CSCI 2330 – Floating Point Exercises

1. Using an 8-bit IEEE 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. Excluding infinity, write down an expression giving the exact decimal value of the largest 32-bit IEEE floating point number (no need to simplify the expression).

4. IEEE 754 encodes the exponent value **E** using the **exp** bits as an unsigned value from which **bias** is subtracted (that is, **E = exp (unsigned) - bias**). A simpler encoding of **E** would be to just make the **exp** bits encode a signed value and get rid of the **bias** term (i.e., **E = exp**). Consider the two bit patterns 01000000 and 00100000 and the same 8-bit format above. Using the alternate, simpler encoding of E, which of these values is larger?

5. Consider the same two bit patterns as above (01000000 and 00100000). Using the actual IEEE 754 encoding of **E**, which of these values is larger? Why might this example explain why IEEE 754 uses this encoding of **E** instead of the simpler encoding described in #4?