Laboratory Report

LABORATORY 5 – REPORT

**Student name: Adeyemo Sobowale**

## Student ID: 18359056

**Programme: ECE**

|  |  |
| --- | --- |
| *I hereby declare that the attached submission is all my own work, that it has not previously been submitted for assessment, and that I have not knowingly allowed it to be used by another student. I understand that deceiving or attempting to deceive examiners by passing off the work of another as one's own is not permitted. I also understand that using another's student’s work or knowingly allowing another student to use my work is against the University regulations and that doing so will result in loss of marks and possible disciplinary proceedings.* | |
| **Signed:** **Adeyemo Sobowale** | **Date:** 7th March **2019** |

**Problem:**

Task 1: write a program that reads the initial height, initial vertical velocity and the initial horizontal velocity. Then the points on the graph were printed.

Task 2: within this task I had to create two additional functions within a program and in addition to this I had to make my program smaller and more manageable. Two of these functions should be file related functions.

**Plan:**

Task 1:

Create the file on textpad

Identifying and using the correct function

Correctly save and name the file

Using the Borland C++ Complier

Compile the file

Executing the file

Task 2:

Create the file on textpad

Counting the spaces to create the pattern on the loop

Correctly save and name the file

Using the Borland C++ Complier

Compile the file

Executing the file

Task 3:

Create the file on textpad

Spacing my initials correctly to get the desired shapes

Correctly save and name the file

Using the Borland C++ Complier

Compile the file

Executing the file

**Development:**

Task 1: I used the required libraries and used float for the variables required: float h, v, x, t, y, Initial\_h;. Then I used printf and scanf commands to get the values for the initial height, horizontal velocity and vertical velocity. Next, I printed the information in a table format using \t. next I used a loop for the t value so it can be printed in the table and the function used to get the values was stated below. And if statement was used so that the function stops when y<=0. Then I printed the values with three decimal points. T=t+0.05 was used to get several decimal points and lastly a thank you message was printed.

Task 2: for task 2 I made the information smaller by reducing the amount of decimal points required so 7 printed in the first task wouldn’t be printed as 7.0000. next I made the program print the information within the main function within a text file called results. FILE\*p; was used to declare the use of this function. Fflush was used to buffer data and fclose was used to close file opened. The text at the start of the program is the third function required but I made it not to print anything.

**Testing:**

Most of my testing came from learning new function to use within the second task. Any errors that were made were due to misplaced brackets or semi colons within the code.

**Conclusion:** a program was written that reads the initial height, initial vertical velocity and the initial horizontal velocity. Then the points on the graph were plotted on excel.

Then I created two additional functions within a program and in addition to this I had to make my program smaller and more manageable. Two of these functions were file related and the last wasn’t related to files.

**Code:**

Task 1:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main(void)

{

float h, v, x, y, t, In\_h;

printf("Welcome to my projectile motion calculator!\n");

printf("\nEnter initial height:\n");

scanf("%f", &In\_h);

printf("Enter initial horizontal velocity:\n");

scanf("%f", &h);

printf("Enter Initial Vertical Velocity:\n");

scanf("%f", &v);

printf("\nTime\t\tX Distance\tY Distance\n");

for (t= 0; t <= 5; t)

{

x = h\*t ;

y = In\_h+((v\*t)+(t\*t)\*(-9.81));

if (y <= 0)

{

break;

}

printf("%1.3f\t\t%3.3f\t\t%3.3f \n\n", t, x, y);

t=t+0.05;

}

printf("\nGoodbye from the most Advanced Projectile Motion Calculator!!\n");

return(0) ;

}

Output:

Welcome to my projectile motion calculator!

Enter initial height:

7

Enter initial horizontal velocity:

9

Enter Initial Vertical Velocity:

5

Time X Distance Y Distance

0.000 0.000 7.000

0.050 0.450 7.225

0.100 0.900 7.402

0.150 1.350 7.529

0.200 1.800 7.608

0.250 2.250 7.637

0.300 2.700 7.617

0.350 3.150 7.548

0.400 3.600 7.430

0.450 4.050 7.263

0.500 4.500 7.047

0.550 4.950 6.782

0.600 5.400 6.468

0.650 5.850 6.105

0.700 6.300 5.693

0.750 6.750 5.232

0.800 7.200 4.722

0.850 7.650 4.162

0.900 8.100 3.554

0.950 8.550 2.896

1.000 9.000 2.190

1.050 9.450 1.434

1.100 9.900 0.630

Thank you for using my projectile motion calculator!

Task 2:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

float vx(float h, float t);

float vy(float In\_h, float v, float t);

void welcome();

float vx(float h, float t)

{

return(h\*t);

}

float vy(float In\_h, float v, float t)

{

return (In\_h+((v\*t)+(t\*t)\*(-9.81)));

}

void welcome()

{

printf("Welcome to my projectile motion calculator!\n");

}

int main()

{

float In\_h, h, v, x, y, t;

FILE\* p;

welcome();

printf("\nEnter Initial height:\n");

scanf("%f", &In\_h);

printf("Enter Initial Horizontal Velocity?\n");

scanf("%f", &h);

printf("Enter Initial Vertical Velocity?\n");

scanf("%f", &v);

p = fopen("hresults.txt","w");

printf("\nTime\t\tX Distance\tY Distance\n", t, x, y);

fprintf(p,"\nTime\t\tX Distance\tY Distance\n", t, x, y);

t = 0;

do

{

x = vx(h, t);

y = vy(In\_h, v, t);

if (y <= 0)

{

break;

}

printf("%1.2f\t\t%3.2f\t\t%4.3f \n\n", t, x, y);

fprintf(p,"%1.2f\t\t%3.2f\t\t%4.3f \n\n", t, x, y);

t+=0.05;

}

while(y > 0);

fflush(p);

fclose(p);

printf("\nThank you for using my motion calculator!\n");

return(0) ;

}

Output:

Time X Distance Y Distance

0.00 0.00 7.000

0.05 0.45 7.225

0.10 0.90 7.402

0.15 1.35 7.529

0.20 1.80 7.608

0.25 2.25 7.637

0.30 2.70 7.617

0.35 3.15 7.548

0.40 3.60 7.430

0.45 4.05 7.263

0.50 4.50 7.047

0.55 4.95 6.782

0.60 5.40 6.468

0.65 5.85 6.105

0.70 6.30 5.693

0.75 6.75 5.232

0.80 7.20 4.722

0.85 7.65 4.162

0.90 8.10 3.554

0.95 8.55 2.896

1.00 9.00 2.190

1.05 9.45 1.434

1.10 9.90 0.630