(d) [15 points] If we modify the vector processor to *support chaining*, how many cycles would be required to execute the same program in part (c)? Explain.

```
VLD  |--100--|--(VLEN-1)--|
VLD                       |---100---|---(VLEN-1)---|
VADD                      |-1-|-5-|---(VLEN-1)---| (this is delayed because the processor
                                                   executes the instructions in order)
VMUL                                |-10-|---(VLEN-1)---|
VST                                      |-100-|---(VLEN-1)---|

    100 + (VLEN-1) + 100 + 10 + 100 + (VLEN-1) = 310 + 2*1000 - 2 = 2308 cycles
```

9 GPUs and SIMD [45 points]

We define the *SIMD utilization* of a program that runs on a GPU as the fraction of SIMD lanes that are kept busy with *active threads* during the run of the program. As we saw in lecture and practice exercises, the SIMD utilization of a program is computed across the *complete run* of the 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 and B are already in vector registers, so there are no loads and stores in this program. (Hint: Notice that there are 3 instructions in each iteration.) A warp in the GPU consists of 32 threads, and there are 32 SIMD lanes in the GPU.

```
for (i = 0; i < 1025; i++) {
    if (A[i] < 33) {         // Instruction 1
        B[i] = A[i] << 1;    // Instruction 2
    }
    else {
        B[i] = A[i] >> 1;    // Instruction 3
    }
}
```

Please answer the following six questions.

(a) [2 points] How many warps does it take to execute this program?

33 warps.

Explanation:
The number of warps is calculated as:
$\#Warps = \lceil \frac{\#Total\_threads}{\#Warp\_size} \rceil$,

where
$\#Total\_threads = 1025 = 2^{10} + 1$ (i.e., one thread per loop iteration),

and
$\#Warp\_size = 32 = 2^5$ (given).

Thus, the number of warps needed to run this program is:
$\#Warps = \lceil \frac{2^{10}+1}{2^5} \rceil = 2^5 + 1 = 33.$

(b) [10 points] What is the *maximum* possible SIMD utilization of this program? (Hint: The warp scheduler does *not* issue instructions when *no* threads are active).

$\frac{1025}{1056}$.

Explanation:
Even though all active threads in a warp follow the same execution path, the last warp will only have one active thread.