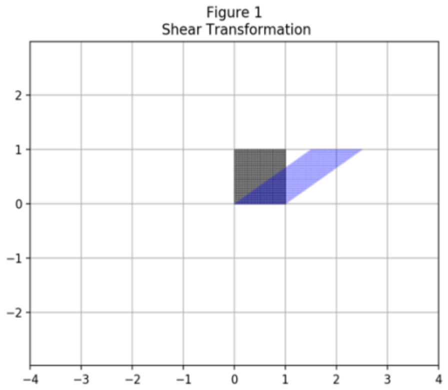
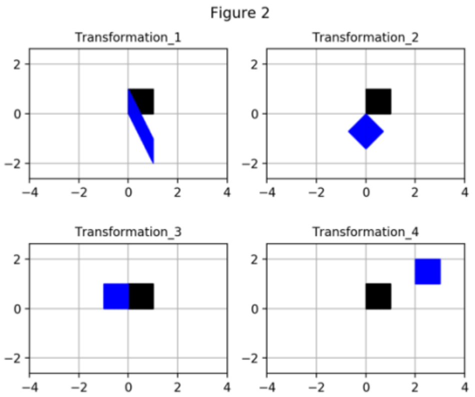
Math 266 project #1

The python program, transform.py when run generates figure 1 shown below. The black box is the input and the transformed box is in blue. The program file is in the file section of canvas. For project #1 you are to modify this program so it outputs what you see in figure 2 below. For the project you are combine your code and output into a single pdf file and submit that. Include a paragraph describing your impressions and experience with this assignment. For the assignment you should:

- First run tranform.py and make sure it runs on your system and produces figure 1.
- Determine the matrix for the various transformations shown. This will involve a combination of theory and trial and error.
- Note that transformation #4 is not linear so a matrix will not work. You will need write a python function to do the transformation.
- To get the output of multiple plots, use plt.subplot(2,2,1) etc. Note that plt.subplot has different functionality than plt.subplots.





```python
'''
transform.py
Math 266
Summer 2020
Project #1


    This program will create a square 1 x 1 grid of points.
    Then a tranformation will be applied to the
    "box". Both the box and its image will then be plotted.
'''

# import standard libraries

import numpy as np
import matplotlib.pyplot as plt
plt.ion()

# generate the data points for the box using a list comprehension
# See https://docs.python.org/3/tutorial/datastructures.html

data = np.array([[x, y] for x in np.arange(0, 1.01, .01) for y in
np.arange(0, 1.01, .01)])

# define transformation matrix A for a shear

A = np.array([[1, 1.5], [0, 1]])

#  apply the transformation to the data points (vectors).
#  recall "@" represents matrix multiplication
#  and v.T  represents the transpose of v.

out_data = A @ data.T
out_data = out_data.T


plt.plot(*zip(*data), '.', markersize=0.1, c='k')
plt.plot(*zip(*out_data), '.', markersize=0.1, c='b', alpha=.5)
plt.axis("equal")
plt.axis((-4, 4, -4, 4))
plt.grid(True)
plt.title('Shear Transformation')
plt.show()

# save the plot
plt.savefig('shear.png')
```