

Course: Programming Fundamental - ENSF 337

Lab #: Lab 1

Instructor: M. Moussavi

Student Name: Shanzi Ye

Lab Section: B01

Date submitted: May 10, 2022

Exercise C

$$Z = x + n * y - (x + n) * y;$$

$$Z = 2.5 + 4 * (-1.5) - (2.5 + 4) * (-1.5)$$

$$Z = 2.5 - 6 - 6.5 * (-1.5)$$

$$Z = 2.5 - 6 - (-9.75)$$

$$Z = 2.5 - 6 + 9.75$$

$$Z = 6.25$$

$$z = m / n + m \% n;$$

$$Z = 18 / 4 + 18 \% 4$$

$$Z = 4 + 2$$

$$Z = 6.00$$

$$z = n / m + n \% m$$

$$Z = 4 / 18 + 4 \% 18$$

$$Z = 0 + 4$$

$$Z = 4.00$$

$$z = 5 * x - n / 5$$

$$Z = 5 * 2.5 - 4 / 5$$

$$Z = 12.5 - 0 \quad Z = 12.50$$

$$z = 1 - (1 - (1 - (1 - (1 - n))))$$

$$Z = 1 - (1 - (1 - (1 - (1 - 4))))$$

$$Z = 1 - (1 - (1 - (1 - (-3))))$$

$$Z = 1 - (1 - (1 - 4))$$

$$Z = 1 - (1 + 3)$$

$$Z = 1 - 4$$

$$Z = -3.00$$

$$z = \text{sqrt}(\text{sqrt}((\text{double})n));$$

$$Z = \text{sqrt}(\text{sqrt}((\text{double})4))$$

$$Z = \text{sqrt}(2.00)$$

$$Z = 1.414$$

Exercise D

```
/*
*File Name: lab1_exe_D.c
*Assignment: Lab 1 Exercise d
*Lab section: (B01)
*Completed by: Shanzi Ye
*Submission Date: May 10, 2022
*/

#include <stdio.h>
#include <math.h>
#define PI 3.14159265
int main()
{
    double angle;
    printf("Please input angle in units of radians: "); scanf("%lf",&angle);

    double result_built_in = sin(angle);
    printf("The built-in sin(x) value is %lf\n",result_built_in);

    Double result_taylorseries_expression = angle - pow(angle,3)/(1*2*3) +
    pow(angle,5)/(1*2*3*4*5) - pow(angle,7)/(1*2*3*4*5*6*7);
    //printf("The taylor series of sin value is %lf\n",result_taylorseries_expression );

    double degree = angle * 180 / PI;
    printf("%.1lf radians (approximately %.2lf degrees) of sin value
is %.2lf\n",angle,degree,result_taylorseries_expression );

    return 0;
}
```

```

1  /*
2  *File Name: lab1_exe_D.c
3  *Assignment: Lab 1 Exercise d
4  *Lab section: (B01)
5  *Completed by: Shanzi Ye
6  *Submission Date: May 10, 2022
7  */
8
9  #include <stdio.h>
10 #include <math.h>
11 #define PI 3.14159265
12 int main()
13 {
14     double angle;
15     printf("Please input angle in units of radians: "); scanf("%lf",&angle);
16
17     double result_built_in = sin(angle);
18     printf("The built-in sin(x) value is %lf \n",result_built_in);
19
20     double result_taylorseries_expression = angle - pow(angle,3)/(1*2*3) + pow(angle,5)/(1*2*3*4*5) - pow(angle,7)/(1*2*3*4*5*6*7);
21     //printf("The taylor series of sin value is %lf \n",result_taylorseries_expression );
22
23     double degree = angle * 180 / PI;
24     printf("%.11f radians (approximately %.21f degrees) of sin value is %.21f\n",angle,degree,result_taylorseries_expression );
25
26     return 0;
27 }
28

```

```

jackye@LAPTOP-BRG08KVA /cygdrive/d/ENSF337_SPRING
$ ./a.exe
Please input angle in units of radians: 0
The built-in sin(x) value is 0.000000
0.0 radians (approximately 0.00 degrees) of sin value is 0.00

jackye@LAPTOP-BRG08KVA /cygdrive/d/ENSF337_SPRING
$ ./a.exe
Please input angle in units of radians: 0.5
The built-in sin(x) value is 0.479426
0.5 radians (approximately 28.65 degrees) of sin value is 0.48

jackye@LAPTOP-BRG08KVA /cygdrive/d/ENSF337_SPRING
$ ./a.exe
Please input angle in units of radians: 1
The built-in sin(x) value is 0.841471
1.0 radians (approximately 57.30 degrees) of sin value is 0.84

jackye@LAPTOP-BRG08KVA /cygdrive/d/ENSF337_SPRING
$ ./a.exe1.5
-bash: ./a.exe1.5: No such file or directory

jackye@LAPTOP-BRG08KVA /cygdrive/d/ENSF337_SPRING
$ ./a.exe
Please input angle in units of radians: 1.5
The built-in sin(x) value is 0.997495
1.5 radians (approximately 85.94 degrees) of sin value is 1.00

jackye@LAPTOP-BRG08KVA /cygdrive/d/ENSF337_SPRING
$ ./a.exe
Please input angle in units of radians: 2.5
The built-in sin(x) value is 0.598472
2.5 radians (approximately 143.24 degrees) of sin value is 0.59

jackye@LAPTOP-BRG08KVA /cygdrive/d/ENSF337_SPRING
$ ./a.exe
Please input angle in units of radians: 1.0
The built-in sin(x) value is 0.841471
1.0 radians (approximately 57.30 degrees) of sin value is 0.84

```

Exercise F

```
/*
*File Name: lab1_exe_F.c
*Assignment: Lab 1 Exercise F
*Lab section: (B01)
*Completed by: Shanzi Ye
*Submission Date: May 10, 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("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;
}
```

```
void create_table(double v)
{
```

```
printf("Angle t d\n");
printf("(deg) (sec)(m)\n");
```

```
double angle = 0.000000; int i ; for(i=0;i<19;i++)
{
double totalTime = Projectile_travel_time(angle,v);
double totalDistance = Projectile_travel_distance(angle,v);
printf("%.6lf\t%.6lf\t%.6lf\n",angle,totalTime,totalDistance); angle += 5.000000;
}

}
```

```
double degree_to_radian(double d)
{
double radian = d * PI/180;
return radian;
}
```

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

```
double Projectile_travel_distance(double a, double v)
```

```

{
double radian = degree_to_radian(a);

double distance = pow(v,2)/G * sin(2*radian);
return distance;
}

```

```

1  /*
2  DaENSF337_SPRING\my_first_c_program.c
3  *Assignment: Lab 1 Exercise F
4  *Lab section: (B01)
5  *Completed by: Shanzi Ye
6  *Submission Date: May 10, 2021
7  */
8
9  #include <stdio.h>
10 #include <stdlib.h>
11 #include <math.h>
12
13 const double G = 9.8; /* gravitation acceleration 9.8 m/s^2 */ const double PI = 3.141592654;
14
15 void create_table(double v);
16 double Projectile_travel_time(double a, double v); double Projectile_travel_distance(double a, double v); double degree_to_radian(double d);
17
18 int main(void)
19 {
20     int n;
21     double velocity;
22
23     printf ("Please enter the velocity at which the projectile is launched (m/sec): "); n = scanf("%lf", &velocity);
24
25     if(n != 1)
26     {
27         printf("Invalid input. Bye...\n");
28         exit(1);
29     }
30
31     while (velocity < 0 )
32     {
33         printf ("please enter a positive number for velocity: "); n = scanf("%lf", &velocity);
34
35         if(n != 1)
36         {
37             printf("Invalid input. Bye...\n");
38             exit(1);
39         }
40     }
41
42     create_table(velocity);
43     return 0;
44 }

```

```

41
42 create_table(velocity);
43 return 0;
44 }
45
46 void create_table(double v)
47 {
48
49     printf("Angle   t   d\n");
50     printf("(deg)   (sec)   (m)\n");
51
52
53     double angle = 0.000000; int i ; for(i=0;i<19;i++)
54     {
55         double totalTime = Projectile_travel_time(angle,v);
56         double totalDistance = Projectile_travel_distance(angle,v); printf("%.6lf\t%.6lf\t%.6lf\n",angle,totalTime,totalDistance); angle += 5.000000;
57     }
58
59 }
60
61 double degree_to_radian(double d)
62 {
63     double radian = d * PI/180;
64     return radian;
65 }
66
67 double Projectile_travel_time(double a, double v)
68 {
69     double radian = degree_to_radian(a); double time = 2*v*sin(radian)/G;
70     return time;
71 }
72
73 double Projectile_travel_distance(double a, double v)
74 {
75     double radian = degree_to_radian(a);
76
77     double distance = pow(v,2)/G * sin(2*radian);
78     return distance;
79 }
80
81

```

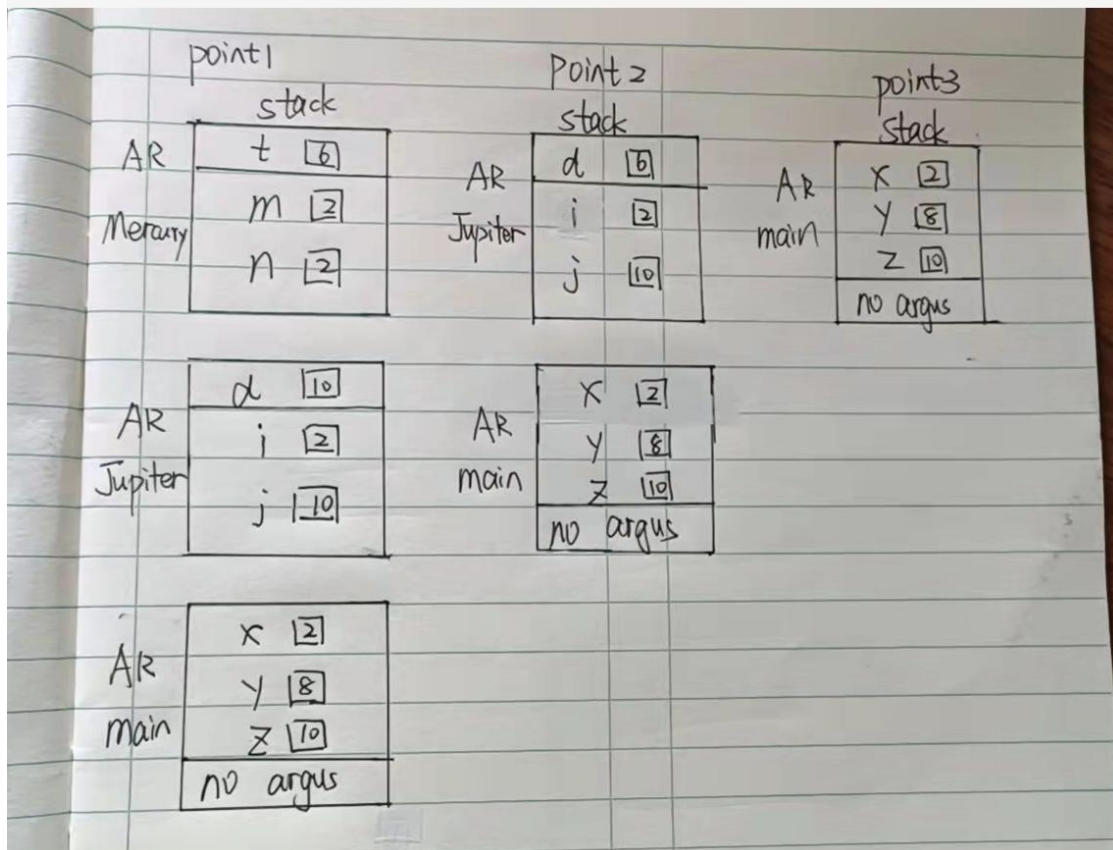
```
jackye@LAPTOP-BRG08KVA /cygdrive/d/ENSF337_SPRING
```

```
$ ./a.exe
```

```
Please enter the velocity at which the projectile is launched (m/sec): 100
```

Angle (deg)	t (sec)	d (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 G



Exercise H

AR

main

point one		
Sam	??	9880
fred	9892	9884
bar	100	9888
foo	200	
no argus		

AR

main

point two		
Sam	9888	9880
fred	9892	9884
bar	130	9888
foo	100	9892
no argus		

AR

main

point three		
Sam	9888	9880
fred	9888	9884
bar	135	9888
foo	100	9892
no argus		

AR

main

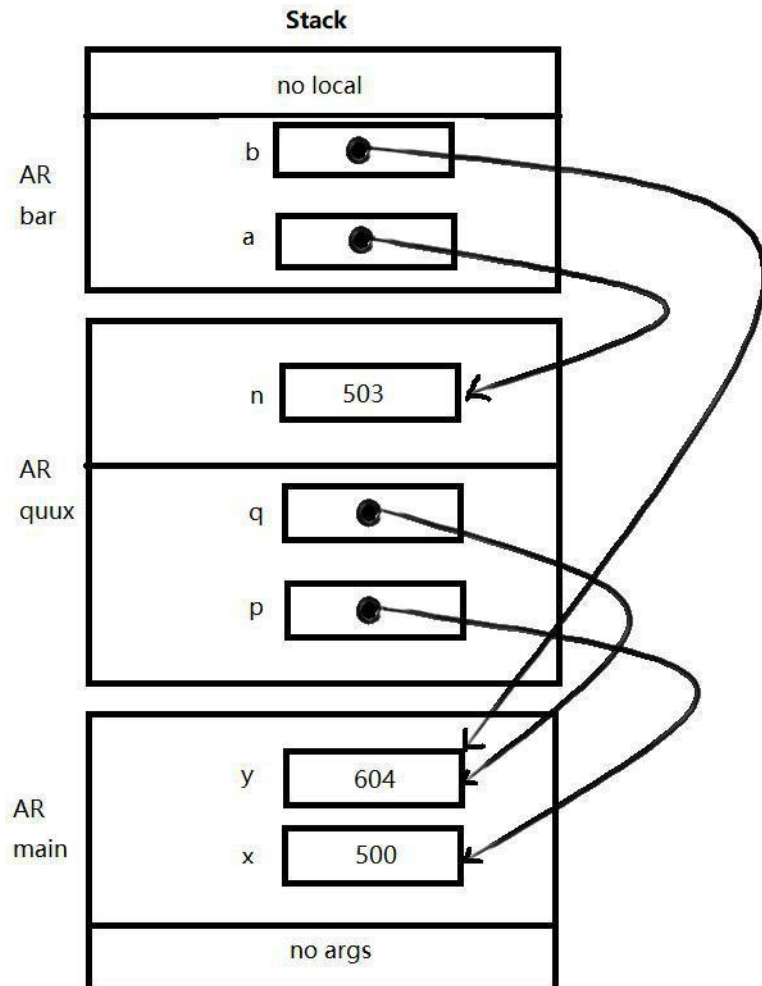
point four		
Sam	9892	9880
fred	9888	9884
bar	135	9888
foo	135	9892
no argus		

AR

main

point five		
Sam	9888	9880
fred	9888	9884
bar	135	9888
foo	13500	9892
no argus		

Exercise 1



Exercise J

```
/*
*File Name: lab2_exe_J.c
*Assignment: Lab 1 Exercise J
*Lab section: (B01)
*Completed by: Shanzi Ye
*Submission Date: May , 2021
*/

#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);

    /* MAKE A CALL TO time_convert HERE. */
    time_convert(millisec,&minutes,&seconds);
    printf("That is equivalent to %d minute(s) and %.2f second(s).\n", minutes, seconds);

    return 0;
}

void time_convert(int ms_time, int *minutes_ptr, double *seconds_ptr)
{
    int t_minutes= ms_time / (1000 * 60);
```

```

double remainder_millsec = ms_time % (1000 * 60); double t_seconds= remainder_millsec / 1000;
*minutes_ptr = t_minutes;
*seconds_ptr = t_seconds;
}

```

```

1  /*
2  *File Name: lab2_exe_J.c
3  *Assignment: Lab 1 Exercise J
4  *Lab section: (B01)
5  *Completed by: Shanzi Ye
6  *Submission Date: May , 2021
7  */
8
9  #include <stdio.h>
10 #include <stdlib.h>
11
12 void time_convert(int ms_time, int *minutes_ptr, double *seconds_ptr);
13
14
15 int main(void)
16 {
17     int millsec; int minutes; double seconds; int nscan;
18
19     printf("Enter a time interval as an integer number of milliseconds: "); nscan = scanf("%d", &millsec);
20
21     if (nscan != 1) {
22
23         printf("Unable to convert your input to an int.\n");
24         exit(1);
25     }
26
27     printf("Doing conversion for input of %d ms ... \n", millsec);
28
29     /* MAKE A CALL TO time_convert HERE. */
30     time_convert(millsec, &minutes, &seconds);
31     printf("That is equivalent to %d minute(s) and %.2f second(s).\n", minutes, seconds);
32
33     return 0;
34 }
35
36 void time_convert(int ms_time, int *minutes_ptr, double *seconds_ptr)
37 {
38     int t_minutes= ms_time / (1000 * 60);
39     double remainder_millsec = ms_time % (1000 * 60); double t_seconds= remainder_millsec / 1000;
40     *minutes_ptr = t_minutes;
41     *seconds_ptr = t_seconds;
42 }

```

```

jackye@LAPTOP-BRG08KVA /cygdrive/d/ENSF337_SPRING
$ ./a.exe
Enter a time interval as an integer number of milliseconds: 16763763
Doing conversion for input of 16763763 ms ...
That is equivalent to 279 minute(s) and 23.76 second(s).

jackye@LAPTOP-BRG08KVA /cygdrive/d/ENSF337_SPRING
$ ./a.exe
Enter a time interval as an integer number of milliseconds: 473845
Doing conversion for input of 473845 ms ...
That is equivalent to 7 minute(s) and 53.84 second(s).

jackye@LAPTOP-BRG08KVA /cygdrive/d/ENSF337_SPRING
$ ./a.exe
Enter a time interval as an integer number of milliseconds: 3463674834
Doing conversion for input of -831292462 ms ...
That is equivalent to -13854 minute(s) and -52.46 second(s).

jackye@LAPTOP-BRG08KVA /cygdrive/d/ENSF337_SPRING
$ ./a.exe
Enter a time interval as an integer number of milliseconds: 64832749
Doing conversion for input of 64832749 ms ...
That is equivalent to 1080 minute(s) and 32.75 second(s).

```