# ExDrop

## LAB 6

## SECTION G

SUBMITTED BY:

RYAN SAMUELSON

SUBMISSION DATE:

10/22/15

## Problem

For part one of the lab, the problem was using correlating the “!” and “.” With the corrispinding the time with the prime function. Deciding on the correct amount of acceptable time in between the marks was not that hard, and a simple while loop solved the problem with amounts of “!” and “.”.

For part two the problem was finding how to get the overall distance with X sub”I” and X sub”i-1”. for most of the week I thought it had to be calculated at the end of the falling portion of the code. Thanks to a helpful TI, I found out that it needed to be in with the code and then just add the partial distances together to get the total distance of the falling object.

## Analysis

For part one, I knew how to solve the amount of ! marks in my head but having to put in on screen and have it work was more difficult than I thought.

For part two, I spent the entire portion of class just trying to get a piece of code to work that would never work in the first place, but once I was told that the distance was accumulative and not a total, it made much more sense.

## Design

The design was simple have two while loops run, one after the other and then find the distance in the second while loop. But as I spend more time on the code, I made a bunch of attempts at the code that didn’t work, eventually I had to get a fresh start for part two. Having the percent after the while loop and the air resistance equations in the loop helped with the total distance because you just add the separate segments together.

## Testing

Testing was the key to this lab and constancy with the tests was curtail for part one. For part two it was less about testing to see if the code worked, and more about if it worked without hindering part one of the lab; while part two works independently. The percent was not that hard to find just total with air over total without air.

## Comments

Answers for part one:

1. My results for the 5 tests were very precise, the only thing that changed was the “.” In the waiting portion of the program.
2. 6 meters

Answers for part two:

1. The difference in the lab drops was much smaller than the drops from the third floor ot the ground floor.
2. 9.1 meters
3. Having part two run without interfering with part one

## LAB 6 PART 1

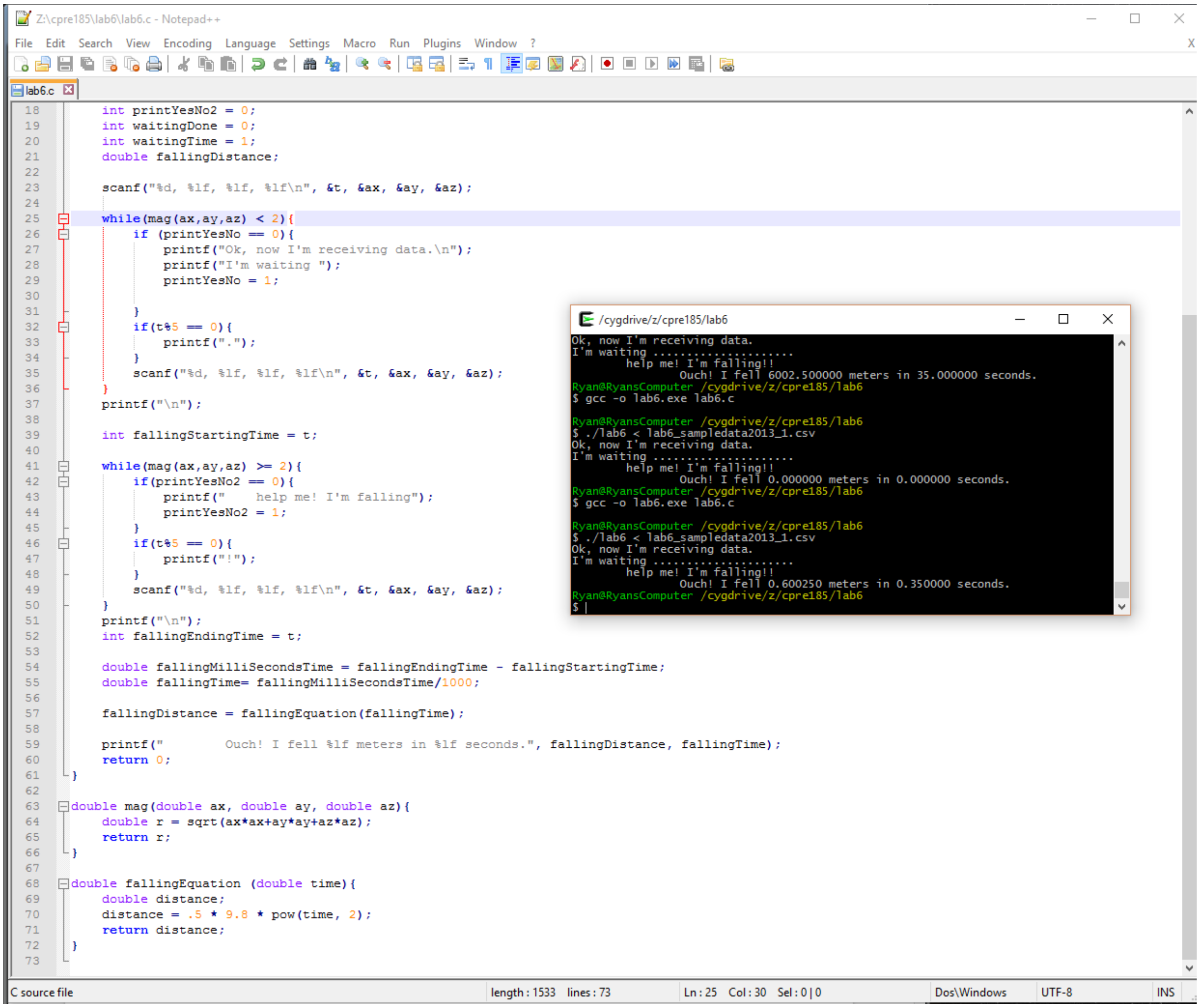
```
#include <stdio.h>
#include <math.h>

#define TRUE 1
#define FALSE 0

int close_to (double tolerance, double point, double value);
double mag(double ax, double ay, double az);
double fallingEquation (double time);

int main (){

    int t;
    double ax, ay, az;
    int printYesNo = 0;

    int printYesNo2 = 0;
    int waitingDone = 0;
    int waitingTime = 1;
    double fallingDistance;

    scanf("%d, %lf, %lf, %lf\n", &t, &ax, &ay, &az);


    while(mag(ax, ay, az) < 2){
        if (printYesNo == 0){
            printf("Ok, now I'm receiving data.\n");
            printf("I'm waiting ");
            printYesNo = 1;

        }
        if(t%10 == 0){
            printf(".");
        }
    scanf("%d, %lf, %lf, %lf\n", &t, &ax, &ay, &az);
    }
    printf("\n\n");

 int fallingStartingTime = t;

    while(mag(ax, ay, az) >= 2){
        if(printYesNo2 == 0){
            printf(" help me! I'm falling");
            printYesNo2 = 1;
        }
        if(t%10 == 0){
```

```
                        printf("!");
                }
                scanf("%d, %lf, %lf, %lf\n", &t, &ax, &ay, &az);
        }
        printf("\n/n");
        int fallingEndingTime = t;

        double fallingMilliSecondsTime = fallingEndingTime - fallingStartingTime;
        double fallingTime = fallingMilliSecondsTime / 1000;

        fallingDistance = fallingEquation(fallingTime);

        printf("                Ouch! I fell %lf meters in %lf seconds.\n\n", fallingDistance, fallingTime);

        return 0;
}

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

double fallingEquation (double time){
        double distance;
        distance = .5 * 9.8 * pow(time, 2);
        return distance;
}



int close_to (double tolerance, double point, double value){
        if(point > value){
                if((point - value) < tolerance){
                        return TRUE;
                }
                else {
                        return FALSE;
                }
        }
        else {
                if((value - point) < tolerance){
                        return TRUE;
                }
                else {
                        return FALSE;
                }
        }
}
```

# Lab 6 Part 2

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

#define TRUE 1
#define FALSE 0

int close_to (double tolerance, double point, double value);
double mag(double ax, double ay, double az);
double fallingEquation (double time);
double vi(double v_minus_1, double g, double acc, double t, double t_minus1);
double xi(double x_minus_1, double vi, double t, double t_minus1);


int main (){

        int t;
        int t_minus_1 = 0;
        int printYesNo = 0;
        int printYesNo2 = 0;
        int waitingDone = 0;
        int waitingTime = 1;
        double gravity = 9.8;
        double fallingDistance;
        double xTotal = 0;
        double tmp_x_minus_1 = 0;
        double tmp_v_minus_1 = 0;
        double ax, ay, az;

        scanf("%d, %lf, %lf, %lf\n", &t, &ax, &ay, &az);


        while(mag(ax, ay, az) < 2){
                if (printYesNo == 0){
                        printf("Ok, now I'm receiving data.\n");
                        printf("I'm waiting ");
                        printYesNo = 1;

                }
                if(t%10 == 0){
                        printf(".");
                }
                t_minus_1 = t;
        scanf("%d, %lf, %lf, %lf\n", &t, &ax, &ay, &az);
        }
        printf("\n\n");

   int fallingStartingTime = t;

        while(mag(ax, ay, az) >= 2){
                if(printYesNo2 == 0){
                        printf(" help me! I'm falling");
                        printYesNo2 = 1;
                }
                if(t%10 == 0){
                        printf("!");
                }
                t_minus_1 = t;
                scanf("%d, %lf, %lf, %lf\n", &t, &ax, &ay, &az);

                double tmp_vi = vi(tmp_v_minus_1, gravity, mag(ax, ay, az), t, t_minus_1);
                tmp_v_minus_1 = tmp_vi;
                double tmp_xi = xi(tmp_x_minus_1, tmp_vi, t, t_minus_1);
                tmp_x_minus_1 = tmp_xi;
                xTotal += tmp_xi;
        }
        printf("\n\n");
        int fallingEndingTime = t;

        double fallingMilliSecondsTime = fallingEndingTime - fallingStartingTime;
        double fallingTime = fallingMilliSecondsTime / 1000;

        fallingDistance = fallingEquation(fallingTime);

        printf("        Ouch! I fell %lf meters in %lf seconds.\n\n", fallingDistance, fallingTime);
        printf("Compensating for air resistance, the fall was %lf meters.\n", xTotal);
        double percentFallingAir = (xTotal / fallingDistance) * 100;
        printf("This is %d% less than computed before.", percentFallingAir);
        return 0;
}

double vi(double v_minus_1, double g, double acc, double t, double t_minus1) {
   return v_minus_1 + (g - acc * 9.8) * (t - t_minus1);
}

double xi(double x_minus_1, double vi, double t, double t_minus1) {
   return x_minus_1 + vi * (t - t_minus1);
}


double mag(double ax, double ay, double az){
        double r = sqrt(ax*ax+ay*ay+az*az);
```

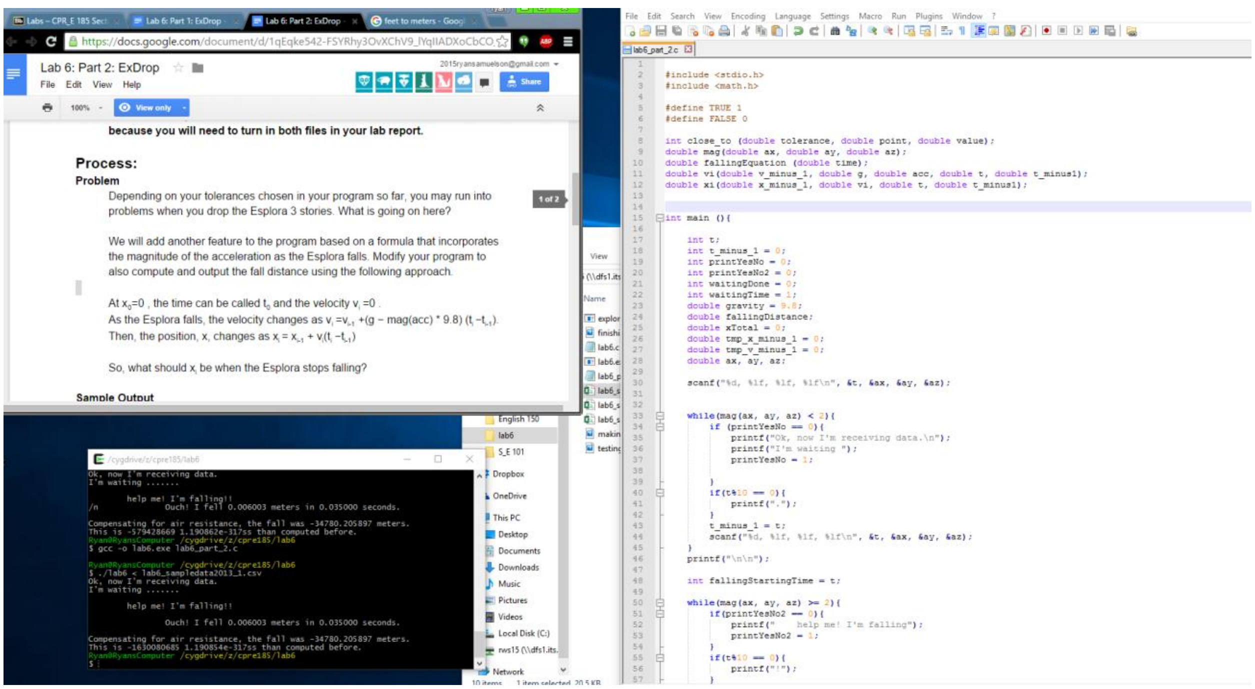
```
    return r;
}

double fallingEquation (double time){
            double distance;
            distance = .5 * 9.8 * pow(time, 2);
            return distance;
}
```



Z:\cpre185\lab6\lab6.c - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
lab6.c
/cygdrive/z/cpre185/lab6
C source file
length : 1533 lines : 73
Ln : 25 Col : 30 Sel : 0 | 0
Dos\Windows
UTF-8
INS

```c
    int printYesNo2 = 0;
    int waitingDone = 0;
    int waitingTime = 1;
    double fallingDistance;

    scanf("%d, %lf, %lf, %lf\n", &t, &ax, &ay, &az);

    while(mag(ax,ay,az) < 2){
        if (printYesNo == 0){
            printf("Ok, now I'm receiving data.\n");
            printf("I'm waiting ");
            printYesNo = 1;

        }
        if(t%5 == 0){
            printf(".");
        }
        scanf("%d, %lf, %lf, %lf\n", &t, &ax, &ay, &az);
    }
    printf("\n");

    int fallingStartingTime = t;

    while(mag(ax,ay,az) >= 2){
        if(printYesNo2 == 0){
            printf("    help me! I'm falling");
            printYesNo2 = 1;
        }
        if(t%5 == 0){
            printf("!");
        }
        scanf("%d, %lf, %lf, %lf\n", &t, &ax, &ay, &az);
    }
    printf("\n");
    int fallingEndingTime = t;

    double fallingMilliSecondsTime = fallingEndingTime - fallingStartingTime;
    double fallingTime= fallingMilliSecondsTime/1000;

    fallingDistance = fallingEquation(fallingTime);

    printf("        Ouch! I fell %lf meters in %lf seconds.", fallingDistance, fallingTime);
    return 0;
}

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

double fallingEquation (double time){
    double distance;
    distance = .5 * 9.8 * pow(time, 2);
    return distance;
}
```

```
Ok, now I'm receiving data.
I'm waiting .....................
       help me! I'm falling!!
              Ouch! I fell 6002.500000 meters in 35.000000 seconds.
Ryan@RyansComputer /cygdrive/z/cpre185/lab6
$ gcc -o lab6.exe lab6.c

Ryan@RyansComputer /cygdrive/z/cpre185/lab6
$ ./lab6 < lab6_sampledata2013_1.csv
Ok, now I'm receiving data.
I'm waiting .....................
       help me! I'm falling!!
              Ouch! I fell 0.000000 meters in 0.000000 seconds.
Ryan@RyansComputer /cygdrive/z/cpre185/lab6
$ gcc -o lab6.exe lab6.c

Ryan@RyansComputer /cygdrive/z/cpre185/lab6
$ ./lab6 < lab6_sampledata2013_1.csv
Ok, now I'm receiving data.
I'm waiting .....................
       help me! I'm falling!!
              Ouch! I fell 0.600250 meters in 0.350000 seconds.
Ryan@RyansComputer /cygdrive/z/cpre185/lab6
$ |
```



**because you will need to turn in both files in your lab report.**

## Process:

### Problem

Depending on your tolerances chosen in your program so far, you may run into problems when you drop the Esplora 3 stories. What is going on here?

We will add another feature to the program based on a formula that incorporates the magnitude of the acceleration as the Esplora falls. Modify your program to also compute and output the fall distance using the following approach.

At $x_0$=0 , the time can be called $t_0$ and the velocity $v_i$ =0 .
As the Esplora falls, the velocity changes as $v_i$ =$v_{i-1}$ +(g − mag(acc) * 9.8) ($t_i$ −$t_{i-1}$).
Then, the position, x, changes as $x_i$ = $x_{i-1}$ + $v_i$($t_i$ −$t_{i-1}$)

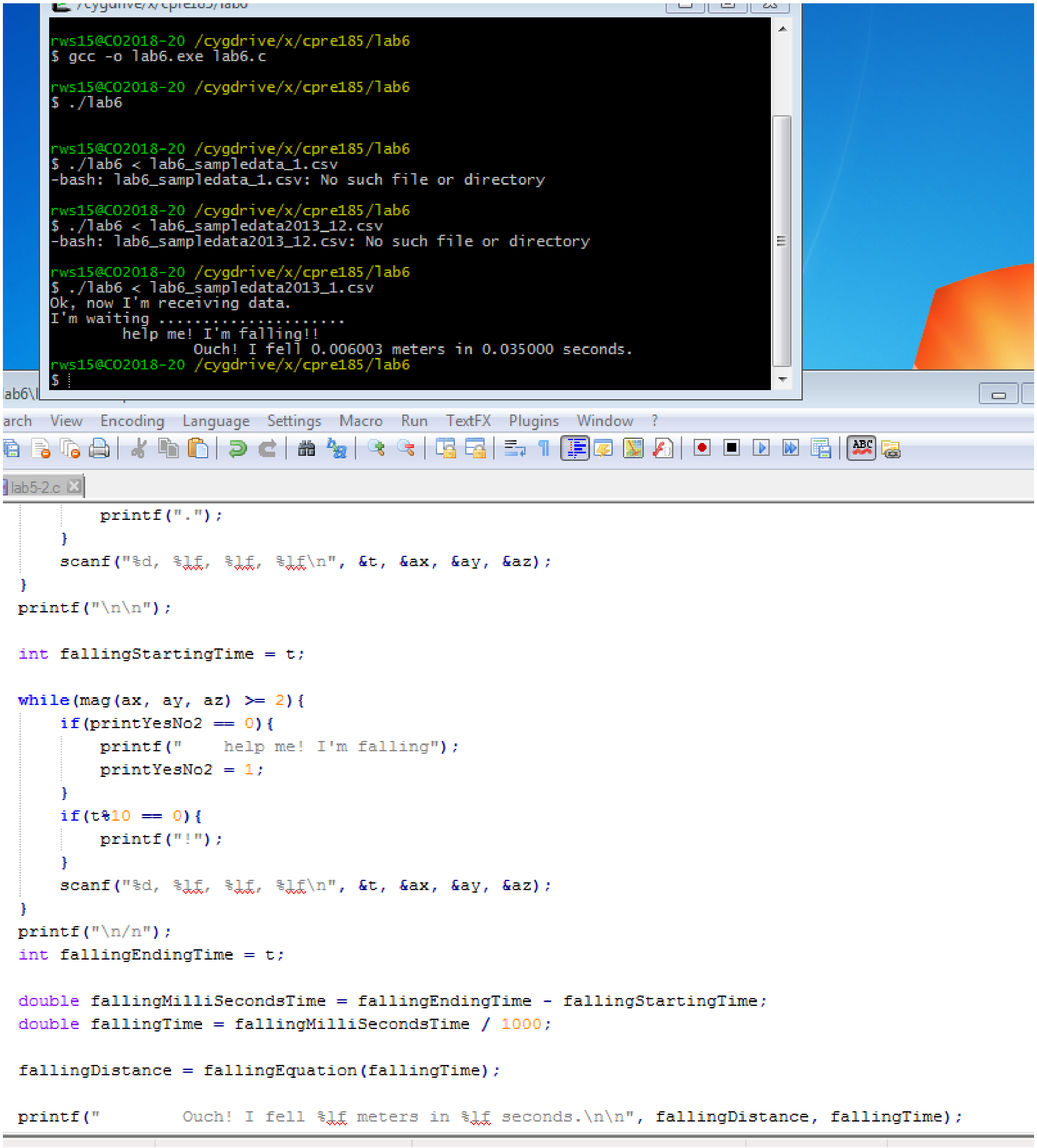
So, what should $x_i$ be when the Esplora stops falling?

### Sample Output

/cygdrive/z/cpre185/lab6

```
Ok, now I'm receiving data.
I'm waiting .......

        help me! I'm falling!!
/n                  Ouch! I fell 0.006003 meters in 0.035000 seconds.

Compensating for air resistance, the fall was -34780.205897 meters.
This is -579428669 1.190862e-317ss than computed before.
Ryan@RyansComputer /cygdrive/z/cpre185/lab6
$ gcc -o lab6.exe lab6_part_2.c

Ryan@RyansComputer /cygdrive/z/cpre185/lab6
$ ./lab6 < lab6_sampledata2013_1.csv
Ok, now I'm receiving data.
I'm waiting .......

        help me! I'm falling!!

                    Ouch! I fell 0.006003 meters in 0.035000 seconds.

Compensating for air resistance, the fall was -34780.205897 meters.
This is -1630080685 1.190854e-317ss than computed before.
Ryan@RyansComputer /cygdrive/z/cpre185/lab6
$
```

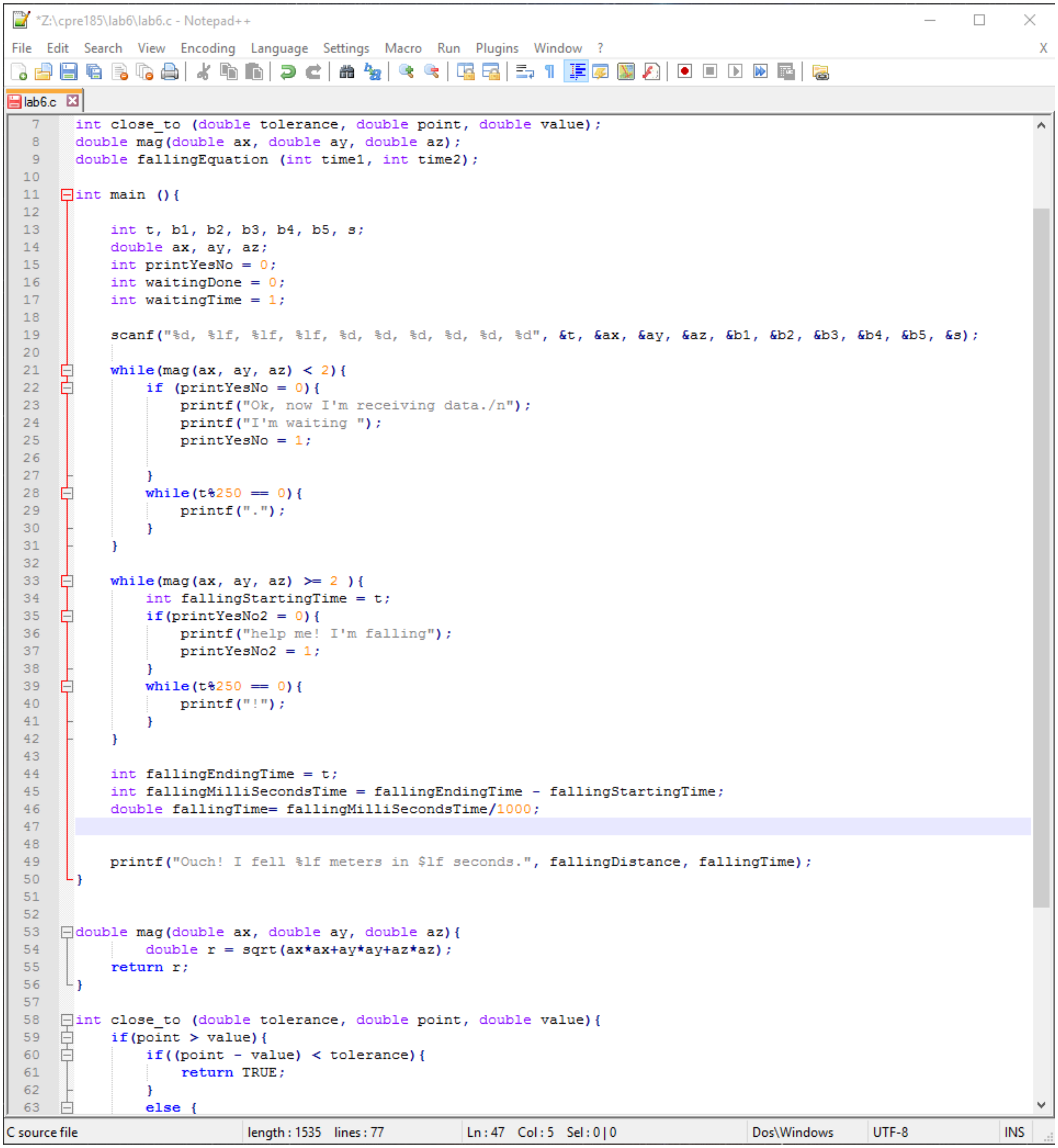
```c

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

#define TRUE 1
#define FALSE 0

int close_to (double tolerance, double point, double value);
double mag(double ax, double ay, double az);
double fallingEquation (double time);
double vi(double v_minus_1, double g, double acc, double t, double t_minus1);
double xi(double x_minus_1, double vi, double t, double t_minus1);


int main (){

    int t;
    int t_minus_1 = 0;
    int printYesNo = 0;
    int printYesNo2 = 0;
    int waitingDone = 0;
    int waitingTime = 1;
    double gravity = 9.8;
    double fallingDistance;
    double xTotal = 0;
    double tmp_x_minus_1 = 0;
    double tmp_v_minus_1 = 0;
    double ax, ay, az;

    scanf("%d, %lf, %lf, %lf\n", &t, &ax, &ay, &az);


    while(mag(ax, ay, az) < 2){
        if (printYesNo == 0){
            printf("Ok, now I'm receiving data.\n");
            printf("I'm waiting ");
            printYesNo = 1;

        }
        if(t%10 == 0){
            printf(".");
        }
        t_minus_1 = t;
        scanf("%d, %lf, %lf, %lf\n", &t, &ax, &ay, &az);
    }
    printf("\n\n");

    int fallingStartingTime = t;

    while(mag(ax, ay, az) >= 2){
        if(printYesNo2 == 0){
            printf("    help me! I'm falling");
            printYesNo2 = 1;
        }
        if(t%10 == 0){
            printf("!");
        }
```



/cygdrive/x/cpre185/lab6
lab6\l
arch View Encoding Language Settings Macro Run TextFX Plugins Window ?
lab5-2.c

```
rws15@CO2018-20 /cygdrive/x/cpre185/lab6
$ gcc -o lab6.exe lab6.c

rws15@CO2018-20 /cygdrive/x/cpre185/lab6
$ ./lab6


rws15@CO2018-20 /cygdrive/x/cpre185/lab6
$ ./lab6 < lab6_sampledata_1.csv
-bash: lab6_sampledata_1.csv: No such file or directory

rws15@CO2018-20 /cygdrive/x/cpre185/lab6
$ ./lab6 < lab6_sampledata2013_12.csv
-bash: lab6_sampledata2013_12.csv: No such file or directory

rws15@CO2018-20 /cygdrive/x/cpre185/lab6
$ ./lab6 < lab6_sampledata2013_1.csv
Ok, now I'm receiving data.
I'm waiting ......................
        help me! I'm falling!!
                Ouch! I fell 0.006003 meters in 0.035000 seconds.
rws15@CO2018-20 /cygdrive/x/cpre185/lab6
$
```

```c
        printf(".");
    }
    scanf("%d, %lf, %lf, %lf\n", &t, &ax, &ay, &az);
}
printf("\n\n");

int fallingStartingTime = t;

while(mag(ax, ay, az) >= 2){
    if(printYesNo2 == 0){
        printf("    help me! I'm falling");
        printYesNo2 = 1;
    }
    if(t%10 == 0){
        printf("!");
    }
    scanf("%d, %lf, %lf, %lf\n", &t, &ax, &ay, &az);
}
printf("\n/n");
int fallingEndingTime = t;

double fallingMilliSecondsTime = fallingEndingTime - fallingStartingTime;
double fallingTime = fallingMilliSecondsTime / 1000;

fallingDistance = fallingEquation(fallingTime);

printf("        Ouch! I fell %lf meters in %lf seconds.\n\n", fallingDistance, fallingTime);
```



```c
int close_to (double tolerance, double point, double value);
double mag(double ax, double ay, double az);
double fallingEquation (int time1, int time2);

int main (){

    int t, b1, b2, b3, b4, b5, s;
    double ax, ay, az;
    int printYesNo = 0;
    int waitingDone = 0;
    int waitingTime = 1;

    scanf("%d, %lf, %lf, %lf, %d, %d, %d, %d, %d, %d", &t, &ax, &ay, &az, &b1, &b2, &b3, &b4, &b5, &s);

    while(mag(ax, ay, az) < 2){
        if (printYesNo = 0){
            printf("Ok, now I'm receiving data./n");
            printf("I'm waiting ");
            printYesNo = 1;

        }
        while(t%250 == 0){
            printf(".");
        }
    }

    while(mag(ax, ay, az) >= 2 ){
        int fallingStartingTime = t;
        if(printYesNo2 = 0){
            printf("help me! I'm falling");
            printYesNo2 = 1;
        }
        while(t%250 == 0){
            printf("!");
        }
    }

    int fallingEndingTime = t;
    int fallingMilliSecondsTime = fallingEndingTime - fallingStartingTime;
    double fallingTime= fallingMilliSecondsTime/1000;


    printf("Ouch! I fell %lf meters in $lf seconds.", fallingDistance, fallingTime);
}


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

int close_to (double tolerance, double point, double value){
    if(point > value){
        if((point - value) < tolerance){
            return TRUE;
        }
        else {
```