1. top-left 5 x 5 matrix of input
    a. (h_a,h_b) -> (0.000000, 1.000000) (h_a,h_b) -> (0.000000, 0.540302)

    (h_a,h_b) -> (0.000000, -0.416147) (h_a,h_b) -> (0.000000, -0.989992)

    (h_a,h_b) -> (0.000000, -0.653644) (h_a,h_b) -> (0.000000, 0.283662)

    (h_a,h_b) -> (0.000000, 0.960170) (h_a,h_b) -> (0.000000, 0.753902)

    (h_a,h_b) -> (0.000000, -0.145500) (h_a,h_b) -> (0.000000, -0.911130)

    (h_a,h_b) -> (0.000000, -0.839072) (h_a,h_b) -> (0.000000, 0.004426)

    (h_a,h_b) -> (0.000000, 0.843854) (h_a,h_b) -> (0.000000, 0.907447)

    (h_a,h_b) -> (0.000000, 0.136737) (h_a,h_b) -> (0.000000, -0.759688)

    (h_a,h_b) -> (0.000000, -0.957659) (h_a,h_b) -> (0.000000, -0.275163)

    (h_a,h_b) -> (0.000000, 0.660317) (h_a,h_b) -> (0.000000, 0.988705)

    (h_a,h_b) -> (0.000000, 0.408082) (h_a,h_b) -> (0.000000, -0.547729)

    (h_a,h_b) -> (0.000000, -0.999961) (h_a,h_b) -> (0.000000, -0.532833)

    (h_a,h_b) -> (0.000000, 0.424179)

    b. top-left 5 x 5 matrix of output

(0,0) -> 536346624.000000 (0,1) -> 536346624.000000 (0,2) -> 536346624.000000

(0,3) -> 536346624.000000 (0,4) -> 536346624.000000 (1,0) -> 536346624.000000

(1,1) -> 536346624.000000 (1,2) -> 536346624.000000 (1,3) -> 536346624.000000

(1,4) -> 536346624.000000 (2,0) -> 536346624.000000 (2,1) -> 536346624.000000

(2,2) -> 536346624.000000 (2,3) -> 536346624.000000 (2,4) -> 536346624.000000

(3,0) -> 536346624.000000 (3,1) -> 536346624.000000 (3,2) -> 536346624.000000

(3,3) -> 536346624.000000 (3,4) -> 536346624.000000 (4,0) -> 536346624.000000

(4,1) -> 536346624.000000 (4,2) -> 536346624.000000 (4,3) -> 536346624.000000

(4,4) -> 536346624.000000

2. **Problem 4**
    a. (ceil) 2000 = 512 = 4 blocks, 4 * 512 = 2048 threads,
    2048 / 32 = 64 warps
    There are 2048 threads in the grid and 64 warps

3. **Problem 5**
    a. Depending on the different situations, if it is a big equation, such as 2+3-5+8, one kernel is better choice because it has less times to read and write back to global memory which improve performance. But it is a simple equation, such as 3+2, 9-5, two kernels, I think, is better choice. Because if we only use one kernel, then control flow divergence in one kernel is more complex.

4. **Problem 6**
    a. BLOCK_SIZE <= 5.
    b. Insert '__syncthreads()' between the code that read and write the shared memory.