Graeme Knowles
CSC 196U
Assignment 2

1. Name 3 applications of convolution
    a. Edge detection in images
    b. Blurring/Sharpening
    c. Artificial Intelligence - Convolutional Neural Networks

2. How many floating operations are being performed in your convolution kernel in terms of variables that indicate mask width, input image size, and number of channels (utilize the same variable names used in the template.cu e.g., utilize MASK_WIDTH to indicate mask width)? Explain.
   Per Thread: 2 * MASK_WIDTH * MASK_WIDTH
    1. val = Input Element * Mask Element
    2. Accumlated value + val (from 1)

   Num Threads Participating = Number of pixels * Num Channels
   Total FLOPs = imageWidth * imageHeight * imageChannels * 2 * MASK_WIDTH * MASK_WIDTH

   The convolution works on each pixel of the image and each channel of each pixel independently. The convolution itself multiplies each element within the mask by a single input value.

3. How many global memory reads are being performed by your kernel in terms of variables that indicate output tile width, mask width, input image size, and number of channels (utilize the same variable names used in the template.cu e.g., utilize O_TILE_WIDTH to indicate output tile width)? explain. Assume that ghost elements also require global reads.

   Number of Blocks: x = (imageWidth - 1) / O_TILE_WIDTH + 1
   y = (imageHeight - 1) / O_TILE_WIDTH + 1
   z = imageChannels
   Threads per Block = O_TILE_WIDTH + (MASK_WIDTH-1)
   Reads per thread: 1

   Total Reads = ((imageWidth - 1) / O_TILE_WIDTH + 1) * ((imageHeight - 1) / O_TILE_WIDTH + 1) * imageChannels * (O_TILE_WIDTH + (MASK_WIDTH-1))

   All threads in the input tile (which equals the block size) will load an element (including ghost elements, which are defined by the inputTile - outputTile). The difference between flops and this is that not all threads that load an element will participate in the convolution.

4. How many global memory writes are being performed by your kernel in terms of variables that indicate input image size and number of channels (utilize the same variable names used in the template.cu e.g., utilize imageWidth to indicate image width)? explain.

   Each channel of each pixel is written exactly once. So total number of writes = imageWidth * imageHeight * imageChannels

5. What do you think happens as you increase the mask width (e.g., you increase it from 5 to 1024) in your kernel while you keep your output tile width at 16? What do you end up spending most of your time doing? Do you think using a tiled convolution approach would still be a good solution in this case? Hint: think of the shared memory size.

   In this case, 1040x1040 elements (mask width + output tile width) would have to be loaded. Because the warp size is 32, 1040x1040 / 32 = 33,800 separate loads would have to be made before any calculations could start. With a 16x16 output tile, 2 FLOPS * 256 / 32 = 16 separate calculation sections would have to be made. This means that in terms of number of instructions, 99.9% of the time would be spent loading data.

   A tiled approach would still be the best way to reuse data, however it would be much simpler to use exclusively global memory. Because the constant memory size is 64KB, the mask would have to be loaded into global memory: 1024*1024 * 4bytes = ~4000KB. The same goes for the input tile, with the shared memory at 48KB. Each input tile would itself have to be tiled. As much of the the mask data and input data as possible could be loaded, then the elements in the output tile that use those elements can compute the partial convolution. After the computation is done, another set of mask data and input data could be loaded and the computation repeated until the entire tile has been computed. This extra level of tiling would add more overhead to the entire process however.

6. Do you have to have a separate output memory buffer, e.g., called P, in your kernel? Put it in another way, why can't you perform the convolution in place (by just using the input memory buffer, e.g. called N)?

   In the kernel's current form, yes, a separate memory buffer is required. This is because each block not only uses the input elements in the output grid, the values the output would overwrite, it uses the values of the halo elements whose corresponding output indices are being calculated by another block. With no way to perform inter-block synchronization, the values of the halo elements could not be guaranteed.

7. What is the identity mask? What would be the values in it for a mask width of 5 for 2D convolution?

   The identity mask is the mask who, when a convolution is performed with it, produces an output identical to the input.

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0