1. How many positions are needed to represent the following numbers in base-2 with no error?

a. $(1.5)_{10}$

$\log_2(0.5) = 1$ fraction bit

$\therefore$ only 2 bits (one integer, one fraction) are needed to accurately represent $(1.5)_{10}$:
$(1.1)_2 = (1 \times 2^0) + (1 \times 2^{-1}) = (1.5)_{10}$

b. $(1.05)_{10}$

Impossible to store with complete accuracy, since $\log_2(0.05) \approxeq -4.321$ is not an integer. This indicates there is no finite number of bits that can be picked to store it. Someone call the IEEE wizards!

2. Given n=4 positions, put the fraction point in a position for the following numbers in base-2 to minimize the conversion error.

a. $(1.5)_{10}$

xxx.x will have precision 0.5 (2^{-1}), thus at least one bit must be allocated to the fraction to have perfect precision. Thus, the following are all valid and completely accurate representations:

- $(001.1)_2$
- $(01.10)_2$
- $(1.100)_2$

b. $(5.5)_{10}$

xxx.x will still have precision 0.5 (2^{-1}), so at least one fraction bit is required, however $(5)_{10}$ requires 3 bits so the only perfect representation is $(101.1)_2$.

c. $(10.5)_{10}$

The glaring issue with this number is that there are only 4 bits to represent the number, but 4 bits are needed for the integer. Thus, the only solution is to minimize error. 0.5 is less than 3 (10 - 7, the max for a 3-digit binary number), so we must forego the fraction bit. Therefore the most accurate binary representation possible is $(1010.)_2$.