(Spring 2024) Due: **Sunday, April 8**

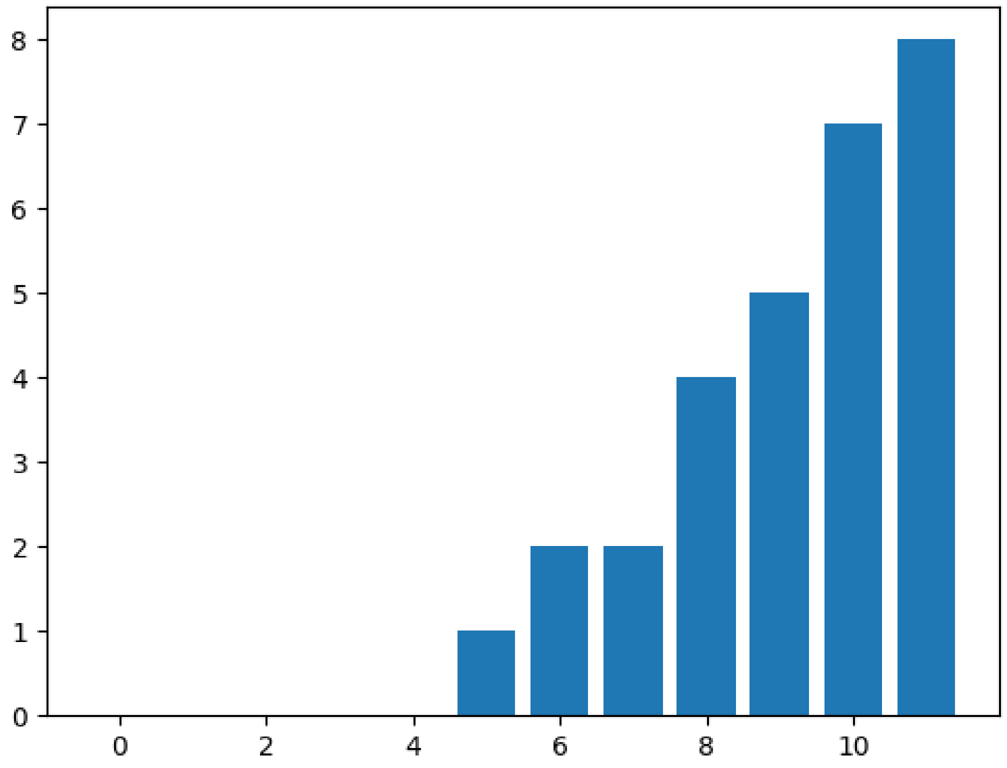
On to the problems!

1 Basic Matrix/Vector Manipulation (20 points)

(e) Without using a loop, multiply each row of M element-wise by a. **Briefly explain the logic of your code in your written report.**

We can use NumPy broadcasting for this purpose: M has shape $(4, 3)$ and a has shape $(3,)$. Therefore, `M*a` should do the required job.

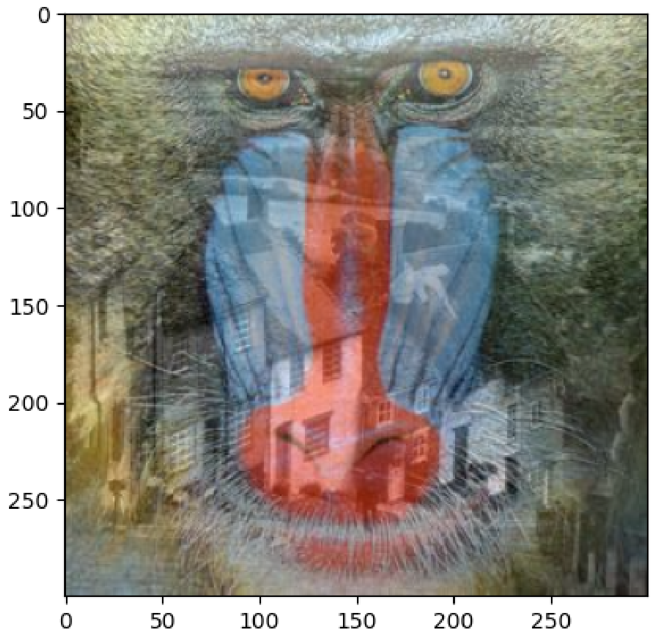
(f) Without using a loop, sort all of the values of the new M from (e) in increasing order and plot them in your report. **Briefly explain the logic of your code in your written report.** We can use `np.sort(newM, axis=None)`. `axis=None` will flatten the array before sorting and thereby give the desired result.



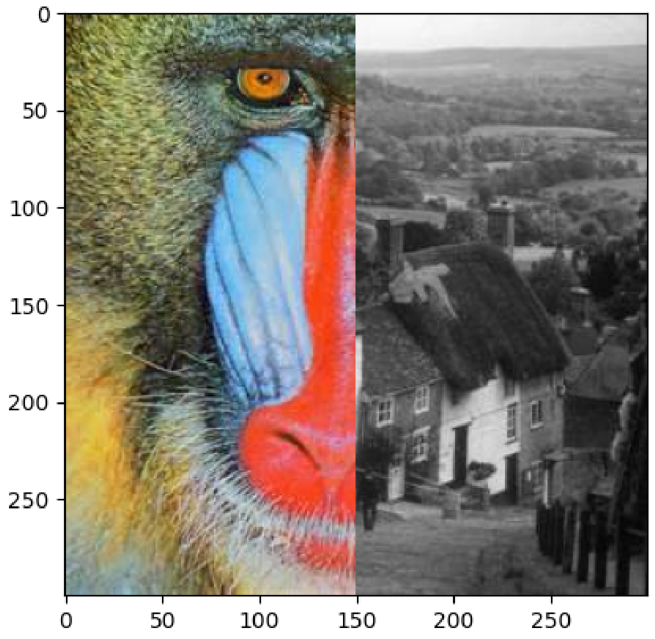
2 Basic Image Manipulations (40 points)

Do the following by filling out p2.py:

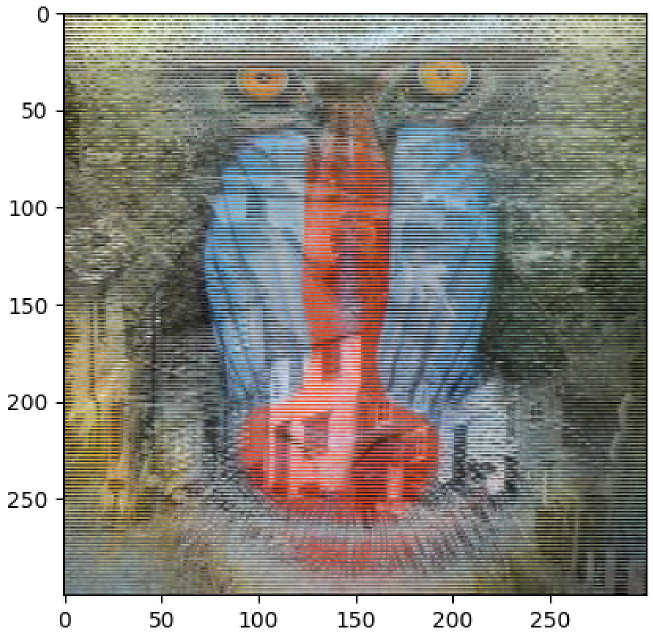
(c) Add the images together and re-normalize them to have minimum value 0 and maximum value 1. **Save and include this image in your report.**



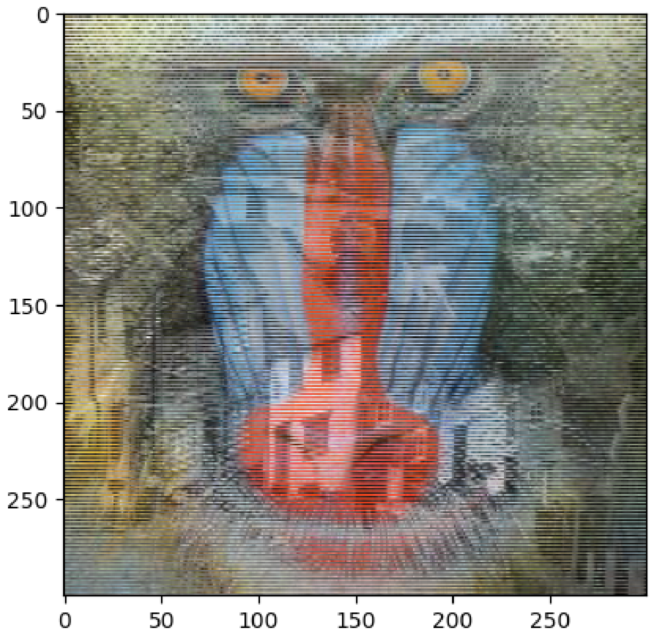
(d) Create a new image such that the left half of the image is the left half of image1 and the right half of the image is the right half of image2. **Save and include this image in your report.**



(e) Using a for loop, create a new image such that every odd numbered row is the corresponding row from image1 and the every even row is the corresponding row from image2 (Hint: Remember that indices start at 0 and not 1 in Python). **Save and include this image in your report.**

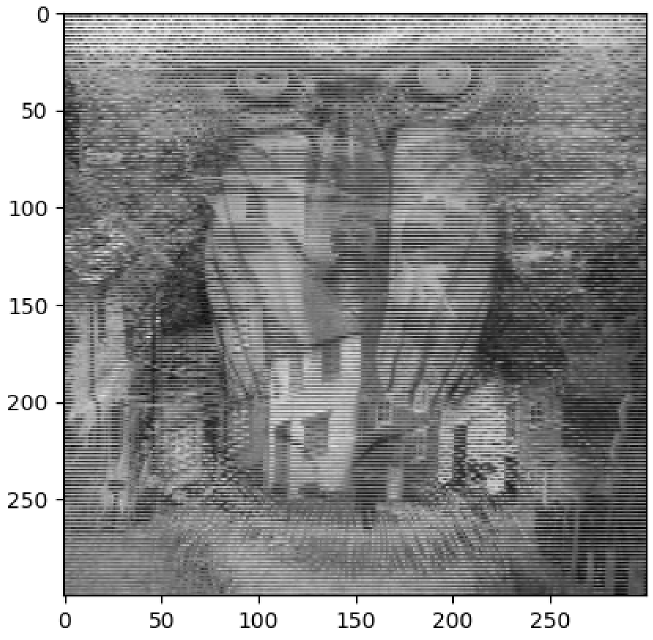


(f) Accomplish the same task as part e without using a for-loop (the functions reshape and tile may be helpful here). **Briefly explain the logic of your code in your written report.**



We can create a mask for each of the image with 0 at the location where the image doesn't contribute to the final image and 1 at the location where the image does contribute to the final image. Since, this mask is a repetition of 0 and 1, we can use `np.tile` function. The final image can then be obtained using `newImage2 = img1*img1_mask + img2*img2_mask`.

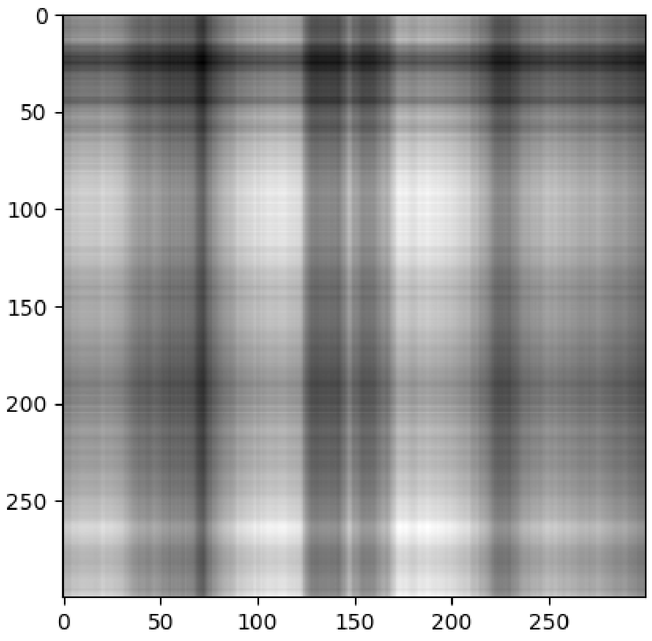
(g) Convert the result from part f to a grayscale image. **Save and include the grayscale image in your report.**



3 Singular Value Decomposition (40 points)

Do the following by filling out p3.py:

(b) **Save and Include the best rank 1 approximation of the (grayscale) image1 in your report.**



(c) **Save and Include the best rank 20 approximation of the (grayscale) image1 in your report.**