Course: Programming Fundamental – ENSF 337

Lab #: Lab 2  
Instructor: Mansouri Habibabadi  
Student Name: Drew Hengehold

Lab Section: B01

Date submitted: Sept 28, 2022

UCID: 30151823

Lab1\_Excersize A:

/\*

\* lab2exe\_A.c

\* Created by Mahmood Moussavi

\* Completed by: Drew Hengehold

\* Completed on: 24 September 2022

\*/

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

const double G = 9.8; /\* gravitation acceleration 9.8 m/s^2 \*/

const double PI = 3.141592654;

void create\_table(double v);

double Projectile\_travel\_time(double a, double v);

double Projectile\_travel\_distance(double a, double v);

double degree\_to\_radian(double d);

int main(void)

{

int n;

double velocity;

printf ("Please enter the velocity at which the projectile is launched (m/sec): ");

n = scanf("%lf" ,&velocity);

//This checks to make sure you entered properly in the scan

if(n != 1)

{

printf("Invlid input. Bye...");

exit(1);

}

while (velocity < 0 )

{

printf ("please enter a positive number for velocity:");

n = scanf("%lf", &velocity);

if(n != 1)

{

printf("Invlid input. Bye...");

exit(1);

}

}

create\_table(velocity);

return 0;

}

double Projectile\_travel\_time(double a, double v)

{

return 2\*v\*sin(a)/G;

}

double Projectile\_travel\_distance(double a, double v)

{

return pow(v, 2)/G\*sin(2\*a);

}

double degree\_to\_radian(double d)

{

return d\*PI/180;

}

void create\_table(double v)

{

printf("Angle\t\tt\t\t\td\n(deg\t\t(sec)\t\t(m)\n");

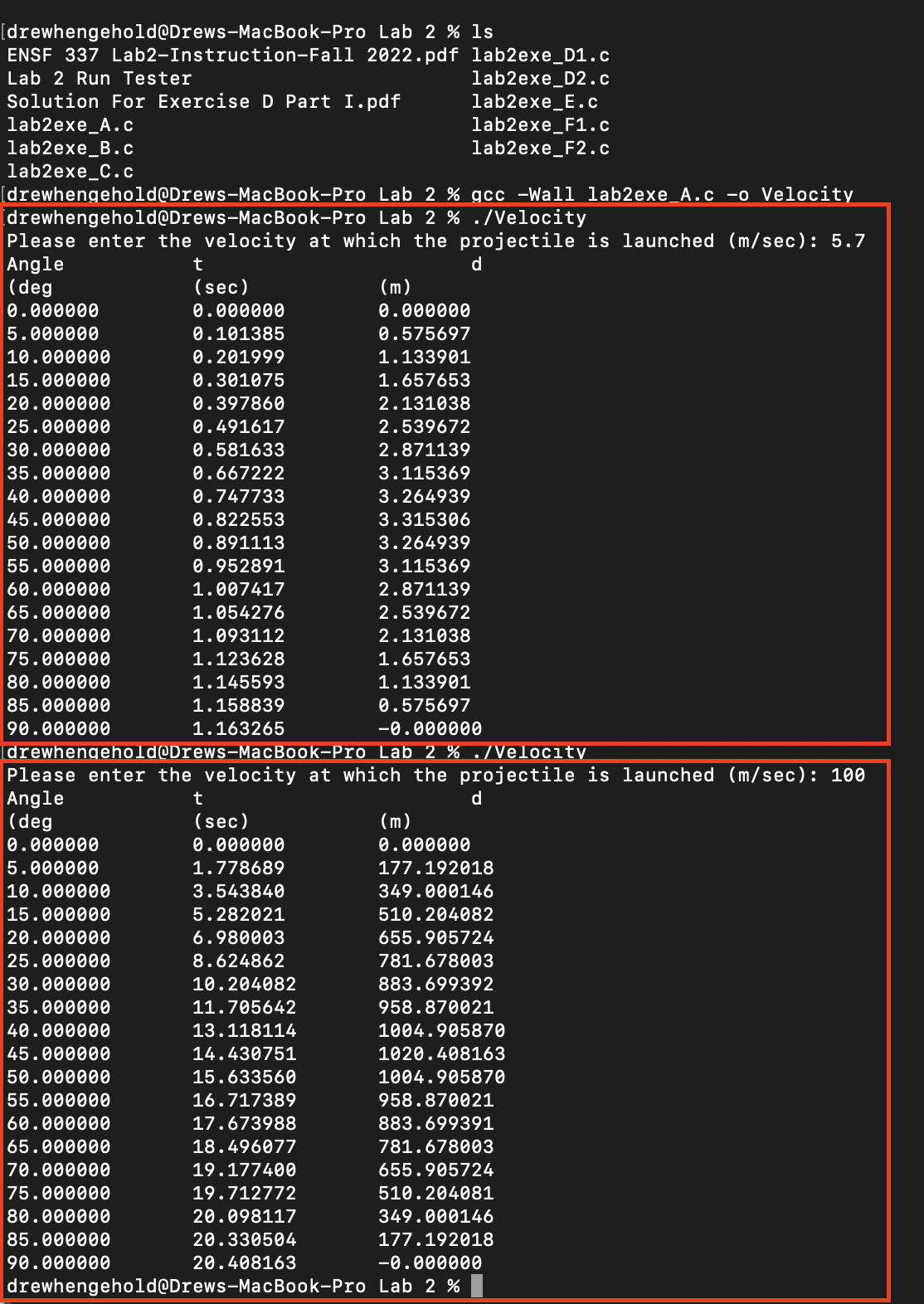
for(double i=0.0; i<=90.0; i+=5)

{

printf("%lf\t%lf\t%lf\n", i, Projectile\_travel\_time(degree\_to\_radian(i),v), Projectile\_travel\_distance(degree\_to\_radian(i), v));

}

}



**Outputs screenshot Exercise A**

Lab1\_Excersize B:





Lab1\_Excersize C:



Lab1\_Excersize D:

2

Lab1\_Excersize E:

**SOURCE CODE:**

/\*

\*

\* lab2exe\_E.c

\* ENSF 337 - Lab 2 - Execise E

\*Author of time convert function: DREW HENGEHPLD

\*/

#include <stdio.h>

#include <stdlib.h>

void time\_convert(int ms\_time, int \*minutes\_ptr, double \*seconds\_ptr);

/\*

\* Converts time in milliseconds to time in minutes and seconds.

\* For example, converts 123400 ms to 2 minutes and 3.4 seconds.

\* REQUIRES

\* ms\_time >= 0.0

\* minutes\_ptr and seconds\_ptr point to variables.

\* PROMISES

\* 0 <= \*seconds\_ptr & \*seconds\_ptr < 60.0

\* \*minutes\_ptr minutes + \*seconds\_ptr seconds is equivalent to

\* ms\_time ms.

\*/

int main(void)

{

int millisec;

int minutes;

double seconds;

int nscan;

printf("Enter a time interval as an integer number of milliseconds: ");

nscan = scanf("%d", &millisec);

if (nscan != 1) {

printf("Unable to convert your input to an int.\n");

exit(1);

}

while (millisec < 0){

printf ("please enter a positive number for milliseconds: ");

nscan = scanf("%d", &millisec);

if(nscan != 1)

{

printf("Unable to convert your input to an int.\n");

exit(1);

}

}

printf("Doing conversion for input of %d ms ... \n", millisec);

time\_convert(millisec, &minutes, &seconds);

printf("That is equivalent to %d minute(s) and %f second(s).\n", minutes, seconds);

return 0;

}

void time\_convert(int ms, int \*m, double \*s)

{

\*m = ms / 60000;

\*s = ms%60000 / 1000.0;

}

**SCREENSHOT OF OUTPUT:**

A picture containing graphical user interface

Description automatically generated

Lab1\_Excersize F:

**Table of Inputs and Outputs:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Run #** | **Your inputs** | **Value N** | **Value I** | **Value D** |
| 1 | 12 0.56 | 2 | 12 | 0.56 |
| 2 | 5.12 9.56 | 2 | 5 | 0.12 |
| 3 | 12 ab | 1 | 12 | 1234.5 |
| 4 | ab 12 | 0 | 333 | 1234.5 |
| 5 | 5ab 9.56 | 1 | 5 | 1234.5 |
| 6 | 13 67 | 2 | 13 | 67.0 |