1. Given n bits, how many distinct combinations of the n bits exist?
2. What is the 6-bit 2’s complement representation of -32? What is the 12-bit representation of the same number?
3. What is the largest positive number we can present with an 8-bit 2’s complement code? Write the result in both binary and decimal.
4. Suppose that we are writing a program and we define a variable x of type character (i.e. of size 1 byte). What happens if you assign to x a value larger than the one you calculated in question# 3 above?
5. Given n bits, how many unsigned integers can be represented with the n bits? What is the range of these integers?
6. Suppose A=0110, B=0000, and C=1111. These values are all in binary. Calculate the following and leave the result in binary.
   a. \(\text{NOT}(A \text{ AND } B)\)
   b. \(\text{NOT}(A) \text{ OR } \text{NOT}(B)\)
   c. \(\text{NOT}(A \text{ OR } B)\)
   d. \(\text{NOT}(A) \text{ AND } \text{NOT}(B)\)
7. Write the IEEE floating point representation (single precision) of 3.75
8. Convert the following hexadecimal number to binary: 0x0F1E2D
9. Why do you think we need to learn binary? Why do we need to learn hexadecimal then?
10. Why computers use binary instead of, say, decimal?
11. What is the minimum number of bits required to present the following numbers (If there are no signs before the number, it is an unsigned number.). For each one, give the binary presentation too:
   a. -7
   b. +14
   c. 7
   d. 14
12. Can we add a signed number to an unsigned number? Why?
13. Why do we need sign extension?