1) What is the range of values for the PC offset for the following instructions: JMP, LEA, STI?
JMP: N/A, address taken from register LEA: +255 to -256 STI: +255 to -256

2) What information is stored in the Processor State Register? what bit locations hold this information?
Bit 15 privilege, bits 10-8 priority level, bits 2-0: condition codes

3) What does the JMP R2 instruction do? How is this different from the RET instruction?
Sets the PC to the value contained in register R2, RET instruction always uses R7 as the source for the PC value

4) What is PC offset for the following BR instruction? Now show the 16 bits that make up the BR instruction.

```
        BRz data1
        ADD R2, R2, #5
        JSRR R4
data1   .FILL 0x1234
```

PC offset: +2 Bits: 0000 0100 0000 0010

5) Given the following information what will be the value in the PC after the following instruction TRAP x23, What is the first instruction executed by the TRAP routine?

location	data
x0020	x0420
x0021	x0251
x0022	x046C
x0023	x0326
x0024	x0324

location	data
x0230	x2A43
x0231	x32AC
x0232	x5E1F
x0233	x8FB2
x0234	xE8A1

location	data
x0323	xFADC
x0324	x32AC
x0325	xAE1F
x0326	x330F
x0327	x98A1

PC: x0326 Instruction: ST R1, #-241

6) What change signifies that the keyboard is in interrupt vs polling mode?
Bit 14 of the KBSR is set to a 1

7) What is the difference between the frame pointer and the stack pointer?
Stack pointer always points to the top of the stack, Frame pointer always points to the location of the first local variable regardless of the number of local variables. This is useful because offsets are know when the code is written.

8) List the 10 parts on the stack protocol in order. What parts have to be done by the caller or callee, and what parts were a choice made by the protocol designer?
Caller pushes arguments (last to first).
Caller invokes subroutine (JSR).
Callee allocates return value, pushes R7 and R5.
Callee sets up new R5; allocates space for local variables.
Callee executes function code.
Callee stores result into return value slot.
Callee pops local vars, pops R5, pops R7.
Callee returns (JMP R7).
Caller loads return value and pops arguments.
Caller resumes computation

Allocation of return value can be done by caller or callee, other parts are fixed

9) What do the following instructions have in common? LDI, LDR, STI, STR, JSRR, JMP, RET
The address they use can be any address in the LC3's memory

10) What instructions make up a PUSH? What instructions make up a POP?
PUSH Rx: ADD R6, R6, #-1
STR Rx, R6, #0
POP Rx: LDR Rx, R6, #0
ADD R6, R6, #1

11) If the LC3 had 32 registers how would the ADD instruction be affected?
32 registers takes 5 bits to specify a specific register, so ADD with two source registers would no longer fit in 16 bits and ADD immediate would only be left with 1 bit for an immediate value

12) How many times does the LDR instruction access memory? What about STI? What about TRAP?
LDR: once STI: twice TRAP: once (Add one to these values if including memory access to load instruction)

13) What is the purpose of bit 15 int the KBSR? Bit 14?
Bit 15: 1 = character ready to be read from keyboard. Bit 14: 1 = Interrupt mode enabled

14) Besides an interrupt signal what else does the interrupting device need to send the LC3 processor for an interrupt to be handled?
An (8 bit) interrupt vector that tells the LC3 where to find the interrupt service routine.