Software Development Environments Base Representation and basic binary operations Tutorial

1) Convert the following base 10 numbers to base 2 using the minimum number of bits
    a. 123 1110011
    b. 257 100000001
    c. 1035 10000001011
    d. 265 100000000
    e. 22 10110
2) Convert the following base 10 numbers to base 2 using an 8 bit binary representation
    a. 123 01111011
    b. 220 11011100
    c. 105 011`01001
    d. 99 01100011
    e. 7 00000111
    f. 17 00010001
3) Convert the following base 10 numbers to base 2 using a maximum of 8 bits
    a. 0.3125 (0.0101 b2)
    b. 0.75 0.11 b2
    c. 0.1875 0.0011 b2
    d. 0.2 0.00110011 b2

    corrected
4) Convert the following binary numbers to decimal
    a. 10100010 162
    b. 00000111 3
    c. 10101011 171
    d. 00011001 25 corrected
    e. 10101011.11000110 171.7734375 b10
    f. 00000000.10001000 0.53125 b10
    g. .10101111 .68359375 b10
5) Convert the following hexadecimal numbers to binary
    a. AB1F 1010101100011111 b2
    b. 1200 0001001000000000
    c. A1B2 1010000110110010
    d. 12BE 0001001010111110
    e. 1CD0 0001110011010000
6) Convert the following binary numbers to hexadecimal
    a. 11010101 D5 base 16
    b. 10111100 BC base 16
    c. 101011 3B b16
    d. 110000 30 b16
    e. 10010101 95 b16
7) Add the following binary numbers
    a. 10101010 + 11011110 110001000
    b. 10011011 + 10110111 101010010
    c. 10101010 + 1010 10110100
    d. 10101 + 1111 100100

8) subtract the following binary numbers
   a. 10111011 – 00101111 10001100
   b. 10110110 – 01011001 01011101
   c. 10000010 – 00001010 01111000
   d. 10011100 – 10000001 00011011

9) If we have an 8-bit representation and we are using 2s complement notation. Convert the following numbers to decimal
   a. 00101010 -11 / -22 *
   b. 00011111 -16 / -1 *
   c. 10000000 -1 / -128 *
   d. 10000001 -127
   e. 10111000 -72
   f. 10101000 -88

10) Convert the following decimal numbers to 2s complement using an 8-bit representation
   a. 64 01000000
   b. 127 01111111
   c. 21 00010101
   d. -21 11101011
   e. -127 10000001
   f. -64 11000000
   g. -128 10000000 *

11) Carry out the subtraction of the following decimal numbers using two's complement notation
   a. 66 – 125 11000101
   b. 120 – 36 0010100 *ignore carry 1
   c. 155 – 172 11101111
   d. 99 – 128 11100011

12) Convert the following decimal numbers to the 32 bit IEEE representation
   a. 25.3125 0 10000011 10010101 000000000000000
   b. 256.65625 0 10000111 00000000 000000000000000
   c. -0.03125 1 01111111 00001 00000000000000000
   d. -150.1875 1 01000110 00101100011 000000000000

13) Covert the following 32 bit IEEE numbers into decimal
   a. 11000001010001000000000000000000 -12.25 base10
   b. 10111111000000000000000000000000 -0.5 base 10
   c. 01000010000011000110000000000000 35.09375 base 10
   d. 01000010100001010000000000000000 66.5 base 10

14) Convert the decimal numbers to an 8 bit binary representation and carry out the following bitwise operation
   a. 23 & 45 where & bitwise AND 000101 = 5
   b. 23 | 45 where | bitwise OR 111111 = 64
   c. 23 ^ 45 where ^ bitwise XOR 111010 = 58
   d. ~67 where ~ bitwise NOT 67 = 1000011 NOT = 0111100 = 60
   e. 115 >> 2 where >> right bit shift 1110011 shift 2 to the right = 00011100
   f. 16 << 3 where << left bit shift 10000 shift 3 to the left = 10000000