EXERCISE #1

Your computer system has a 5-stage pipeline (but not super-pipeline) processor (like the one we discussed in the classroom). We use the five-stage pipeline processor to execute the following loop structure in assembly instructions.

**Question:** How much can this loop structure be faster if we apply loop unrolling and delayed branch using the following assumptions? Show all your work (especially for partial credit).

(a) The same number of stalled cycles for each RAW data hazard as we discussed in the classroom (i.e., 3 stalled cycles for each RAW data dependency).
(b) The delayed branch slot is 3 cycles.
(c) The loop structure repeats exactly 1,000 times.
(d) The 5-stage processor has an infinite number of registers (as many registers as you need).

```
LOOP: lw $t1, 0($t2)     // $t1 ← MEM[$t2]
      add $t3, $t1, 10    // $t3 ← $t1 + 10
      sw $t3, 0($t2)      // MEM[$t2] ← $t3
      add $t2, $t2, 4     // $t2 ← $t2 + 4
      bne $t2, $t6, LOOP  // if $t2 ≠ $t6, jump to LOOP
```
EXERCISE #2

Assume that we have a CPU with a pipeline datapath that has 20 \((k = 20)\) stages. The size of the branch slot in this CPU is 19 cycles (i.e., every time a conditional branch takes a jump, the processor loses 19 cycles). Now we are applying delayed-branch to assembly programs to be executed by this CPU. For the programs executed by this CPU, branches are taken for 56\% of their executions. It was also found that in 23\% of time, delayed-branch could not be applied because of data dependencies between the instructions. To simplify this question, let’s assume that any of the 19 instructions could not be moved after a branch instruction as long as any one of the 19 instructions has a data dependency (i.e., for 23\% of time, none of the 19 instructions can be moved to the behind of a conditional branch instruction). Assume that the number of the instructions the CPU executes is huge (you can safely consider that is infinity \((\infty)\)).

**Question:** in order for the slow-down due to conditional branch instructions to be less than 20\% (after the delayed-branch is used), what should be the frequency of branch instructions in your programs (i.e., how much of the assembly instructions are conditional branch instructions)?

*Show your work* (if you are running out of time, you can present a formula or equation, that will provide a solution for this question).

**Note:** For any meaningful effort, decent partial credit will be given.