Problem 1:

1. We have one block that synchronizes each thread twice in the kernel function.
2. The minimum would be 0 if nothing is passed, the maximum would be 8 if count the number of iterations. The average case would be min+max/2 = 4

Problem 2:

In this problem we get the elements from the user and divide it by 1024, not sure if this is the right solution, since it works till 1024 i.e one block but it is not able to expand it to 2 blocks.

Problem 3:

1. The number of threads will be 1024\*256= 262,144 in total.
2. 32
3. 256
4. 3 (on lines 17, 18 and 30)
5. 5 \*256= 1280. We access the shared memory 5 times 27, 30, 21.s
6. Line 27 will cause memory bank conflicts.
7. The number of iterations will be 8 :
   1. 256/2 = 128
   2. 128 /2 = 64
   3. 32/2 = 16
   4. 16/2 = 8
   5. 8/2 = 4
   6. 4/2 =2
   7. 2/2 =1
8. We just need to add an if loop over the lines 30, if we don’t this will keep writing the same thing to the global memory location, we just need it to be written once.