CMSC 411—Homework 2

1. Cite one advantage and one disadvantage of each of the following binary representation formats:

- sign/magnitude
- two's complement
- biased (a.k.a. “excess”) notation (e.g.: “excess 127” for IEEE 754 floating point)

2. What is the difference between the signed and unsigned versions of the ALU operations in the MIPS processor when both are available (e.g. **`add`** (signed add) vs. **`addu`** (add unsigned)?

3. The MIPS assembler allows you to use the pseudo-instruction **`blt`**: for "branch-if-less-than". However, the MIPS processor does not actually have such an instruction; the assembler simulates the instruction by substituting a sequence of one or more real MIPS instructions. Give one possible substitution. (Hint: look up the semantics of the **`slt`** instruction.)

4. The x86 instruction set uses the 2-operand model for ALU operations: all store the result in one of the original source operand locations, replacing what was there. This can be convenient for doing things like incrementing a variable in-place. By contrast, the MIPS ISA’s ALU operations are 3-operand: they require the destination operand to be specified separately from the two source operands. Why is this not an additional complication for a compiler writer? (This is a simple question—don’t overthink it.)

5. The MIPS has a "zero register" ($0, a.k.a. $zero) that always supplies a 0 when read. As a designer, enumerate at least one pro and one con of this design choice. (Hint: what is that register used for, and if there was *not* a zero register, how would you make up for it? Would the replacement be completely equivalent?)