Course: ENSF 337 - Programming Fundamentals for Software and Computer

Lab #: Lab 2

Instructor: Dr. Maan Khedr

Student Name: Sadia Khandaker

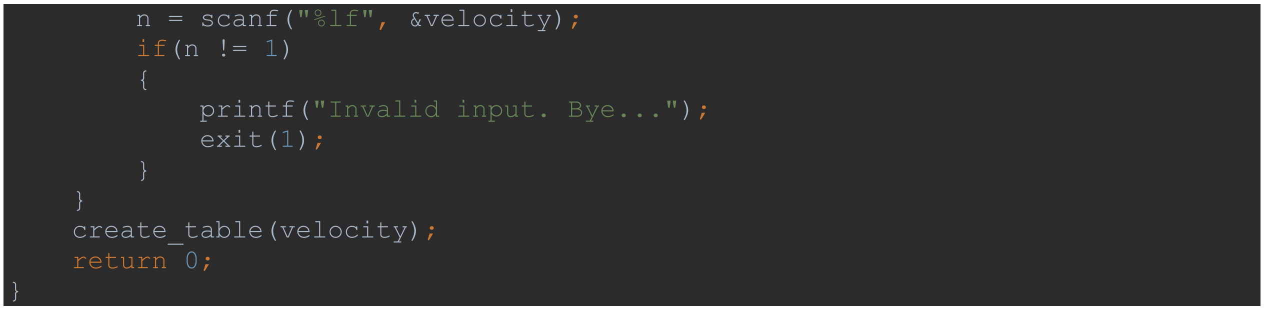
Lab Section: B04

Date Submitted: September 28, 2021

Exercise A – Projectile Time and Motion Calculator

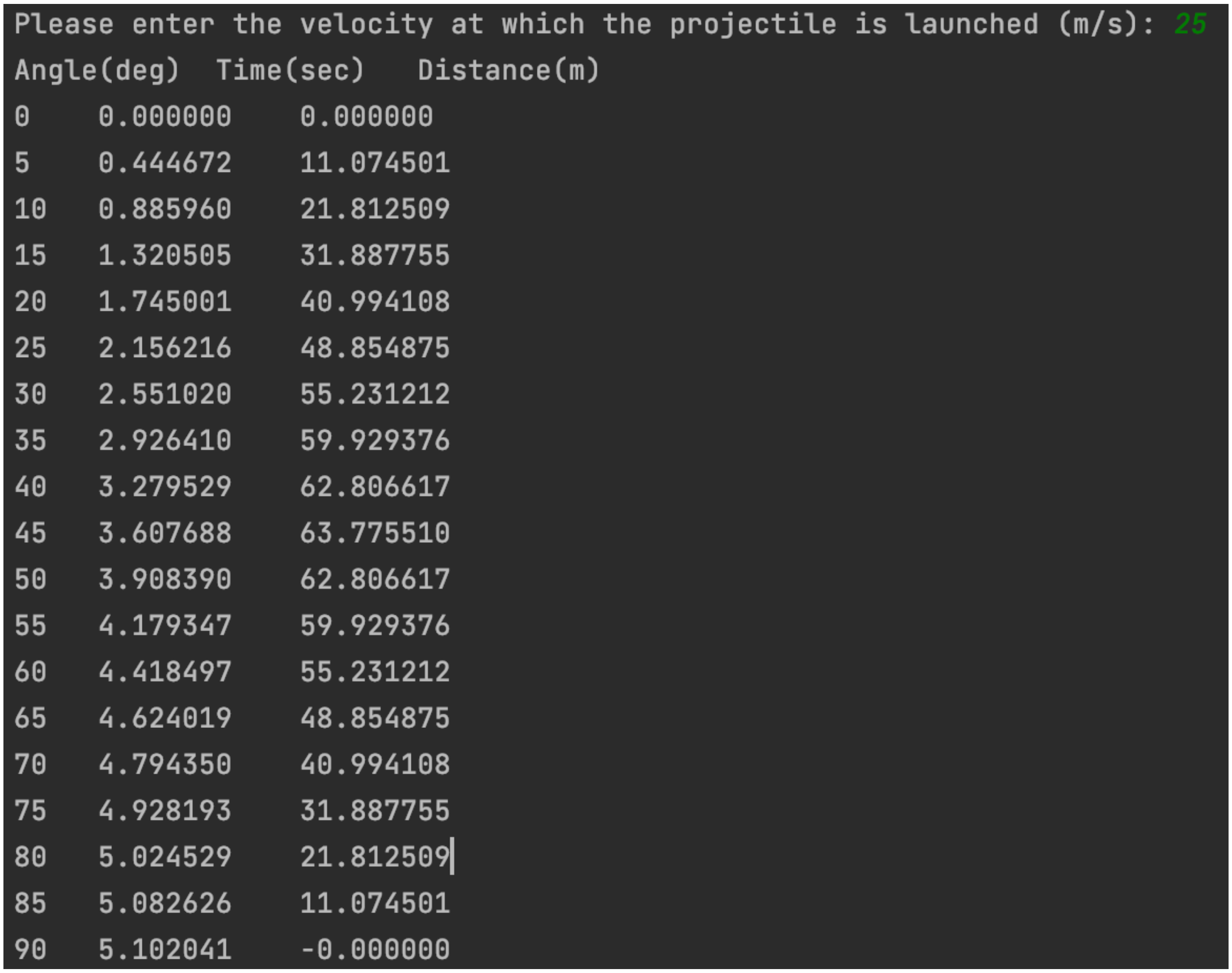
Program to Calculate Projectile Motion and Time:

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

const double g = 9.8;
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);

void create_table(double v) {
    int degree;
    double radian, time, distance;
    printf("Angle(deg)\tTime(sec)\tDistance(m)\n");

    for (degree=0; degree<=90; degree +=5) {
        radian = degree_to_radian(degree);
        time = projectile_travel_time(radian,v);
        distance = projectile_travel_distance(radian,v);
        printf("%d \t %lf \t %lf\n", degree , time, distance);
    }

}

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)* sin(2*a))/(g);
}
double degree_to_radian(double d) {
    return d*(pi/180);
}

int main() {
    int n;
    double velocity;
    printf ("Please enter the velocity at which the projectile is launched
(m/s): ");
    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;
}
```

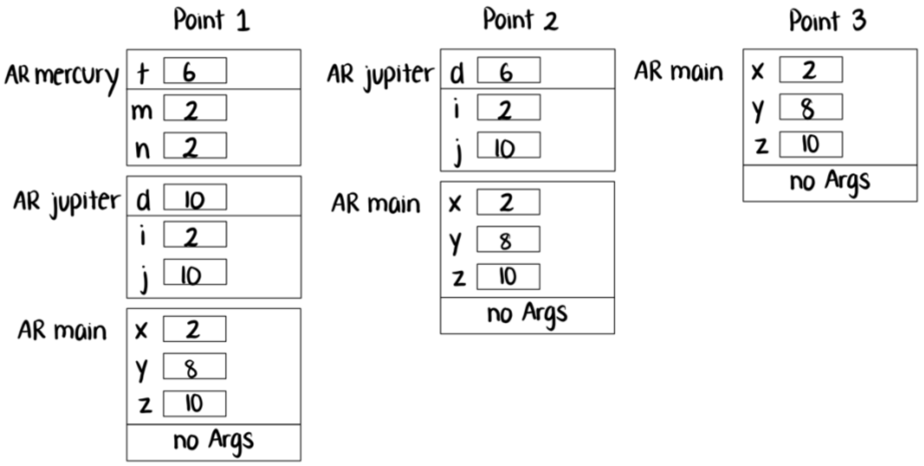
Output:



```
Please enter the velocity at which the projectile is launched (m/s): 25
Angle(deg)  Time(sec)   Distance(m)
0    0.000000    0.000000
5    0.444672    11.074501
10   0.885960    21.812509
15   1.320505    31.887755
20   1.745001    40.994108
25   2.156216    48.854875
30   2.551020    55.231212
35   2.926410    59.929376
40   3.279529    62.806617
45   3.607688    63.775510
50   3.908390    62.806617
55   4.179347    59.929376
60   4.418497    55.231212
65   4.624019    48.854875
70   4.794350    40.994108
75   4.928193    31.887755
80   5.024529    21.812509
85   5.082626    11.074501
90   5.102041    -0.000000
```

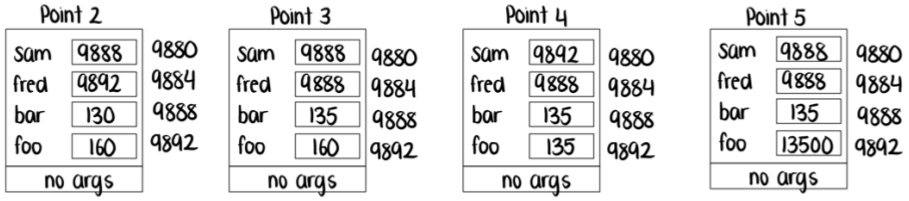
Exercise B - Drawing AR Diagrams for a Simple C Program

AR Diagrams for Point 1, 2 and 3:



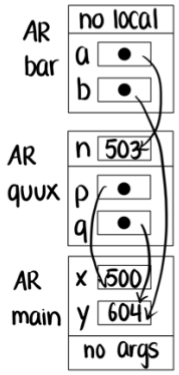
Exercise C – Introduction to Pointers

AR Diagrams from Point 2 to 5:



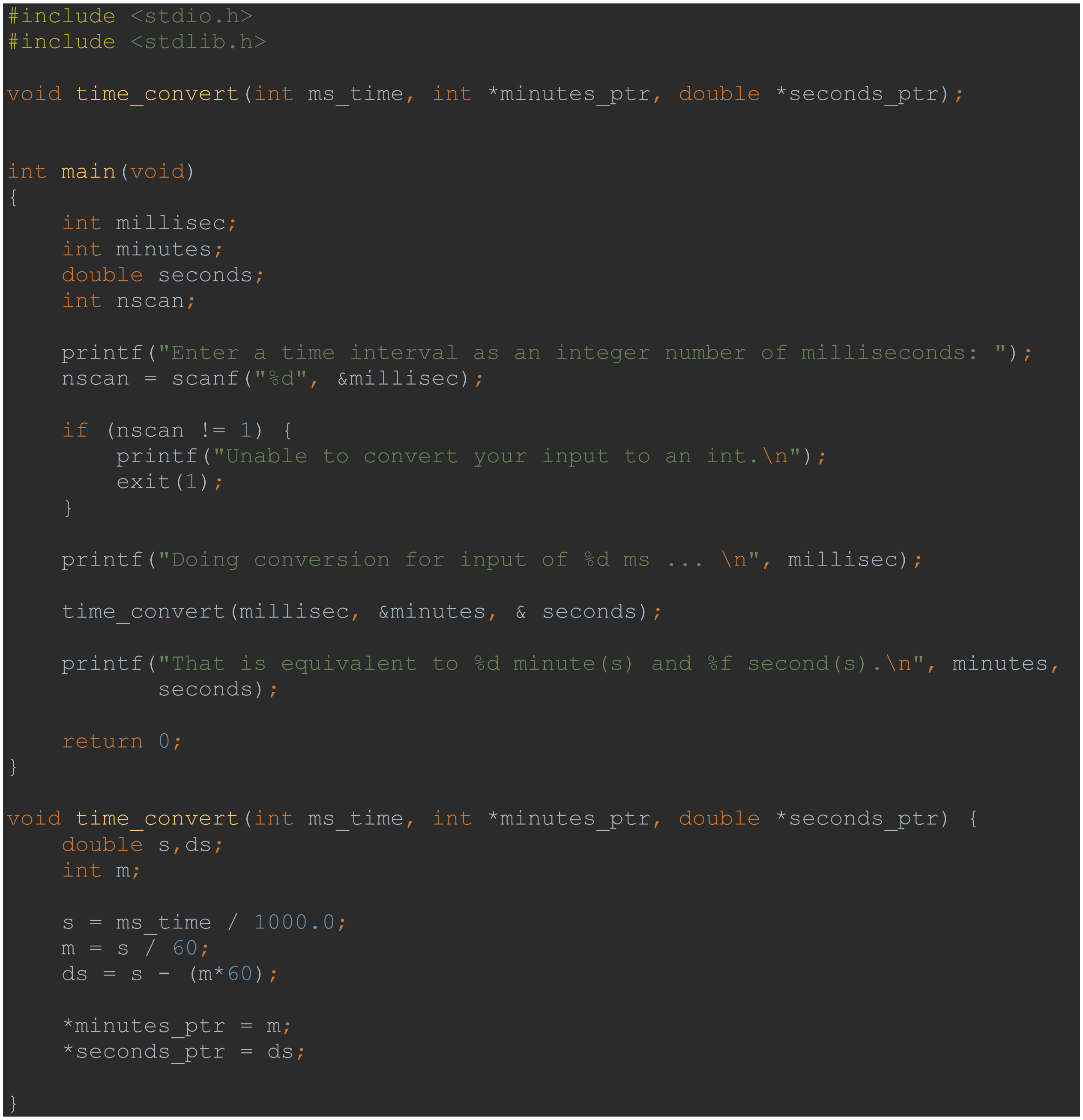
Exercise D - Pointers as Function Arguments

AR Diagram using Arrow Notation:



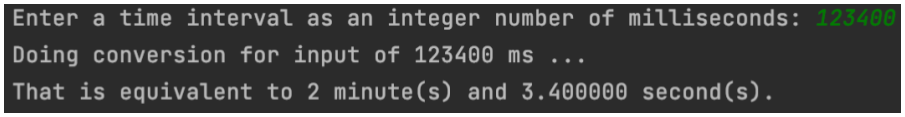
Exercise E - Using Pointers to Get a Function to Change Variables

Program to Convert Milliseconds to Minute and Seconds:



```c
#include <stdio.h>
#include <stdlib.h>

void time_convert(int ms_time, int *minutes_ptr, double *seconds_ptr);


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);

    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_time, int *minutes_ptr, double *seconds_ptr) {
    double s,ds;
    int m;

    s = ms_time / 1000.0;
    m = s / 60;
    ds = s - (m*60);

    *minutes_ptr = m;
    *seconds_ptr = ds;

}
```

Output:



```
Enter a time interval as an integer number of milliseconds: 123400
Doing conversion for input of 123400 ms ...
That is equivalent to 2 minute(s) and 3.400000 second(s).
```

Exercise F: More on scanf

Table of Values of n, i, and d:

Run #	First Input	Second Input	Value of n	Value of i	Value of d
1	12	0.56	2	12	0.560000
2	5.12	9.56	2	5	0.120000
3	12	ab	1	12	1234.500000
4	ab	12	0	333	1234.500000
5	5ab	9.56	1	5	1234.500000
6	13	67	2	13	67.000000