Programming Fundamentals – ENSF 337

Lab 2

M. Moussavi

Jaxon Braun

B01

Submitted on September 28, 2021

Exercise A: Projectile Time and Distance Calculator

Source code:

```
/*
 *  File Name: lab2exe_A.c
 *  Created by Mahmood Moussavi, Sept 2017 for ENCM 339
 *  Assignment: Lab 2 Exercise A
 *  Completed by: Jaxon Braun
 *  Submission Date: September 28. 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("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;
}
void create_table(double v){
    printf("Angle(deg)\t time(s)\t distance(m) \n");
    for (double deg; deg <= 90; deg += 5){
        double a = degree_to_radian(deg);
        double t = Projectile_travel_time(a, v);
        double d = Projectile_travel_distance(a, v);
        printf("%f\t %f\t %f\t \n", deg, t, d);
    }
    return ;
```

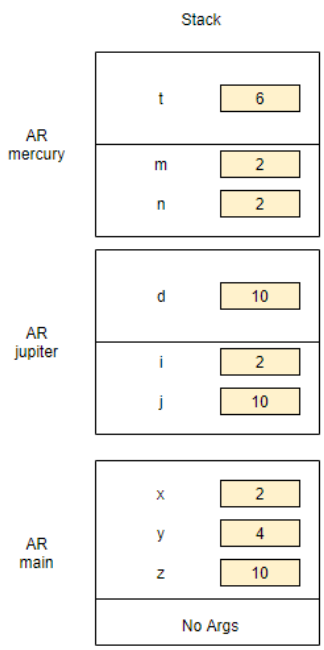
```
}
double Projectile_travel_time(double a, double v){
    double t = (2 * v * sin(a)) / G;
    return t;
}
double Projectile_travel_distance(double a, double v){
    double d = (pow(v, 2) / G) * sin(2*a);
    return d;
}
double degree_to_radian(double d){
    double a = d * (PI / 180);
    return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

Terminal Output:

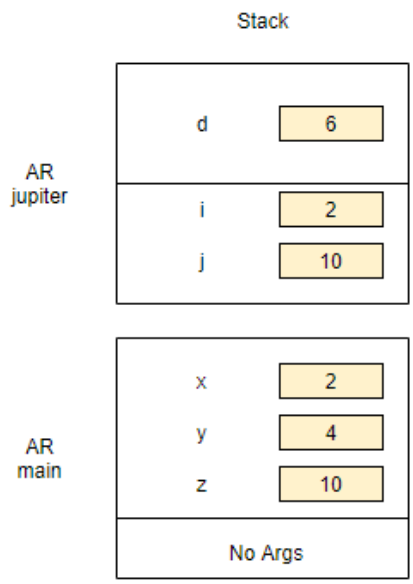
```
Please enter the velocity at which the projectile is launched (m/sec): 100
Angle(deg)      time(s)         distance(m)
0.000000        0.000000        0.000000
5.000000        1.778689        177.192018
10.000000       3.543840        349.000146
15.000000       5.282021        510.204082
20.000000       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
```

Exercise B: Drawing AR Diagrams for a Simple C Program

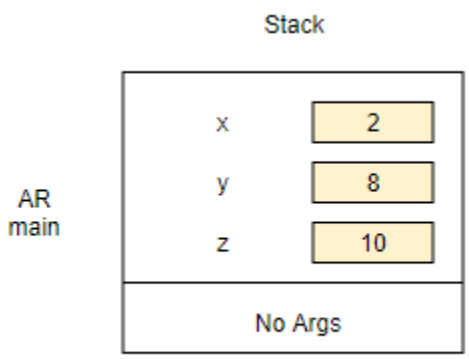
Point 1:



Point 2:

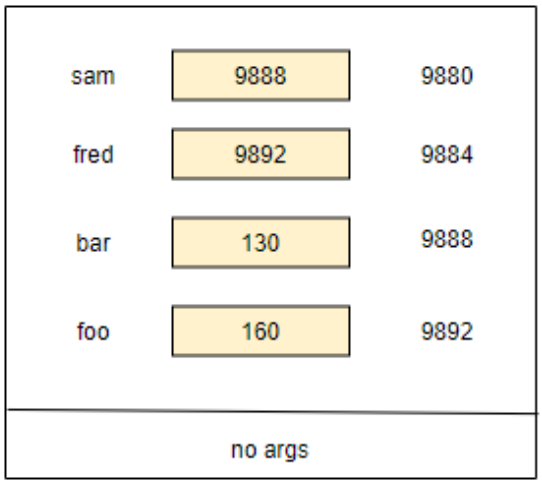


Point 3:

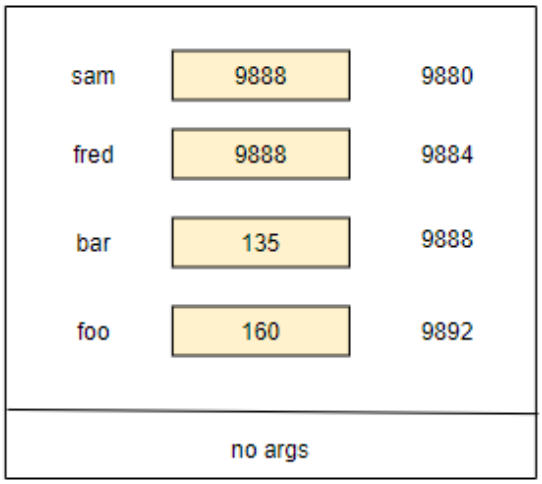


Exercise C: Introduction to Pointers

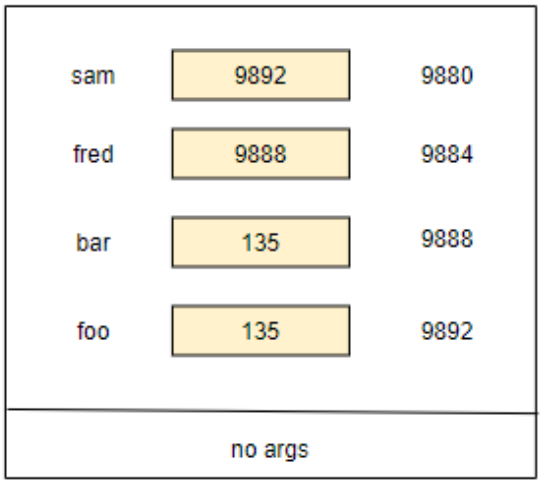
Point 2 Stack



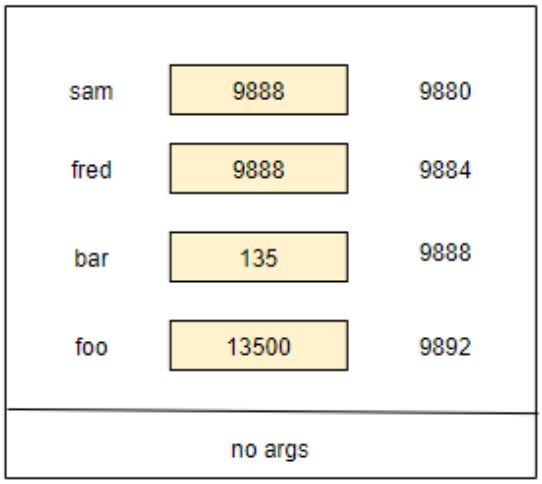
Point 3 Stack



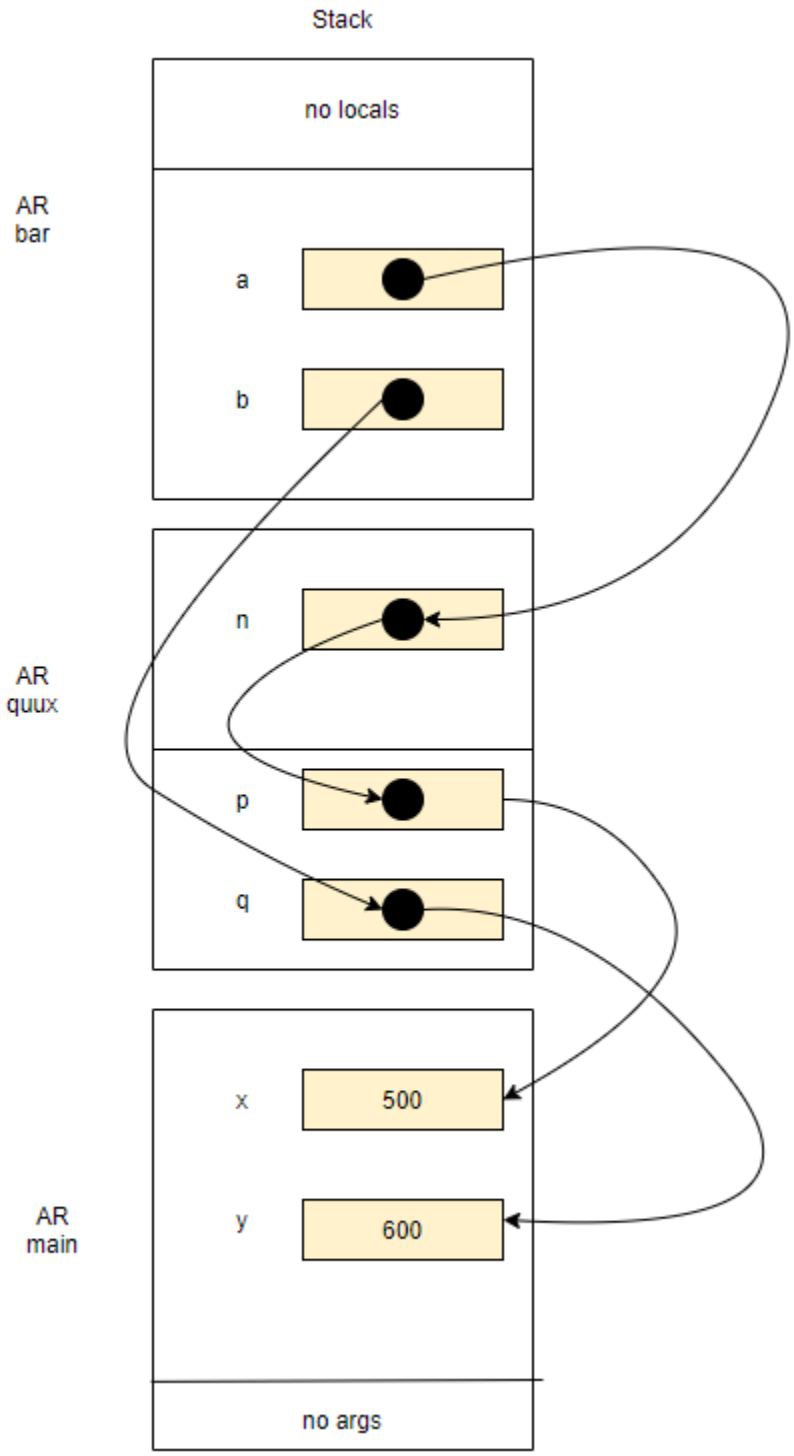
Point 4 Stack



Point 5 Stack



Exercise D: Pointers as Function Arguments



Exercise E: Using Pointers to get a Function to Change Variables

```
/*
 *  File Name: lab2exe_E.c
 *  Created by Mahmood Moussavi
 *  Assignment: Lab 2 Exercise E
 *  Completed by: Jaxon Braun
 *  Submission Date: September 28, 2021
 */

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

```
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).
```

```
  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)
{
  *minutes_ptr = (ms_time/1000)/60;
  *seconds_ptr = fmod((double)ms_time/1000, 60);
}
```

Exercise F: More on scanf

Run #	Inputs	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