1. Unsigned Integers

1.1. Conversions

1.1.a. Convert the following from their initial radix to the other two common radices: 0b10001110, 0xC3BA, 81, 0b100100100, 0xBCA1, 0, 42, 0xBAC4

```
0b10001110 to Decimal:
     10001110
     (0 * 2^0) + (1 * 2^1) + (1 * 2^2) + (1 * 2^3) + (0 * 2^4) + (0 * 2^5) + (0 * 2^6) + (1 * 2^7)
     0 + 2^1 + 2^2 + 2^3 + 0 + 0 + 0 + 2^7
     0 + 2 + 4 + 8 + 0 + 0 + 0 + 128
     2 + 4 + 8 + 128
     142
0b10001110 to Hexadecimal:
     10001110
     1000 1110
     1000 = 8; 1110 = E
     0x8E
0xC3BA to Decimal:
     C3BA
     C = 12; 3 = 3; B = 11; A = 10
     (10 * 16^0) + (11 * 16^1) + (3 * 16^2) + (12 * 16^3)
     10 + 176 + 768 + 49152
     50106
0xC3BA to Binary:
     C3BA
     C = 1100; 3 = 0011; B = 1011; A = 1010
     1100 0011 1011 1010
     0b1100001110111010
81 to Binary:
     81
     81 / 2 = 40 R1
     40 / 2 = 20 R0
     20 / 2 = 10 R0
     10 / 2 = 5  R0
     5 / 2  = 2  R1
     2 / 2  = 1  R0
     1 / 2  = 0  R0
     0010001
     0b10001
81 to Hexadecimal:
     81
     81 / 16 = 5 R1
     5 / 16  = 0 R5
     0x51
0b100100100 to Decimal:
     100100100
     (0 * 2^0) + (0 * 2^1) + (1 * 2^2) + (0 * 2^3) + (0 * 2^4) + (1 * 2^5) + (0 * 2^6) + (0 * 2^7) + (1 * 2^8)
     0 + 0 + 2^2 + 0 + 0 + 2^5 + 0 + 0 + 2^8
     0 + 0 + 4 + 0 + 0 + 32 + 0 + 0 + 256
     4 + 32 + 256
     292
0b100100100 to Hexadecimal:
     100100100
     000100100100
     0001 0010 0100
     0001 = 1; 0010 = 2; 0100 = 4
     0x124
```

```
            0 to Binary:
                  0
                  0 / 2 = 0 R0
                  0b0
            0 to Hexadecimal:
                  0
                  0 / 16 = 0 R0
                  0x0
            42 to Binary:
                  42
                  42 / 2 = 21 R0
                  21 / 2 = 10 R1
                  10 / 2 = 5  R0
                  5 / 2  = 2  R1
                  2 / 2  = 1  R0
                  1 / 2  = 0  R1
                  0b101010
            42 to Hexadecimal:
                  42
                  42 / 16 = 2 R10; 10 = A
                  2 / 16  = 0 R2 ; 2  = 2
                  0x2A
            0xBAC4 to Decimal:
                  BAC4
                  B = 11; A = 10; C = 12; 4 = 4
                  (4 * 16^0) + (12 * 16^1) + (10 * 16^2) + (11 * 16^3)
                  4 + 192 + 2560 + 45056
                  47812
            0xBAC4 to BINARY:
                  BAC4
                  B = 1011; A = 1010; C = 1100; 4 = 0100
                  1011 1010 1100 0100
                  0b1011101011000100
      1.1.b. Write the following using IEC prefixes: 2^14, 2^43, 2^23, 2^58, 2^64, 2^42
            2^14 to IEC Prefix:
                  2^14
                  2^4 * 2^10
                  2^10 = Ki
                  2^4 Ki
                  16 Ki
            2^43 to IEC Prefix:
                  2^43
                  2^3 * 2^40
                  2^40 = Ti
                  2^3 Ti
                  8 Ti
            2^23 to IEC Prefix:
                  2^23
                  2^3 * 2^20
                  2^20 = Mi
                  2^3 Mi
                  8 Mi
            2^58 to IEC Prefix:
                  2^58
                  2^8 * 2^50
                  2^50 = Pi
                  2^8 Pi
                  256 Pi
```

2^64 to IEC Prefix:
2^64
2^4 * 2^60
2^60 = Ei
2^4 Ei
16 Ei

2^42 to IEC Prefix:
2^42
2^2 * 2^40
2^40 = Ti
2^2 Ti
4 Ti

1.1.c. Write the following as powers of 2: 2 Ki, 512 Pi, 256 Ki, 32 Gi, 64 Mi, 8 Ei

2 Ki to Power of 2:
2 Ki
2 = 2^1
Ki = 2^10
2^1 * 2^10
2^11

512 Pi to Power of 2:
512 Pi
512 = 2^9
Pi = 2^50
2^9 * 2^50
2^59

256 Ki to Power of 2:
256 Ki
256 = 2^8
Ki = 2^10
2^8 * 2^10
2*18

32 Gi to Power of 2:
32 Gi
32 = 2^5
Gi = 2^30
2^5 * 2^30
2^35

64 Mi to Power of 2:
64 Mi
64 = 2^6
Mi = 2^20
2^6 * 2^20
2^26

8 Ei to Power of 2:
8 Ei
8 = 2^3
Ei = 2^60
2^3 * 2^60
2^63

2. Signed Integers

2.1. Two’s complement

2.2. Exercises (Two’s Complement Number and Unsigned Number)

2.2.1. What is the largest integer? The largest integer + 1?

Two's Complement:

The largest integer is 127 as filling all bits with 1 except the sign bit results in 01111111 = 127.

The largest integer + 1 is 128 as 127 + 1 = 128.

Unsigned:

The largest integer is 255 as filling all bits with 1 results in 11111111 = 255.

The largest integer + 1 is 256 as 255 + 1 = 256.

2.2.2. How do you represent the numbers 0, 3, and -3?

Two's Complement:

0 can be represented as 0b00000000 = 0.
3 can be represented as 0b00000011 = 3.
-3 can be represented as 0b11111101 = -3.

Unsigned:

0 can be represented as 0b00000000 = 0.
3 can be represented as 0b00000011 = 3.
-3 can NOT be represented as an unsigned 8-bit integer.

2.2.3. How do you represent 42, -42?

Two's Complement:

42 can be represented as 0b00101010 = 42.
-42 can be represented as 0b11010110 = -42.

Unsigned:

42 can be represented as 0b00101010 = 42.
-42 can NOT be represented as an unsigned 8-bit integer.

2.2.4. What is the largest integer that can be represented by any encoding scheme that only uses 8 bits?

Two's Complement:

The largest integer is 127 as filling all bits with 1 except the sign bit results in 01111111 = 127.

Unsigned:

The largest Integer is 255 as filling all bits with 1 results in 0b11111111 = 255.

Therefore, the largest integer that can be reprsented in 8 bits is an unsigned number with value 0b11111111 = 255.

2.2.5. Prove that the two’s complement inversion trick is valid (i.e. that x and x’+ 1 sum to 0).

For any given number x we have x + x’ = 0b1...1. And for a regular hand calculation we have 0b1...1 + 0b1 = 0.

2.2.6. Explain where each of the three radices shines and why it is preferred over other bases in a given context

Decimal is the preferred radix for regular hand calculations, because it is easier to calculate on two hands.

Binary based radix is the most useful for computers because the transistors inside the circuits in computers requires electronic signals to operate. The on and off states for the switches are determined by the 0 and 1 set by binary and are the easiest way to send signals across the transistors,

Hexadecimal numbers are a convenient for people who program and need to point out locations in the program memory. This system can represent each byte as two hex digits instead of the 8 digits that binary requires.

3. Counting

3.1. Exercises

3.1.1. How many bits do we need to represent a variable that can only take on the values 0, π or e?

To represent a variable that can only take on the values 0, p or e, we need 2 bits.

Since there are 3 things to represent, we need n = [log(base2)3] = 2 bits.

3.1.2. If we need to address 2TiB of memory and we want to address every byte of memory, how long does an address need to be?

2 TiB = 2 x (2^40)
{log(base2)[2 x 2^40]}

{log(base2)[2] + log(base2)[2^40]

1 + 40

41

3.1.3. If the only value a variable can take on is e, how many bits are needed to represent it.

If the only value a variable can take on is e, 0 bits are needed to represent it.