w03-Lec

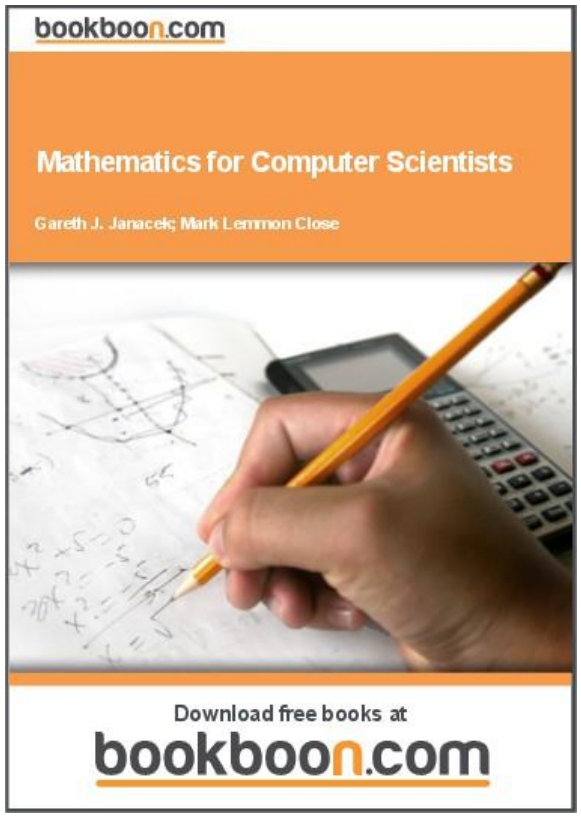
Mathematics and Computer Science: Numbers

for 204111

by Kittipitch Kuptavanich

Numbers

- **Integers**
- Factors and Primes
- Modular Arithmetic
- The Euclidean Algorithm
- Rational and Reals
- Ceiling and Floor functions
- Number Systems: decimal, binary, octal, hexadecimal



Integers

$1, 2, 3, 4, \ldots, 101, 102, \ldots, n, \ldots, 2^{32582657}-1, \ldots$

- **Integer หรือจำนวนเต็ม**
- **เริ่มจากจำนวนนับ (Natural/Counting Number)**
- **Then we add** 0 (zero), **defined as**

$$0 + \text{any integer } n = 0 + n = n + 0 = n$$

- **Negative integer:** $-n$, **defined as**
  - $-n$ **is the number which when added to** n **gives zero**

$$n + (-n) = (-n) + n = 0$$

Simple Rules for Integers

- **For integers a and b**

1. $a + b = b + a$
2. $a \times b = b \times a$ or $ab = ba$

} **commutative**

3. $-a \times b = -ab$
4. $(-a) \times (-b) = ab$
5. a^k = **shorthand for a multiplied by itself k times.**

   $3^4 = 3 \times 3 \times 3 \times 3$

   Note: $a^n \times a^m = a^{n+m}$
6. $n^0 = 1$

Numbers

- Integers
- **Factors and Primes**
- Modular Arithmetic
- The Euclidean Algorithm
- Rational and Reals
- Ceiling and Floor functions
- Number Systems: decimal, binary, octal, hexadecimal

Factors and Primes

- **Many integers are products (ผลคูณ) of smaller integers, for example**

$$2 \times 3 \times 7 = 42$$

- **Here 2, 3 and 7 are called the factors (ตัวประกอบ) of 42**
- **factorization = การแยกตัวประกอบ**

Factors and Primes [2]

- **Not all integers have factors such as**

$$3, 5, 7, 11, 13, \ldots, 2^{216091}-1, \ldots$$

- **These number are called primes (จำนวนเฉพาะ)**
- **พิจารณาการหาร (division)**
  - **ในกรณีหารไม่ลงตัว จะเหลือเศษของการหาร (remainder)**

$$9 = 2 \times 4 + 1$$

Factors and Primes [3]

- เมื่อนำ 9 มาหารด้วย 4 จะเหลือเศษ 1

$$9 = 2 \times 4 + 1$$

- For any integers x and y

$$y = k \times x + r$$

- where r is the remainder (เศษของการหาร)
  - กรณี r เป็น 0 (ศูนย์) เรากล่าวได้ว่า x หาร y ลงตัว (x เป็นตัวหาร)
    - หรือ y หารด้วย x ลงตัว
    - x *divides* y หรือ $x \mid y$ โดยเส้นตั้งใช้แสดงการหารลงตัว
    - เช่น $2 \mid 128$ หรือ $7 \mid 49$ (*ตัวหารอยู่ด้านหน้า)
    - กรณี 3 หาร 4 ไม่ลงตัว แทนด้วยสัญลักษณ์ $3 \nmid 4$

$\nmid$ indivisible symbol

Factorization

- ในการหาตัวประกอบของ integer n เราสามารถใช้วิธีการลองหาร n ด้วยจำนวนเฉพาะ

$$k = 2, 3, 5, 7, 11, 13, \ldots.$$

- ถ้า n หารด้วย k ลงตัว $\rightarrow$ k เป็น factor ของ n
  - ทำการหารอีกครั้งด้วย k
- ถ้า n หารด้วย k ไม่ลงตัว
  - ลองจำนวนเฉพาะตัวถัดไป

Factorization [2]

- ตัวอย่าง 2394

List of primes: 2, 3, 5, 7, 11, 13, ...
more at: http://primes.utm.edu/lists/small/1000.txt

Pseudocode

1. 2394/2 = 1197
2. **Can't divide** by 2 **again so try** 3
3. 1197/3 = 399
4. 399/3 = 133
5. **Can't divide** by 3 **again so try** 5
6. **Can't divide** by 5 so try 7
7. 133/7 = 19 (19 **is prime so we are done**)

$$2394 = 2 \times 3 \times 3 \times 7 \times 19$$

Factorization [3]

Notes:

- จำนวนเฉพาะมีมากมายไม่จำกัด
- เป็นไปไม่ได้ ที่จะมี **list** ของจำนวนเฉพาะทั้งหมด
- หรือไม่สามารถหา **list** ของจำนวนเฉพาะได้
- **Solution:**
  - ใช้เลขคี่ตัวถัดไป จากตัวหารปัจจุบัน
  - ทำไมถึงไม่ใช้เลขคู่?
- ทฤษฎี : ตัวประกอบเฉพาะ*ตัวแรก*ของจำนวนเต็มใด ๆ จะต้องมีค่าน้อยกว่าหรือเท่ากับรากที่สองของจำนวนเต็มนั้น ๆ
  - **Why?**
  - ดังนั้นหากตัวหาร k มากกว่า $\sqrt{n}$ แล้วยังไม่สามารถหา k ที่ $k \mid n$ แสดงว่า n เป็นจำนวนเฉพาะ (ควรหยุดหาตัวประกอบต่อ)

Numbers

- Integers
- Factors and Primes
- **Modular Arithmetic**
- The Euclidean Algorithm
- Rational and Reals
- Ceiling and Floor functions
- Number Systems: decimal, binary, octal, hexadecimal

Modular Arithmetic [1]

- **Operator ที่ใช้ในการหาร กรณีสนใจเศษของการหาร คือ modulo หรือ mod**
  - **Operator: % (C/C++, Java, python)**
- **The operator simply gives the remainder after division. For example,**

1. $25 \bmod 4 = 1$ **because** $25 \div 4 = 6$ **remainder** 1.
2. $19 \bmod 5 = 4$ **because** $19 = 3 \times 5 + 4$.
3. $24 \bmod 5 = 4$.
4. $99 \bmod 11 = 0$.

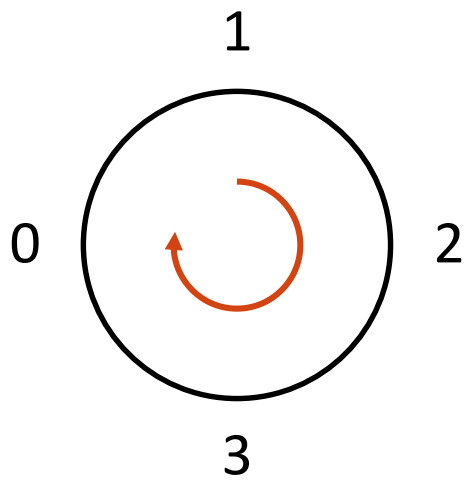
Modular Arithmetic [2]

- **We will ignore cases with negative number for now.**
- **These results can be written in a different ways**

$$25 = 1 \bmod 4 \text{ OR } 25 \bmod 4 = 1$$

We will use this notation in this class

- **Modular arithmetic is sometimes called clock arithmetic.**
- $47 \bmod 4$
  - **Going around 11 times**
  - **And** $\frac{3}{4}$
  - **Stops at** 3



Numbers

- Integers
- Factors and Primes
- Modular Arithmetic
- **The Euclidean Algorithm**
- Rational and Reals
- Ceiling and Floor functions
- Number Systems: decimal, binary, octal, hexadecimal

The Euclidean Algorithm

Euclid

- ในคณิตศาสตร์ ตัวหารร่วมมาก หรือ ห.ร.ม. (อังกฤษ: greatest common divisor: gcd) ของจำนวนเต็มสองจำนวนซึ่งไม่เป็นศูนย์พร้อมกัน คือจำนวนเต็มที่มากที่สุดที่หารทั้งสองจำนวนลงตัว เช่น
  - gcd ของ 15 และ 25 คือ 5
- The Euclidean algorithm for finding the gcd is one of the oldest algorithms known, it appeared in Euclid's Elements around 300 BC.

Image: http://www.glogster.com/codster72011/fun-facts-about-euclid/
Ref: http://th.wikipedia.org/wiki/ตัวหารร่วมมาก

The Euclidean Algorithm [2]

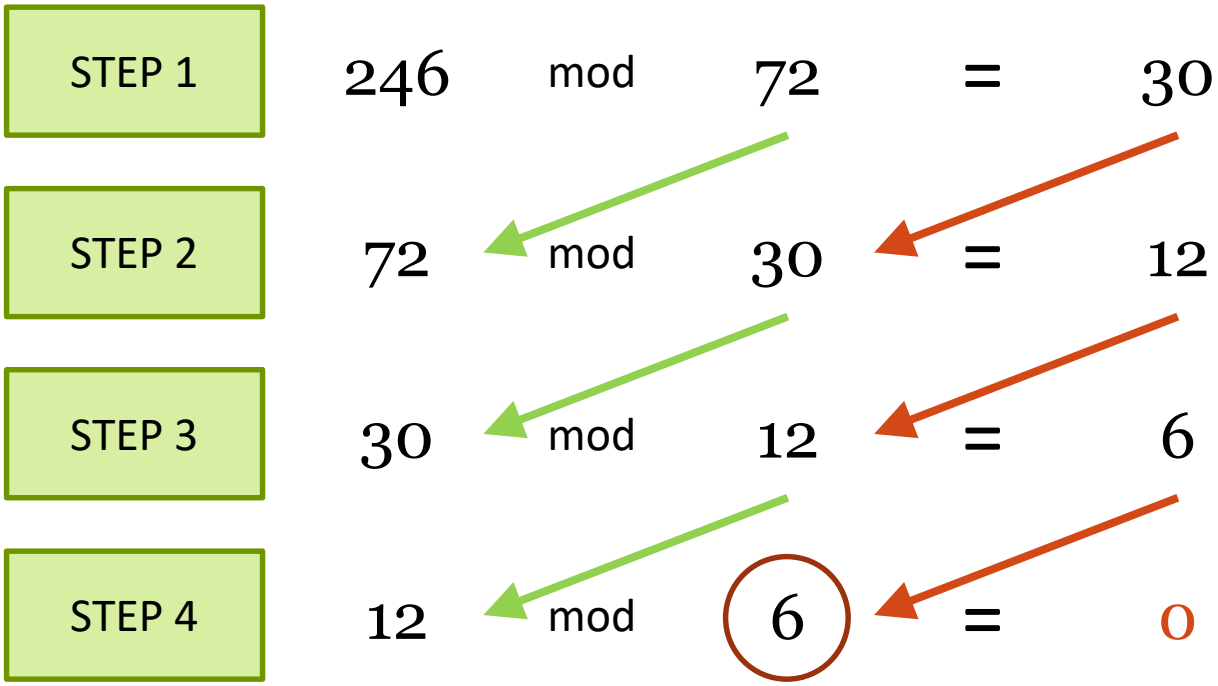
- ให้ a เป็นจำนวนเต็ม ขนาดเล็กกว่า b.

1. ให้ b เป็นตัวตั้ง a เป็นตัวหาร
2. ถ้าเศษจากการหาร r คือ 0, b คือผลคูณของ a (and we are done. - a is the gcd)
3. ถ้าเศษไม่ใช่ 0, นำ a มาหารด้วยเศษ r ดังกล่าว
4. ทำซ้ำข้อ 1 – 3 จน เศษ r จากการหารเป็น 0
5. เศษ r ตัวสุดท้ายที่ไม่ใช่ 0 คือ gcd

$$\gcd(a, b) \times \mathrm{lcm}(a, b) = a \times b$$

The Euclidean Algorithm [3]

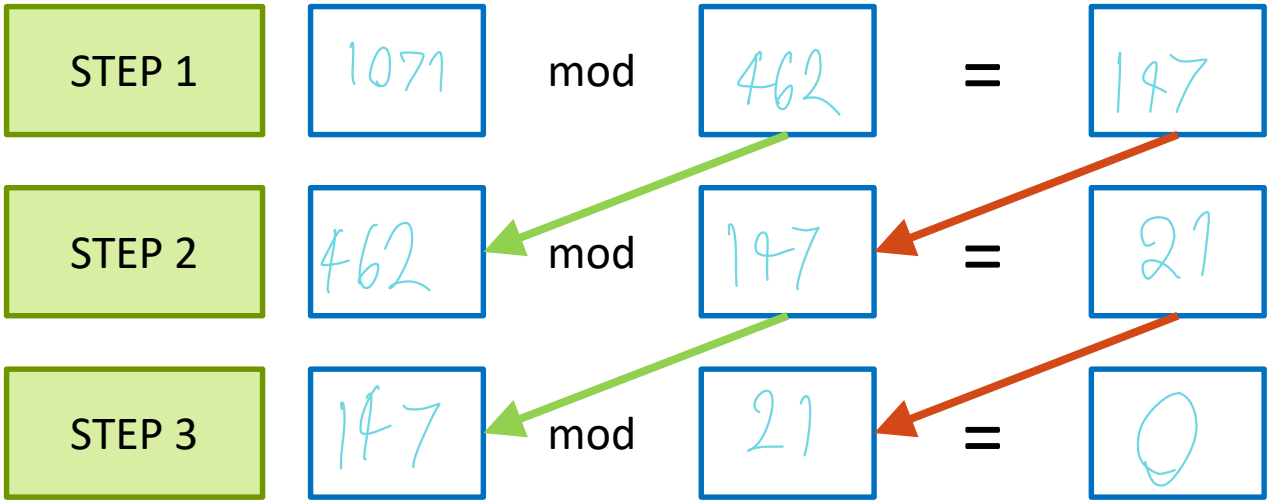
- **For example** 246 **and** 72



- **So the** gcd **of** 246 **and** 72 **is** 6

The Euclidean Algorithm [4]

- **Now let's try** 1071 **and** 462



- **So the gcd is** 21

Numbers

- Integers
- Factors and Primes
- Modular Arithmetic
- The Euclidean Algorithm
- **Rational and Reals**
- Ceiling and Floor functions
- Number Systems: decimal, binary, octal, hexadecimal

Rationals and Reals

- จำนวนตรรกยะ (rational number) คือจำนวนที่สามารถเขียนในรูป $\frac{P}{Q}$ where P and Q are integers. Examples are:

$$\frac{1}{2} \quad \frac{3}{4} \quad \frac{7}{11} \quad \frac{7}{6}$$

- สำหรับจำนวนเต็ม n, except zero, จะมี inverse (อินเวอร์ส), ในรูป $\frac{1}{n}$ โดยมีคุณสมบัติคือ

$$n \times \frac{1}{n} = \frac{1}{n} \times n = 1$$

- เมื่อคูณ $\frac{1}{n}$ ด้วย m จะได้เศษส่วน $\frac{m}{n}$. These are called rational numbers

Rationals and Reals [2]

- **นอกจากนี้ ยังมีตัวเลขที่ไม่ใช่ทั้งจำนวนเต็ม และไม่ใช่จำนวนตรรกยะ เช่น $\sqrt{2}$ ที่ไม่สามารถเขียนให้อยู่ในรูปเศษส่วนได้ เรียกว่า จำนวนอตรรกยะ (irrational numbers)**
- **จำนวนจริง (real numbers)**
  - **Irrational:** $\pi, \sqrt{2}, \sqrt{3}, \ldots$
  - **Rational:** $-3.4, \frac{3}{4}, 9.454545\ldots, \ldots$
  - **Integer:** $-2, 5, -9, 0, \ldots$
  - **Whole:** $0, 1, 2, 3, \ldots$
  - **Natural:** $1, 2, 3, \ldots$

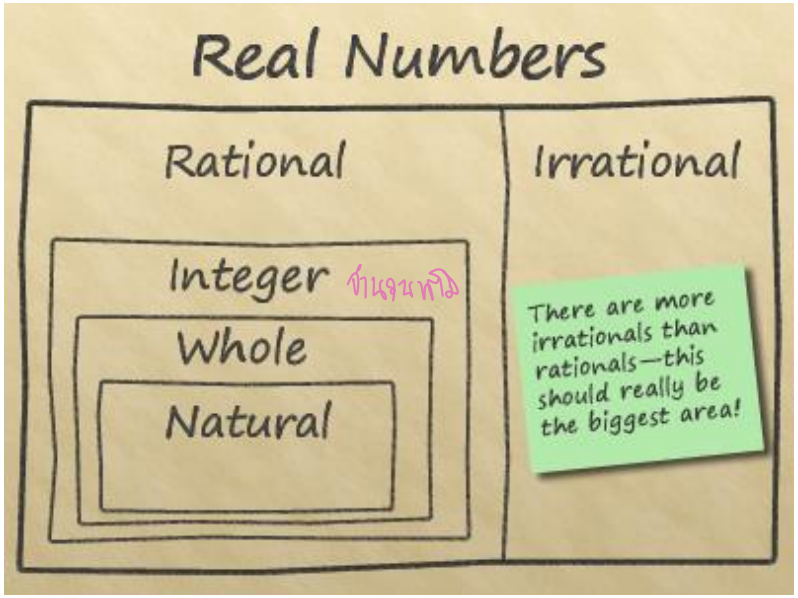


Image: http://leferemath.weebly.com/rational-numbers.html

Notations

เครื่องหมายอื่น ๆ

- **If x is less than (น้อยกว่า) y**
  - **then we write $x < y$. If there is a possibility that they might be equal then $x \leq y$ (น้อยกว่าหรือเท่ากับ)**
  - **We can also write $y > x$ or $y \geq x$**
    - y **is greater than (มากกว่า)** x **or greater than or equal to (มากกว่าหรือเท่ากับ)** x

Numbers

- Integers
- Factors and Primes
- Modular Arithmetic
- The Euclidean Algorithm
- Rational and Reals
- **Ceiling and Floor functions**
- Number Systems: decimal, binary, octal, hexadecimal

Notations [2]

- **Floor function (ฟังก์ชันพื้น) of a real number x, denoted by $\lfloor x \rfloor$ or `floor(x)`, เป็นฟังก์ชันที่ให้ผลลัพธ์เป็นจำนวนเต็มที่มากที่สุดที่น้อยกว่า หรือเท่ากับ x เช่น**
  - **`floor(2.7)` หรือ $\lfloor 2.7 \rfloor$ มีค่าเท่ากับ 2**
    - $\lfloor 5 \rfloor$ มีค่าเท่ากับ 5
  - **แต่ `floor(-3.6)` หรือ $\lfloor -3.6 \rfloor$ มีค่าเท่ากับ -4**
  - **ปัดลงไปทางด้านซ้ายของเส้นจำนวนหากไม่ใช่ integer**
- **Ceiling function (ฟังก์ชันเพดาน) $\lceil x \rceil$ ทำหน้าที่ตรงข้ามกับ floor**
  - **`ceiling(2.7)` หรือ $\lceil 2.7 \rceil$ มีค่าเท่ากับ 3**
  - **ปัดขึ้นไปทางด้านขวาของเส้นจำนวนหากไม่ใช่ integer**

Notations [3]

- **The absolute value (ค่าสัมบูรณ์ หรือ *modulus*) of x written $|x|$ is just x when $x \geq 0$ and $-x$ when $x < 0$ so $|2| = 2$ and $|-6| = 6$**
- **The famous result about the absolute value is that for any x and y**

$$|x + y| \leq |x| + |y|$$

Notations [4]

- **We met a^b when we discussed integers and in the same way we can have x^y when x and y are not integers e.g. $2.5^{3.67}$ or $0.25^{1/2}$**
- **Note however that**

$a^0 = 1$ for all a except zero
$0^b = 0$ for all values of b where $b > 0$
0^0 is undefined mathematically (in python/C you might get 1)

Numbers

- Integers
- Factors and Primes
- Modular Arithmetic
- The Euclidean Algorithm
- Rational and Reals
- Ceiling and Floor functions
- **Number Systems: decimal, binary, octal, hexadecimal**

Number Systems

- ระบบจำนวนที่เราคุ้นเคยและพบมากที่สุดในชีวิตประจำวันคือเลขฐาน 10 (Decimal System)
- 3459 is shorthand (รูปย่อ) for

$$3 \times 1000 + 4 \times 100 + 5 \times 10 + 9$$

OR

$$3 \times 10^3 + 4 \times 10^2 + 5 \times 10^1 + 9 \times 10^0$$

- ตำแหน่ง (position) ของตัวเลขมีความสำคัญ

Number Systems [2]

- เราทราบว่า $10^3 = 1000$ กรณีเลขยกกำลังเป็นจำนวนลบ (negative) เช่น 10^{-3} หมายถึงเศษส่วนในรูป $\frac{1}{10^3}$
- ในเลขฐานสิบ เราใช้จุดทศนิยม (Decimal Point) และตำแหน่งตัวเลขหลังจุดทศนิยมเพื่อแสดงเศษส่วนในกรณีที่มีส่วนเป็น 10^n
- เราสามารถเขียน **123.456** ในรูป

$$1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + . + 4 \times 10^{-1} + 5 \times 10^{-2} + 6 \times 10^{-3}$$

Decimal Point

Number Systems [3]

- **Today the common number systems are**
  - **Decimal number system: ใช้สัญลักษณ์ 0 – 9; ฐาน (base) 10**
  - **Binary number system: ใช้สัญลักษณ์ 0,1; ฐาน 2**
  - **Hexadecimal number system: ใช้สัญลักษณ์ 0-9 และ A-F; ฐาน 16**
    - **here $A \equiv 10$, $B \equiv 11$, $C \equiv 12$, $D \equiv 13$ $E \equiv 14$, $F \equiv 15$.**
  - **Octal number system: ใช้สัญลักษณ์ 0-7; ฐาน 8**

Binary

- **เช่นเดียวกันกับในกรณีเลขฐาน 10 ที่ตำแหน่งแต่ละตำแหน่งแทน 10^n ในระบบเลขฐานสอง แต่ละตำแหน่งแทนด้วย 2^n**

Decimal number		in powers of 2	power of 2				Binary number
			3	2	1	0	
8	=	2^3	1	0	0	0	1000
7	=	$2^2 + 2^1 + 2^0$	0	1	1	1	111
6	=	$2^2 + 2^1$	0	1	1	0	110
5	=	$2^2 + 2^0$	0	1	0	1	101
4	=	2^2	0	1	0	0	100
3	=	$2^1 + 2^0$	0	0	1	1	11
2	=	2^1	0	0	1	0	10
1	=	2^0	0	0	0	1	1

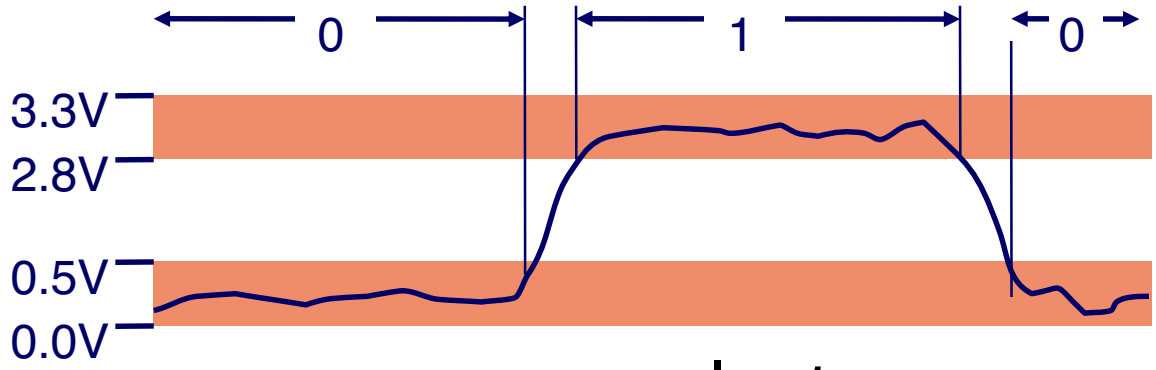
from Carnegie Mellon University's 15213 course slide by Greg Kesden

Why Binary

- รูปแบบของเลขฐาน 2 ประกอบด้วยเลข 0 และ 1
  - Represent 15213_{10} as 11101101101101_{2}
  - Represent 1.20_{10} as $1.0011001100110011[0011]\ldots_{2}$
- เลข 0 และ 1 สามารถตีความได้เป็นสถานะ เปิด/ปิด ของกระแสไฟฟ้า
  - การส่งสัญญาณในคอมพิวเตอร์เป็นการไหลของกระแสไฟฟ้า

Digital Data

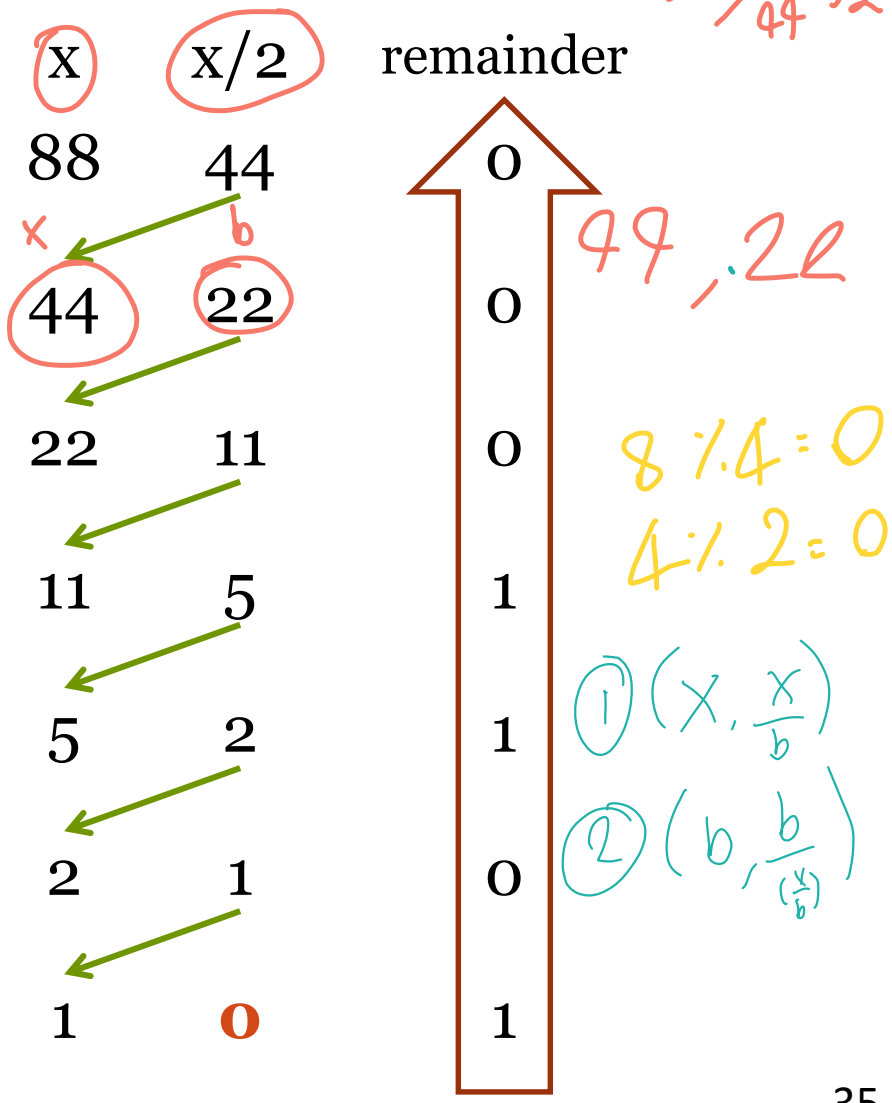
- ข้อมูลต่าง ๆในระบบคอมพิวเตอร์ มีลักษณะดังนี้
  - ระดับไฟฟ้า **binary physical states (high or low voltages, etc.)**
  - สามารถแปลความหมายได้ในรูปแบบบิต **(bit -1s and 0s)**



- เราสามารถแปลความหมายบิต เพื่อเก็บข้อมูลประเภทต่าง ๆ เช่น **integers, real numbers, text, …**

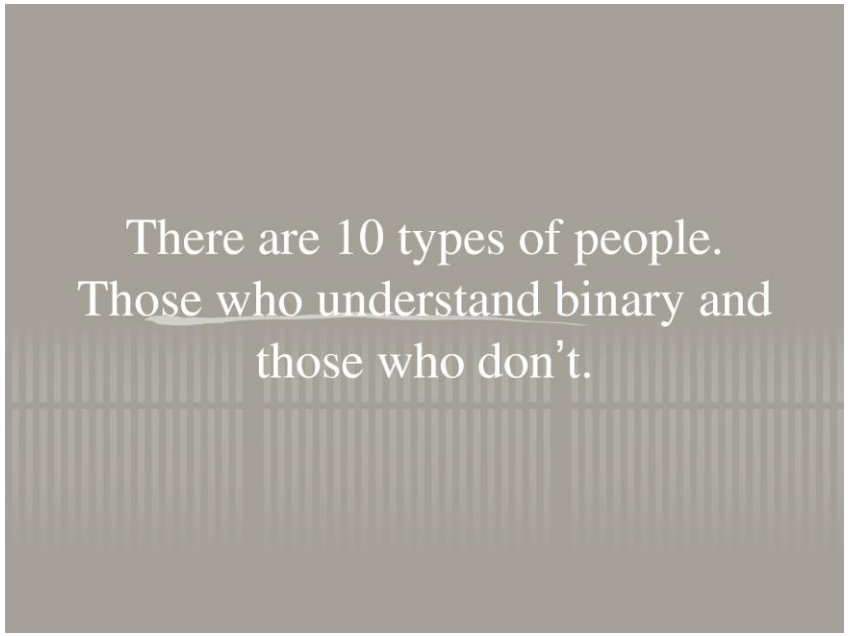
Binary Conversion

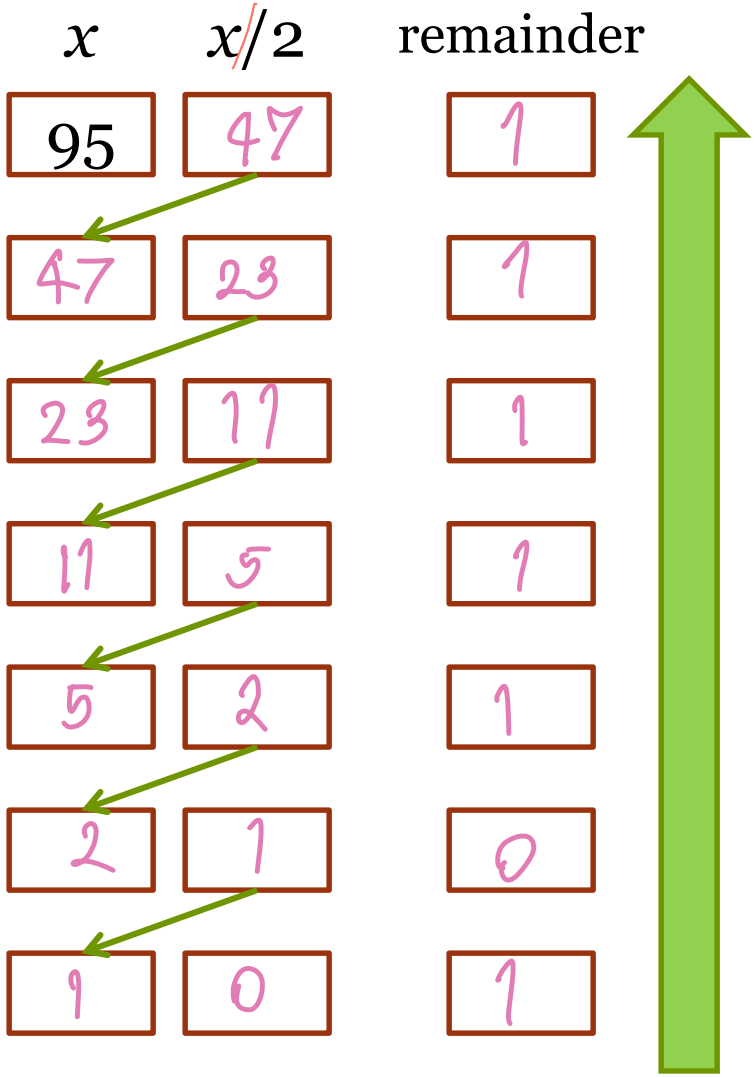
- เราสามารถใช้ **modulo** (การหารเอาเศษ) ในการเปลี่ยนเลขฐาน 10 เป็นเลขฐาน 2 ตัวอย่างเช่น 88
- เมื่อ **x/2 = 0** ให้เขียน **column** สุดท้ายจากล่างขึ้นบน
- จะได้ว่า
  - $88_{10} = 1011000_2$
- วิธีนี้สามารถใช้ แปลงเลขฐาน 10 เป็นฐานอื่น ๆ เช่นกัน



Binary Conversion [2]

- **Let's try with 95**
- **95_{10} =** 1011111 **$_2$**





Binary Conversion Tips

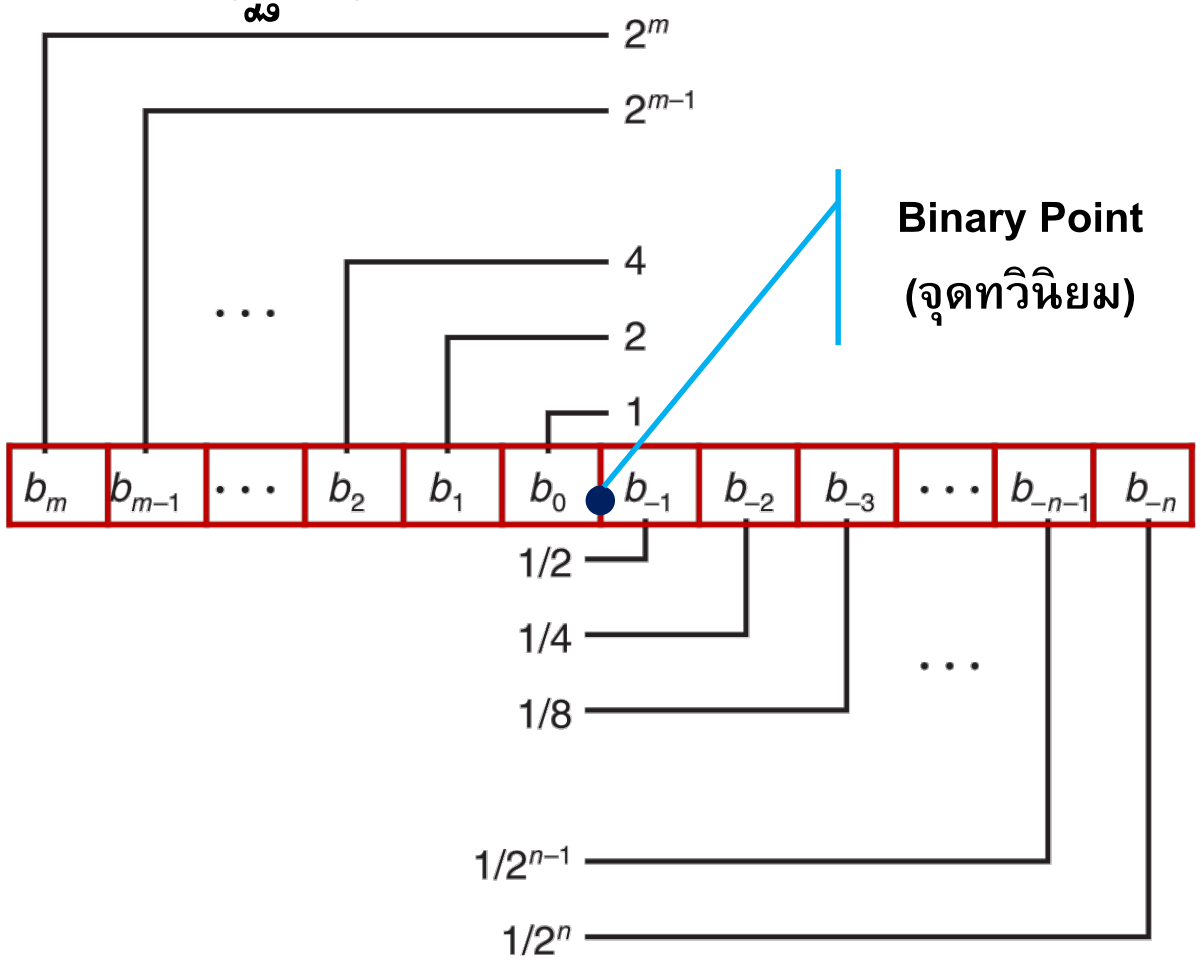
- ในกรณีการแปลงเลขฐาน 10 จำนวนที่ไม่มากนักเป็นฐาน 2 เราอาจใช้วิธีการลบเลขแทน

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

- เช่น 213
  - จากตารางพบว่า 213 มากกว่า 2^7 (1 ตามด้วย 0 ทั้งหมด 7 ตัว)
  - ใส่ 1 _ _ _ _ _ _ _
- เหลือ 213 – 128 = 85
  - พบว่า 85 มากกว่า 2^6
  - ใส่ 1 1 _ _ _ _ _ _
- เหลือ 85 – 64 = 21
  - พบว่า 21 มากกว่า 2^4
  - ใส่ 1 1 _ 1 _ _ _ _
- เหลือ 21 – 16 = 5
  - พบว่า 5 มากกว่า 2^2
  - ใส่ 1 1 _ 1 _ 1 _ _
- เหลือ 5 – 4 = 1
  - ใส่ 1 1 _ 1 _ 1 _ 1
  - ได้ 1101 0101

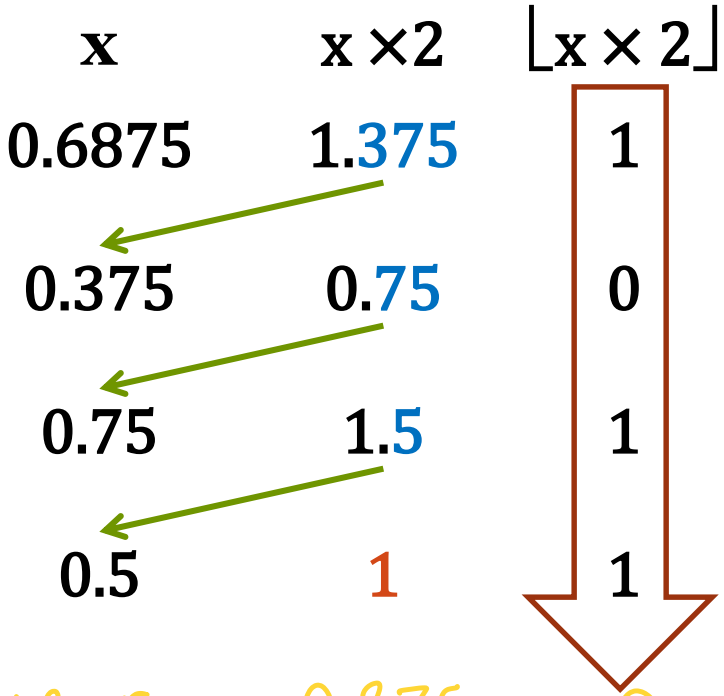
Fractional Binary Numbers

- เลขเศษส่วนฐานสอง



Fractional Binary Numbers [2]

- การเปลี่ยนเลขทศนิยมจากฐาน 10 เป็น ฐาน 2 เราจะใช้ฟังก์ชัน floor
- เมื่อ x × 2 = 1 ให้เขียน column สุดท้ายจากบนลงล่าง
- จะได้ว่า
  - $0.6875_{10} = 0.1011_2$



0.1875 0.375 0
0.375 0.75 0
0.75 1.5 1
0.5 1 1
0.0110$_2$

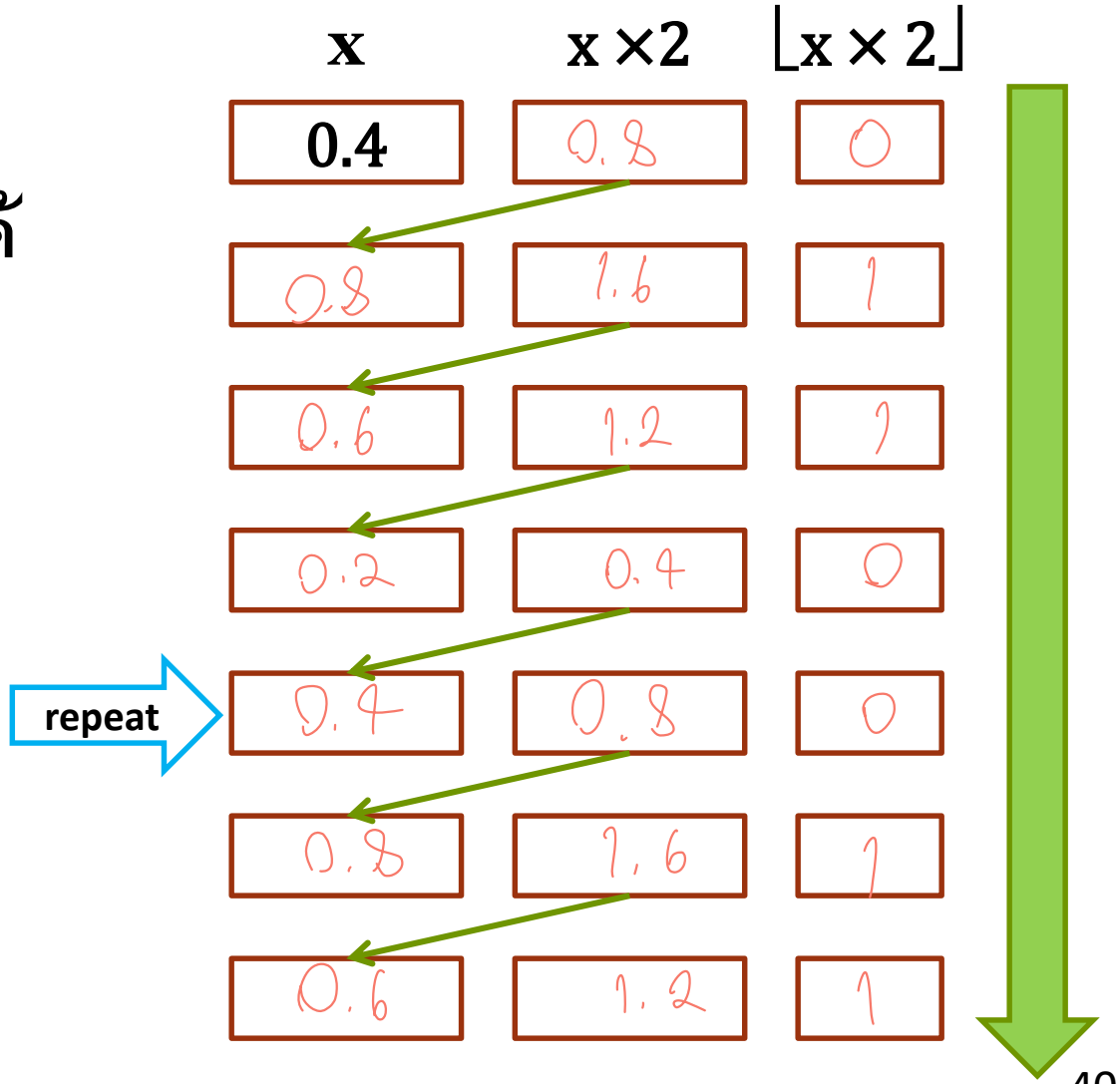
x	$x \cdot 3$	$[x \cdot 3]$
0.1415	0.4245	0
0.4245	1.2735	1
0.2735	0.8205	0
0.8205	2.4615	2
0.4615	1.3845	1
0.3845	1.1535	1
0.1535		

0102117_3

Fractional Binary Numbers [3]

- **Let's try with 0.4**
- **ในบางกรณีเราจะได้ทศนิยมไม่รู้จบ**
- **0.4_{10} =**

0110̇ $_2$



Fractional Binary Numbers [4]

- ในระบบเลขฐาน 2 นั้นมีข้อจำกัด
  - แทน**ค่าจริง**ของตัวเลขได้ เฉพาะตัวเลขที่สามารถเขียนในรูป $x \times 2^y$ เท่านั้น
  - นอกจากนั้นจะเป็น**ค่าประมาณ** – ความใกล้เคียงกับค่าจริง***ขึ้นกับ***จำนวนตำแหน่งที่ใช้แสดงค่า

ตำแหน่งหลังจุดทศนิยมมาก = ค่าใกล้เคียงจำนวนจริงมากขึ้น

Fractional Binary Numbers [5]

Representation	Value	Decimal
0.0_2	$\frac{0}{2}$	0.0_{10}
0.01_2	$\frac{1}{4}$	0.25_{10}
0.010_2	$\frac{2}{8}$	0.25_{10}
0.0011_2	$\frac{3}{16}$	0.1875_{10}
0.00110_2	$\frac{6}{32}$	0.1875_{10}
0.001101_2	$\frac{13}{64}$	0.203125_{10}
0.0011010_2	$\frac{26}{128}$	0.203125_{10}
0.00110011_2	$\frac{51}{256}$	0.19921875_{10}

Conversion Exercise 1

เติมตารางต่อไปนี้ให้สมบูรณ์

Fractional value	Binary representation	Decimal representation
$\frac{1}{8}$	0.001	0.125
$\frac{3}{4}$	_____	_____
$\frac{25}{16}$	_____	_____
_____	10.1011	_____
_____	1.001	_____
_____	_____	5.875
_____	_____	3.1875

Addition in Binary

0+0 = 0

0+1 = 1

1+1 = 10 (เนื่องจาก 1 + 1 มีค่าเกิน 1 ใส่ 0 และ ทดไปหลักถัดไป)

1+1+1 = 1+(1+1) =1+10 =11

- **การบวกเลขในเลขฐาน 2 มีลักษณะคล้ายในฐาน 10**

	1	1	0	1	0	1	
+	1	0	1	1	1	0	
1	1	0	0	0	1	1	Sum

↑ ↑

ตำแหน่งที่เกิดการทดเลข

Subtraction in Binary

	1	1	0	1	0	1	
-	1	0	1	1	1	0	
	0	0	0	1	1	1	difference

- **การลบเลขในเลขฐานสองมีลักษณะคล้ายในฐาน 10 หากตัวตั้งในหลักใด ๆ ไม่พอสำหรับการลบ ก็ให้ "ขอยืม" จากหลักถัดไป**

Multiplication in Binary

การคูณเลขฐาน 10

		1	2	5	6	7	8	ตัวตั้ง
×					3	8	7	ตัวคูณ
		8	7	9	7	4	6	คูณ 7
1	0	0	5	4	2	4		ขยับซ้าย 1 ตำแหน่งแล้วคูณ 8
3	7	7	0	3	4			ขยับซ้าย 2 ตำแหน่งแล้วคูณ 3
4	8	6	3	7	3	8	6	ผลบวกสามบรรทัด

การคูณเลขฐาน 2

		1	0	0	1	1	1	0	Multiplicand
×						1	0	1	Multiplier
		1	0	0	1	1	1	0	times 1
	0	0	0	0	0	0	0		Shift left one and times 0
1	0	0	1	1	1	0			Shift left two and times 1
1	1	0	0	0	0	1	1	0	Add to get the product

Tips

ข้อสังเกต

$$111_2 = 7 \text{ and } 111_2 \times 2 = 14 = 1110_2$$
$$101_2 = 5 \text{ and } 101_2 \text{ x } 2 = 10 = 1010_2$$

- **การคูณเลขใด ๆ ในฐาน 2 ด้วย 2 ให้ขยับเลขนั้นไปทางซ้าย 1 ตำแหน่งแล้วเติม 0**

Octal

- การเปลี่ยนเลขฐาน 2 เป็นฐาน 8 (= 2^3) ให้แบ่งเลขฐานเป็นกลุ่ม กลุ่มละ 3 ตัวเริ่มจากหลัก 2^0
- เช่น 11000010001
- แล้วจึงเปลี่ยนเลขในแต่ละกลุ่มเป็นค่าในฐาน 10 (0-7)

```
0 = 000
1 = 001
2 = 010
3 = 011
4 = 100
5 = 101
6 = 110
7 = 111
```

11000010001

11 000 010 001

3 0 2 1

Hexadecimal

- การเปลี่ยนเลขฐาน 2 เป็นฐาน 16 (= 2^4) ให้แบ่งเลขฐานเป็นกลุ่ม กลุ่มละ 4 ตัวเริ่มจากหลัก 2^0
- เช่น 01011110101101010010
- แล้วจึงเปลี่ยนเลขในแต่ละกลุ่มเป็นค่าในเลขฐาน 16

A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

1011110101101010010

101 1110 1011 0101 0010

5 14 11 5 2

5EB52

Conversion Exercise

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

Fill in the missing entries

Decimal	Binary	Hexadecimal
0	0000 0000	0x00
167		
62		
188		
	0011 0111	
	1000 1000	
	1111 0011	
		0x52
		0xAC
		0xE7

Final Notes

[Math] is a little like programming, it takes time to understand a lot of code and you never understand how to write code by just reading a manual - you have to do it!

Mathematics is exactly the same, you need to do it.

Homework

- **Memorize these table**

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

Hex	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Summary

- Integers
- Factors and Primes
- Modular Arithmetic
- The Euclidean Algorithm
- Rational and Reals
- Ceiling and Floor functions
- **Number Systems: decimal, binary, octal, hexadecimal**