ENSF 614 – Fall 2021

Lab 1 – Tuesday, September 14

Student Name: Bhavyai Gupta

Submission date: September 14, 2021

# Exercise B – Source Code

/\*

 \* File Name:               lab1exe\_B.c

 \* Course:                  ENSF 614 - Fall 2021

 \* Lab # and Assignment #:  Lab 1 Exercise B

 \* Lab section:             B01

 \* Completed by:            Bhavyai Gupta

 \* Submission Date:         September 14, 2021

 \*/

#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);

    if (n != 1)

    {

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

        exit(1);

    }

    while (velocity < 0)

    {

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

        n = scanf("%lf", &velocity);

        if (n != 1)

        {

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

            exit(1);

        }

    }

    create\_table(velocity);

    return 0;

}

/\*\*

 \* Prints the table with relationship between trajectory angle and projectile maximum

 \* travel distance and time for a given initial velocity

 \*/

void create\_table(double v)

{

    printf("\nBelow is the table showing time (t) and distance (d) for various angles -\n\n");

    // print the header and subheader and dividers

    printf("+---------+-----------------+-----------------+\n");

    printf("| %7s | %15s | %15s |\n", "Angle", "t  ", "d ");

    printf("| %7s | %15s | %15s |\n", "(deg)", "(sec)", "(m)");

    printf("+---------+-----------------+-----------------+\n");

    // loop from 0 deg to 90 deg

    for (int deg = 0; deg <= 90; deg = deg + 5)

    {

        // convert deg to rad

        double rad = degree\_to\_radian((double)deg);

        printf("| %7d | %15.5lf | %15.5lf |\n", deg, projectile\_travel\_time(rad, v), projectile\_travel\_distance(rad, v));

    }

    // print the dividers

    printf("+---------+-----------------+-----------------+\n");

    return;

}

/\*\*

 \* Converts degrees to radians

 \*/

double degree\_to\_radian(double d)

{

    return (d \* M\_PI) / 180;

}

/\*\*

 \* Calculates travelling time for the projectile for a given angle a and velocity v

 \*/

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

{

    return ((2 \* v \* sin(a)) / G);

}

/\*\*

 \* Calculates travelling distance for the projectile for a given angle a and velocity v

 \*/

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

{

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

}

/\*\*

 \* Converts degrees to radians

 \*/

double degree\_to\_radian(double d)

{

    return (d \* M\_PI) / 180;

}

/\*\*

 \* Calculates travelling time for the projectile for a given angle a and velocity v

 \*/

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

{

    return ((2 \* v \* sin(a)) / G);

}

/\*\*

 \* Calculates travelling distance for the projectile for a given angle a and velocity v

 \*/

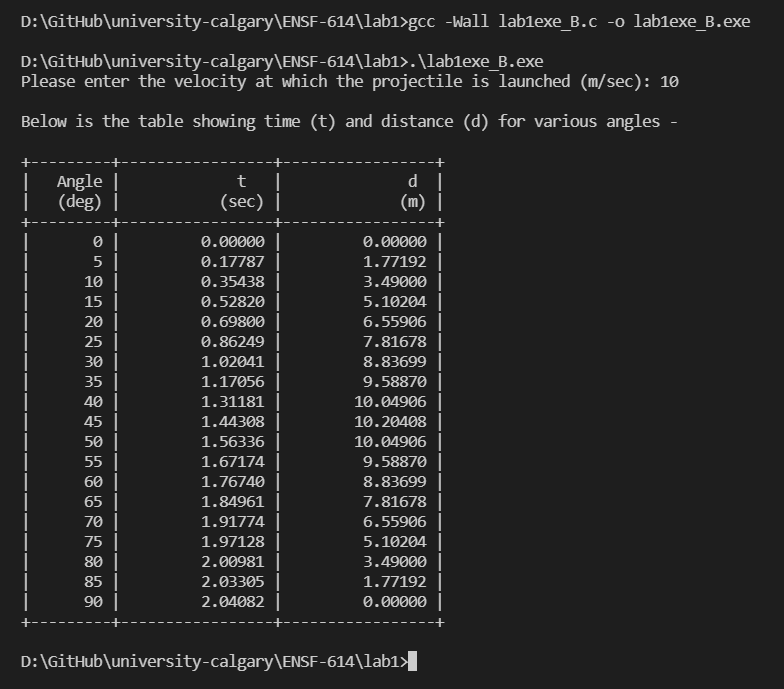
double projectile\_travel\_distance(double a, double v)

{

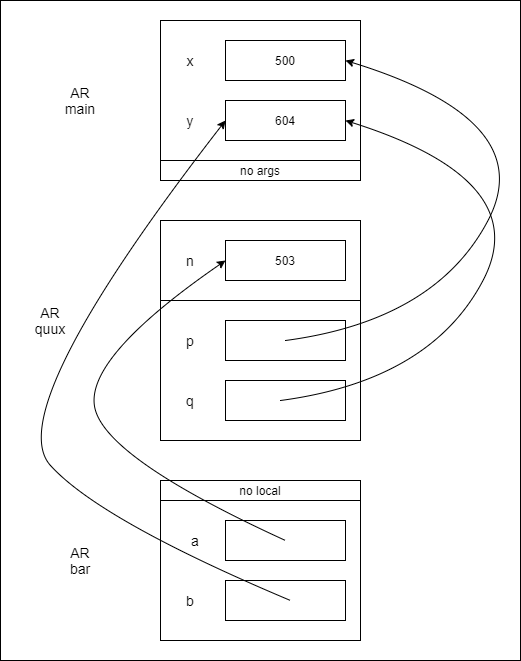
    return ((v \* v \* sin(2 \* a)) / G);

}

# Exercise B – Program Output



# Exercise D Part Two – AR Diagram



# Exercise E – Source Code

/\*

 \* File Name:               lab1exe\_E.c

 \* Course:                  ENSF 614 - Fall 2021

 \* Lab # and Assignment #:  Lab 1 Exercise E

 \* Lab section:             B01

 \* Completed by:            Bhavyai Gupta

 \* Submission Date:         September 14, 2021

 \*/

#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.

 \*    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);

    }

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

    /\* MAKE A CALL TO time\_convert HERE. \*/

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

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

           seconds);

    return 0;

}

/\* PUT YOUR FUNCTION DEFINITION FOR time\_convert HERE. \*/

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

{

    \*minutes\_ptr = ms\_time / (60 \* 1000);

    \*seconds\_ptr = ((double) (ms\_time - (\*minutes\_ptr \* 60 \* 1000))) / 1000;

}

# Exercise E – Program Output

