Assignment 3: Parallel Convolution

Let us consider the 2D convolution operation for a 2D Matrix of floating-point numbers. It is a neighborhood operation where each output element in the matrix is the weighted sum of a collection of neighboring elements of the input matrix. The weights used in the weighted sum are typically stored in an array called the convolution mask. This mask is applied across each element of the input to yield a convolved output. Let us focus on specific convolution operations, namely average convolution. This operation considers a 3x3 mask and for every element, **A[i][j],** considers the nearest 8 neighbors contained in the **3x3 mask** centered at the element **A[i][j],** takes the average and stores in the output matrix **B[i][j].**

Implement a CUDA program which takes as input:
i) the number of test cases and for each test case
ii) the integer n,
iii) the values of each element of an n x n matrix;
apply the average convolution operation and produce an **n x n** output matrix.
Refer to the following input-output example.
Input
1
3
3.0 3.0 3.0
3.0 3.0 3.0
3.0 3.0 3.0
Output
1.3 2.0 1.3
2.0 3.0 2.0
1.3 2.0 1.3

Note for boundary elements, assume a padding of 0.0. In the above example, for element input[0][0], placing the 3x3 mask with its center at input[0][0], produces output[0][0] = 0.0/9 +0.0/9 +0.0/9 + 0.0/9 + input[0][0]/9 + input[0][1]/9 + 0.0/9 + input[1][0]/9 + input[1][1]/9 =0.0/9 +0.0/9 +0.0/9 + 0.0/9 + 3.0/9 + 3.0/9 + 0.0/9 + 3.0/9 + 3.0/9 = 1.3.
Further note that the above input-output was a representative example depicting what the convolution operation is. Expect sample input matrices to be of the order NxN where N =512,1024,2048 etc. Your task would be to generate random floating-point matrices of this order and test your implementation while adhering to the input format discussed above. The code should be general enough to handle large matrices. Accordingly, your implementation should be able to automatically select a suitable grid and block dimensions for processing the matrix.
Ensure that you print your output matrices using %.2f format.