Number Systems Day 2

So far we've learned:

$\left(\text{whole } \#\right)_r \rightarrow$ base 10 — expand it out

$\left(\text{fractional parts}\right)_r \rightarrow$ base 10 — expand it out using negative exponents

whole # base 10 $\rightarrow$ base r — ① logical AND ② algorithm

EX: $1A0.1_{12} \rightarrow$ base 10

$$1A0.1_{12} = \left(1\times 12^2 + \overset{10}{A}\times 12^1 + 0\times 12^0 + 1\times 12^{-1}\right)_{10}$$
$$= \left(144 + 120 + 0 + \tfrac{1}{12}\right)_{10}$$
$$= \left(264 + .083333\ldots\right)_{10}$$
$$= 264.08\overline{3}_{10}$$

EX: $14_{10} \rightarrow$ base 3

logical method

$$\underline{}\;\underset{3^3}{}\quad \underset{3^2}{\underline{1}}\quad \underset{3^1}{\underline{1}}\quad \underset{3^0}{\underline{2}}$$

$14 - 9 = 5$, $5 - 3 = 2$

$\therefore\ 14_{10} = 112_3$

algorithm method

14/3	4	2
4/3	1	1
1/3	0	1

$\therefore 14_{10} = 112_3$

How do we convert fractional base 10 numbers to an arbitrary base r?

1. logical thinking method
2. algorithm method

EX: $0.25_{10} \rightarrow$ base 2

logical method

$$.25 = \frac{1}{4} = \frac{1}{2^2} = 2^{-2}$$

$$.\ \underset{2^{-1}}{\underline{0}}\ \underset{2^{-2}}{\underline{1}}\ \underset{2^{-3}}{\underline{0}}\ \underset{2^{-4}}{\underline{0}}$$

$\therefore\ .25_{10} = .01_2$

algorithm method

decimal * base you want	whole #	decimal part
(.25) 2	0	.50
(.50) 2	1	.00 Stop

answer

$.25_{10} = .01_2$

EX: Convert $.3_{10}$ to base 2.

logical method

repeats

$$.\ \frac{0}{2^{-1}}\ \ \underbrace{\frac{1}{2^{-2}}\ \ \frac{0}{2^{-3}}\ \ \frac{0}{2^{-4}}\ \ \frac{1}{2^{-5}}}_{\text{repeats}}\ \ \frac{1}{2^{-6}}\ \cdots$$

$.5 \quad .25 \quad .125 \quad .0625 \quad .03125 \quad .015625$

$\cancel{.3 - .5}$ $\quad .30 - .25 = .05 \quad$ $\cancel{.05 - .125}$ $\quad$ $\cancel{.05 - .0625}$ $\quad .05000 - .03125 = .018750$

$.018750 - .015625 = .003125$

$\cdots$

algorithm method

decimal part * 2	whole # part	decimal part
(.3)2	0	.6
(.6)2	1	.2
(.2)2	0	.4
(.4)2	0	.8
(.8)2	1	.6
(.6)2	1	.2
(.2)2	0	.4
(.4)2	0	.8
(.8)2	1	.6

repeats

repeats

$$\therefore\ .3_{10} = .0100110011001\ldots_2$$

$$= .0\overline{1001}_2$$

Just because a number is finite in one base does not mean it is finite in another base.

EX: $.3_{10} \rightarrow$ base 3

algorithm method

(.3)3	0	.9
(.9)3	2	.7
(.7)3	2	.1
(.1)3	0	.3
(.3)3	0	.9
(.9)3	2	.7
(.7)3	2	.1
(.1)3	0	.3

repeats

repeats

$\therefore .3_{10} = .\overline{0220}_3$

EX: $75.5_{10} \rightarrow$ base 5
using algorithm method

(A) $75_{10} \rightarrow$ base 5

whole#/5	whole #	remainder
75/5	15	0
15/5	3	0
3/5	0	3
	stop	

$75_{10} = 300_5$

(B) $.5_{10} \rightarrow$ base 5

decimal part * 5	whole #	decimal part
(.5)5	2	.5
(.5)5	2	.5
(.5)5	2	.5

$.5_{10} = .\overline{2}_5$

$$\therefore\ 75.5_{10} = 300.\overline{2}_5$$

base r → base 10

base 10 → base r

How about base r → base b ?

base r → base 10 → base b

expand it out — algorithm

EX: $35_6 \rightarrow$ base 3

(A) $35_6 \rightarrow$ base 10

$35_6 = (3 \times 6^1 + 5 \times 6^0)_{10}$

expand it out

$= (18 + 5)_{10}$

$= 23_{10}$

(B) $23_{10} \rightarrow$ base 3

algorithm

23/3	7	2
7/3	2	1
2/3	0 stop!	2

$\therefore 35_6 = 23_{10} = 212_3$

How do computers do arithmetic?

they do it in binary!

We will learn how to add, subtract, multiply, and divide in binary.

two methods:
1. convert to decimal
   do arithmetic
   convert back to binary
2. do it binary

Learn the Facts:

Addition Facts in Binary:

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 0$ plus "carry" a 1 to next column

$2_{10} = 10_2$

EX: Add

$$\begin{array}{r} {}^{1}\ 101_2 \\ +\ 110_2 \\ \hline 1011_2 \end{array}$$

$\rightarrow 4 + 1 = 5_{10}$

$\rightarrow 4 + 2 = 6_{10}$

$+$

11_{10}

$\frac{1}{2^3}\ \frac{0}{2^2}\ \frac{1}{2^1}\ \frac{1}{2^0}$

$\therefore 1011_2$

EX: Add

$$\begin{array}{r} 1111_2 \\ +\ 10100_2 \\ \hline 100011_2 \end{array}$$

EX: Add in binary:

$$\begin{array}{r} 111.01 \\ +\ 101.11 \\ \hline 1001.00 \end{array}$$

EX: Add in binary

$1+1+1+1 = ?$

$$\begin{array}{r} 1 \\ 1 \\ 1 \\ +\ 1 \\ \hline 100_2 \end{array}$$

Subtraction in Binary:

$0 - 0 = 0$

$1 - 0 = 1$

$0 - 1 = 1$ BUT have to "borrow" a 1 from next column

$1 - 1 = 0$

EX: Subtract in binary.

$$\begin{array}{r} 1001 \\ -\ 101 \\ \hline 100 \end{array}$$

value 2

EX:

$$\begin{array}{r} 10000 \\ -\ 11 \\ \hline 1101 \end{array}$$

* recall that $10_2 = 2_{10}$

EX: Subtract in binary.

$$\begin{array}{r} 110.01 \\ -\ 100.10 \\ \hline 1.11 \end{array}$$