Reading:
PP, Chp. 5: The LC-3 ISA, datapath, and controller.
PP, Chp. 6.2: Using the LC-3 simulator

(NB--PP describe their Simulate.exe, which is very close to what we are using, PennSim.jar. The major difference is that PennSim will not execute an instruction with all zeroes. This is a flaw as such an instruction is a perfectly valid LC3 branch instruction; it just doesn't get used much.)

PP, Chp 5, problems:

5.4 (#address bits, PC-relative offset size and value)
5.5 (addressing modes and operand locations)
5.6 (BUSYNESS vector, bit masking)
5.7 (largest pos. immed. value)
5.9 (cc+br = nop?)
5.10 (given instr. bits, instr. diff?)
5.11 (immed. data limits)
5.13 (reg-reg transfer, sub, cc set, cc codes, clear reg)
5.14 (OR)
5.18 (#mem accesses LDR, STI, TRAP)
5.21 (#trap routines)
5.28 (remove STI from ISA)
5.31 [NB-"top of next page" => "below"] (instr bits from reg contents)
5.40 (ctl logic)
5.41 (cc logic)

PP, Chp 6, problems:

6.3 (set BUSYNESS vector according to x4000 content)
6.4 (prog. to compare R1 and R2: GT, EQ, LT)

Do 6.3 in parts:

(NB-The BUSYNESS vector in section 2.7.1 would more logically be called the IDLENESS vector because if bit_i is 1 then machine_i is idle. This problem states the opposite: a 1 indicates a machine is busy.)

(A) Write some code to turn ON an arbitrary bit in the BUSYNESS vector (memory location x4001). Assume register R1 has the appropriate bit pattern; for example, supposing one wants to turn on bit_12 in x4001, assume R1 has bit_12 on and all other bits are off.

(B) Assume R1 has only bit_0 = 1. Write code to left shift the bit to the position indicated by the contents of R2. R2 contains an integer ***i*** indicating we are to left shift ***i*** times.

(C) Combine the two code pieces with whatever additional code is needed to complete the solution. Turn in all three.

(D, Extra Credit) A-C only allow us to turn bits on in the BUSYNESS vector. Write code to turn the indicated bit on or off. Assume memory location x4002 contains all zeros if nothing is to be done, only bit_0 is on if the machine indicated in x4000 has become busy, and only bit_1 is on if the machine indicated is now idle.