Ashkan Vedadi Gargary | aveda002 | 862352576

1. **How many total thread blocks do we use?**

   Number of threads per blocks = BLOCKSIZE * BLOCKSIZE

   For our problem 16 * 16

   GridSize (number of blocks) = (dim-1)/BLOCK_SIZE + 1 * (dim-1)/BLOCK_SIZE + 1

   For 1000*1000 Matrix → (1000/16 + 1)*(1000/16 + 1) = 63*63

2. **Are all thread blocks full? That is, do all threads in the thread block have data to operate on?**

   No, All the thread blocks are not full. For 1000*1000 matrices we have 1,016,064 threads (63*63 * 16*16) but we only need 1,000,000 threads for our calculations. This show us that 16,064 threads are not using

3. **How can this basic Matrix Addition program be improved? (What changes do you think can be made to speed up the code?)**
   - Change in memory management that we can use all of the thread blocks and reduce the memory transfers between host and device.
   - Decreasing the number of blocks and respectively increasing the number of threads will help us in improving the performance of our codes. By decreasing the number of blocks, the number of threads per blocks will increase (block size will increase), then we would be able to improve the performance by this.
   - According to the hardware, we must care about the specs and see how many threads a block can have and use as many threads as we can per block. I have tested BLOCK SIZE of 32 and I have got a better performance by it.