Last name _____

First name _____

LARSON—MATH 310–CLASSROOM WORKSHEET 08
Convex Combinations of Images.

1. Set up your CoCalc JUPYTER notebook for today's work.

   (a) Start the Chrome browser.

   (b) Go to `https://cocalc.com`

   (c) Log in.

   (d) You should see an existing Project for our class. Click on that.

   (e) Make sure you are in your Home directory (if you work in your Handouts directory, your work could get overwritten).

   (f) Click "New", then "Jupyter Notebook", then call it **310-c08**.

   (g) Make sure you have PYTHON as the *kernel*.

   Review

   (a) What is a *convex combination*?

2. (**Task 2.6.9**) Write a python procedure `segment(pt1, pt2)` that, given points represented as 2-element lists, returns a list of a hundred points spaced evenly along the line segment whose endpoints are the two points.

3. Find the hundred points resulting when pt1 = [3.5, 3] and pt2 = [0.5, 1]. Call your list `pts`.

4. We'll use the built-in `matplotlib` library for plotting functions. These functions do not take lists of pairs but rather a list of x-coordinates, together with a list of y-coordinates.

```
import matplotlib.pyplot as plt
x=[p[0]for p in pts]
y=[p[1] for p in pts]
plt.scatter(x,y)
```

   Image Lab

5. Copy the libraries `png.py`, `img.py`, and the folder `faces` from your CoCalc project `310-handouts/CodingTheMatrix-originals` folder to your `Home` directory.

6. Import the required packages:

```
import image as img
import png
```

7. Code the following procedure and test it on any of the images in your faces folder (for instance, let: `image0 = display_png("faces/img00.png", True))`.

```
def display_png(png, hide=None):
    '''
    Returns a list of lists of pixel values (tuples with three values)
    read from a png and displays it unless hide is true.
    '''
    image = img.file2image(png)
    if not hide:
        plt.imshow(image)
        plt.show()
    return image
```

8. Code the following function which defines a convex combination of img1 and img2.

```
def avg_img(img1, img2, pctg):
    '''Creates a composite image from two image types and displays the
   result'''
    new_img = []
    for i in range(len(img1)):
        row1 = img1[i]
        row2 = img2[i]
        new_row = []
        for j in range(len(row1)):
            pixel1 = row1[j]
            pixel2 = row2[j]
            new_pixel = tuple(int(pixel1[k] * pctg + pixel2[k] * (1 -
   pctg)) for k in range(3))
            new_row.append(new_pixel)
        new_img.append(new_row)
    plt.imshow(new_img)
    plt.show()
    return new_img
```

9. Define `image2` to be another faces from your `faces/` folder. Then find various convex combinations of them (by evaluating `avg_img` and setting the `pctg` parameter to various values between 0 and 1).

10. How can you get a sequence of convex combinations of these images for `pctg` parameters from 0 to 1?

Klein's Vec() class

```
class Vec:
    def __init__(self, labels, function):
        self.D = labels
        self.f = function

```

11. Once Python has processed this definition, you can create an instance of Vec as follows. Code and run.

```
Vec({'A','B','C'}, {'A':1})
```

12. (**Quiz 2.7.1**) Write a procedure `zero_vec(D)` with the following spec:
    • input: a set D
    • output: an instance of Vec representing a D-vector all of whose entries have value zero.

13. Look in the Handouts folder for the *vec.py* file and the *class* definition of the vector class `Vec`.

Getting your classwork recorded

When you are done, before you leave class...

(a) Click the "Print" menu choice (under "File") and make a pdf of this worksheet (html is OK too).

(b) Send me an email (`clarson@vcu.edu`) with an informative header like "Math 310 - c08 worksheet attached" (so that it will be properly recorded).

(c) Remember to attach today's classroom worksheet!