DOCUMENTATION FOR ECPROG FINALS PROJECT

5	Nuñez	Complex Numbers	Header File : Implementation File : Test Program : Documentation :	nunez_complexnum.h nunez_implementation.c nunez_test.c nunez_documentaion.pdf
Write a header file that defines a structure for complex numbers and functions for basic operations: addition, subtraction, multiplication, and division of complex numbers.			Addition of complex numbers Subtraction of complex numbers Multiplication of complex numbers Division of complex numbers	

Figure1.0 & 1.1 indicates the unique problem assigned to me

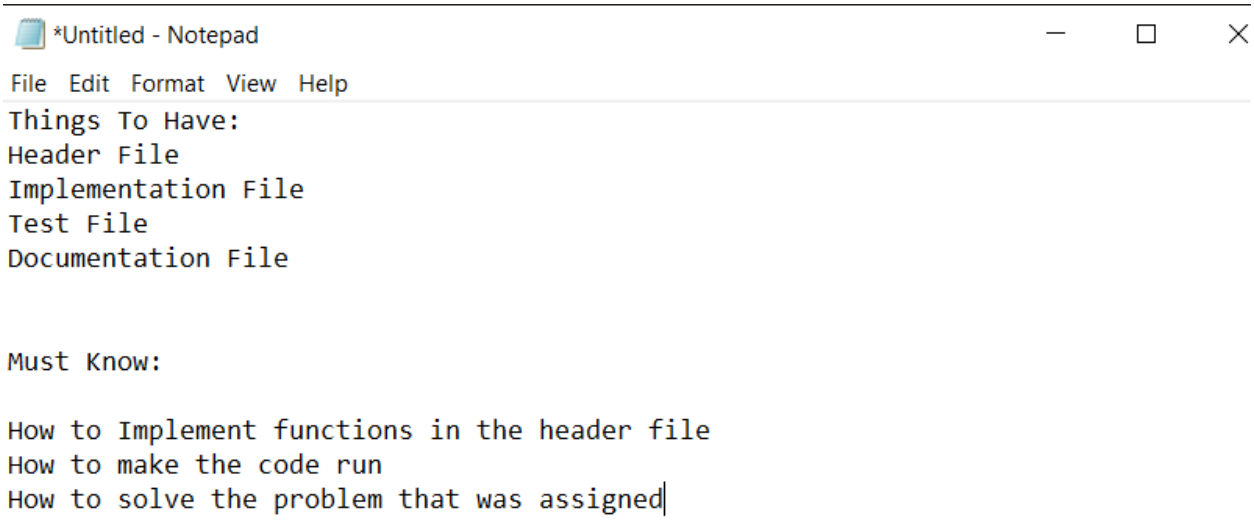


Figure 1.2 shows the notes the important aspects that needs to be applied during the initial run of the project

```c
#include <stdio.h>
int main () {
    int num1; int num2;
    int choice;

    printf("Enter the first complex number: ");
    scanf("%.2lf", num1);
    printf("Enter the second complex number: ");
    scanf("%.2lf", num2);

    printf("Choose an operation:\n");
    printf("1. Addition\n");
    printf("2. Subtraction\n");
    printf("3. Multiplication\n");
    printf("4. Division\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);
}
```

Figure 1.3 shows the first test for the finals project. (may have a lot of errors)

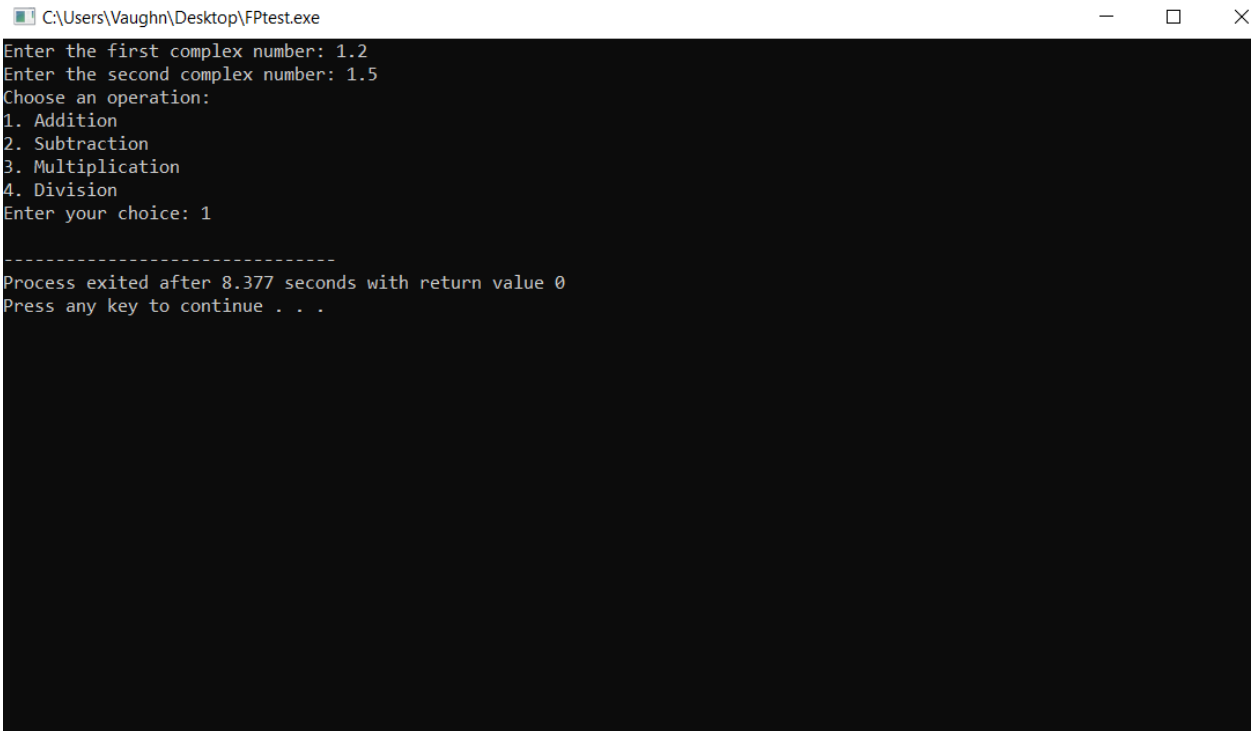


Figure 1.4 shows the first run of the code and has made a successful run despite the incompleteness of code

```c
switch (choice) {
    case 1:
        printf("Result of addition: %.2f\n", num1 + num2);
        break;
    case 2:
        printf("Result of subtraction: %.2f\n", num1 - num2);
        break;
    case 3:
        printf("Result of multiplication: %.2f\n", num1 * num2);
        break;
    case 4:
        if (num2 != 0) {
            printf("Result of division: %.2f\n", num1 / num2);
        } else {
            printf("Error: Division by zero\n");
        }
        break;
    default:
        printf("Invalid choice\n");
```

```
C:\Users\Vaughn\Desktop\FPtest.exe
Enter the first complex number: 1.5
Enter the second complex number: 2
Choose an operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
Enter your choice: 3
Result of multiplication: 3.00

--------------------------------
Process exited after 7.295 seconds with return value 0
Press any key to continue . . .
```

Figure 1.5 & 1.6 show the switch case code and its successful run of the code

```
Code must Have:
Function Prototypes
Complex Numbers
Imaginary Numbers
printf for complex and imaginary (undecided whether to make it in 1 printf or separate)
```

Figure 2.0 shows the next obstacle that needs to be tackled

```c
#ifndef NUNEZ_COMPLEXNUM_H
#define NUNEZ_COMPLEXNUM_H



typedef struct {
    double real;
    double imag;
} Complex;


// Functions that will be used in the implementation of the header file
Complex inputComplex();
Complex addComplex(Complex a, Complex b);
Complex subtractComplex(Complex a, Complex b);
Complex multiplyComplex(Complex a, Complex b);
Complex divideComplex(Complex a, Complex b);

#endif
```

Figure 2.1 shows the sample header file that is being used (will also edit the previous codes to match the current)

```c
Complex inputComplex() {
    Complex c;
    printf("Enter the real part: ");
    scanf("%lf", &c.real);
    printf("Enter the imaginary part: ");
    scanf("%lf", &c.imag);
    return c;
```

Figure 2.2 shows the code that will input the real complex number and the imaginary complex number

>decided to separate it to not cause any errors (tackled from the activities given)

```
// Function to add two complex numbers
Complex addComplex(Complex a, Complex b) {
    Complex result;
    result.real = a.real + b.real;
    result.imag = a.imag + b.imag;
    return result;
}

// Function to subtract two complex numbers
Complex subtractComplex(Complex a, Complex b) {
    Complex result;
    result.real = a.real - b.real;
    result.imag = a.imag - b.imag;
    return result;
}
```

```
// Function to multiply two complex numbers
Complex multiplyComplex(Complex a, Complex b) {
    Complex result;
    result.real = a.real * b.real - a.imag * b.imag;
    result.imag = a.real * b.imag + a.imag * b.real;
    return result;
}

// Function to divide two complex numbers
Complex divideComplex(Complex a, Complex b) {
    Complex result;
    double denominator = b.real * b.real + b.imag * b.imag;
    result.real = (a.real * b.real + a.imag * b.imag) / denominator;
    result.imag = (a.imag * b.real - a.real * b.imag) / denominator;
    return result;
}
```

Figure 2.3 & 2.4 shows the functions for each operator when given complex numbers

```c
int main() {
    Complex a, b, result;
    int choice;

    printf("Enter the first complex number:\n");
    a = inputComplex();

    printf("Enter the second complex number:\n");
    b = inputComplex();

    printf("Choose an operation:\n");
    printf("1. Addition\n");
    printf("2. Subtraction\n");
    printf("3. Multiplication\n");
    printf("4. Division\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch (choice) {
        case 1:
            result = addComplex(a, b);
            printf("Result: %.2f + %.2fi\n", result.real, result.imag);
            break;
        case 2:
            result = subtractComplex(a, b);
            printf("Result: %.2f + %.2fi\n", result.real, result.imag);
            break;
        case 3:
            result = multiplyComplex(a, b);
            printf("Result: %.2f + %.2fi\n", result.real, result.imag);
            break;
        case 4:
            result = divideComplex(a, b);
            printf("Result: %.2f + %.2fi\n", result.real, result.imag);
            break;
        default:
            printf("Invalid choice\n");
            break;
    }
```

Figure 2.5 & 2.6 shows the previous code (Figure 1.3 & 1.5) being edited to be more in line with the function and headers

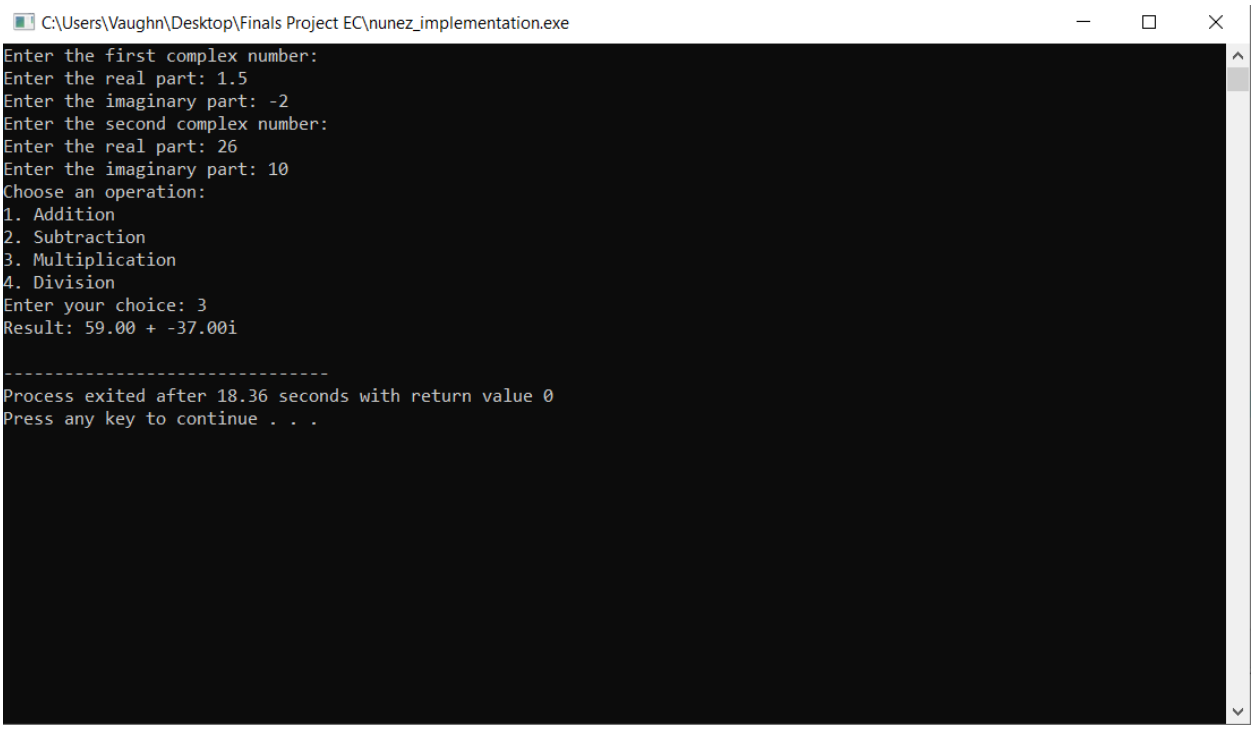


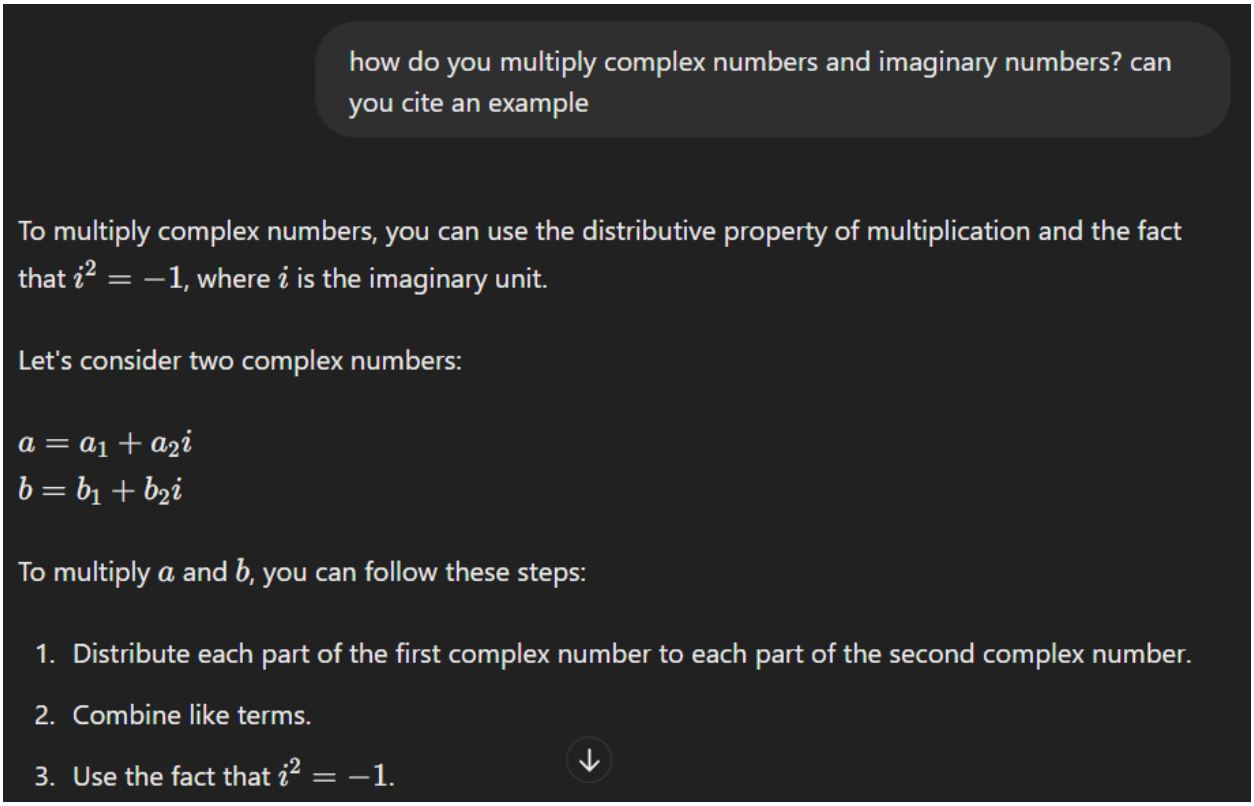
Figure 2.7 shows the code being run through and successfully performs its given task

```c
#include <stdio.h>
#include "nunez_complexnum.h"

// Function to input a complex number
Complex inputComplex() {
    Complex c;
    printf("Enter the real part: ");
    scanf("%lf", &c.real);
    printf("Enter the imaginary part: ");
    scanf("%lf", &c.imag);
    return c;
}
```

Figure 2.8 shows the implementation file calling the header file

Bonus Figures:



Shows how the user asked ChatGPT how to multiply complex and imaginary numbers

>TLDR so had to rephrase it and was given the next figure

Example

Let's multiply $2 + 3i$ and $1 + 4i$:

$$a = 2, \quad b = 3, \quad c = 1, \quad d = 4$$

Using the formula:

1. $ac = 2 \cdot 1 = 2$
2. $ad = 2 \cdot 4 = 8$
3. $bc = 3 \cdot 1 = 3$
4. $bd = 3 \cdot 4 = 12$

So,

$$(2 + 3i)(1 + 4i) = (2 - 12) + (8 + 3)i$$
$$= -10 + 11i$$

Tested it out and

```
Enter the first complex number:
Enter the real part: 2
Enter the imaginary part: 3
Enter the second complex number:
Enter the real part: 1
Enter the imaginary part: 4
Choose an operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
Enter your choice: 3
Result: -10.00 + 11.00i

--------------------------------
Process exited after 22.5 seconds with return value 0
Press any key to continue . . .
```

Given the same results