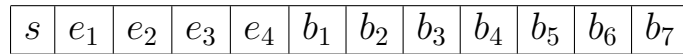


Worksheet: If IEEE 754 had a 12-bit standard ...

A floating point system \mathbb{F} described in Lecture 13 of the textbook (L. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM Press 1997) is, in reality, implemented in bits. The actual IEEE 754 standards for 32-bit single precision and 64-bit double precision representations are cumbersome to play with, so for convenience we pretend here that the standard has a 12-bit version. It might look like this:



These 12 bits are organized to represent a *nonzero* number:

$$x = (-1)^s (1.b_1b_2b_3b_4b_5b_6b_7)_2 2^{(e_1e_2e_3e_4)_2 - (0111)_2}$$

Note that $(1.b_1b_2b_3b_4b_5b_6b_7)_2$ is called the *mantissa*. The power on the 2 is the *exponent*. The special offset $(0111)_2$, equal to 7 in base ten, is called the *exponent bias*. We also define some exceptional cases:

- exponent bits $(0000)_2$ define the number zero or subnormal numbers
- exponent bits $(1111)_2$ define the other exceptions: $\pm\infty$ and NaN

We will say nothing further about the $(1111)_2$ exceptions.

(a) What is the largest real number that this system can represent? Show the bits.

--	--	--	--	--	--	--	--	--	--	--	--

(b) Not considering subnormal numbers, what is the smallest positive number that this system can represent? (*The first normal number to the right of zero.*) Show the bits.

--	--	--	--	--	--	--	--	--	--	--	--

(c) If we define $\epsilon_{\text{machine}}$ as the gap between 1 and the next representable number greater than 1, what is the value of $\epsilon_{\text{machine}}$ in this system?

