In this lab, you will be implementing some simple image processing commands in CUDA. Begin by opening the Colab notebook:

`https://github.com/TeachingUndergradsCHC/modules/blob/master/Programming/cuda/cudaBlur.ipynb`

Follow the directions in that notebook. The files to upload are available from the module repo:

`https://github.com/TeachingUndergradsCHC/modules/blob/master/Programming/cuda/`

The last two code cells of the notebook run the "Hello World" program (recommended to check that everything is working) and then a first version of our image processing program. This version removes all the red from our sample image (640x426.ppm) and creates a new file `out.ppm`. (You may have to refresh the file panel to the left by hitting the button with circular arrow to see it.) Download this file and observe that it looks basically the same as our original image, but the colors are slightly shifted. The image is stored as an array of RGB values, each color in a separate *channel* of the pixel. Look through the kernel to see how it works.

As a first task, let's change this kernel to convert an image into grayscale (black and white). To do this, each pixel should be set to the average of its input red, green, and blue values. In the kernel, this corresponds to adding up the values for each channel, dividing by three, and then setting each channel of the output image to the result. Do this and make sure it's working before proceeding.

Creating the blur effect is the final challenge. This is also an average, but a different kind. Each channel is blurred separately— each gets the average of that channel's value for nearby pixels. "Nearby" in this case means all pixels for which each coordinate is within a specified radius of the pixel being set (i.e. its the L_∞ distance $\leq$ the radius).