UCF

BASE CONVERSION

COP 3502

The Decimal Numbering System

- What is the decimal numbering system?

- Have you ever thought about why we use this system or how it came about?

Base 10

- Our regular counting system is decimal
  - i.e. Base 10
    - This is because we use 10 distinct digits, 0 → 9
  - In general the numerical value of a number, is what you were taught in elementary school:
    - 2713
    - $= 2x10^3 + 7x10^2 + 1x10^1 + 3x10^0$
    - Where each digit's value is determined by what place it's in, going from right to left:
      - **3** is in the **1's** place — **3**
      - **1** is in the **10's** place — **10**
      - **7** is in the **100's** place — **700**
      - **2** is in the **1000's** place — **+2000**
      - **2713**

Note: each 'place', is a perfect power of the base, i.e. 10

Base Conversion

- Although this may seem to be the only possible numbering system,
  - it turns out that the number of digits used is arbitrary!
  - We could just have easily chosen 5 digits (0→4),
    - in which case the value of a number would be:
    - 314_5 =
    - = $3x5^2 + 1x5^1 + 4x5^0 = 84_{10}$
  - Thus, this is how we convert from a different base to base 10

Base Conversion

- $314_5 = 3x5^2 + 1x5^1 + 4x5^0 = 84_{10}$

- In general, we can use the following formula:

$$d_{n-1}d_{n-2}\ldots d_2d_1d_{0\text{ (in base b)}} = d_{n-1}xb^{n-1} + d_{n-2}xb^{n-2} + \ldots + d_2xb^2 + d_1xb + d_0$$

  - (Note: b raised to the 1 and 0 powers were simplified)

Base Conversion to Base 10

- Given, **781_9** what would this be base 10?
  - **$781_9 = 7x9^2 + 8x9^1 + 1x9^0$**
  - **$= 640_{10}$**

- Given, **1110101_2** what would this be base 10?
  - **$1110101_2 = 1x2^6 + 1x2^5 + 1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0$**
  - **$= 117_{10}$**
  - Base 2 is so common it's called binary
    - (heard of it before?)

Binary and Hexadecimal

- The uses of Binary and Hexadecimal
  - A “digital” computer, (vs analog) operates on the principle of 2 possible states – ON and OFF
    - This corresponds to there being an electrical current present, or absent.
  - So ON is “1” and OFF is “0”
  - Each binary digit, or “bit” corresponds to a single “switch” in a circuit
  - And if we add up enough switches we can represent more numbers, so instead of one digit we can get 8 to make a byte.

Binary and Hexadecimal

- So why do all computers use binary then?
  - Simple answer – computer weren't designed to use binary, rather, binary was determined to be the most practical system to use with the computers we did design

Binary and Hexadecimal

- So this explains why we would want to use binary, but why would we ever want hexadecimal?
  - Octal and hexadecimal are simply a shorter representation of binary, or a more human readable version of binary.
  - Usually used more memory addresses or RGB color values
  - So if you want to specify this color:
    - You don't have to type: 1111 1111 0000 0000 1111 1111
    - You can type: #FF00FF

Hexadecimal

- The most common base above 10 is 16
  - Which is known as Hexadecimal
  - The 16 digits are:
  - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
    - Where A = 10, B = 11, C = 12, D = 13, E = 14, F = 15

  - Note that converting from Hexadecimal to Decimal, is the same as the previous slide

  - **$A3D_{16}$ =**
  - **$= A \times 16^2 + 3 \times 16^1 + D = 10 \times 16^2 + 3 \times 16 + 13$**
  - **$= 2621_{10}$**

Converting from Hexadecimal to Binary

- Since 16 is a perfect power of 2, converting to base 2 is relatively easy ☺
  - We find that each hexadecimal digit is perfectly represented by 4 binary digits (since $16 = 2^4$)
- Here's a chart with the conversions between each hexadecimal digit and base 2:

Hex	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Bin	0000	0001	0010	0011	0100	0101	0110	0111
Hex	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
Bin	1000	1001	1010	1011	1100	1101	1110	1111

Converting from Hexadecimal to Binary

- Using the chart below,
  - What is **ACD_{16}** in binary?
    - **= $1010\ 0011\ 1101_2$**

Hex	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Bin	0000	0001	0010	0011	0100	0101	0110	0111
Hex	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
Bin	1000	1001	1010	1011	1100	1101	1110	1111

Converting from Decimal to Another Base

- Let's consider converting a number in decimal to another base, say binary
  - For example convert 117_{10} to binary
  - We know that the format is going to look something like this:
  - $117_{10} = d_6 x 2^6 + d_5 x 2^5 + d_4 x 2^4 + d_3 x 2^3 + d_2 x 2^2 + d_1 x 2^1 + d_0 x 2^0$
- So, what we know is this:
  - All of the terms on the right hand side are divisible by 2 except the last.
    - Sooo all of them will give a remainder of 0, when divided by 2
    - Thus we know that d_0 = 117%2, this is the remainder

Converting from Decimal to Another Base

- $117_{10} = d_6x2^6 + d_5x2^5 + d_4x2^4 + d_3x2^3 + d_2x2^2 + d_1x2^1 + d_0x2^0$

- Now if we divide the right hand side by 2, using int division we get:
  - $\mathbf{d_6x2^5 + d_5x2^4 + d_4x2^3 + d_3x2^2 + d_2x2^1 + d_1x2^0}$
  - This number must equal 117/2 = 58,
  - So we've come up with a process of continually reducing the equation by a factor of 2…

Converting from Decimal to Another Base

- $117_{10} = d_6 x 2^6 + d_5 x 2^5 + d_4 x 2^4 + d_3 x 2^3 + d_2 x 2^2 + d_1 x 2^1 + d_0 x 2^0$

- Here is the whole process:
  - **117 % 2 = 1 (d_0) 117/2 = 58**
  - **58 % 2 = 0 (d_1) 58/2 = 29**
  - **29 % 2 = 1 (d_2) 29/2 = 14**
  - **14 % 2 = 0 (d_3) 14/2 = 7**
  - **7 % 2 = 1 (d_4) 7/2 = 3**
  - **3 % 2 = 1 (d_5) 3/2 = 1**
  - **1 % 2 = 1 (d_6) 1/2 = 0 (we can stop)**

Now, read the answer from the bottom up!

Converting from Decimal to Another Base

- $117_{10} = d_6 x 2^6 + d_5 x 2^5 + d_4 x 2^4 + d_3 x 2^3 + d_2 x 2^2 + d_1 x 2^1 + d_0 x 2^0$

- Here is the whole process:
  - **117 % 2 = 1 (d_0) 117/2 = 58**
  - **58 % 2 = 0 (d_1) 58/2 = 29**
  - **29 % 2 = 1 (d_2) 29/2 = 14**
  - **14 % 2 = 0 (d_3) 14/2 = 7**
  - **7 % 2 = 1 (d_4) 7/2 = 3**
  - **3 % 2 = 1 (d_5) 3/2 = 1**
  - **1 % 2 = 1 (d_6) 1/2 = 0 (we can stop)**

Now, read the answer from the bottom up!

1 1 1 0 1 0 1

Practice Problem

- Convert the following from decimal to the given base:

175_{10} = ____$_3$

175 % 3 = 1 175/3 = 58
58 % 3 = 1 58/3 = 19
19 % 3 = 1 19/3 = 6
6 % 3 = 0 6/3 = 2
2 % 3 = 2 2/3 = 0,

so $175_{10} = 20111_3$.

381_{10} = ____$_{16}$

381 % 16 = 13 (D) 381/16 = 23
23 % 16 = 7 23/16 = 1
1 % 16 = 1 1/16 = 0,

so $381_{10} = 17D_{16}$.

Converting from any base

- Converting from any base (B1) to and other base (B2) where neither base is base 10
  - What do we do?
  - Don't panic!
    - We can:
    1. Convert from base B1 to base 10
    2. Convert from base 10 to base B2
    - Phew! We know how to do both of those!

Converting from any base

- If you are converting between two bases that are perfect powers of 2,
  - the following procedure works more quickly:
    1. Convert from base B1 to base 2
    2. Convert from base 2 to base B2
  - Consider the following example
    - We want to know what **ACD_{16}** is in base 8
    - We know **ACD_{16} = $1010\ 0011\ 1101_2$ from before**
    - **Know we must convert the right-hand side to base 8.**

Converting from any base

- We know **ACD_{16} = $1010\ 0011\ 1101_2$** from before
- Now we must convert the right-hand side to base 8.
  - Remember that 8 = 2^3, so thre3 binary digits perfectly represent one octal (base 8) digit.
  - So let's just group the binary digits in sets of 3, from right to left
    - **$101\ \ 000\ \ 111\ \ 101_2$**
  - And finally convert each set of three binary digits to its octal equivalent:
    - **5075_8**
  - **Note: only works when one base is a perfect power of the second base.**

Why this works:

$A3D_{16} = 10x16^2 + 3x16^1 + 13x16^0$

$= (1x2^3 + 0x2^2 + 1x2^1 + 0x2^0)x16^2 +$

$(0x2^3 + 0x2^2 + 1x2^1 + 1x2^0)x16^1 +$

$(1x2^3 + 1x2^2 + 0x2^1 + 1x2^0)$

$= (1x2^3 + 0x2^2 + 1x2^1 + 0x2^0)x2^8 +$

$(0x2^3 + 0x2^2 + 1x2^1 + 1x2^0)x2^4 +$

$(1x2^3 + 1x2^2 + 0x2^1 + 1x2^0)$, by rewriting 16 as a power of 2.

$= 1x2^{11} + 0x2^{10} + 1x2^9 + 0x2^8 +$

$0x2^7 + 0x2^6 + 1x2^5 + 1x2^4 +$

$1x2^3 + 1x2^2 + 0x2^1 + 1x2^0$

$= 101000111101_2$.

Practice Problem

- Consider writing a function that takes in a number in decimal, and prints out the equivalent value in binary.
  - We can utilize what we learned about base conversion.
  - The key is as follows:
    - If we are converting 78 from base 10 to base 2, we calculate 78%2 = 0.
    - This is the LAST digit we want to print, since it's the units digit of our answer.
    - Preceding that zero, we must take the decimal number 78/2 = 39, and convert THAT to binary. **But, this is a recursive task!!!**