Course: Programming Fundamental – ENSF 337
Lab #: Lab 2
Instructor: M. Moussavi
Student Name: Aarushi Roy Choudhury
Lab Section: B01
Date submitted: Sept 30,2021

Exercise A

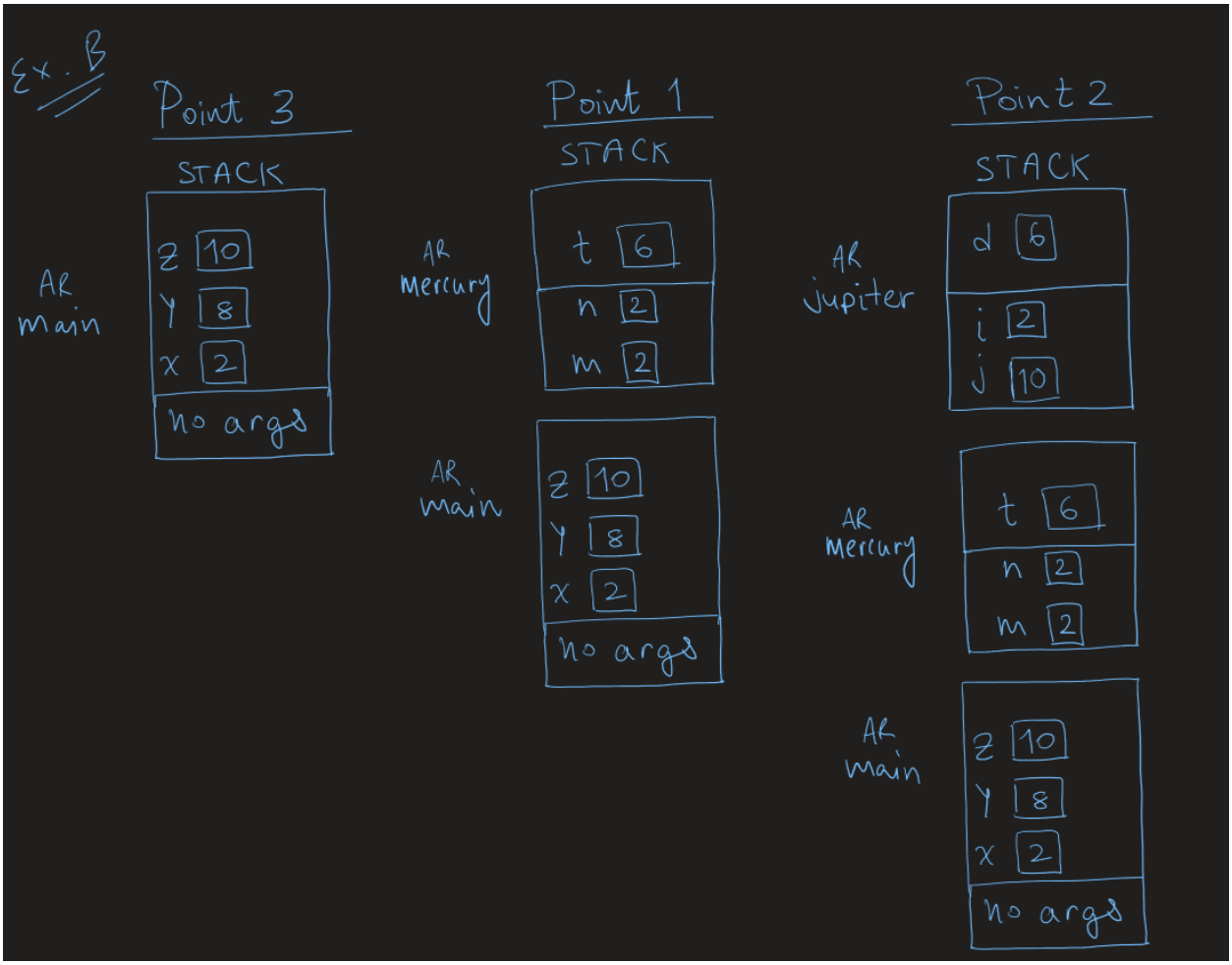
```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

const double G = 9.8;   /* gravitation acceleration 9.8 m/s^2 */
const double PI = 3.141592654;

void create_table(double v);
//REQUIRES: A positive, real number for the value of the velocity
//PROMISES: Displays a table of angle, time and distance of the projectile
double degree_to_radian(double d);
//REQUIRES: A real number in degrees for the value of the angle
//PROMISES: Converts the angle from degrees to radians
double Projectile_travel_time(double a, double v);
//REQUIRES: A positive, real number for the value of the velocity and real number
 for the angle
//PROMISES: Calculates the time and returns the value
double Projectile_travel_distance(double a, double v);
//REQUIRES: A positive, real number for the value of the velocity and real number
 for the angle
//PROMISES: Calculates the distance and returns the value


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("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;
}

/* UNCOMMENT THE CALL TO THE create_table IN THE main FUNCTION, AND COMPLETE THE
PROGRAM */



double degree_to_radian(double a)
{

    return a*(PI/180);
}
double Projectile_travel_time(double a, double v)
{
    double rad = degree_to_radian(a);
    double t= (2*v*sin(rad))/G;
    return t;
}

double Projectile_travel_distance(double a, double v)
{
    double rad = degree_to_radian(a);
    double d= ((v*v)/G)*sin(2*rad);
    return d;
}

void create_table(double v)
{
    int i =0;
    printf(    "Angle              t             d\n");
    printf(    "(deg)            (sec)          (m)\n");
    while (i<=90)
    {
        double ang = (double) i;
        double t= Projectile_travel_time(ang,v);
        double d = Projectile_travel_distance(ang,v);
```

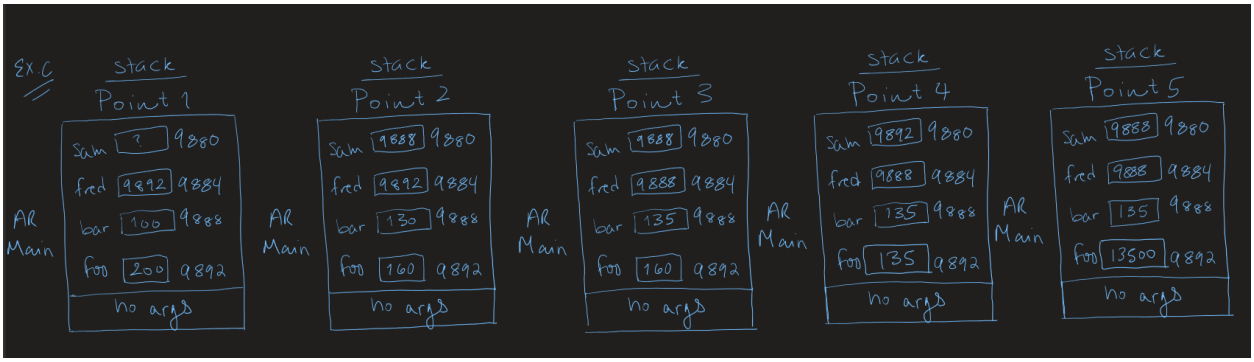
```
        printf("%8.6lf       %7.6lf          %8.6lf\n",ang, t, d );
        i+=5;

    }
}
```

```
Please enter the velocity at which the projectile is launched (m/sec): 22.5
Angle               t              d
(deg)             (sec)           (m)
0.000000       0.000000          0.000000
5.000000       0.400205          8.970346
10.000000       0.797364          17.668132
15.000000       1.188455          25.829082
20.000000       1.570501          33.205227
25.000000       1.940594          39.572449
30.000000       2.295918          44.737282
35.000000       2.633769          48.542795
40.000000       2.951576          50.873360
45.000000       3.246919          51.658163
50.000000       3.517551          50.873360
55.000000       3.761412          48.542795
60.000000       3.976647          44.737282
65.000000       4.161617          39.572449
70.000000       4.314915          33.205227
75.000000       4.435374          25.829082
80.000000       4.522076          17.668132
85.000000       4.574363          8.970346
90.000000       4.591837          -0.000000
```

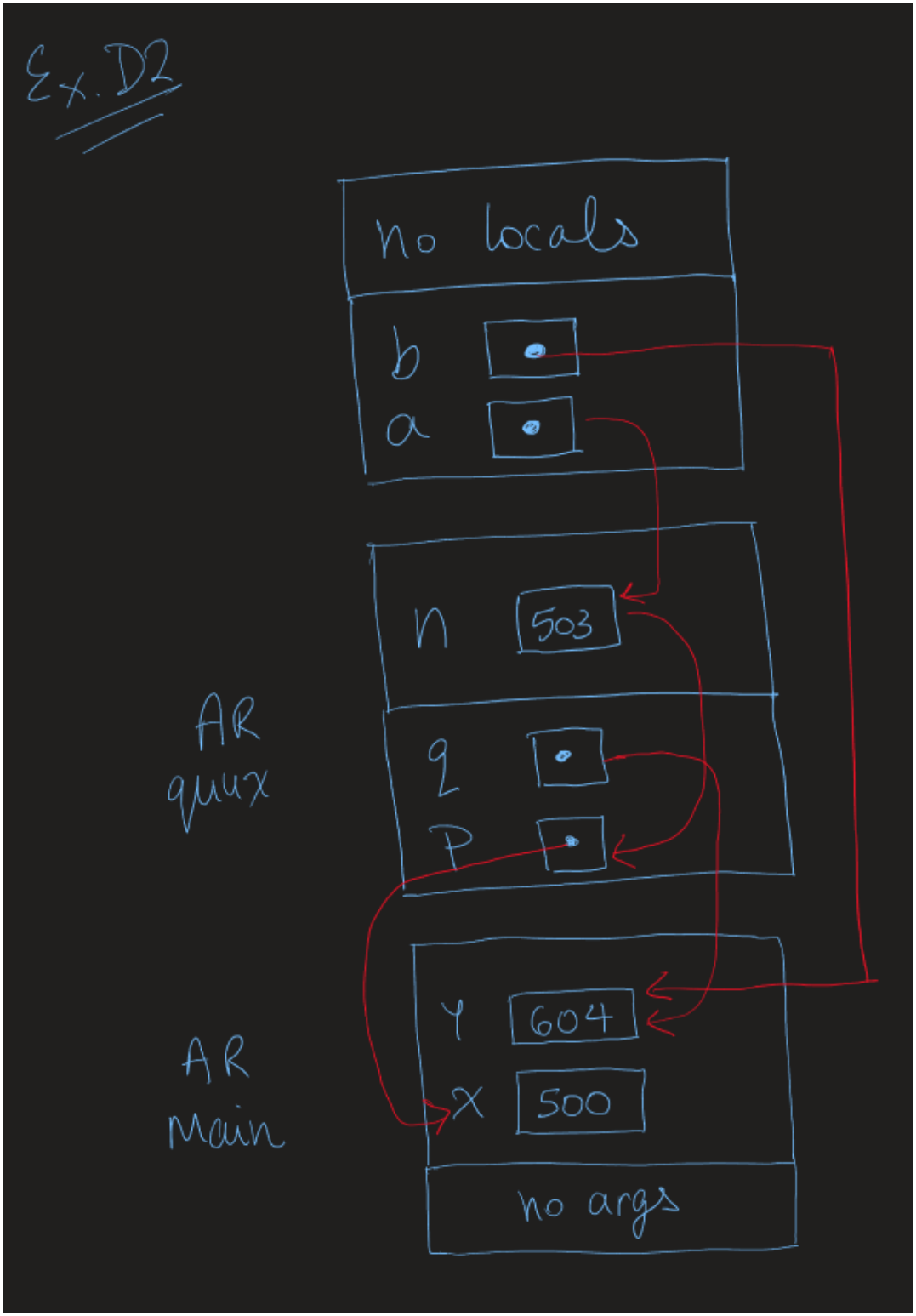
Exercise B



Exercise C



Exercise D Part II



Exercise E

```c
#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);


  time_convert(millisec,&minutes, &seconds);

  /* MAKE A CALL TO time_convert HERE. */

  printf("That is equivalent to %d minute(s) and %4.2f second(s).\n", minutes,
   seconds);

  return 0;
```

```c
}

/* WRITE YOUR FUNCTION DEFINITION FOR time_convert HERE. */
void time_convert(int millisec, int *p_min, double *p_sec)
{
  *p_min= millisec/60;

  *p_sec = millisec%60;

}
```

```
Enter a time interval as an integer number of milliseconds: 120
Doing conversion for input of 120 ms ...
That is equivalent to 2 minute(s) and 0.00 second(s).
PS C:\Users\Aarus\Downloads>
```

Exercise F

Run #	Input	n	i	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