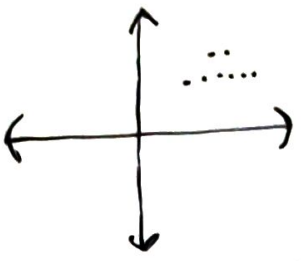
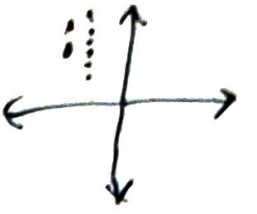
chapter 2 - Rotation

- TO rotate by 90 degrees counter clockwise multiply coordinates by ij.



After 90° counter clockwise rotation:



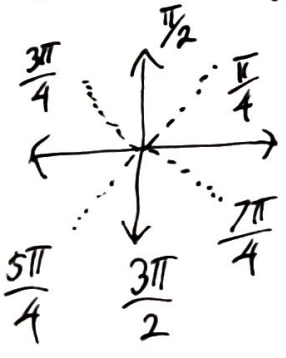
- TO rotate by θ radians, multiply the coordinates by:

$$e^{i\theta}$$

e is a predefined number (similar to how Pi is a predefined number)

i is imaginary $\gamma(\sqrt{-1})$

θ is the amount to rotate counter clockwise in radians



$$e^{i\left(\frac{\pi}{2}\right)} = i$$

```
from plotting import plot
plot({1j/2 * z for z in S}, 4)
plot({0.5j*z for z in S}, 4)
plot({0.5 * (e**((pi/2) * 1j)) * z for z in S}, 4)

from math import pi, e
```

TASK 1.4.8: Create a new plot in which the points of S are rotated by 90 degrees counter clockwise and scaled by 1/2. Use a comprehension in which the points of S are multiplied by a single complex number.

TASK 1.4.18: Write a comprehension whose value is the set consisting of rotations by pi/4 (counter clockwise) of the elements of S. Plot the value of the comprehension.

Degree to radian conversion:

If x is degree value:

(x/360) * 2pi = radian value

(x/180) * pi = radian value

```
plot({2 * (e** ((pi/4) * 1j)) for z in S}, 4)
```

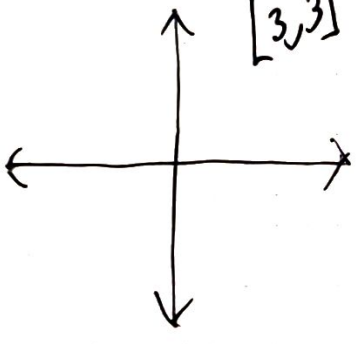
Another example: using the previous answer, scale by 1/4 and shift to the right by 2 and down by 3.

Plot([(2-3j)+(1/4)*z*(e**((Pi/4)*1j)) for z in S],4)

Your plot function's second parameter should be 4.

If the problem says rotate 45 degrees clockwise, make it negative and convert to radians. Then that will be the amount to rotate.

$(-45/180) * Pi = -Pi/4$



Need to connect two coordinates with a series of points.

$3 \times [1,2] = [3,6]$

(python)

[1,2,1,2,1,2]

scalar value

```
def scalar_vector_mult(alpha, V):        # V: the vector (list data type)
    return [alpha * V[i] for i in range(len(V))]
```

ex: $3 \times [1,2] = [3,6]$

```
scalar_vector_mult(3,[1,2])
```

output: [3,6]