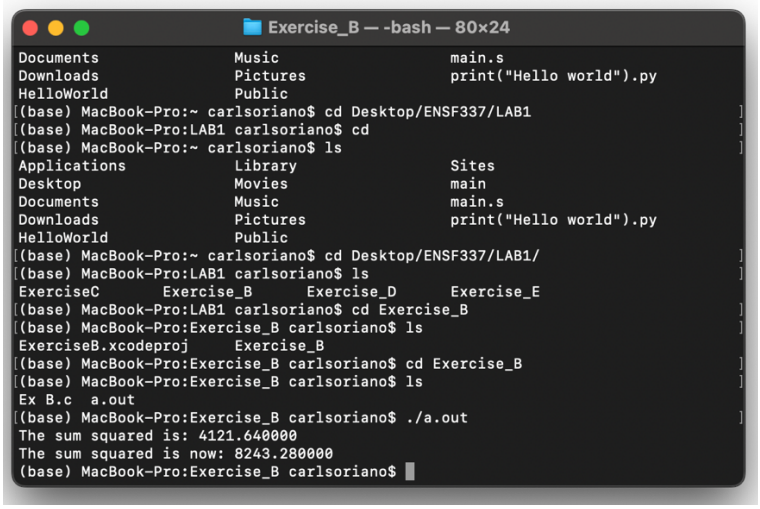
Course: Programming Fundamental – ENSF 337
Lab #: Lab 1
Instructor: M. Moussavi
Student Name: Carl Soriano
Lab Section: B01
Date submitted: Sept 21, 2022

Exercise B

```c
/*
* File Name:
* Assignment: Lab 1 Exercise B
* Lab section: B01
* Completed by: Carl Soriano
* Submission Date: Sept 18, 2021
*/

#include <stdio.h>
int main(){
    // insert code here...
    double num1 = -34.5;
    double num2 = 98.7;
    double sum; // sum of num1 and num2
    double sumSquared; // the square of num2 plus num2

    sum = num1 + num2;

    sumSquared = sum*sum;

    // 1) Add the two numbers and store the result in the variable 'sum'
    // 2) Compute the square of the sum and store the result in the variable 'sumSquared'
    // Use the variable 'sum' (computed above) for this computation
    printf( "The sum squared is: %f \n", sumSquared);

    sumSquared = sumSquared*2;
    // 3) Now double the sum squared value and store the result in 'sumSquared'
    printf( "The sum squared is now: %f \n", sumSquared);

    return 0;
}
```



Exercise C

a) `z = x + n * y - (x + n) * y;` = 6.25
b) `z = m / n + m % n;` = 6.0
c) `z = n / m + n % m;` = 4.0
d) `z = 5 * x - n / 5;` = 12.5
e) `z = 1 - ( 1 - ( 1 - ( 1 - ( 1 -n))));` = -3.0
f) `z = sqrt(sqrt((double)n);` = 1.414214

a) 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 = 6.25
```

b) z = m / n + m % n;

```
z = 18 / 4 + 18 % 4
z = 4 + 2
z = 6.0
```

c) z = n / m + n % m;

```
z = 4 / 18 + 4 % 18
z = 0 + 4
z = 4.0
```

d) z = 5 * x - n / 5;

```
z = 5 * 2.5 - 4 / 5
z = 12.5 - 0
z = 12.5
```

e) 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 = -3.0
```

f) z = sqrt(sqrt((double)n));

```
z = sqrt(sqrt((double)4)
z = sqrt(2.0)
z = 1.414
```

Exercise D

```c
//
//  main.c
//  Exercise_D
//
//  Created by Carl  Soriano on 2022-09-17.
//
/*
* File Name:
* Assignment: Lab 1 Exercise D
* Lab section: B01
* Completed by: Carl Soriano
* Submission Date: Sept 21, 2021
*/
#include <stdio.h>
#include <math.h>

double taylor_series(double x);
double factorial(double x);

int main(){
    double x;
    //prompt user for value
    printf("Enter radian value here: ");
    scanf("%lf", &x);
    //calculate sin(x)
    printf("Value input: %lf \n",(x));
    printf("Calculated sine value: %lf \n", sin(x));
    //calculate taylor series
    double a = taylor_series(x);
    printf("Taylor Series Approximation: %lf \n", a);


    return 0;
}
//taylor series function
double taylor_series(double x)
{
    double c;
    {
        c = x - (pow(x,3)/6) + (pow(x,5)/120) - (pow(x,7)/5040);
    }
    return c;
}
```

```
(base) MacBook-Pro:Exercise_D carlsoriano$ ./a.out
Enter radian value here: .5
Value input: 0.500000
Calculated sine value: 0.479426
Taylor Series Approximation: 0.479426
(base) MacBook-Pro:Exercise_D carlsoriano$ ./a.out
Enter radian value here: 0
Value input: 0.000000
Calculated sine value: 0.000000
Taylor Series Approximation: 0.000000
(base) MacBook-Pro:Exercise_D carlsoriano$
```

```
(base) MacBook-Pro:Exercise_D carlsoriano$ ./a.out
Enter radian value here: 1
Value input: 1.000000
Calculated sine value: 0.841471
Taylor Series Approximation: 0.841468
(base) MacBook-Pro:Exercise_D carlsoriano$ ./a.out
Enter radian value here: 2.5
Value input: 2.500000
Calculated sine value: 0.598472
Taylor Series Approximation: 0.588534
(base) MacBook-Pro:Exercise_D carlsoriano$
```

Radian Value	Sin Value	Taylor Series
.5	0.479426	0.479426
0	0	0
1	0.841471	0.841468
2.5	0.598472	0.588534

Exercise E

```c
//  main.c
//  Exercise_E
//
//  Created by Carl  Soriano on 2022-09-20.
//
/*
* File Name:
* Assignment: Lab 1 Exercise E
* Lab section: B01
* Completed by: Carl Soriano
* Submission Date: Sept 21, 2021
*/

#include <stdio.h>
#include <math.h>
int main() {

    double a,b,c,square,r1,r2,complex_real,complex_imaginary;
    //prompt user for entered values
    printf("Enter Coefficients in order a, b and c:");

    scanf("%lf %lf %lf", &a, &b, &c);
    //under the square root varialble
    square = b * b - 4 * a * c;
    //accounts for if the value/discriminat is negative
    if (square > 0) {
        r1 = (-b + sqrt(square)) / (2 * a);

        r2 = (-b - sqrt(square)) / (2 * a);

        printf("r1 = %lf and r2 = %lf", r1, r2);
    }
    //accounts for if the roots are equal
    else if (square == 0){
        r1 = r2 = -b / (2 * a);

        printf("r1 = r2 = %lf", r1);

    }
    //accounts for imaginary and real roots
    else {

        complex_real = -b / (2 * a);

        complex_imaginary = sqrt(-square) / (2 * a);

        printf("r1 = %lf + %lfi and r2 = %f - %fi \n", complex_real,
complex_imaginary, complex_real, complex_imaginary);

    }

    return 0;
}
```

```
(base) MacBook-Pro:lab1_exe_E.c carlsoriano$ ./a.out
Enter Coefficients in order a, b and c:2
5
7
r1 = -1.250000 + 1.391941i and r2 = -1.250000-1.391941i
(base) MacBook-Pro:lab1_exe_E.c carlsoriano$ ./a.out
Enter Coefficients in order a, b and c:3
5
8
r1 = -0.833333 + 1.404358i and r2 = -0.833333-1.404358i
(base) MacBook-Pro:lab1_exe_E.c carlsoriano$ ./a.out
Enter Coefficients in order a, b and c:1
9
6
r1 = -0.725083 and r2 = -8.274917(base) MacBook-Pro:lab1_exe_E.c carlsoriano$
```

A	B	C	R1	R2
2	5	7	-1.250000+1.391941i	-1.250000-1.391941i
3	5	8	-0.833333+1.404358i	-0.833333-1.404358i
1	9	6	-0.725083	8.274917