ENSF 614 – Fall 2021

Lab 1 – Tuesday, September 14

Student Name: Bhavyai Gupta

Submission date: September 14, 2021

Exercise B – Source Code

```c
/*
 * 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);
    }
```

```c
    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(ra
d, v), projectile_travel_distance(rad, v));
    }

    // print the dividers
    printf("+---------+----------------+----------------+\n");
    return;
}
```

```c
/**
 * 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

```
D:\GitHub\university-calgary\ENSF-614\lab1>gcc -Wall lab1exe_B.c -o lab1exe_B.exe

D:\GitHub\university-calgary\ENSF-614\lab1>.\lab1exe_B.exe
Please enter the velocity at which the projectile is launched (m/sec): 10

Below is the table showing time (t) and distance (d) for various angles -

+---------+-----------------+-----------------+
| Angle |        t        |        d        |
| (deg) |      (sec)      |       (m)       |
+---------+-----------------+-----------------+
|     0 |     0.00000     |     0.00000     |
|     5 |     0.17787     |     1.77192     |
|    10 |     0.35438     |     3.49000     |
|    15 |     0.52820     |     5.10204     |
|    20 |     0.69800     |     6.55906     |
|    25 |     0.86249     |     7.81678     |
|    30 |     1.02041     |     8.83699     |
|    35 |     1.17056     |     9.58870     |
|    40 |     1.31181     |    10.04906     |
|    45 |     1.44308     |    10.20408     |
|    50 |     1.56336     |    10.04906     |
|    55 |     1.67174     |     9.58870     |
|    60 |     1.76740     |     8.83699     |
|    65 |     1.84961     |     7.81678     |
|    70 |     1.91774     |     6.55906     |
|    75 |     1.97128     |     5.10204     |
|    80 |     2.00981     |     3.49000     |
|    85 |     2.03305     |     1.77192     |
|    90 |     2.04082     |     0.00000     |
+---------+-----------------+-----------------+

D:\GitHub\university-calgary\ENSF-614\lab1>
```
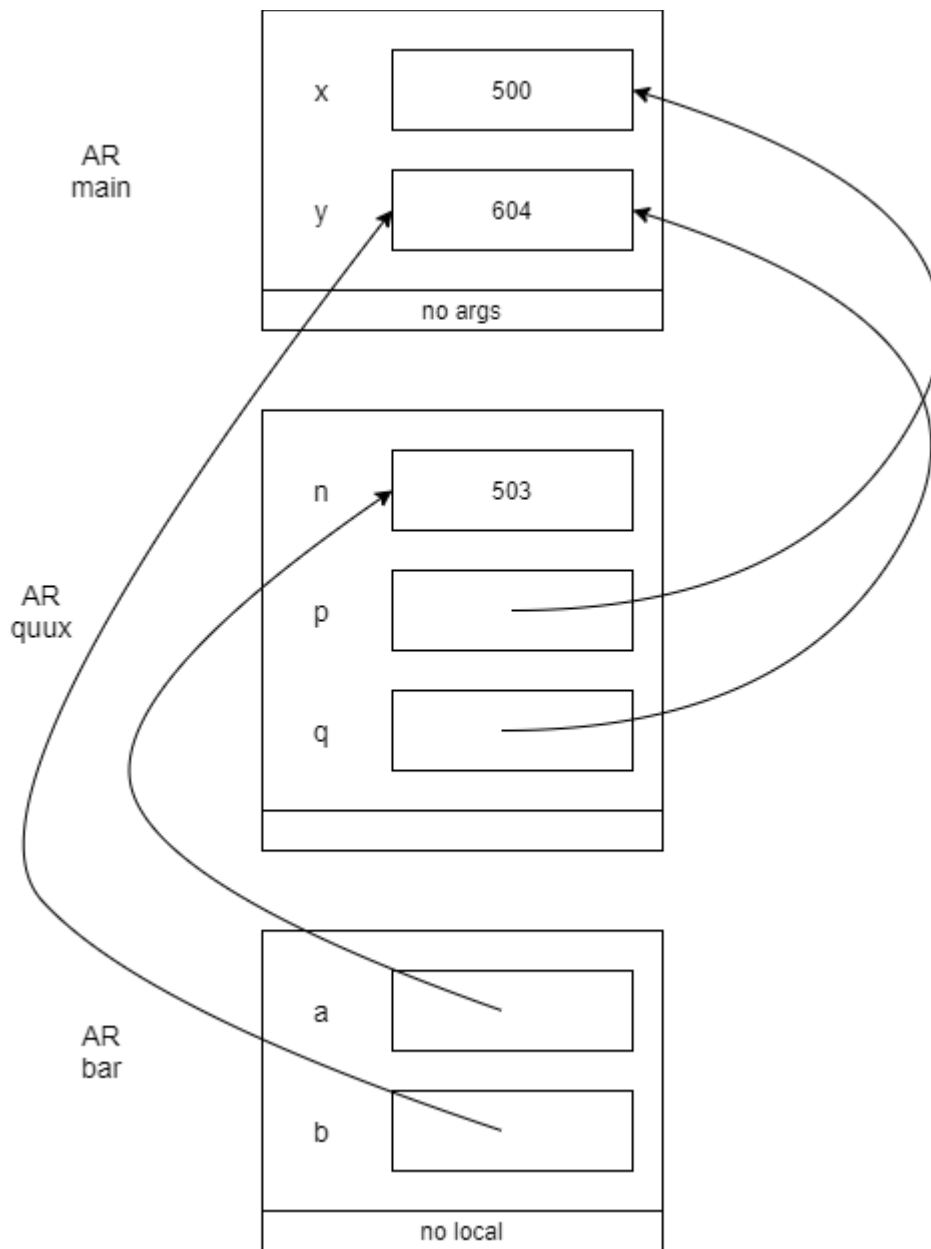
Exercise D Part Two – AR Diagram

AR main		
x	500	
y	604	
	no args	

AR quux		
n	503	
p		
q		

AR bar		
a		
b		
	no local	

Exercise E – Source Code

```c
/*
 * 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

```
D:\GitHub\university-calgary\ENSF-614\lab1>gcc -Wall lab1exe_E.c -o lab1exe_E.exe

D:\GitHub\university-calgary\ENSF-614\lab1>.\lab1exe_E.exe
Enter a time interval as an integer number of milliseconds: 435300
Doing conversion for input of 435300 ms ...
That is equivalent to 7 minute(s) and 15.300000 second(s).

D:\GitHub\university-calgary\ENSF-614\lab1>
```