Image Convolution Workshop **CS 480**

When you are finished share your Python code with me. If you are using Google Colab, share your code with: `lins.brian@gmail.com`

In this workshop we are going to apply different convolution kernels to an image to see what happens. First you need to download an image to work with and convert it to a numpy array using the cv2 library. Pick something that is safe for work and save it onto your computer. Then upload it into your Google colab by clicking on the files tab on the left side. We'll convert the image to grayscale to make things a little easier.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

image = cv2.imread("/content/your_image.png")
image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
image = cv2.resize(image, (128,128)) # it's optional, but recommended, to resize your image.
plt.imshow(image)
plt.show(image)
```

Once your image is uploaded, write a Python function to apply a convolution with kernel K to any grayscale image. Then apply the following convolutions to your image. Send me your code with an image showing how each convolution looks.

1. **(Box blur)** $K = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

2. **(Ridge detection)** $K = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$

3. **(Sharpen)** $K = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$

4. **(3-by-3 Gaussian blur)** $K = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$

5. **(Vertical edge detection)** $K = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$