LAB REPORT

LAB 6 PART 1 & 2
SECTION C

SUBMITTED BY:

SCOTT VLASIC

SUBMISSION DATE:

MARCH 10, 2017

Problem:

There were two issues that I had to solved for this lab since there were two separate parts. For the first part of the lab I had to create a program that would detect the free fall of the esplora. I was supposed to print 4 lines, two for when the esplora was at rest, one for when it was falling, and a final one for when it had stopped falling. To do this I had to create while loops since on some of the lines there needed to be either "." or "!" following the print statements. Part 2 was very similar to part 1 except I had to include air resistance and calculate the difference from the original drop in percentage. I also had to modify my program to output the distance fallen using a different approach than in Part 1.

Analysis:

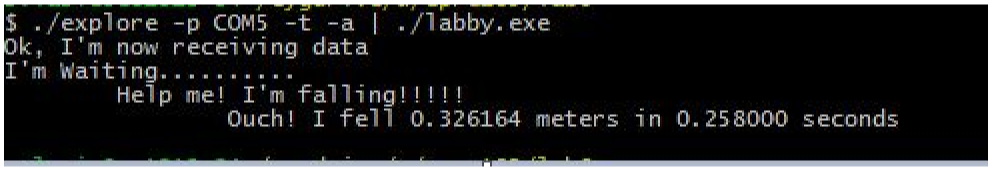
To begin this lab I started by opening lab5.c on Notepad++ so I could edit the code and being solving the problem. I copied over the magnitude function and close_to function from last week's lab as well as I needed them for this one. Then, I looked at the values on the esplora for when it was sitting and for when it began to fall and created loops for it. I also had to have the program stop when it wasn't falling and output the data that was recorded. The same was true for Part 2 except this time I had to also calculate the fall with air resistance and how it affected the fall of the esplora.

Design:

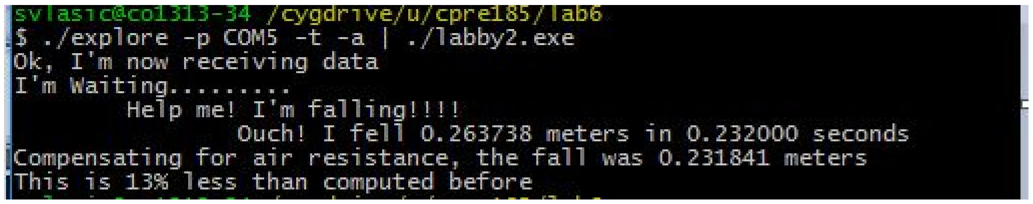
To begin solving this problem I had to realize what I needed to write in my while loops for the program. For example, for the line "I'm waiting", the number of periods should've corresponded to an arbitrary number of inputs from explore.exe. To do this I wrote a while loop that called the close_to function and looked to see if it was within the tolerance of the point I chose. Then I had it look at the mag function and if the value % 200 == 0, the program would print a "." because the esplora would be level waiting to be dropped. A similar design was used for the "Help, I'm falling line", as this needed to be followed by an arbitrary number of "!". I wrote the same sort of while loop but instead of % 200, I did when the value %20 == 0, the program would print a "!" as the esplora would be falling. Finally, to calculate the distance fallen and time elapsed, I wrote equations based on the variables I defined. The total time would be the (endtime-initial time)/1000.0 to convert from milliseconds to seconds and the distance would follow the equation 1.0/2.0*acceleration*time^2. Thus my distance equation was 1.0/2.0*9.8*(totaltime*totaltime). For Part 2, I had to modify the program from the previous week in order to calculate the wind resistance and percentage error. To do this I created a new variable, currentX, that would calculate the wind resistance based on the formula recommended in the lab description. Thus, the equation was equal to the beginningX+(currentVelocity *(endtime- previoustime)/1000.0. To get the percentage error, I calculated the (distancefallen-currentX)*100 / (currentX).

Testing:

To test this program, I had to go through many trials and errors. In testing part 1, I would run the program and make sure the "." and "!" were printing the correct ways I wanted them to and that the distance and time values being returned were feasible. Then, I ran the program using the sample data given to us in the lab description and saw if that matched up in the proper way I wanted it to. The process was very similar for part 2 of the lab. I also tested to see if the lines were producing what I wanted and looked at the percentage values being returned to see if they were feasible. I could test to see if the percentage was correct by taking the percent of the value of the air resistance and comparing it to the value of the esplora when it was dropped. I also tested this by using the sample data and seeing if it was matched up, which the TA's said was correct. Some output of my code is shown below.



```
$ ./explore -p COM5 -t -a | ./labby.exe
Ok, I'm now receiving data
I'm Waiting..........
        Help me! I'm falling!!!!!
                Ouch! I fell 0.326164 meters in 0.258000 seconds
```

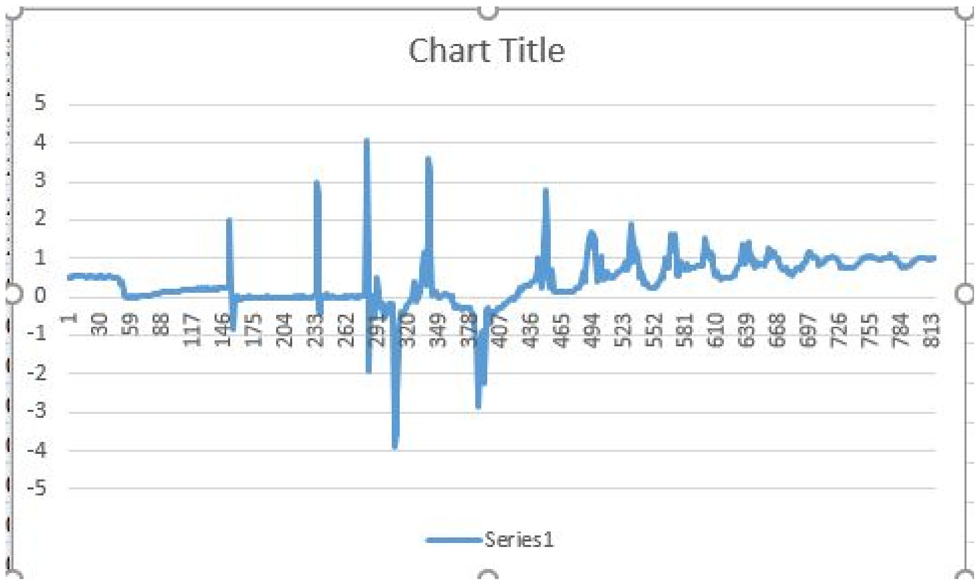


```
svlasic@co1313-34 /cygdrive/u/cpre185/lab6
$ ./explore -p COM5 -t -a | ./labby2.exe
Ok, I'm now receiving data
I'm Waiting.........
        Help me! I'm falling!!!!
                Ouch! I fell 0.263738 meters in 0.232000 seconds
Compensating for air resistance, the fall was 0.231841 meters
This is 13% less than computed before
```

Comments:

I also had to spend more time with the TA's working on both parts of this lab. I feel much more confident with while loops and finding errors in my code. For example at the beginning my distance and time values were in the high thousands and were negative as well which I knew wasn't correct. So I realized that I was dividing by the wrong number and that my tolerance and point values were off. Based on this I was able to correct my program and get the right values. When I dropped the esplora 5 different times, the values were quite close to each other but not exactly the same. I think the cause for this would be air resistance and that not every drop will be the same time at the same distance because of it. According to my code, the drop from the third floor railing to the bottom floor is about 9.023100 meters.

The differences in part 2 tended to be about 0.15 meters and around 20% less than computed before, giving or taking a few meters/percentages. From the third floor, the fall computing air resistance was 8.227160 meters and it was 9% less than before. An issue that arose for me in implementing Part 2 was creating the equation to calculate percentage. The major thing I forgot to do was divide it by the currentX value I had made.



This graph shows the freefall happening at the points where the graph spikes upwards. When the graph is at flat line, the esplora was sitting still. It hits the ground where the spike goes below the x-axis slightly. I used a tolerance of 0.5 since the esplora was at rest at about 1.0 so when it varied too much it was in freefall.

Implementation:

```
/* Lab 6 */
#include <stdio.h>
#include <math.h>


/* Put your function prototypes here */
double mag (double x, double y, double z);
int close_to(double tolerance, double point, double value);

int main(void) {
        int s;
        double  ax, ay, az;
        int d = 0, e = 0;
        double y = 1.0;
        int begintime;
        int endtime;
        double totaltime;
        double distancefell;



        printf("Ok, I'm now receiving data\n");
        printf("I'm Waiting");
        while(close_to(0.5,1.0, y)){
                scanf("%d,%lf,%lf,%lf", &s, &ax, &ay, &az); y = mag(ax, ay, az);{
                        d++;
                        fflush(stdout);
                        if((d % 200) == 0)
                                printf(".");
                }
        }
        begintime = s;
        printf("\n\tHelp me! I'm falling");
        while(close_to(0.5,0.0, y)){
                scanf("%d,%lf,%lf,%lf", &s, &ax, &ay, &az); y = mag(ax, ay, az);{
                        e++;
                        fflush(stdout);
```

```
                    if((e % 20) == 0)
                        printf("!");
        endtime = s;
            }
        }

            totaltime = ((endtime - begintime)/1000.0);
            distancefell = ((1.0/2.0)*9.8*(totaltime*totaltime));
            printf("\n\t\tOuch! I fell %lf meters in %lf seconds\n", distancefell, totaltime);

return 0;
}
/* Put your functions here */
double mag(double x, double y, double z)
{
        return sqrt(x*x + y*y + z*z);
}
int close_to(double tolerance, double point, double value)
{
        if(value >= (point - tolerance) && value <= (point + tolerance))
         return 1;
        else
         return 0;
}



/* Lab 6-2 */
#include <stdio.h>
#include <math.h>



/* Put your function prototypes here */
double mag (double x, double y, double z);
int close_to(double tolerance, double point, double value);

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

```
int d = 0, e = 0;
double y = 1.0;
int begintime;
int endtime;
double totaltime;
double distancefell;
double beginningX = 0.0, currentX=0;
double prevV = 0.0, currentV=0;
double prevT = 0.0;
int percentage;


printf("Ok, I'm now receiving data\n");
printf("I'm Waiting");
while(close_to(0.5,1.0, y)){
        scanf("%d,%lf,%lf,%lf", &s, &ax, &ay, &az);
        y = mag(ax, ay, az);
        {
                d++;
                fflush(stdout);
                if((d % 200) == 0)
                        printf(".");
        }
}
begintime = s;
printf("\n\tHelp me! I'm falling");
prevT=s;
while(close_to(0.5,0.0, y)){
        scanf("%d,%lf,%lf,%lf", &s, &ax, &ay, &az);
        y = mag(ax, ay, az);
                e++;

                fflush(stdout);
                if((e % 20) == 0)
                        printf("!");
        if (mag(ax, ay, az) > 1)
        {
                break;
        }
```

```
                endtime = s;
                currentV = prevV + ((9.8 - mag(ax, ay, az)*9.8) * ((endtime - prevT)/1000.0));
                currentX = beginningX + (currentV * ((endtime - prevT)/1000.0));
                prevT = s;
                prevV = currentV;
                beginningX = currentX;

        }

                totaltime = ((endtime - begintime)/1000.0);
                distancefell = ((1.0/2.0)*9.8*(totaltime*totaltime));
                printf("\n\t\tOuch! I fell %lf meters in %lf seconds\n", distancefell, totaltime);

                printf("Compensating for air resistance, the fall was %lf meters\n", currentX);
                percentage = (distancefell - currentX)* 100 / (currentX);
                printf("This is %d%% less than computed before", percentage);

return 0;
}

double mag(double x, double y, double z)
{
        return sqrt(x*x + y*y + z*z);
}
int close_to(double tolerance, double point, double value)
{
        if(value >= (point - tolerance) && value <= (point + tolerance))
         return 1;
        else
         return 0;
}
```