



X

Classes: Dealing with Complex Numbers *

Your Classes: Dealing with Complex Numbers submission got 20.00 points.

are P

Post

.

Try the next challenge | Try a Random Challenge

Problem

Submissions

Leaderboard

Editorial 🖰

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

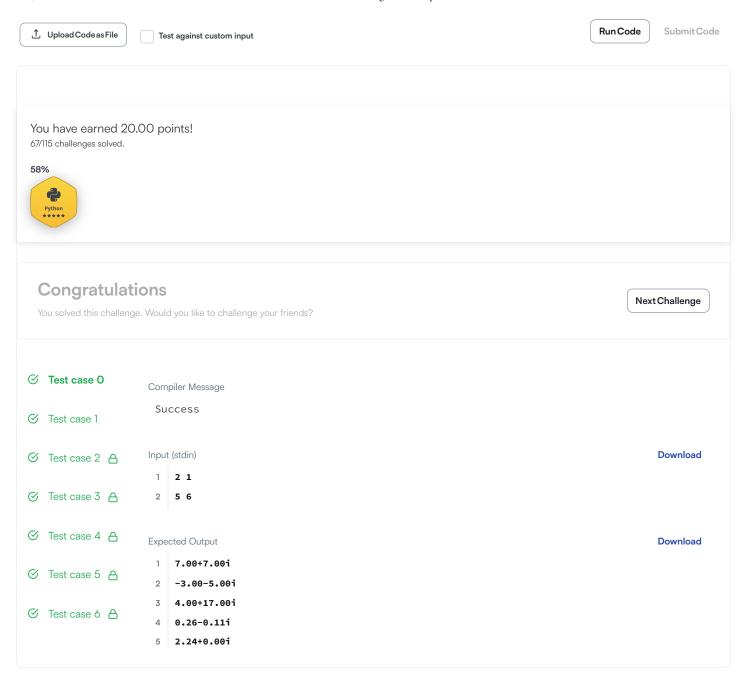
Python is a fully object-oriented language like C++, Java, etc. For reading about classes, refer here.

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.

```
Change Theme Language Python 3
                                                                                                          (O)
                                                                                                              53
          import math
     1
     2
     3
          class Complex(object):
             def __init__(self, real, imaginary):
     4
     5
                  self.real = real
     6
                  self.imaginary = imaginary
     7
     8
             def __add__(self, no):
     9
                  return Complex(self.real + no.real, self.imaginary + no.imaginary)
     10
              def __sub__(self, no):
     11
     12
                  return Complex(self.real - no.real, self.imaginary - no.imaginary)
     13
     14
              def __mul__(self, no):
    15
                  new_real = self.real * no.real - self.imaginary * no.imaginary
                  new_imaginary = self.real * no.imaginary + self.imaginary * no.real
     16
     17
                  return Complex(new_real, new_imaginary)
     18
     19
              def __truediv__(self, no):
     20
                  new_real = (self.real * no.real + self.imaginary * no.imaginary) / (no.real **2 + no.imaginary**2)
                  new_imaginary = (self.imaginary * no.real - self.real * no.imaginary) / (no.real **2 + no.imaginary*
     21
                  return Complex(new_real, new_imaginary)
     22
     23
             def mod(self):
     24
     25
                  return Complex(math.sqrt(self.real**2 + self.imaginary**2),0)
     26
              def __str__(self):
     27
     28
                  if self.imaginary == 0:
                      result = "%.2f+0.00i" % (self.real)
     29
                  elif self.real == 0:
     30
                      if self.imaginary >= 0:
     31
                          result = "0.00+%.2fi" % (self.imaginary)
     32
     33
                      else:
                          result = "A AA-% Ofi" % (abs(self imaginary))
     34
                                                                                                       Line: 26 Col: 1
EMACS
```



Blog | Scoring | Environment | FAQ | About Us | Helpdesk | Careers | Terms Of Service | Privacy Policy