Problem 1:

Using good coding practices, design your own `ComplexVector` class for vectors whose elements are complex numbers. Provide an overloading implementation for term by term addition, subtraction, multiplication, division, and the stream output operator. Such that if `v1` and `v2` are `ComplexVector` objects, then one can compute `v1 + v2`, etc. *Hint* : you may want to create a `Complex` class separately.

Complex numbers are expressed in the form $a+bi$, where a is the real part, b is the imaginary part, and i is the imaginary unit satisfying $i^2 = -1$. Print your complex numbers using this representation. When adding (or subtracting) two complex numbers, add (or subtract) their real parts and imaginary parts separately. The multiplication of two complex numbers is defined by,

$$(a + bi)(c + di) = (ac - bd) + (bc + ad)i, \tag{1}$$

while division is given by,

$$\frac{a + bi}{c + di} = \frac{ac + bd}{c^2 + d^2} + \frac{bc - ad}{c^2 + d^2}i. \tag{2}$$

Using your newly defined `ComplexVector` class, write a recursive function to print the first 6 terms in the sequence from equation (3) to a file titled "ComplexSequence.txt".
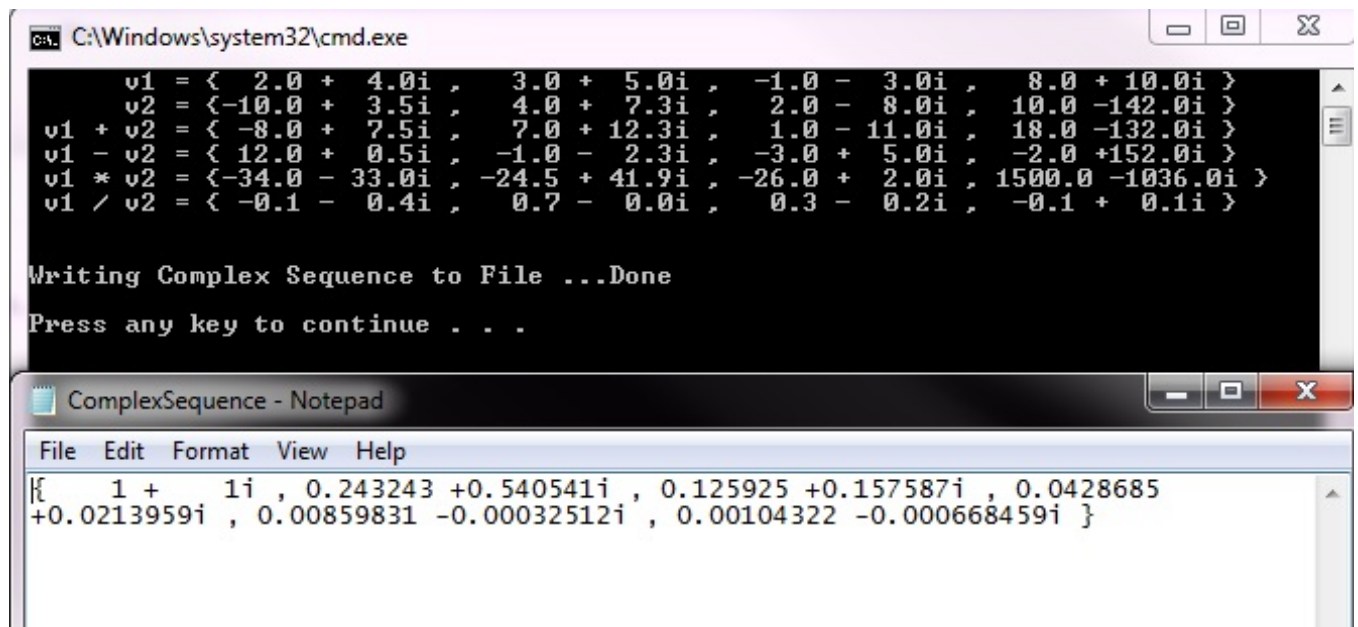
$$f_{n+1} = \frac{(2 + 3\,i)\,n}{7 + 5\,n^2\,i} f_n, \tag{3}$$

$$f_1 = 1 + i \tag{4}$$

Using the vectors `v1` and `v2` ,

```
v1 = { 2 + 4 i, 3 + 5 i, -1 - 3 i , 8 + 10 i }
v2 = { -10 + 3.5 i, 4 + 7.3 i, 2 - 8 i, 10 -142 i},
```

write a main function that checks `v1 + v2`, `v1 - v2`, `v1 * v2`, and `v1 / v2` by printing them to the console. Figure 1 shows the sample outputs.

Figure 1: Sample output.

Instructions for submission:

- Name your main file exactly `hw4.cpp`, and the rest being `Complex.h`, `Complex.cpp`, `ComplexVector.h`, and `ComplexVector.cpp`.

- You may not use `#include "stdafx.h"`.

- Add code description in the comment at the beginning of the file.
  A sample description may look like:

  ```
  /*
      PIC 10B 2A, Homework 1
      Purpose: Tic-tac-toe game
      Author: Hanqin Cai
      Date: 10/10/2019
  */
  ```

- Submit your header files and source codes to CCLE in separate files. Only .h and .cpp files should be uploaded.

Grading

- Good Coding Practices (5 points)

- Compiling (5 points)

- Takes Correct Input (3 points)

- Gives Correct Output (3 points)

- Code (4 points)