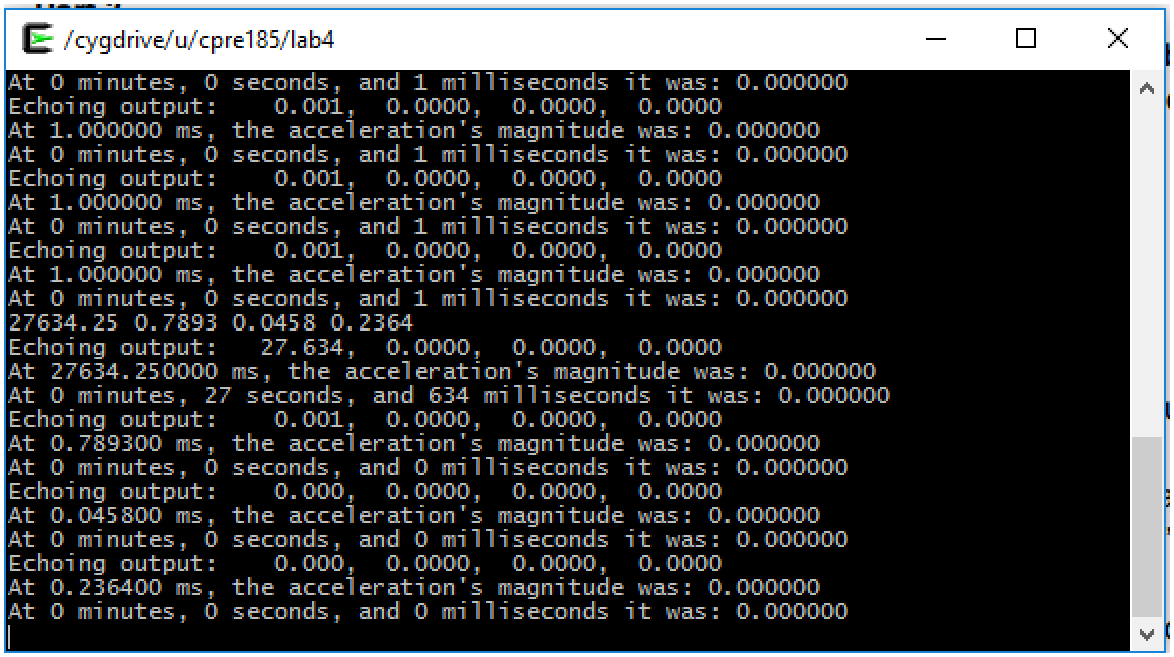
This lab really helped me become with functions, having to use prototypes, and also becoming familiar with the operators. I thought I had a good grasp on the operators, but I struggled using them to convert milliseconds into minutes, seconds, and milliseconds.


U:\CprE185\lab4\lab4.c - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
lab4.c
C source file
length : 1086 lines : 52
Ln : 1 Col : 1 Sel : 0 | 0
Dos\Windows
UTF-8
INS
9:13 AM
9/29/2016

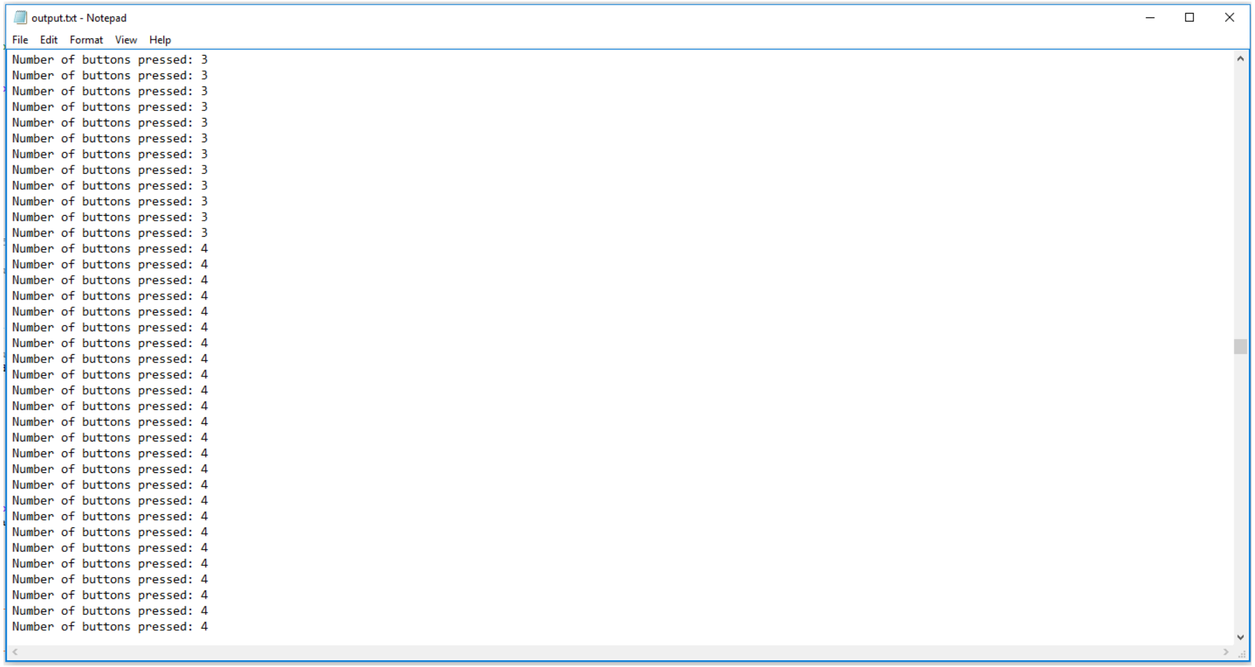
```
/* Lab 4 Wrapper Program */

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

#define TRUE 1

/* Put your function prototypes here */

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

int main(void) {
    double t;
    double  ax, ay, az;


    while (TRUE) {
        scanf("%lf,%lf,%lf,%lf", &t, &ax, &ay, &az);
/* CODE SECTION 0 */
        printf("Echoing output: %8.3lf, %7.4lf, %7.4lf, %7.4lf\n", timeSec(t), ax, ay, az);

/*  CODE SECTION 1 */
        printf("At %lf 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;
}

double timeSec(double t){
    return t / 1000;
}
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;
}
```


U:\CprE185\lab4\lab4-4.c - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
lab4.c
lab4-4.c
C source file
length : 378 lines : 24
Ln : 1 Col : 1 Sel : 0 | 0
Dos\Windows
UTF-8
INS
9:15 AM
9/29/2016

```c
/* Lab 4 Wrapper Program */

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

#define TRUE 1

/* Put your function prototypes here */

int main(void) {
    int a, b, c, d, e, f;


    while (TRUE) {
        scanf("%d, %d, %d, %d, %d, %d", &a, &b, &c, &d, &e, &f);


        printf("Number of buttons pressed: %d\r\n", (a + b + c + d + e) );
        fflush(stdout);
    }

    return 0;
}
```



```
At 0 minutes, 0 seconds, and 1 milliseconds it was: 0.000000
Echoing output:    0.001,  0.0000,  0.0000,  0.0000
At 1.000000 ms, the acceleration's magnitude was: 0.000000
At 0 minutes, 0 seconds, and 1 milliseconds it was: 0.000000
Echoing output:    0.001,  0.0000,  0.0000,  0.0000
At 1.000000 ms, the acceleration's magnitude was: 0.000000
At 0 minutes, 0 seconds, and 1 milliseconds it was: 0.000000
Echoing output:    0.001,  0.0000,  0.0000,  0.0000
At 1.000000 ms, the acceleration's magnitude was: 0.000000
At 0 minutes, 0 seconds, and 1 milliseconds it was: 0.000000
27634.25 0.7893 0.0458 0.2364
Echoing output:   27.634,  0.0000,  0.0000,  0.0000
At 27634.250000 ms, the acceleration's magnitude was: 0.000000
At 0 minutes, 27 seconds, and 634 milliseconds it was: 0.000000
Echoing output:    0.001,  0.0000,  0.0000,  0.0000
At 0.789300 ms, the acceleration's magnitude was: 0.000000
At 0 minutes, 0 seconds, and 0 milliseconds it was: 0.000000
Echoing output:    0.000,  0.0000,  0.0000,  0.0000
At 0.045800 ms, the acceleration's magnitude was: 0.000000
At 0 minutes, 0 seconds, and 0 milliseconds it was: 0.000000
Echoing output:    0.000,  0.0000,  0.0000,  0.0000
At 0.236400 ms, the acceleration's magnitude was: 0.000000
At 0 minutes, 0 seconds, and 0 milliseconds it was: 0.000000
```



```
Number of buttons pressed: 3
Number of buttons pressed: 3
Number of buttons pressed: 3
Number of buttons pressed: 3
Number of buttons pressed: 3
Number of buttons pressed: 3
Number of buttons pressed: 3
Number of buttons pressed: 3
Number of buttons pressed: 3
Number of buttons pressed: 3
Number of buttons pressed: 3
Number of buttons pressed: 3
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
Number of buttons pressed: 4
```