Q1. Row major index for row 20 and column 10 = row \* width + column = 20 \* 400 + 10 = 8010. Column major index for row 20 & column 10 = column \* height + row = 10 \* 500 + 20 = 5020

Q2. Index = z \* (width \* height) + y \* width + x for row-major order index. Therefore, we have index = 5 \* (400 \* 500) + 20 \* 400 + 10 = 5 \* 200,000 + 8000 + 10 = 108,010

Q3. The vector length is 2000 and will need 2000 threads since each thread compute one element. Since each warp contains 32 threads, we will have 2000/32 = 62.5 or 62 full warps. The last warp is only half full at 16 threads out of 32 and will experience divergence due to boundary check on vector length so 1 warp will be divergence.

Q4.

Case 1: Threads per SM will be the limiting factor because the kernel use 128 \* 30 = 3840 registers per block and only 65,536 / 3840 = 17.6 max blocks based on registers. However, max # of blocks based on threads is 2048 / 128 = 16 max blocks. The kernel can only fit 16 blocks based on the maximum number of threads per SM. Even though the register usage would allow for 17 blocks, it is constrained by the thread limit.

Case 2: It can achieve full occupancy because kernel will use 32 \* 29 = 928 registers per block and 65,536 / 928 =  70.6 max blocks based on registers. Max # of blocks based on threads is 2048 / 32 = 64 max blocks. In this case, while the kernel can theoretically support up to 70 blocks based on register usage, it is limited by the maximum number of blocks that can fit based on threads, which is 64.

Case 3: No full occupancy with limiting factor is the number of registers available per SM. Kernel use 256 \* 34 = 8704 registers per block and 65,536 / 8704 = 7.5 or 7 blocks. Max # of blocks based on threads is 2048 / 256 = 8 blocks. The kernel can fit up to 8 blocks based on the maximum threads per SM, but it is limited to only 7 blocks based on the available registers.