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:

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

```c
        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));
        }
}
```

```
[drewhengehold@Drews-MacBook-Pro Lab 2 % ls
ENSF 337 Lab2-Instruction-Fall 2022.pdf  lab2exe_D1.c
Lab 2 Run Tester                         lab2exe_D2.c
Solution For Exercise D Part I.pdf       lab2exe_E.c
lab2exe_A.c                              lab2exe_F1.c
lab2exe_B.c                              lab2exe_F2.c
lab2exe_C.c
[drewhengehold@Drews-MacBook-Pro Lab 2 % gcc -Wall lab2exe_A.c -o Velocity
drewhengehold@Drews-MacBook-Pro Lab 2 % ./Velocity
Please enter the velocity at which the projectile is launched (m/sec): 5.7
Angle          t                    d
(deg           (sec)                (m)
0.000000       0.000000             0.000000
5.000000       0.101385             0.575697
10.000000      0.201999             1.133901
15.000000      0.301075             1.657653
20.000000      0.397860             2.131038
25.000000      0.491617             2.539672
30.000000      0.581633             2.871139
35.000000      0.667222             3.115369
40.000000      0.747733             3.264939
45.000000      0.822553             3.315306
50.000000      0.891113             3.264939
55.000000      0.952891             3.115369
60.000000      1.007417             2.871139
65.000000      1.054276             2.539672
70.000000      1.093112             2.131038
75.000000      1.123628             1.657653
80.000000      1.145593             1.133901
85.000000      1.158839             0.575697
90.000000      1.163265             -0.000000
[drewhengehold@Drews-MacBook-Pro Lab 2 % ./Velocity
Please enter the velocity at which the projectile is launched (m/sec): 100
Angle          t                    d
(deg           (sec)                (m)
0.000000       0.000000             0.000000
5.000000       1.778689             177.192018
10.000000      3.543840             349.000146
15.000000      5.282021             510.204082
20.000000      6.980003             655.905724
25.000000      8.624862             781.678003
30.000000      10.204082            883.699392
35.000000      11.705642            958.870021
40.000000      13.118114            1004.905870
45.000000      14.430751            1020.408163
50.000000      15.633560            1004.905870
55.000000      16.717389            958.870021
60.000000      17.673988            883.699391
65.000000      18.496077            781.678003
70.000000      19.177400            655.905724
75.000000      19.712772            510.204081
80.000000      20.098117            349.000146
85.000000      20.330504            177.192018
90.000000      20.408163            -0.000000
drewhengehold@Drews-MacBook-Pro Lab 2 % █
```
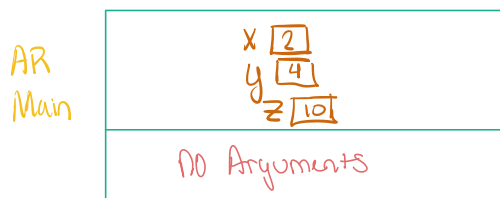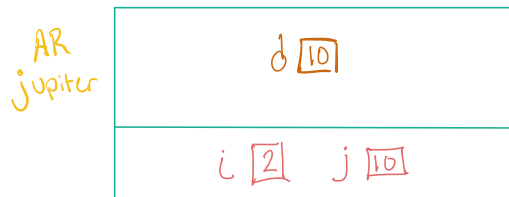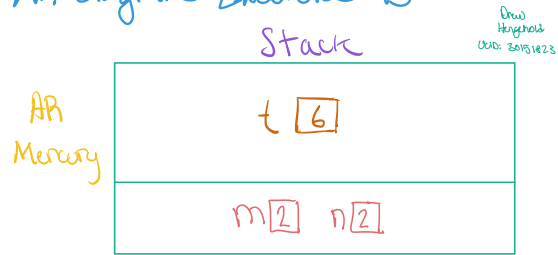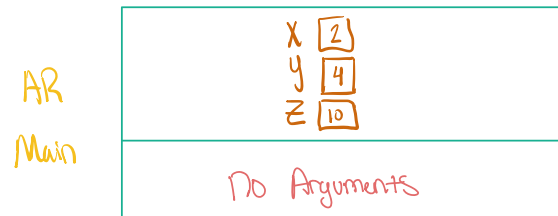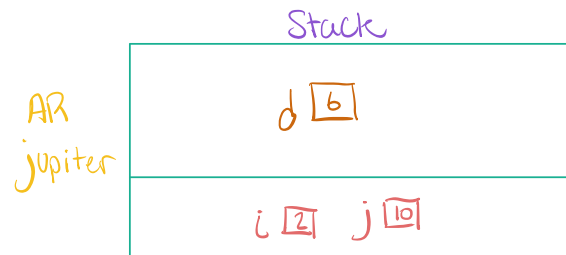
Outputs screenshot Exercise A

```
Lab1_Excersize B:
```

AR diagrams Excercise B: Point 1

Drew
Hergehold
UCID: 30151823

Stack

AR
Mercury

| t 6 |
| m 2   n 2 |

AR
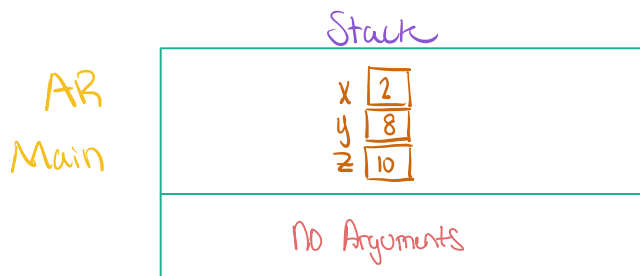jupiter

| d 10 |
| i 2   j 10 |

AR
Main

| x 2 |
| y 4 |
| z 10 |
| No Arguments |

Point 2

Drew
Hergehold
UCID: 30151823

Stack

AR
jupiter

| d 6 |
| i 2   j 10 |

AR
Main

| x 2 |
| y 4 |
| z 10 |
| No Arguments |

Point 3

Drew
Hergehold
UCID: 30151823

Stack

AR
Main

| x 2 |
| y 8 |
| z 10 |
| No Arguments |

Lab1_Excersize C:

Eercise C : AR Diagrams

Stack
Sum	9888
fred	9892
bar	130
foo	160

AR

Main

No Arguments

Stack
Sum	9888
fred	9888
bar	135
foo	160

AR

Main

No Arguments

Stack
Sum	9892
fred	9888
bar	135
foo	135

AR

Main

No Arguments

Stack
Sum	9888
fred	9888
bar	135
foo	13500

AR

Main

No Arguments

Lab1_Excersize D:
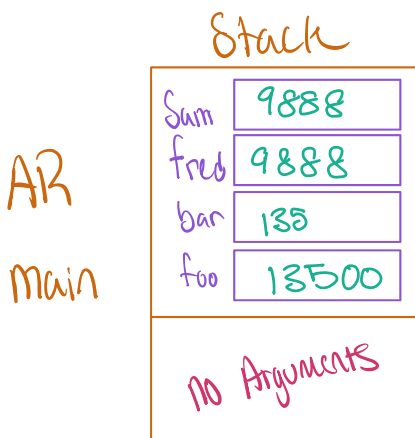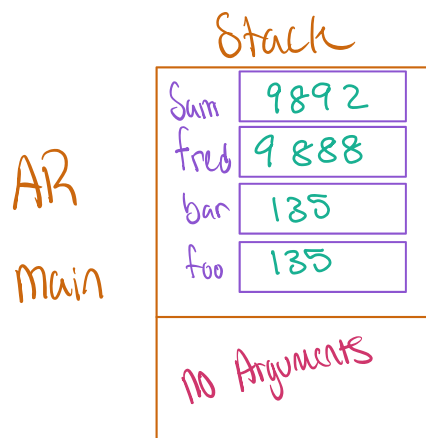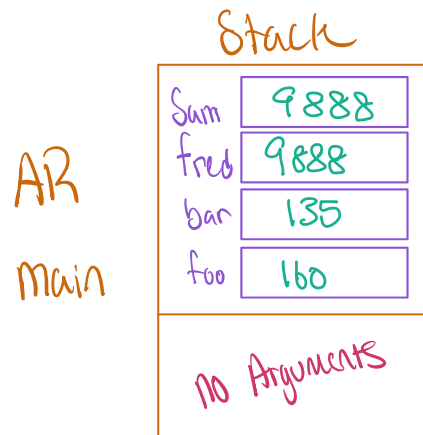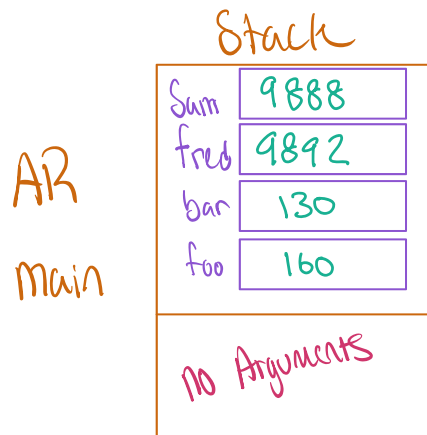
Exercise D: Part 2, AR Diagrams

Stack

AR
bar

No Arguments

a ▢   b ▢

AR
quux

n ▢

p ▢   q ▢

AR
Main
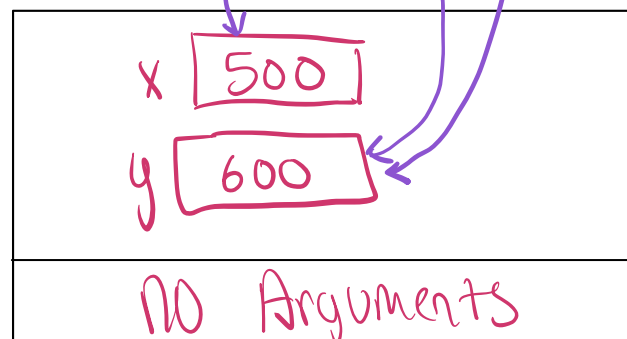
x [500]

y [600]

No Arguments

Lab1_Excersize E:
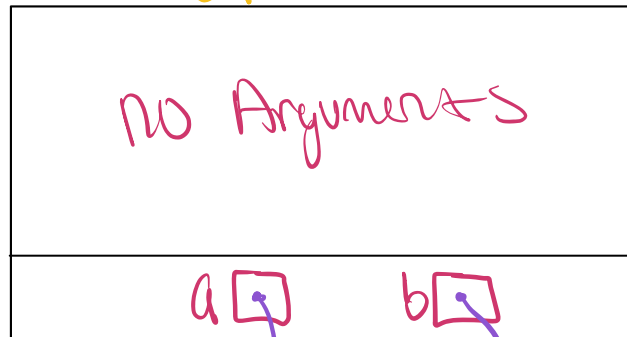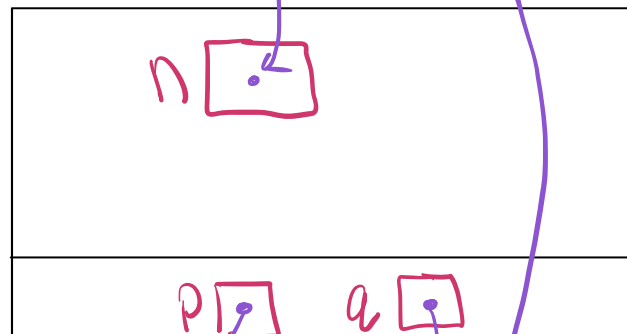
SOURCE CODE:

```c
/*
 *
 * 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:


```
drewhengehold@Drews-MacBook-Pro ~ % cd /Users/drewhengehold/Documents/Software\ Engineering/XCode\ Projects/Lab\ 2
drewhengehold@Drews-MacBook-Pro Lab 2 % gcc -Wall lab2exe_E.c -o Milliseconds
drewhengehold@Drews-MacBook-Pro Lab 2 % ./Milliseconds
Enter a time interval as an integer number of milliseconds: 60000
Doing conversion for input of 60000 ms ...
That is equivalent to 1 minute(s) and 0.000000 second(s).
drewhengehold@Drews-MacBook-Pro Lab 2 % ./Milliseconds
Enter a time interval as an integer number of milliseconds: -5000
please enter a positive number for milliseconds: 59000
Doing conversion for input of 59000 ms ...
That is equivalent to 0 minute(s) and 59.000000 second(s).
drewhengehold@Drews-MacBook-Pro Lab 2 %
```

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