CS2211a Assignment 5

Issued on: Thursday, November 20, 2014

Due by: Thursday November 27, 2014 at 11:55 pm,

- For this assignment, ***only electronic submission*** at owl.uwo.ca is required.
- ONLY user ***Courier New*** (*size = 11 pts.*)
- ***Start each question in a NEW PAGE***
- ***Write the question number is a separate line followed by an empty line***
- After finishing the assignment, you have to do the following:
  - Type your report and convert it to the PDF format (*no handwriting*),
  - The report should include:
    - Answers to *all* questions/requirements in the assignment
    - Enough test cases (as well as sample outputs) to demonstrate and cover all possible options in your program
    - A copy of all programs that you have written
  - Prepare a soft-copy submission, including:
    - A copy of your *typed* report
    - All programs that you wrote (*each program in a file*)—*2 in total* (use meaningful program names). These files **MUST BE** text ASCII files. Do not submit them as PDF.
  - Upload the soft-copy submission file-by-file (*3 in total*), or as an archived directory.

Failure to follow the above format may cost you 10% of the total assignment mark.

- Late assignments are strongly discouraged
  - 10% will be deducted from a late assignment (up to 24 hours after the due date/time)
  - After 24 hours from the due date/time, late assignments will receive a zero grade.

(100 marks)

A *complex number* is a number that can be expressed in the form $a + i\,b$, where a and b are real numbers and i is the imaginary unit, which satisfies the equation $i^2 = -1$. In this expression, a is the *real part* and b is the *imaginary part* of the *complex number*.

Complex numbers extend the concept of the one-dimensional number line to the two-dimensional complex plane by using the horizontal axis for the real part and the vertical axis for the imaginary part.

Declare a tag named `complex_t` for a structure with two members, `real` and `imaginary`, of type `double`.

Write a function that accepts two parameters of type `complex_t` and returns a `complex_t` of their multiplication.

Note that: if c_1 and c_2 are two complex numbers, where $c_1 = a_1 + i\,b_1$, $c_2 = a_2 + i\,b_2$, then

$$c_1 \times c_2 = (a_1 \times a_2 - b_1 \times b_2) + i\,(a_2 \times b_1 + a_1 \times b_2)$$

Write a function that accepts two parameters of type **pointer** to `complex_t` and returns a **pointer** to a `complex_t` of their division as a **pointer** to a `complex_t`.

Hint, you will need to ***malloc*** a memory for the returned **pointer** to a `complex_t`. Make sure that ***malloc*** is successful before using its output. Otherwise, you need to print an error message and exit the program.

Note that: if c_1 and c_2 are two complex numbers, where $c_1 = a_1 + i\,b_1$, $c_2 = a_2 + i\,b_2$, then

$$c_1 \div c_2 = (a_1 \times a_2 + b_1 \times b_2) \div (a_2^2 + b_2^2) + i\,(a_2 \times b_1 - a_1 \times b_2) \div (a_2^2 + b_2^2), \quad \textbf{where } (a_2^2 + b_2^2) \neq 0$$

Write a main program that

- Declares four variables of type `complex_t`.
- Prompts the user to input values from the standard input to initialize two of these structure variables.
- Passes these initialized two structure variables to the two functions described above and store the results in the other two variables.
- Prints the returned structure values that indicate the multiplication and the division of the two complex numbers.

Modify your program and your two functions by declaring `complex_t` using *typedef*.
Make sure that your program is up and correctly running after doing the modifications.

You should include as many test cases as possible to demonstrate various cases, e.g., dealing with

- ✓ two complex numbers,
- ✓ two real numbers,
- ✓ two imaginary numbers,
- ✓ a real number and an imaginary number,
- ✓ an imaginary number and a real number,
- ✓ a zero and a complex number, and
- ✓ a complex number and a zero.