1) Calculate the speedup that is achieved by a pipelined datapath versus a multi-cycle datapath under the following assumptions:
   - in the multi-cycle datapath, branches take 3 cycles, load words take 5 cycles and all other instructions take 4 cycles
   - 20% of instructions are load words
   - 15% of instructions are branches
   - in the pipelined datapath:
     - all data hazards except load word hazards are handled by forwarding
     - load word hazards cause a 1-cycle penalty and 50% of the load word instructions cause a hazard
     - branches cause a 2-cycle penalty
     - the time to initially fill the pipeline can be ignored

(Hint: Speedup can be calculated as $CPI_{\text{non-pipelined}}$ divided by $CPI_{\text{pipelined}}$.)

2) Consider the following MIPS assembly program executing in a pipelined datapath with no hardware for hazard handling:

   ```
   add $5, $0, $0
   lw $10, 1000($5)
   xor $4, $3, $2
   addi $20, $20, -4
   bne $20, $0, Sum
   ```

   a) Add as few NOPs as possible to this instruction sequence to ensure that hazards are eliminated and the code executes correctly. Do not reorder any instructions.

   b) Now reorder the instructions to eliminate as many NOPs as possible without affecting the correctness of execution.

3) Problem 6.13 from Patterson and Hennessy, 3rd edition.

4) Problem 7.25 from Patterson and Hennessy, 3rd edition.

5) Problem 7.32 from Patterson and Hennessy, 3rd edition.

(Not part of the assignment: for those who want additional practice problems on caches, do Problems 7.9, 7.10, 7.26, and 7.27 from Patterson and Hennessy.)