5 GPUs and SIMD [50 points]

We define the *SIMD utilization* of a program run on a GPU as the fraction of SIMD lanes that are kept busy with *active threads* during the run of a program.

The following code segment is run on a GPU. Each thread executes **a single iteration** of the shown loop. Assume that the data values of the arrays A, B, and C are already in vector registers so there are no loads and stores in this program. (Hint: Notice that there are 6 instructions in each thread.) A warp in the GPU consists of 64 threads, and there are 64 SIMD lanes in the GPU.

```
for (i = 0; i < 4096; i++) {
    if (B[i] < 8888) {
        A[i] = A[i] * C[i];
        A[i] = A[i] + B[i];
        C[i] = B[i] + 1;
    }
    if (B[i] > 8888) {
        A[i] = A[i] * B[i];
    }
}
```

(a) [10 points] How many warps does it take to execute this program?

Warps = (Number of threads) / (Number of threads per warp)
Number of threads = 2^{12} (i.e., one thread per loop iteration).
Number of threads per warp = 64 = 2^6 (given).
Warps = $2^{12}/2^6 = 2^6$

(b) [20 points] When we measure the SIMD utilization for this program with one input set, we find that it is 134/320. What can you say about arrays A, B, and C? Be precise (Hint: Look at the "if" branch).

A: Nothing

B: 2 in every 64 of B's elements are less than 8888, the rest are 8888

C: Nothing

(c) [10 points] Is it possible for this program to yield a SIMD utilization of 100% (circle one)?

YES NO

If YES, what should be true about arrays A, B, C for the SIMD utilization to be 100%? Be precise. If NO, explain why not.

Yes. All consecutive 64 elements of B should be either:
(1) All of B's elements are equal to 8888, or
(2) All of B's elements are less than 8888, or
(3) All of B's elements are greater than 8888.

(d) [10 points] What is the lowest SIMD utilization that this program can yield? Explain.

132/384. 1 in every 64 of B's elements are greater than 8888, and 1 in every 64 of B's elements are less than 8888, and the rest of the elements are 8888.