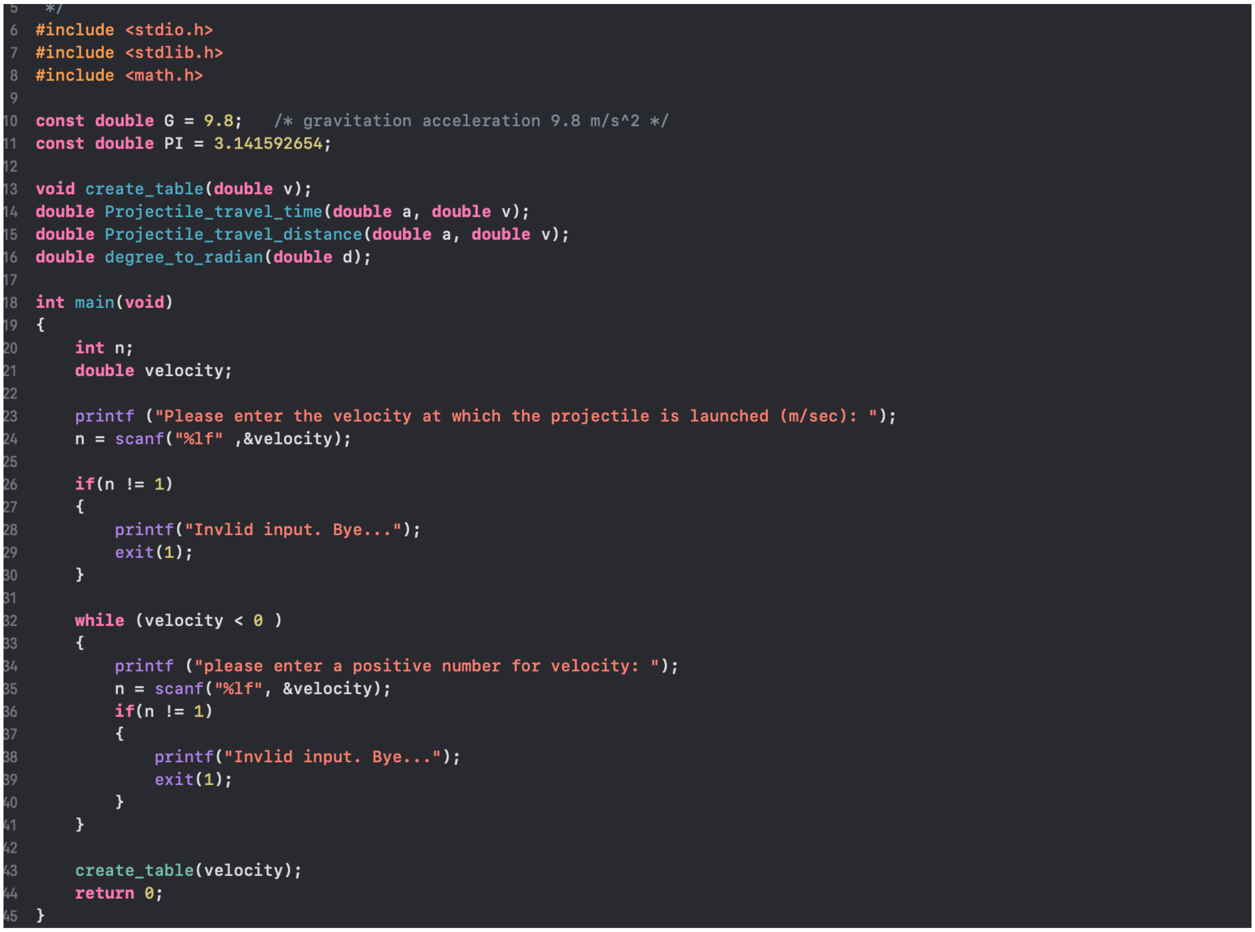
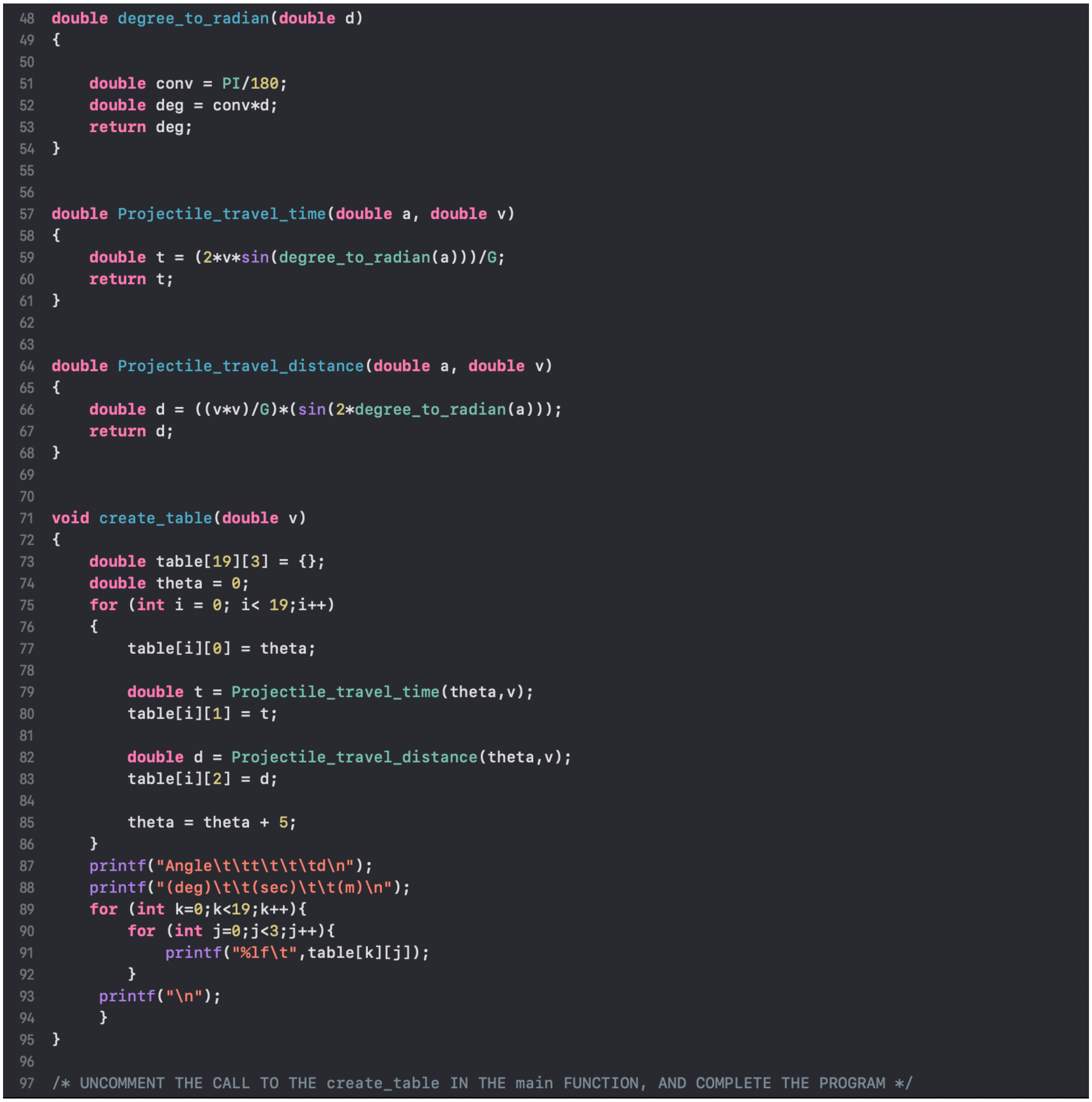
Course	: Programming Fundamental – ENSF 337
Lab #	: Lab 2
Instructor	: M. Moussavi
Student Names	: Nimna Wijedasa, Mehvish Shakeel
UCID	: 30146042 , 30161318
Lab Section	: B02
Date submitted	: Sept 29, 2022

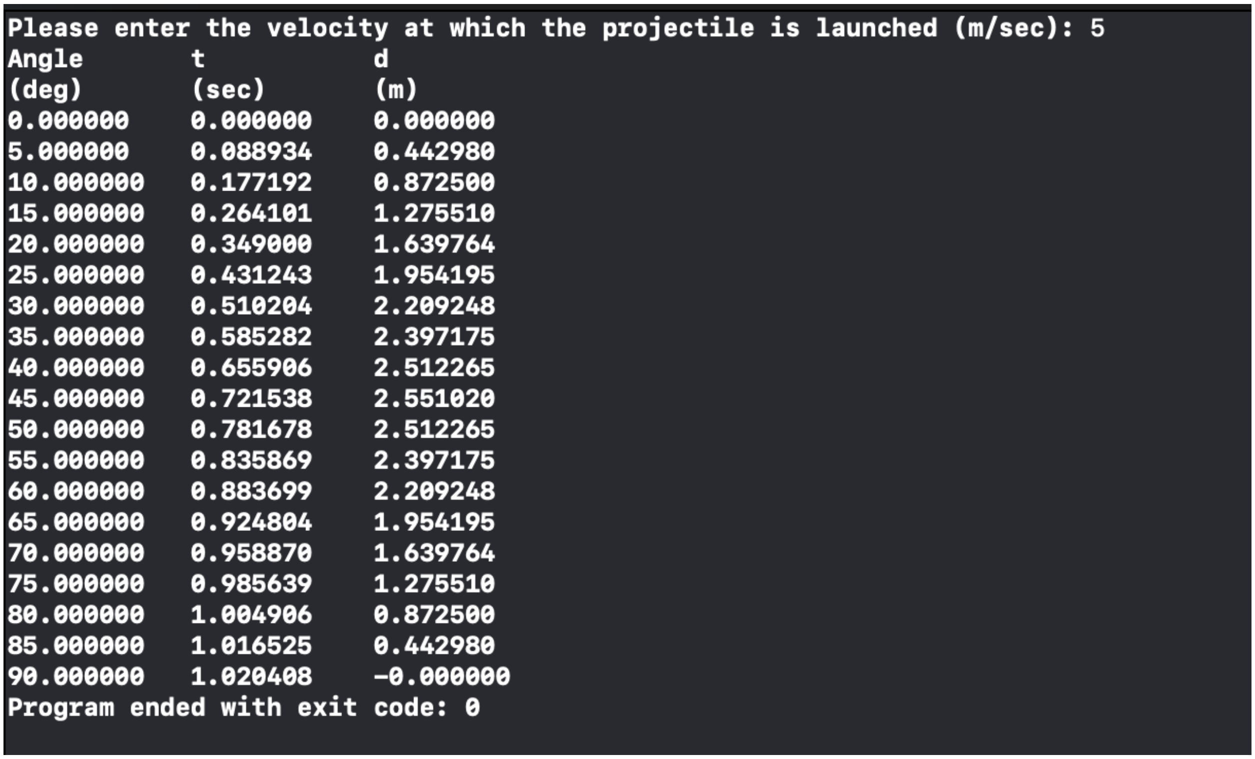
Exercise A



```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);
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("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;
}
```



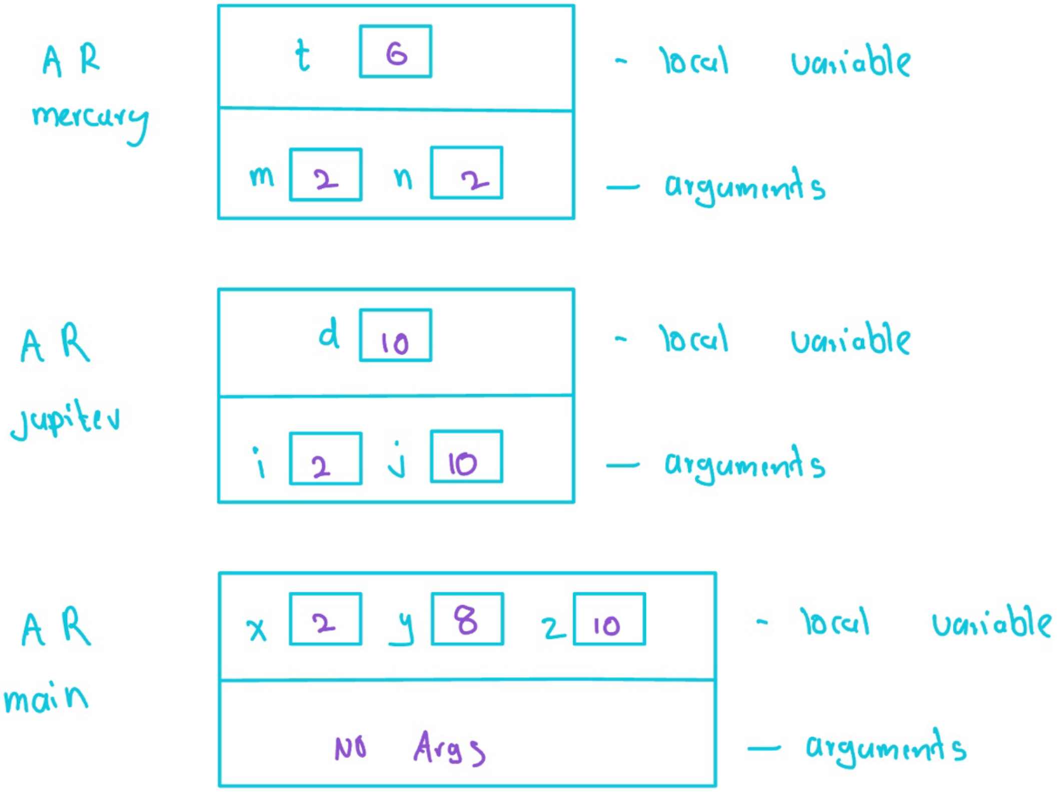
```c
double degree_to_radian(double d)
{

    double conv = PI/180;
    double deg = conv*d;
    return deg;
}


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


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


void create_table(double v)
{
    double table[19][3] = {};
    double theta = 0;
    for (int i = 0; i< 19;i++)
    {
        table[i][0] = theta;

        double t = Projectile_travel_time(theta,v);
        table[i][1] = t;

        double d = Projectile_travel_distance(theta,v);
        table[i][2] = d;

        theta = theta + 5;
    }
    printf("Angle\t\tt\t\t\td\n");
    printf("(deg)\t\t(sec)\t\t(m)\n");
    for (int k=0;k<19;k++){
        for (int j=0;j<3;j++){
            printf("%lf\t",table[k][j]);
        }
     printf("\n");
     }
}

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



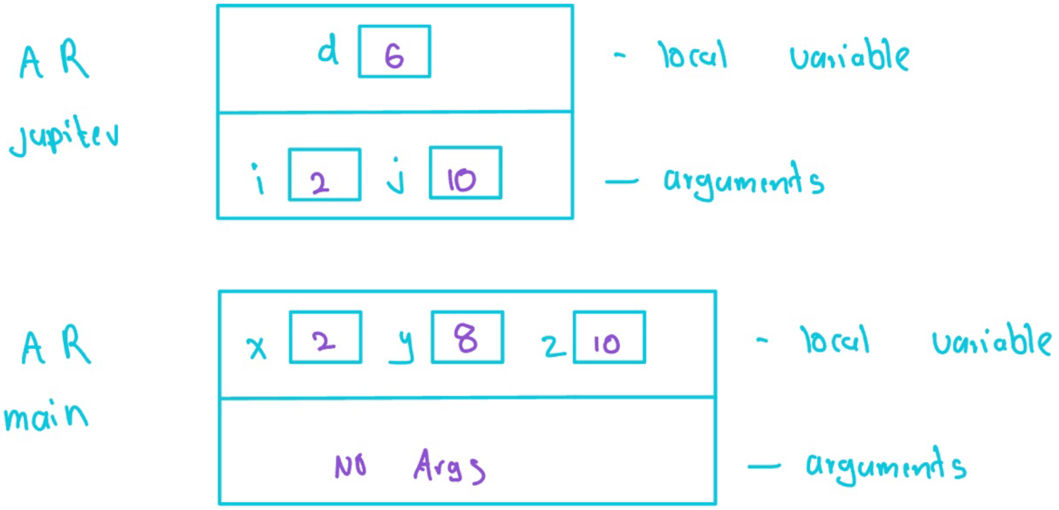
```
Please enter the velocity at which the projectile is launched (m/sec): 5
Angle       t           d
(deg)       (sec)       (m)
0.000000    0.000000    0.000000
5.000000    0.088934    0.442980
10.000000   0.177192    0.872500
15.000000   0.264101    1.275510
20.000000   0.349000    1.639764
25.000000   0.431243    1.954195
30.000000   0.510204    2.209248
35.000000   0.585282    2.397175
40.000000   0.655906    2.512265
45.000000   0.721538    2.551020
50.000000   0.781678    2.512265
55.000000   0.835869    2.397175
60.000000   0.883699    2.209248
65.000000   0.924804    1.954195
70.000000   0.958870    1.639764
75.000000   0.985639    1.275510
80.000000   1.004906    0.872500
85.000000   1.016525    0.442980
90.000000   1.020408    -0.000000
Program ended with exit code: 0
```

Exercise B

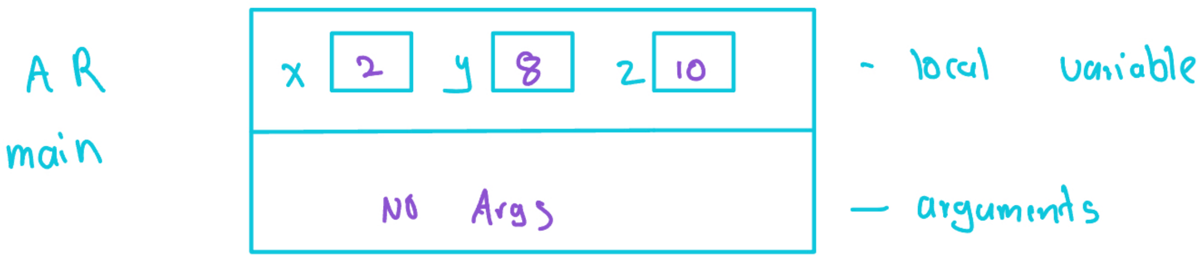
point 1



point 2

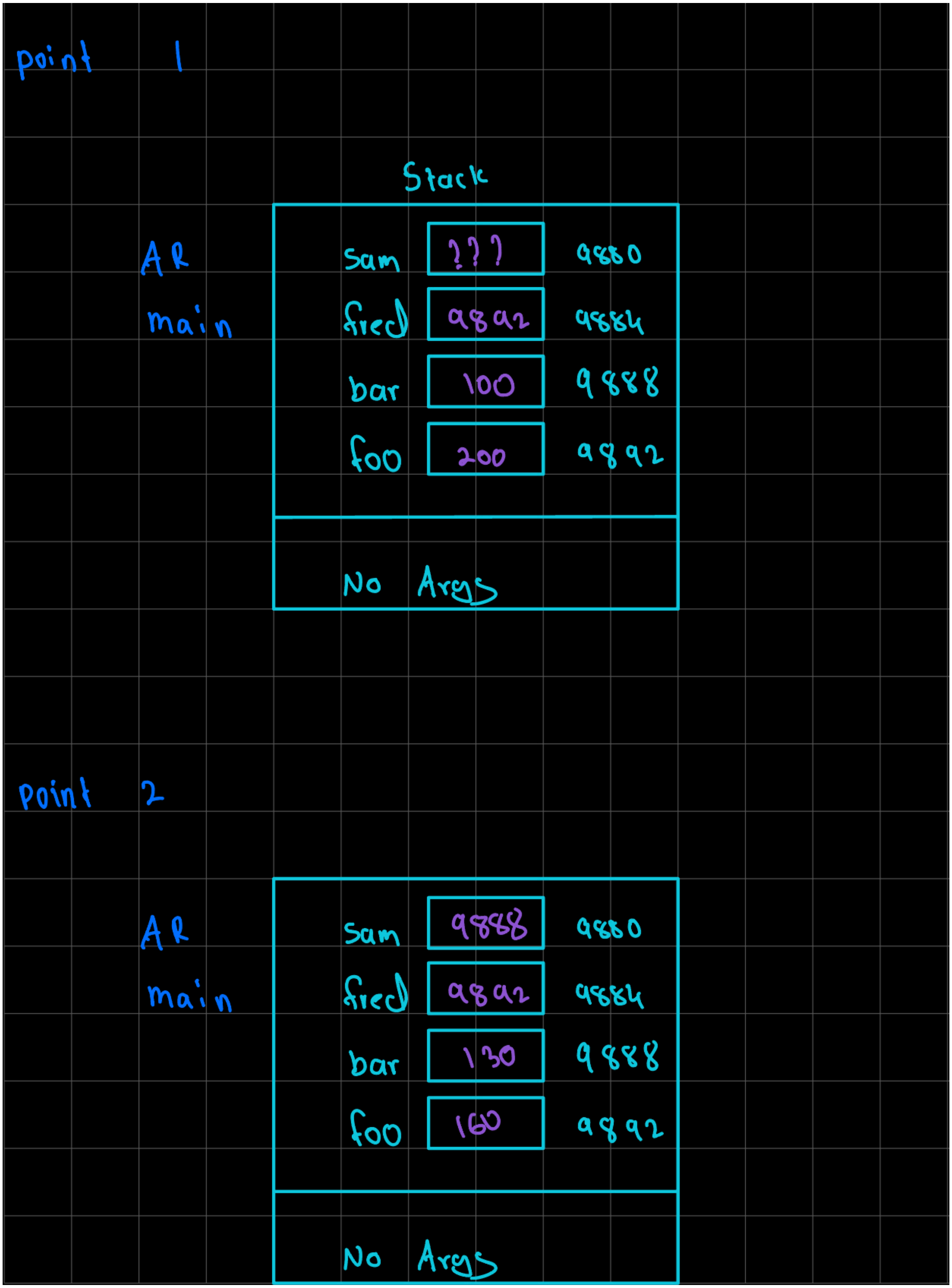


point 3



$-4 + 6 \times 2$

Exercise C



point 1

Stack

AR main

sam	???	9880
fred	9892	9884
bar	100	9888
foo	200	9892

No Args

point 2

AR main

sam	9888	9880
fred	9892	9884
bar	130	9888
foo	160	9892

No Args

Point 3

AR main

sam	9888	9880
fred	9888	9884
bar	135	9888
foo	160	9892

No Args

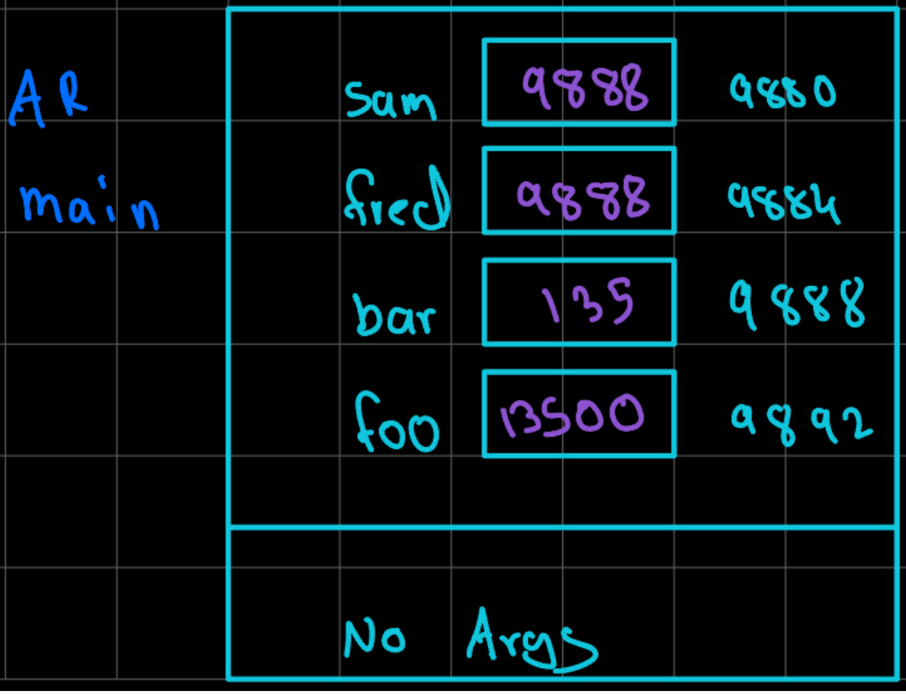
Point 4

AR main

sam	9892	9880
fred	9888	9884
bar	135	9888
foo	135	9892

No Args

Point 5

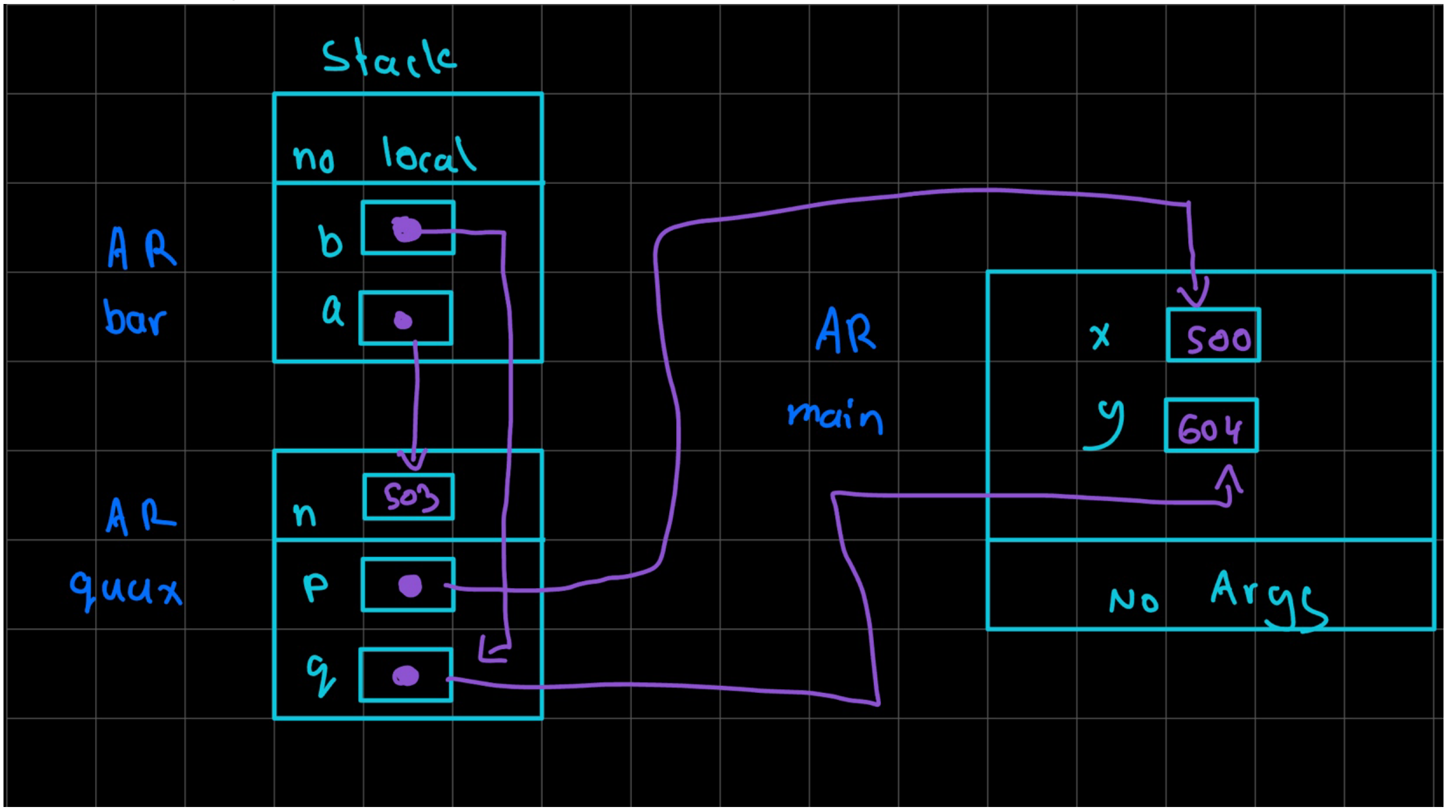


AR
main

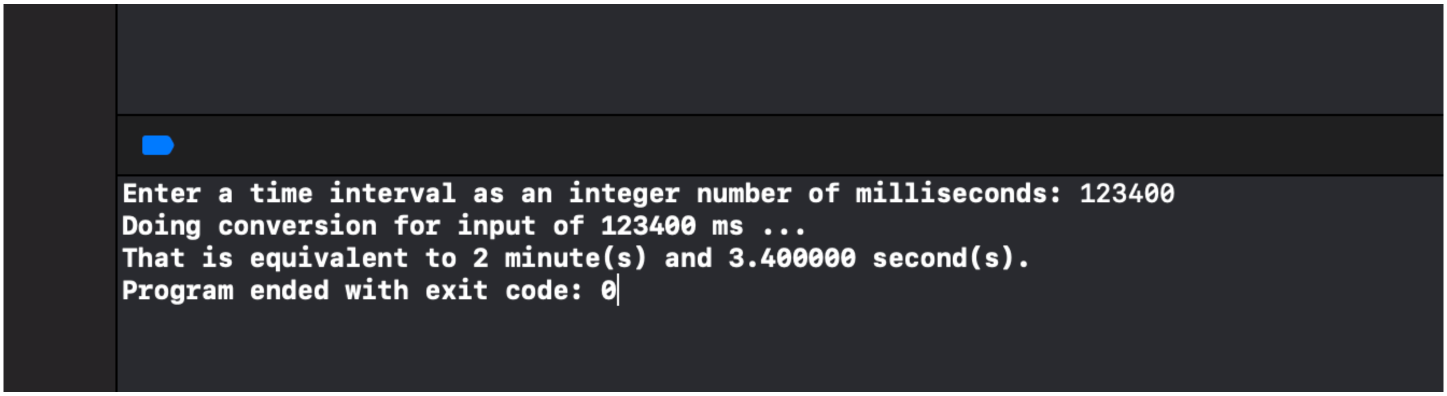
sam	9888	9880
fred	9888	9884
bar	135	9888
foo	13500	9892

No Args

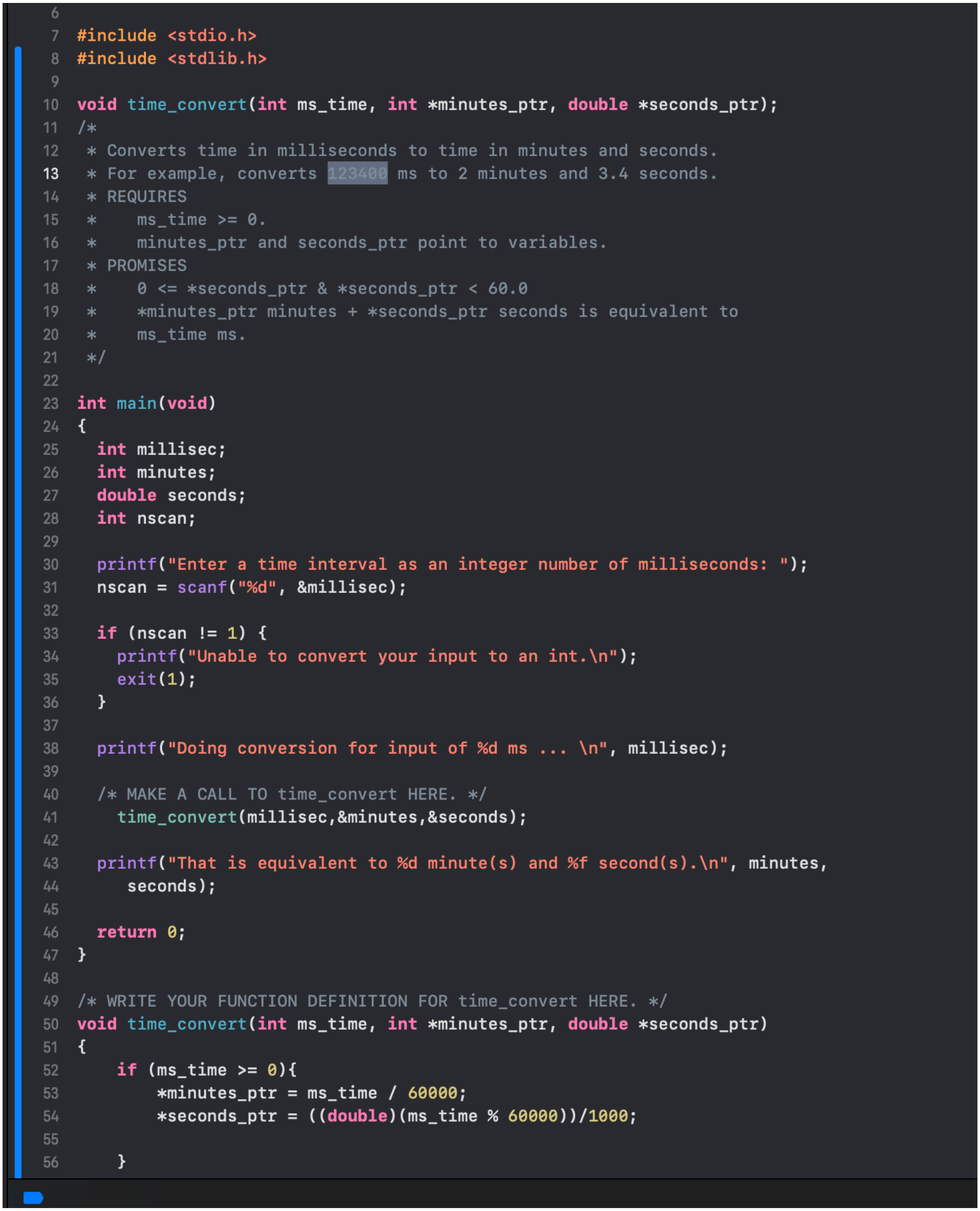
Exercise D (p ii)



Exercise E



```
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).
Program ended with exit code: 0
```



```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);

  /* 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;
}

/* WRITE YOUR FUNCTION DEFINITION FOR time_convert HERE. */
void time_convert(int ms_time, int *minutes_ptr, double *seconds_ptr)
{
    if (ms_time >= 0){
        *minutes_ptr = ms_time / 60000;
        *seconds_ptr = ((double)(ms_time % 60000))/1000;

    }
```

Exercise F (p ii)

Run #	Your inputs	What is the value of n	What is the value of i	What is the value d
1	12 0.56	2	12	.56
2	5.12 9.56	2	5	.12
3	12 ab	1.	12	1234.5
4	ab 12	0	33	1234.5
5	5ab 9.56	1	5	1234.5
6	13 67	2	13	67.0