1) For the LC3, how do you convert an assembly language program to machine code?

2) How many bits are there in an LC3 machine code instruction?

3) How many bits are there in an LC3 op code?

4) And given that there are this many bits, how many different possible instructions?

5) For the ADD and AND instructions, you can choose between taking the second value from the register file, or from sign extended bits from the instruction. Which bit in the instruction controls that?

6) And that bit (from question 5). What sort of circuit does it get routed to? (Hint. It’s drawn like a trapezoid).

7) The BR instruction can check for three bits to decide if you should take the branch. What do N, Z, and P stand for?

8) The BR instruction has a PCoffset9. How far can you branch forward? How far backwards?

9) The LC3 assembler takes two passes over a source code file. In the first pass, it builds up the symbol table. In the second pass, it uses the symbol table to determine the offsets it needs to insert to create the machine code. This really isn’t a question. It’s just a trick to get you to read something.

10) Write a “hello world” program for the LC3.

11) On a scale of 1 to 10... How much of this is making sense? Totally lost, or ok?