

EECS 10: Assignment 6

October 30, 2009

Due Monday 11/16/09 at 12:00pm Note: This is a two-week assignment.

1 Calculations with Complex Numbers [80 points]

1.1 Background knowledge of complex numbers

In mathematics, a **complex number** is a number of the form:

$$x + yi$$

where x and y are real numbers, and i is the imaginary unit, with the property $i^2 = -1$. The real number x is called the *real part* of the complex number, and the real number y is the *imaginary part*. For example, $6 + 7i$ is a complex number, with real part 6 and imaginary part 7.

To refresh your memory on complex numbers, consult your math books and/or use one of the following links:

<http://www.clarku.edu/~djoyce/complex/>
<http://mathworld.wolfram.com/ComplexNumber.html>

Alternatively to the regular form $z = x+yi$, the complex number z can be specified by polar coordinates:

$$r @ \Theta \text{ degrees}$$

where r is the *magnitude* and Θ is the *phase*. Conversion from the regular form to the polar form:

$$r = |z| = \text{sqrt}(x^2 + y^2)$$

x	Y	Θ (in radians)
$x > 0$		$\arctan(y/x)$
$x < 0$	$y \geq 0$	$\arctan(y/x) + \pi$
$x < 0$	$y < 0$	$\arctan(y/x) - \pi$
$x = 0$	$y > 0$	$+\pi/2$

x	Y	Θ (in radians)
$x=0$	$y<0$	$-\pi/2$
$x=0$	$y=0$	0

Note that we will use degrees as the units for the phase (not radians), so you need to convert the phase from radians to degrees. (Recall that 180 degrees = π radians, using $\pi = 3.1415926$ for our case)

Since $x + yi = r \cos\theta + i r \sin\theta$, to convert the polar coordinates into regular form, you may use:

$$\begin{aligned}x &= r * \cos\theta \\y &= r * \sin\theta\end{aligned}$$

where r is the magnitude and θ is the phase.

Hint:

C provides functions that perform sqrt, arctan, cos and sin operations. To use those functions, you need to include math.h as a header file:

```
#include <math.h>
```

The functions C provides to perform sqrt, arctan, cos and sin operations are:

```
double sqrt(double value);
double atan(double value);
double cos(double value); /*The unit of value is radians*/
double sin(double value); /*The unit of value is radians*/
```

You will need to include an extra option “-lm” when compiling programs that use functions from “math.h”. This option tells the linker where to find the libraries required to use the math functions.

```
gcc complex.c -o complex -lm -ansi -Wall
```

1.2 Problem Description

Your program should be a menu driven program. At the start of your program, the user is prompted to input a complex number by providing the real part and the imaginary part. For example:

```
Welcome to my complex number calculator!
Enter a complex number.
Enter the real part: 3
Enter the imaginary part: 5
```

Then the program should print out the current complex number in both regular and polar forms, followed by a menu that allows users to select an operation.

In our example, it looks like this:

```
Current number is 3.000 + 5.000i
Polar Notation: 5.831 @ 59.036 degrees
```

- (1) Set the complex number in regular format
- (2) Set the complex number in polar format
- (3) Compute the absolute value of the number
- (4) Compute the reciprocal of the number
- (5) Add another complex number
- (6) Subtract another complex number
- (7) Multiply by another complex number
- (8) Divide by another complex number
- (9) Exit

Enter your choice:

Your program should perform the following operations for the above ten options:

(1) Set the complex number in regular format

This function should allow users to set the value of the complex number by entering the real and imaginary parts.

```
Enter your choice: 1
Enter the real part: 2.0
Enter the imaginary part: 7.0
```

```
Current number is 2.000 + 7.000i
Polar Notation: 7.280 @ 74.055 degrees
```

- (1) Set the complex number in regular format
- (2) Set the complex number in polar format
- (3) Compute the absolute value of the number
- (4) Compute the reciprocal of the number
- (5) Add another complex number
- (6) Subtract another complex number
- (7) Multiply by another complex number
- (8) Divide by another complex number
- (9) Exit

Enter your choice:

(2) Set the complex number in polar format

This function should allow the user to set the value of the complex number by entering the magnitude and phase (in degrees).

```
Enter your choice: 2
Enter the magnitude: 8.5
Enter the phase: 60
```

```
Current number is 4.250 + 7.361i
Polar Notation: 8.500 @ 60.000 degrees
```

- (1) Set the complex number in regular format
- (2) Set the complex number in polar format
- (3) Compute the absolute value of the number
- (4) Compute the reciprocal of the number
- (5) Add another complex number
- (6) Subtract another complex number
- (7) Multiply by another complex number
- (8) Divide by another complex number
- (9) Exit

```
Enter your choice:
```

Then your program should print out the current complex number in both regular and polar forms and print out the menu again.

(3) Compute the absolute value of the current complex number

Note: your program should compute the absolute value of the current number and display it on the screen. The current complex number should not change. For example

```
Enter your choice: 3
The absolute value of the current number is 8.500
Current number is 4.250 + 7.361i
Polar Notation: 8.500 @ 60.000 degrees
```

- (1) Set the complex number in regular format
- (2) Set the complex number in polar format
- (3) Compute the absolute value of the number
- (4) Compute the reciprocal of the number
- (5) Add another complex number
- (6) Subtract another complex number
- (7) Multiply by another complex number
- (8) Divide by another complex number
- (9) Exit

```
Enter your choice:
```

Hint: The absolute value $|z|$ can be calculated using the following formula:

$$|z| = \text{sqrt}(x^2 + y^2)$$

(4) Compute the reciprocal of the current complex number

Note: your program should calculate the reciprocal of the current number and store the result back as the new current number. For example:

```
Current number is 4.250 + 7.361i
Polar Notation: 8.500 @ 60.000 degrees
```

- (1) Set the complex number in regular format
- (2) Set the complex number in polar format
- (3) Compute the absolute value of the number
- (4) Compute the reciprocal of the number
- (5) Add another complex number
- (6) Subtract another complex number
- (7) Multiply by another complex number
- (8) Divide by another complex number
- (9) Exit

```
Enter your choice: 4
```

```
Current number is 0.059-0.102i
Polar Notation: 0.118 @ -60.000 degrees
```

Hint: The reciprocal is computed as follows:

$$1 / (x + yi) = x / (x^2 + y^2) - yi / (x^2 + y^2)$$

Error handling: if x^2+y^2 is zero, in the main function an error message should be printed because the denominator in a division operation cannot be zero.

```
Current number is 0.000 + 0.000i
Polar Notation: 0.000 @ 0.000 degrees
```

- (1) Set the complex number in regular format
- (2) Set the complex number in polar format
- (3) Compute the absolute value of the number
- (4) Compute the reciprocal of the number
- (5) Add another complex number
- (6) Subtract another complex number
- (7) Multiply by another complex number
- (8) Divide by another complex number
- (9) Exit

```
Enter your choice: 4
```

```
Error! The selected reciprocal operation cannot be
performed!
Current number is 0.000 + 0.000i
```

Polar Notation: 0.000 @ 0.000 degrees

(5) Add another complex number

First, user should enter a complex number that will be added to the current complex number. Then, the program should compute the addition of the two numbers (the current complex number and the newly entered one) and assign the result back to the current complex number.

Current number is 0.000 + 0.000i
Polar Notation: 0.000 @ 0.000 degrees

- (1) Set the complex number in regular format
- (2) Set the complex number in polar format
- (3) Compute the absolute value of the number
- (4) Compute the reciprocal of the number
- (5) Add another complex number
- (6) Subtract another complex number
- (7) Multiply by another complex number
- (8) Divide by another complex number
- (9) Exit

Enter your choice: 5

Enter the real part of the operand:2
Enter the imaginary part of the operand:7

Current number is 2.000 + 7.000i
Polar Notation: 7.280 @ 74.055 degrees

Hint: this is the formula to calculate the sum of two complex numbers

$$(a + bi) + (c + di) = (a + c) + (b + d)i$$

(6) Subtract another complex number

This should follow the very same procedure as for option (5), except that the second complex number is subtracted from the current number.

$$(a + bi) - (c + di) = (a - c) + (b - d)i$$

(7) Multiply by another complex number

Again, this is very similar to option (5). You should use the following formula to perform the multiplication operation.

$$(a + bi)(c + di) = ac + bci + adi + bdi^2 = (ac - bd) + (bc + ad)i$$

(8) Divide by another complex number

Again, this is very similar to option (5). Here, however, this time the current number should be divided by the second complex number.

$$\frac{(a + bi)}{(c + di)} = \left(\frac{ac + bd}{c^2 + d^2} \right) + \left(\frac{bc - ad}{c^2 + d^2} \right) i$$

Error handling: if c^2+d^2 is zero, in the main function an error message should be printed because the denominator in a division operation cannot be zero.

Current number is 2.000 + 7.000i
Polar Notation: 7.280 @ 74.055 degrees

- (1) Set the complex number in regular format
- (2) Set the complex number in polar format
- (3) Compute the absolute value of the number
- (4) Compute the reciprocal of the number
- (5) Add another complex number
- (6) Subtract another complex number
- (7) Multiply by another complex number
- (8) Divide by another complex number
- (9) Exit

Enter your choice: 8

Enter the real part of the operand:0
Enter the imaginary part of the operand:0

Error! The selected division operation cannot be performed.

Current number is 2.000 + 7.000i
Polar Notation: 7.280 @ 74.055 degrees

(9)Exit

This option will terminate your program.

Note:

If the user enters an option that is not among the specified options, you should prompt the user to enter again.

- (1) Set the complex number in regular format
- (2) Set the complex number in polar format
- (3) Compute the absolute value of the number
- (4) Compute the reciprocal of the number
- (5) Add another complex number
- (6) Subtract another complex number
- (7) Multiply by another complex number
- (8) Divide by another complex number
- (9) Exit

Enter your choice: 10

Invalid Input. Try Again!
Current number is 2.000 + 7.000i
Polar Notation: 7.280 @ 74.055 degrees

- (1) Set the complex number in regular format
- (2) Set the complex number in polar format
- (3) Compute the absolute value of the number
- (4) Compute the reciprocal of the number
- (5) Add another complex number
- (6) Subtract another complex number
- (7) Multiply by another complex number
- (8) Divide by another complex number
- (9) Exit

Enter your choice:

1.3 Implementation

1.3.1 Functions Definitions

To implement the program, you need to define some functions; each function does a specific job, such as addition, subtraction, and division etc. Here are the functions you need to define:

```
/*Set a complex number in regular form*/  
void cSetInRegular(double rPart, double iPart);  
  
/*Set a complex number in polar coordinates*/  
void cSetInPolar(double mag, double phase);  
  
/*Calculate the absolute value of the current complex number and  
return it*/  
double cAbsolute();  
  
/*Calculate the reciprocal of the current complex number*/  
/* If the operation cannot be performed, i.e., denominator in a  
division operation is zero, this function should return -1 as the  
error code*/  
int cReciprocal ();  
  
/*do complex number addition*/  
void cAdd(double rPart, double iPart);  
  
/*do complex number subtraction*/  
void cSubtract(double rPart, double iPart);  
  
/*do complex number multiplication*/
```



```

void cMultiply(double rPart, double iPart);

/*do complex number division*/
/* If the operation cannot be performed, i.e., denominator in a
division operation is zero, this function should return -2 as the
error code*/
int cDivide(double rPart, double iPart);

/*display the current complex number in both regular and polar
form*/
void DisplayComplex();

```

1.3.2 Use Global Variables

To make the current complex number accessible by any of those above functions, you should define the real and imaginary parts of the current complex number as global variables. The definition should be put before the main function like this:

```

/* currentR and current I are the real part and imaginary part of
the current complex number */
double currentR, currentI;
int main (void)
{
    ...
}

```

And here is the example showing how to assign values to those global variables in cSetInRegular() function:

```

void cSetInRegular(double rPart, double iPart)
{
    ...
    currentR = rPart;
    currentI = iPart;
    ...
}

```

Note:

As already explained, there is some error handling required in the 'cReciprocal' and 'cDivide' functions. If an error occurs, these functions return a non-zero error code to the main function. The main function on receiving a non-zero error code prints the error message to the screen. (Remember a return value of 0 indicates success)

Always use the -Wall and -ansi parameters while compiling your code.

1.4 What to Submit

The files that you should submit for this assignment are: **complex.c**, **complex.txt**, and **complex.script**.

Your **complex.script** should show the output of your program after performing the following steps:

1. Set the current complex number to $3.0 + 5.0i$
2. Add the following complex number to the current value: $4.3 + 1.44i$
3. Subtract the following complex number from the current value: $6.0 + 6.66i$
4. Reset the current value with the following number in polar notation: 7.0 at angle of 45 degrees
5. Multiply by $5.5 + 5.3i$
6. Divide by $8 + 17i$
7. Compute the absolute value of the current complex number
8. Compute the reciprocal of the current complex number
9. Divide by $0 + 0i$
10. Exit

Your **complex.txt** should contain a brief description about the design of your program.

1.5 Test Example

This example helps you to verify your program. **Please note that this is just a sample example and your script should not include this example.**

First column shows the operation to be performed; center and right column shows the result in general and polar format respectively.

Operation	Input	General Form Result	Polar Form result
1. Set Complex No.	$0.0 + 4.0i$	$0.000 + 4.000i$	4.000 @ 90.000 degrees
2. Set Complex No.	$3.0 + 4.0i$	$3.000 + 4.000i$	5.000 @ 53.130 degrees
3. Add	$2.5 + 2.5i$	$5.500 + 6.500i$	8.515 @ 49.764 degrees
4. Absolute value:	8.515(absolute value)		
5. Reciprocal		$0.076 - 0.090i$	0.117 @ -49.764 degrees
6. Set Complex No.	$-1.0 + 0.0i$	$-1.000 + 0.000i$	1.000 @ 180.000 degrees
7. Set No. in Polar	5.0 @ 45 degrees	$3.536 + 3.536i$	5.000 @ 45.000 degrees
8. Divide	$1.0 + 2.0i$	$2.121 - 0.707i$	2.236 @ -18.435 degrees
9. Multiply	$6.0 + 10.0i$	$19.799 + 16.971i$	26.077 @ 40.601 degrees
10. Set Complex No.	$0.0 + 0.0i$	$0.000 + 0.000i$	0.000 @ 0.000 degrees

2 Extra Credit: Rotate the Complex Number [10 bonus points]

In the bonus question, you should add a new option to the menu which allows users to rotate the current complex number by N degrees counterclockwise.

The execution of the program should look as follows:

```
Current number is 2.000 + 7.000i
```

Polar Notation: 7.280 @ 74.055 degrees

- (1) Set the complex number in regular format
- (2) Set the complex number in polar format
- (3) Compute the absolute value of the number
- (4) Compute the reciprocal of the number
- (5) Add another complex number
- (6) Subtract another complex number
- (7) Multiply by another complex number
- (8) Divide by another complex number
- (9) "Rotate" the number by N degrees counterclockwise
- (10) Exit

Enter your choice: 9

Enter the degrees you want to rotate: 90

Current number is $-7.000 + 2.000i$

Polar Notation: 7.280 @ 164.055 degrees

After rotation, your program should print out the rotated complex number in both regular and polar forms.

This is the definition of the function you need to implement:

```
/* Rotate the current complex number*/  
void cRotate(double degrees);
```

The files that you should submit for this assignment are: **complex.c**, **complex.txt**, and **complex.script**.

Your **complex.script** should show the output of your program after performing the following steps:

1. Set the current complex number to $3.0 + 5.0i$
2. Add the following complex number to the current value: $4.3 + 1.44i$
3. Subtract the following complex number from the current value: $6.0 + 6.66i$
4. Reset the current value with the following number in polar notation: 7.0 at angle of 45 degrees
5. Multiply by $5.5 + 5.3i$
6. Divide by $8 + 17i$
7. Compute the absolute value of the current complex number
8. Compute the reciprocal of the current complex number
- 9. Rotate the current complex number by 45 degrees counterclockwise.**
10. Exit

3 Submission

The submission is similar to the previous homeworks. You need to create a directory called **hw6**; put all the files `complex.c`, `complex.script`, and `complex.txt` in the directory and run the command `/ecelib/bin/turnin` to submit your homework.