Question 1

So, sign is - because of the first bit. Then, we flip the bits, add 1 and we add up the power of twos to get the number in decimal representing the magnitude. Combine with the negative symbol, and we have our number!

1. Flip the bits

$$(11111111111111111111111111101010)_2 = (00000000000000000000000000010101)_2$$

2. Add 1

$$(00000000000000000000000000010101 + 1)_2 = (00000000000000000000000000010110)_2$$

3. Calculate the magnitude

$$2^4 + 2^2 + 2^1 = 22$$

4. Combine

$$-22$$

Thus, our answer is -22.

Question 2

a.

The exponent bias should be 7.

b.

The machine espilon is

$$(00111001 - 00111000)_2 = \epsilon = 2^{-3} = 0.125$$

c.

Largest: $(01110111)_2 = 2^7 * (1 + 2^{-1} + 2^{-2} + 2^{-3}) = 128 * 1.875 = 240$
Smallest: $(00001000)_2 = 1 * 2^{-6} = 0.015625$

d.

No, not all integers. Proof by counterexample that not any integer can be represented. For example, 239 cannot be stored. We can store 240, as seen above. However, the smallest step under would be $(01110110)_2$ which is $2^7 * (1 + 2^{-1} + 2^{-2}) = 128 * 1.75 = 224$. Thus, since 239 is not representable, by contraditction we prove that not all ints are representable.

e.

largest: $(00000111)_2 = 2^{-6} * (2^{-1} + 2^{-2} + 2^{-3}) = 0.013671875$
smallest: $(00000001)_2 = 2^{-6} * 2^{-3} = 0.001953125$

f.

5 is 101_2 so exponent is 2+7 = 9 so $(1001)_2$. Then, we have $(1.01)_2$ left so $(010)_2$ is the fraction, because of hidden bit normalization! So, $(5)_{10} = (01001010)_2$.

- largest step before: $(01001001)_2$
- smallest step after: $(01001011)_2$

g.

- round down
  So, we know the first bit is negative, thus we need to round up our magnitude so it rounds towards $-\infty$. Let's then focus on rounding $(10.1011)_2$ up. $(10.1011)_2 = (1.01011)_2 * 2^1$ then
  - Exponent: $(1000)_2$ since bias is 7.
  - Fraction: $(011)_2$ since we round up
  Thus: $(11000011)_2$ is the number in binary8.
- round up
  Here, we must round down the magnitude since it is negative.
  $(11000010)_2$

- round to zero
  This would be x_+ since it is closer to 0 so $(11000010)_2$
- round towards nearest
  The nearest would be $(11000011)_2$.

h.

Lets try x smaller than subnormal, even if we might not be able to represent it. The smallest subnormal representable in binary8, as established before, is $2^{-6} * 2^{-3} = 2^{-9}$. We also established that the machine epsilon is 0.125. So, let's try with $x = 2^{-12}$.

$$\frac{|round(x) - x|}{|x|} > 12\epsilon$$

$$\frac{2^{-9} - 2^{-12}}{2^{-12}} > 1.5$$

assuming we round 2^{-12} to nearest subnormal so 2^{-9}

$$7 > 1.5$$

which holds. So x = 2^{-12} works!

Question 3

a.

True. Since $x \oplus x = \text{round}(x + x) = \text{round}(2x) = 2x$ since x is a finite fpn.

b.

False. Let us pick $x = 1$ and $y = 2^{-4} = 0.0625$. We'll assume binary8.

$$x \ominus y = \text{x - y} = \text{round}(1 - 0.0625) = \text{round}(0.9375) = (00111000)_2 = 1$$

using round up.

Then,

$$x \ominus x = -(y \ominus x) = -(\text{round}(0.0625 - 1))$$

Then, assuming round up again,

$$= -(\text{round}(0.0625 - 1)) = -\text{round}(-0.9375) = (00000111)_2 = 0.875$$

but $0.875 \neq 1$. Thus, this does not hold by counterexample.

Question 4

a.

$-\infty/0 = -\infty$ assuming positive 0.

b.

$\infty/(-\infty) = \text{NaN}$

c.

$3 * \text{NaN} - \text{NaN} = \text{NaN}$

d.

$0/(-\text{NaN}) = \text{NaN}$