In [139…

```python
# Defining the complex number system
import math
from fractions import Fraction as frac

class myComplex:

    def __init__(self,real,imaginary):
        self.x = real
        self.y = imaginary

    def disp(self):
        print(str(self.x)+" + "+"("+str(self.y)+")" + "i")

    def sum(self,a):
        r = self.x + a.x
        i = self.y + a.y
        return myComplex(r,i)

    def prod(self,a):
        r = self.x*a.x - self.y*a.y
        i = self.x*a.y + self.y*a.x
        return myComplex(r,i)

    def conjugate(self):
        self.y = -1*self.y

    def find_conjugate(self):
        r = self.x
        i = -self.y
        return myComplex(r,i)

    def modulus(self):
        mod = ((self.x)*(self.x) + (self.y)*(self.y))**(0.5)
        return mod

    def reciprocal(self):
        self.conjugate()
        p = self.modulus()
        self.x = self.x/(p**2)
        self.y = self.y/(p**2)
        return self

    def find_reciprocal(self):
        u = self.find_conjugate()
        p = self.modulus()
        i = u.x/p**2
        r = u.y/p**2
        return myComplex(i,r)

    def div(self,b):
        u = self.find_reciprocal()
        p = u.prod(b)
        p.reciprocal()
        return p

    def phase(self):
        print(math.atan(self.y/self.x),end=" ")
        print("i.e. tan inverse of "+ str(frac(self.x,self.y)))
```

```
r = myComplex(5,3)
s = myComplex(1,2)
t = myComplex(1,0)
u = myComplex(0,4)

s.disp()
t.sum(u).disp()
u.prod(t).disp()
r.find_conjugate().disp()
print(s.modulus())
u.find_reciprocal().disp()
s.find_reciprocal().disp()
t.div(u).disp()
s.div(t).disp()
r.div(s).disp()
```

```
1 + (2)i
1 + (4)i
0 + (4)i
5 + (-3)i
2.23606797749979
0.0 + (-0.25)i
0.19999999999999996 + (-0.3999999999999999)i
0.0 + (-0.25)i
1.0000000000000002 + (2.0000000000000004)i
2.2000000000000006 + (-1.4000000000000001)i
```