Tutorial problems for Lectures 6/7. Due Monday May 7, 2018. Please put in a new repository named Tutorial_3

Problem 1: Complete the complex class definition (available in overload.py from the lecture 7 repository) to support -,*, and / (__sub__, __mul__, and __div__). Recall that a/b = a*conj(b)/(b*conj(b)). Show from a few sample cases that your functions work. (10)

Problem 2: Let's write a direct-solving N-body solver. First, write a class that contains masses and x and y positions for a collection of particles. The class should also contain a dictionary that can contain options. Two entries in the dictionary should be the number of particles and G (gravitational constant). The class should also contain a method that calculates the potential energy of every particle, $\sum(Gm_1m_2/r_{12})$. (10)

Problem 3: Now lets add some methods to the class so we can actually use it in an n-body simulation.
(A): First, add a method to initialize the number of particles with random positions in 2-D (numpy.random.randn() will get gaussian random numbers). (5)
(B): Next, write a method that calculates the forces on the particles using a softened potential (10)
(C): Write a method that will update the particle positions and velocities using a timestep. (5)
(D): Finally, plot the total kinetic and potential energies as a function of time, and show that the total energy is approximately conserved (5)

Bonus: Extend the complex class to also support arbitrary (i.e. non-integer) powers (keyword is __pow__). +3 if the routine works if ab works for complex a and real b, +5 if it works for complex a and complex b. (you may ignore branch cuts).