CSCI 2330 – Floating Point Exercises

1. Using our toy 8-bit floating point representation (with k=4 exponent bits and 3 fractional bits), convert **00110100** into a decimal value.

2. Using the same 8-bit representation, convert **10000101** into a decimal value (working with a fraction here is advisable).

3. If **d** is a double in C, does $d < 0.0$ imply $((d * 2) < 0.0)$? (remember this is not true for ints)

4. Excluding infinity, what is the decimal value of the largest 32-bit IEEE floating point number? You should be able to write down the exact (unsimplified) expression.

5. IEEE 754 encodes the exponent value **E** using an unsigned **exp** field from which a **bias** value is subtracted. An alternate approach would be to just make **exp** encode a signed number and dispense with the bias term. Is there a reason to prefer the unsigned - bias approach?

(*Hint*: one of the design goals of IEEE 754 was to have floating point numbers ordered in the same way as if they were ints, to allow for easy comparisons -- e.g., the binary values $001 < 010 < 011$ are ordered accordingly regardless of whether they are ints or floating point numbers. Think about what an ordering of floating point values would look like if **exp** encoded a signed number.)