1. Problem 1
   - Convert 623 from decimal to binary
     i. 623/2 => R = 1
     ii. 311/2 => R = 1
     iii. 155/2 => R = 1
     iv. 77/2 => R = 1
     v. 38/2 => R = 0
     vi. 19/2 => R = 1
     vii. 9/2 => R = 1
     viii. 4/2 => R= 0
     ix. 2/2 => R = 0
     x. ½ => R = 1
     xi. Therefore, the binary representation of 623 (decimal) is 1111011001

   - Convert 11011010 from binary to decimal
     i. $1 * 2^7 = 128$
     ii. $1 * 2^6 = 64$
     iii. $0 * 2^5 = 0$
     iv. $1 * 2^4 = 16$
     v. $1 * 2^3 = 8$
     vi. $0 * 2^2 = 0$
     vii. $1 * 2^1 = 2$
     viii. $0 * 2^0 = 0$
         1. The final answer is the sum of these values: (128 + 64 + 0 + 16 + 8 + 0 + 2 + 0) = 218
         2. Therefore the decimal representation of 11011010 (binary) is 218

   - Convert the decimal number 39123 to base 16 (hexadecimal).
     i. 39123/16 => R = 3
     ii. 2445/16 => R = D
     iii. 152/16 => R = 8
     iv. 9/16 => R = 9
     v. Therefore, the hexadecimal representation of 39123 (decimal) is 98D3
   - $(BEAD)_{16} = (x)_{10,}$ solve for x
     i. B = 11, $11 * 16^3 = 45056$
     ii. E = 14, 14 * 162 = 3584
     iii. A = 10, $10 * 16^1 = 160$
     iv. D = 13, $13 * 16^0 = 13$
         1. Final answer is the sum: (45056 + 3584 + 160 + 13) = 48813
         2. We can plug this back into the equation to get :
             a. x = 48813

2. Problem 2
   Part A)
   - First convert 5E to binary
     i. $(5)_{16} = (101)_2$
     ii. $(E)_{16} = (1110)_2$
     iii. $(5E)_{16} = (01011110)_2$
   - Convert $(34)_{16}$ to binary
     i. $(34)_{16} = (00110100)_2$

- Convert $(AD)_{16}$ to binary
    i. $(A)_{16} = (1010)_2$
    ii. $(D)_{16} = (1101)_2$
    iii. $(AD)_{16} = (10101101)_2$
- Convert 91 to binary
    i. $(91)_{16} = (10010001)_2$
- IP address in binary
    i. 01011110:00110100:10101101:10010001
- IP in dotted decimal format
    i. 46.52.173.145

Part B)

- With an 8-bit binary number there is 256 possible combinations. This is due to the fact that each address is made of 4 8-bit numbers to create a 32 bit address.
- There are 4,294,967,296 possible combinations of IP address as 256^4
- Due to this, there are not enough 32-bit IP addresses for over 5 billion users to have their own.

3. Problem 3

Part A)

- $(10000110)_2$ => 01111001 + 1 = 01111010
    - $(01111010)_2$= 2 + 8 + 16 + 32 + 64 = $(-122)_{10}$

- $(10010111)_2$ => 01101000+ 1 = 01101001
    - $(01101001)_2$= 1+8+32+64 = $(-105)_{10}$

- $(01101011)_2$ => 10010100 + 1 = 10010101
    - $(10010101)_2$= -128+16+4+1= $(107)_{10}$

Part B)

- $(-45)_{10}$ => $(00101101)_2$ => 11010010 + 1 = $(11010011)_2$
- $(38)_{10}$ => $(01111001)_2$ => 01111001 + 1 = $(01111010)_2$
- $(-86)_{10}$ => $(01010110)_2$ => 10101001+ 1 = $(10101010)_2$

4. Problem 4
    - The range of a 13 bit twos complement would be -4096 to 4095. This can be found using the formula that tells us the lower limit is $-(2^{(n-1)})$ with the higher limit being $2^{(n-1)} - 1$.
    - The minimum number of bits that would be needed in two complement is 15 bit number. We can do this by setting $8888 = 2^{(n-1)} - 1$. When we do this, we get answer of 14.11. Due to the fact that there cannot be a half of a bit, this means that we need at least a 15 bit number to represent 8888.
5. Problem 5
    - When we do the multiplication of the $(5)_{10}$ and $(2)_{10}$, we first have to convert the number to binary, and we get a calculated answer of $(1010)_2$, which is not correct. This is due to the fact that the range of 3-bit is from 0 to 7, mean that it the multiplication cannot be represented.
    - When we use the 4-bit 2 complement, we get a calculated answer of $(1000)_2$. This is also incorrect, because the range for a 4 bit 2s complement is from -8 to 7.

- When we use the 8 bit 2 complement we get an answer of $(0001\ 0000)_2$, which is correct. The reason for this is because the range for an 8-bit 2's complement is -128 and 127, meaning that the answer for of 10 will be within the range and correctly represented.
- There is not a value x > 2, where multiplying two numbers will always have the correct answer. We can solve for this using the formula to find the largest number represented by a x-bit number: $2^{(x-1)} - 1$. If we want to find the number at which this is the largest, we can multiply both numbers: $(2^{(x-1)} - 1)^2$. We know that the largest number represented by x-bit is $2^{(x-1)} - 1$, so we can set $(2^{(x-1)} - 1)^2 = 2^{(x-1)} - 1$, to get an x value of 1 and 2, which is not over 2.

6. Problem 6
    - The weight set can be seen as an equivalent to a 6-bit binary number. With the each of the weights corresponding to each place of the binary number. We can represent the 32lb weight to be the 2^6 place, the 16lb weight can be represented by the 2^5 place, and so on. We can use the 1 and 0 in the binary number to represent the weight being placed or not, sort of like a Boolean. Through this we can visualize the scenario to be like finding a range of numbers that a 6 bit can correctly represent allowing us to see what weights this scale can measure. If we use the range formula 2^n-1, we can plug 6 in for n to get a max 63. This means that the weights can measure from the range of 0 to 64lbs.
    - The X that will be the max weight to do is 26. This is due to the fact that there are 3 different weights and three different options as to where to place them. You can either place it on the right, left, or not on the scale at all. We can exclude the case where the weights are not paced on the scale, and ten we can use the $3^3 - 1$ to get 26.