Classes: Dealing with Complex Numbers

For this challenge, you are given two complex numbers, and you have to print the result of their addition, subtraction, multiplication, division and modulus operations.

The real and imaginary precision part should be correct up to two decimal places.

Input Format

One line of input: The real and imaginary part of a number separated by a space.

Output Format

For two complex numbers C and D, the output should be in the following sequence on separate lines:

- $C+D$
- $C-D$
- $C*D$
- C/D
- $mod(C)$
- $mod(D)$

For complex numbers with non-zero real(A) and complex part(B), the output should be in the following format:

$A+Bi$

Replace the plus symbol $(+)$ with a minus symbol $(-)$ when $B<0$.

For complex numbers with a zero complex part i.e. real numbers, the output should be:

$A+0.00i$

For complex numbers where the real part is zero and the complex part(B) is non-zero, the output should be:

$0.00+Bi$

Sample Input

```
2 1
5 6
```

Sample Output

```
7.00+7.00i
-3.00-5.00i
4.00+17.00i
0.26-0.11i
2.24+0.00i
7.81+0.00i
```

Concept

Python is a fully object-oriented language like C++, Java, etc. For reading about classes, refer here.

Methods with a double underscore before and after their name are considered as built-in methods. They are used by interpreters and are generally used in the implementation of overloaded operators or other built-in functionality.

```
__add__-> Can be overloaded for + operation



__sub__ -> Can be overloaded for - operation



__mul__ -> Can be overloaded for * operation
```

For more information on operator overloading in Python, refer here.

Current Buffer (saved locally, editable) Python 3

```python
import math
class Complex(object):
    def __init__(self, real, imaginary):
        self.real=real
        self.imaginary=imaginary
    def __add__(self, no):
        return Complex(self.real+no.real,self.imaginary+no.imaginary)
    def __sub__(self, no):
        return Complex(self.real-no.real,self.imaginary-no.imaginary)
    def __mul__(self, no):
        return Complex((self.real*no.real-self.imaginary*no.imaginary),(self.real*no.imaginary+self.imaginary*no.real))
    def __truediv__(self, no):
        a=(self.real*no.real+self.imaginary*no.imaginary)
        b=(self.imaginary*no.real-self.real*no.imaginary)
        c=(no.real**2+no.imaginary**2)
        return Complex(a/c,b/c)
    def mod(self):
        return Complex(math.sqrt(self.real**2+self.imaginary**2),0)
    def __str__(self):
        if self.imaginary == 0:
            result = "%.2f+0.00i" % (self.real)
        elif self.real == 0:
            if self.imaginary >= 0:
```

```
                result = "0.00+%.2fi" % (self.imaginary)
            else:
                result = "0.00-%.2fi" % (abs(self.imaginary))
        elif self.imaginary > 0:
            result = "%.2f+%.2fi" % (self.real, self.imaginary)
        else:
            result = "%.2f-%.2fi" % (self.real, abs(self.imaginary))
        return result
if __name__ == '__main__':
```

Line: 14 Col: 58

Upload Code as File | Test against custom input | Run Code | Submit Code

You have earned 20.00 points!
114/115 challenges solved. 99%

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Testcase 0 | Testcase 1 | Testcase 2 | Testcase 3

10 Testcases

Input (stdin)	Download	Expected Output	Download
2 1 5 6		7.00+7.00i -3.00-5.00i 4.00+17.00i 0.26-0.11i	

Compiler Message

Success