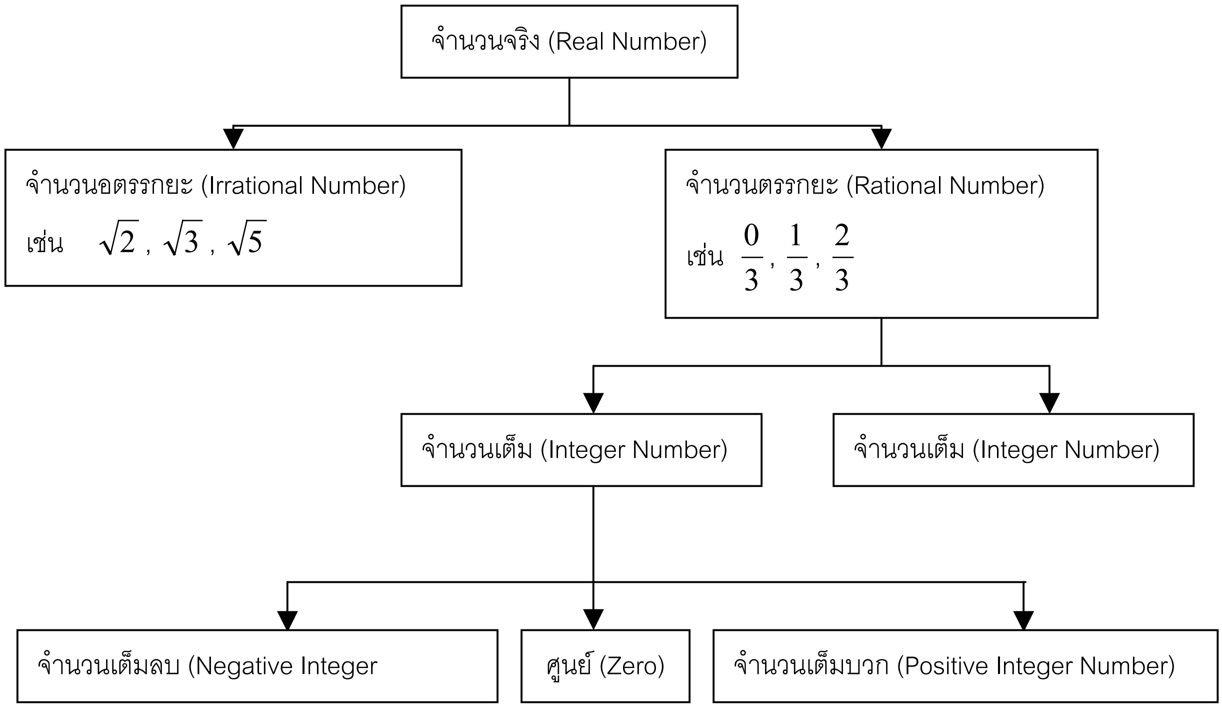
ระบบจำนวน ระบบเลขฐานต่าง ๆ
ระบบเลขฐานสอง ระบบเลขฐานแปด และฐานสิบหก

ระบบจำนวน



ระบบเลขฐาน (SCALE OF NOTATION)

เลขฐานสอง (Binary Number) มี 2 สัญลักษณ์ คือ 0 และ 1
เลขฐานแปด (Octal Number) มี 8 สัญลักษณ์ คือ 0, 1, 2, 3, 4, 5, 6 และ 7
เลขฐานสิบหก (Hexadecimal Number) มี 16 สัญลักษณ์ คือ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E และ F

อธิบายสาระที่ต้องเรียนรู้เรื่องเลขฐานสอง เพราะโยงใยกับการรับรู้ของเครื่องคอมพิวเตอร์ ส่วนฐานแปดและฐานสิบหกเพียงแต่สามารถจัดกลุ่มเลขฐานสองให้คนเข้าใจง่ายขึ้น โดยเลขฐานแปด 1 หลัก เท่ากับเลขฐานสอง 3 หลักเลขฐานสิบหก 1 หลัก เท่ากับเลขฐานสอง 4 หลัก

ในระยะแรกเราใช้เลขฐานสอง 6 หลักในการแทนสัญลักษณ์ 1 สัญลักษณ์ (แทนได้ เท่ากับ $2^6 = 64$ สัญลักษณ์) จึงเขียนแทนได้ด้วยเลขฐานแปด 2 หลัก พอดี

แต่ในปัจจุบันเราใช้เลขฐานสอง 8 หลักในการแทนสัญลักษณ์ 1 สัญลักษณ์เพื่อให้สามารถแทนสัญลักษณ์ได้มากขึ้น (แทนได้ เท่ากับ $2^8 = 256$ สัญลักษณ์) จึงเขียนแทนได้ด้วยเลขฐานสิบหก 2 หลัก พอดี

ดังนั้นเลขฐานแปดจึงไม่มีความจำเป็นสำหรับการใช้ในคอมพิวเตอร์ปัจจุบัน

เน้นความเข้าใจในเรื่องตำแหน่งของเลขฐานต่างและสอนให้สามารถเปลี่ยนฐานได้ทุกฐาน โดยเฉพาะ ระหว่าง ฐานสอง ฐานสิบหก และฐานสิบ (Decimal Number)
ตัวอย่างเช่น

ฐานสิบ
$111.11 = 1x10^2 + 1x10^1 + 1x10^0 + 1x10^{-1} + 1x10^{-2}$
ฐานสอง
$111.11_2 = 1x2^2 + 1x2^1 + 1x2^0 + 1x2^{-1} + 1x2^{-2}$
ฐานสิบหก
$111.11_{16} = 1x16^2 + 1x16^1 + 1x16^0 + 1x16^{-1} + 1x16^{-2}$

เน้นการแปลงเลขฐานสิบเป็นฐานสองโดย
กรณีเป็นเลขจำนวนเต็มให้หารเลขฐานสิบด้วยสองแล้วบันทึกเศษไว้ (เศษจะมีได้ 2 กรณี คือ 0 กับ 1) เศษตัวแรกจะเป็นหลักหน่วย เศษตัวต่อไปก็จะเป็นหลักสิบ หลักร้อย หลักพัน ตามลำดับ

กรณีเป็นเลขหลังจุดทศนิยมให้คูณเลขฐานสิบด้วยสองแล้วบันทึกจำนวนเต็มไว้ (จำนวนเต็มมีได้ 2 กรณี คือ 0 กับ 1) จำนวนเต็มตัวแรกจะเป็นหลักของทศนิยมหลักแรก จำนวนเต็มตัวต่อไปก็จะเป็นทศนิยมหลักต่อไปคือหลักที่สอง หลักที่สาม หลักที่สี่ ตามลำดับ

กรณีเปลี่ยนไปเป็นฐานแปดและสิบหกให้ใช้จับกลุ่มจากเลขฐานสอง

แบบฝึกหัดที่ 1

1. จงเปลี่ยนเลขฐานต่อไปนี้ โดยเติมลงในช่องที่ว่างไว้ให้ตรงกับเลขฐานที่หัวตาราง

ฐานสอง	ฐานสิบ	ฐานสิบหก
1011	…………………………	…………………………
…………………………	1011	…………………………
…………………………	…………………………	1011
…………………………	…………………………	0.001
…………………………	0.001	…………………………
0.001	…………………………	…………………………
11.11	…………………………	…………………………
…………………………	11.11	…………………………
…………………………	…………………………	11.11

2. คนใช้เลขฐานสิบ คอมพิวเตอร์ใช้เลขฐานสอง ทำไมเราต้องเรียนรู้เลขฐานสิบหก ยกตัวอย่างให้เห็นจริง

………………………………………………………………………………………………..

……………………………………….

………………………………………………………………………………………………..

……………………………………….

………………………………………………………………………………………………..

……………………………………….

………………………………………………………………………………………………..

ทบทวนความเป็นมาและความสำคัญของเลขฐานสองกับคอมพิวเตอร์

1859 George Boole

1886 Herman Hollerith

1939 Howard Aiken MARK

1952 John Von Nuemann ใช้ระบบเลขฐานสอง สร้าง EDVAC (Electronic Discrete Variables Automatic Computer) บิดา digital computer

ทำไมต้องใช้ เลขฐานสอง ฐานแปด และฐานสิบหก ในคอมพิวเตอร์

คอมพิวเตอร์สร้างขึ้นด้วยวงจรไฟฟ้าและอิเล็กทรอนิก ซึ่งรับรู้วงจรเปิดและปิดเท่านั้น ซึ่งสามารถสื่อกับเลขฐานสองได้พอดี โดยวงจรปิดแทน 0 วงจรเปิดแทน 1

การใช้เลขฐานสองกับคอมพิวเตอร์เหมาะกับคอมพิวเตอร์ก็จริงสำหรับคนแล้วลำบากมากที่จะจดจำและเข้าใจ เพราะเราคุ้นเคยกับเลขฐานสิบ เนื่องจากในระยะแรกคอมพิวเตอร์ใช้หลักในการแทนสัญลักษณ์จำนวน 6 หลัก เพราะสามารถแทนสัญลักษณ์ที่แตกต่างกันได้ถึง $2^6 = 64$ ตัว ซึ่งเพียงพอในขณะนั้นเพราะคิดถึงแต่อักษรในภาษาอังกฤษเท่านั้น และตรงนี้เองถ้าแยกเลขฐานสองออกเป็นกลุ่มละ 3 ตัว ก็จะแทนได้ด้วยเลขฐานแปดพอดี ทำให้เขียนได้สั้นลงและผิดพลาดน้อยลง แต่เมื่อวงการคอมพิวเตอร์ขยายตัวขึ้นการใช้หลักในเลขฐานสองจำนวน 6 หลักไม่เพียงพอ ในการสื่อความหมายแทนสัญลักษณ์อย่างน้อย 2 ภาษา จึงขยายหลักในเลขฐานสองเป็น 8 หลัก เพื่อให้แทนสัญลักษณ์ที่แตกต่างกันได้ถึง $2^8 = 256$ ตัว ตรงนี้เองถ้าแยกเลขฐานสองออกเป็นกลุ่มละ 4 ตัว ก็จะแทนได้ด้วยเลขฐานสิบหกพอดี ดังนั้นเพื่อเป็นการพบกันครึ่งทางระหว่างคนกับคอมพิวเตอร์ก็โดยผ่านเลขฐานสิบหกที่คนพอจะรู้เรื่องไปสู่เลขฐานสองที่คอมพิวเตอร์รู้เรื่อง

Over Flow และ Under Flow

สภาวะจำนวนมากเกินกว่าจะรับได้ กับจำนวนน้อยเกินกว่าที่รับได้ ขึ้นอยู่กับจำนวนหลักในเลขฐานสอง (บิท) ที่ใช้แทนตัวเลขนั้น ๆ และขึ้นการเอาจำนวนบิทเหล่านั้นกับรูปแบบการแทนค่าจำนวน เช่น อาจจะนำมาแทนค่าจำนวนเต็มบวกหรือจำนวนนับ เอามาแทนจำนวนเต็ม หรือเอามาแทนจำนวนจริง เพราะรูปแบบการแทนค่าไม่เหมือนกันส่งผลให้ค่าที่แทนได้มีขนาดไม่เท่ากันแม้ว่าจะใช้จำนวนบิทเท่ากัน ตัวอย่างเช่น ใช้ 3 บิท แทนจำนวนนับ จะได้ ค่าจากต่ำสุดไปสูงสุดดังนี้คือ 000, 001, 010, 011, 100, 101, 110, 111, ได้แก่ ค่า 0, 1, 2, 3, 4, 5, 6, 7 ในเลขฐานสิบที่เราใช้อยู่นั่นเอง แต่ถ้า 3 บิท เดียวกันนำมาแทนจำนวนเต็ม แบบคอมพลีเมนท์ (one's complement) จะแทนค่า ได้ตั้งแต่ -3 ถึง +3 ดังนี้

ฐานสิบ	ฐานสอง
-3	100
-2	101
-1	110
-0	111
+0	000
+1	001
+2	010
+3	011

และถ้า 3 บิท เดียวกันนำมาแทนจำนวนเต็ม แบบสองคอมพลีเมนท์ (two's complement) จะแทนค่าได้ตั้งแต่ -4 ถึง +3 ดังนี้

ฐานสิบ	ฐานสอง
-4	100
-3	101
-2	110
-1	111
0	000
+1	001
+2	010
+3	011

และหลักการแทนจำนวนเต็มแบบสองคอมพลีเมนท์นี้เองเหมาะกับการแทนค่าในคอมพิวเตอร์มากที่สุดเพราะสัญลักษณ์ที่แตกต่างทั้งหมด แทนด้วยค่าที่แตกต่างกัน และนอกจารกนั้นการนำค่าบวกและลบมาบวกกันก็สื่อความหมายว่าเป็นการลบกันทำให้คอมพิวเตอร์สามารถลบได้จากการเอาตัวคอมพลีเมนท์มาบวกกัน ตัวอย่างเช่น

ตัวอย่างที่ 1

ฐานสิบ	ฐานสอง
-3	101
+3	011
0	000

ถูกต้องทั้งบวกในฐานสิบและฐานสอง

ตัวอย่างที่ 2

ฐานสิบ	ฐานสอง
-3	101
+2	010
-1	111

ถูกต้องทั้งบวกในฐานสิบและฐานสอง

ตัวอย่างที่ 3

ฐานสิบ	ฐานสอง
+3	011
+2	010
+1	101

จะเห็นว่าค่าที่ได้ไม่ตรงกัน แต่คอมพิวเตอร์รู้ว่าผิดพลาดเพราะจำนวนบวกสองจำนวนบวกกันค่าต้องเป็นบวก แต่ค่าที่ได้เป็นลบทำให้รู้ว่ามีการ Over Flow เกิดขึ้น

แบบฝึกหัด

ถ้ากำหนดให้คอมพิวเตอร์เครื่องหนึ่งแทนค่าจำนวนเต็มด้วยหลักในเลขฐานสอง 8 ตัว (8 บิท) โดยให้บิทซ้ายสุดเป็นบิทเครื่องหมายและใช้หลักการของ 2's complement ในการแทนตัวเลข จงหาค่าสูงสุดและต่ำสุดของจำนวนเต็มของเครื่องนี้

1. ถ้ากำหนดให้คอมพิวเตอร์เครื่องหนึ่งแทนค่าจำนวนเต็มด้วยหลักในเลขฐานสอง 16 ตัว (16 บิท) โดยให้บิทซ้ายสุดเป็นบิทเครื่องหมายและใช้หลักการของ 2's complement ในการแทนตัวเลข จงหาค่าสูงสุดและต่ำสุดของจำนวนเต็มของเครื่องนี้
2. (ABCDEF) ฐาน 16 มีค่า เท่ากับ เท่าใดในเลขฐาน 2 เลขฐาน 8 และฐาน 10

การแทนจำนวนบนคอมพิวเตอร์

1. **ตัวเลขจำนวนเต็ม** ใช้หลักการของ 2's complement ในการเก็บข้อมูล โดยกันบิทแรกเป็นบิทเครื่องหมาย เช่น กรณี 16 บิท ค่าสูงสุดที่เก็บได้ คือ

   111 1111 1111 1111 มีค่าเท่ากับ $2^{15} - 1 = 32{,}768 - 1 = 32{,}767$

   ค่าต่ำสุดที่เก็บได้ คือ

   1000 0000 0000 0000 มีค่าเท่ากับ $-2^{5} = -32{,}768$

2. ตัวเลขทศนิยมแบบคงที่ (Fixed-point number)

กรณี 16 บิท แทน 0.15625 ได้ดังนี้ บิทแรกจะเป็นบิทเครื่องหมาย บิทต่อมาเป็น Normal binary point และ Implied binary point เป็นจุดเดียวกับ Normal binary point

$$0.15625 = 0.1255 + 0.03125 = \frac{1}{8} + \frac{1}{32} = \mathbf{0.00101_2}$$ จัดอยู่ในคอมพิวเตอร์ คือ

0001 0100 0000 0000

$$57.8125 = 32 + 16 + 8 + 1 + .5 + .25 + .0625 = \mathbf{111001.1101_2}$$

แทนได้ดังนี้ บิทแรกจะเป็นบิทเครื่องหมาย บิทต่อมาเป็น Normal binary point และ Implied binary point คนละจุดกับ Normal binary point โดยต้องอยู่หลังบิทที่ 7 เพราะมีจำนวนเต็ม 6 บิท และเป็นบิทเครื่องหมายเสีย 1 บิท

0001 0100 0000 0000

3. ตัวเลขทศนิยมแบบตัวลอย (Floating-point number)

การกำหนดการแทนจำนวนเต็มในคอมพิวเตอร์แทนตัวเลขได้น้อย เช่น ในการใช้ 8 บิท แทน จะแทนค่าสูงสุดได้เพียงค่า 127 เท่านั้น ถ้าเพิ่มจำนวนบิทเท่าตัว เป็น 16 บิท ก็แทนได้สูงสุดเพียง 32,767 เท่านั้น แต่จำนวนในความเป็นจริงที่ต้องใช้งานนั้นต้องการมากกว่านี้ จึงคิดวิธีการเก็บข้อมูลแบบจำนวนจริงขึ้น โดยใช้จำนวนบิทเท่าเดิมแต่สามารถเก็บค่าได้มากกว่า

บิทเครื่องหมาย	เก็บค่ายกกำลังจริง (True exponential)	เก็บค่าหลังทศนิยม (Mantissa)

16	15 14 13 12 11 10 9	8 7 6 5 4 3 2 1
S	_ _ _ _ _ _ _	_ _ _ _ _ _ _ _

ตัวอย่าง เช่น 0 001 0111 11001010 หมายถึง $+(0.11001010)_2 \times 2^{(10111)_2}$ มีค่าเท่ากับ

$= (2\text{-}1 + 2\text{-}2 + 2\text{-}5 + 2\text{-}7) \times 2^{(2^4 + 2^2 + 2^1 + 2^0)}$

$= (0.5 + 0.25 + 0.03125 + 0.0078125) \times 2^{23}$

$= 0.7890625 \times 8388608 = 6619136$

หรือ $+(0.11001010)_2 \times 2^{(10111)_2} = +(0.11001010)_2 \times 2^{23}$

$= (11001010000000000000000)_2$

$= 6619136$

จาก 0 001 0111 1100 1010 อธิบายได้ดังนี้

บิทแรก คือ บิทเครื่องหมายของจำนวน 0 หมายถึง + 1 หมายถึง -

7 บิท ถัดมา คือ จำนวนยกกำลัง (exponential) ถ้าแทนในลักษณะนี้เป็นการแทนแบบ 2's complement ในรูปแบบเลขจำนวนเต็ม เรียกว่า true exponential

8 บิท ถัดมา คือ จำนวนหลังจุดทศนิยม ของจำนวนที่แทนในเลขฐานสองโดยหลักที่มีนัยสำคัญสูงกว่าอยู่บิทซ้ายมือ ดังนี้ถ้าจำนวนบิทที่แทนเลขหลังจุดทศนิยมนี้ยิ่งมากความถูกต้องก็ยิ่งมากขึ้นด้วย

ตัวอย่าง จงหาค่าสูงสุดของการเก็บข้อมูลแบบ Floating point จำนวน 16 บิท

0 011 1111 1111 1111 มีค่าเท่ากับ $0.9960937 \times 2^{63} = 9.187342779 \times 10^{18}$

ตัวอย่าง จงหาค่าน้อยสุดของการเก็บข้อมูลแบบ Floating point จำนวน 16 บิท

1 011 1111 1111 1111 มีค่าเท่ากับ $-0.9960937 \times 2^{63} = -9.187342779 \times 10^{18}$

ตัวอย่าง จงหาค่าน้อยที่สุดที่เป็นค่าบวกของการเก็บข้อมูลแบบ Floating point จำนวน 16 บิท

0 100 0000 0000 0001 มีค่าเท่ากับ $0.00390625 \times 2^{-64} = 2.117582368 \times 10^{-22}$

ตัวอย่าง จัดจำนวน 15.2 ให้อยู่ในรูปของ floating point จำนวน 16 บิท และคำนวณร้อยละของความคลาดเคลื่อน

$15 = 1111_2$

$0.2 = (0.00110011)_2$

ฉะนั้น $15.2 = 1111.00110011_2 = 0.111100110011 \times 2^4$

จึงเก็บได้ดังนี้ 0 000 0100 1111 0011

คำนวณกลับจะได้ $0.11110011 \times 2^4 = 1111.0011_2 = 15.1875$

ร้อยละความคลาดเคลื่อน $= \dfrac{15.2 - 15.1875}{15.2} x\, 100 = 0.08223$

แบบฝึกหัด

จัดจำนวนต่อไปนี้ให้อยู่ในรูปของ floating point จำนวน 16 บิท และคำนวณร้อยละของความคลาดเคลื่อน

1. 12.34
2. –13.42
3. 0.000125
4. –0.0625
5. –0.01234
6. 15.1246
7. –18.132
8. –0.008132

ทบทวนการแทนค่าตัวเลขในคอมพิวเตอร์อย่างละเอียด 1'Complement และ 2's Complement สามารถบอกรูปแบบในการเก็บข้อมูลแบบต่าง ๆ เช่น integer, long integer, float, double, long double พร้อมทั้งสามารถคำนวณเพื่อตรวจสอบค่าสูงสุดและต่ำสุดของการเก็บข้อมูลแต่ละประเภทได้

ตารางช่วยคำนวณในการแปลงระหว่างเลขฐานสองและฐานสิบ

2^0 =	1	2^0 =	1
2^1 =	2	2^-1 =	0.5
2^2 =	4	2^-2 =	0.25
2^3 =	8	2^-3 =	0.125
2^4 =	16	2^-4 =	0.0625
2^5 =	32	2^-5 =	0.03125
2^6 =	64	2^-6 =	0.015625
2^7 =	128	2^-7 =	0.007813
2^8 =	256	2^-8 =	0.003906
2^9 =	512	2^-9 =	0.001953
2^10 =	1024	2^-10 =	0.000977
2^11 =	2048	2^-11 =	0.000488
2^12 =	4096	2^-12 =	0.000244
2^13 =	8192	2^-13 =	0.000122
2^14 =	16384	2^-14 =	6.1E-05
2^15 =	32768	2^-15 =	3.05E-05
2^16 =	65536	2^-16 =	1.53E-05
2^17 =	131072	2^-17 =	7.63E-06
2^18 =	262144	2^-18 =	3.81E-06
2^19 =-	524288	2^-19 =	1.91E-06
2^20 =	1048576	2^-20 =	9.54E-07
2^21 =	2097152	2^-21 =	4.77E-07
2^22 =	4194304	2^-22 =	2.38E-07
2^23 =	8388608	2^-23 =	1.19E-07
2^24 =	16777216	2^-24 =	5.96E-08
2^25 =	33554432	2^-25 =	2.98E-08

2^26 =	67108864	2^-26 =	1.49E-08
2^27 =	1.34E+08	2^-27 =	7.45E-09
2^28 =	2.68E+08	2^-28 =	3.73E-09
2^29 =	5.37E+08	2^-29 =	1.86E-09
2^30 =	1.07E+09	2^-30 =	9.31E-10
2^31 =	2.15E+09	2^-31 =	4.66E-10
2^32 =	4.29E+09	2^-32 =	2.33E-10
2^33 =	8.59E+09	2^-33 =	1.16E-10
2^34 =	1.72E+10	2^-34 =	5.82E-11
2^35 =	3.44E+10	2^-35 =	2.91E-11
2^36 =	6.87E+10	2^-36 =	1.46E-11
2^37 =	1.37E+11	2^-37 =	7.28E-12
2^38 =	2.75E+11	2^-38 =	3.64E-12
2^39 =	5.5E+11	2^-39 =	1.82E-12
2^40 =	1.1E+12	2^-40 =	9.09E-13
2^41 =	2.2E+12	2^-41 =	4.55E-13
2^42 =	4.4E+12	2^-42 =	2.27E-13
2^43 =	8.8E+12	2^-43 =	1.14E-13
2^44 =	1.76E+13	2^-44 =	5.68E-14
2^45 =	3.52E+13	2^-45 =	2.84E-14
2^46 =	7.04E+13	2^-46 =	1.42E-14
2^47 =	1.41E+14	2^-47 =	7.11E-15
2^48 =	2.81E+14	2^-48 =	3.55E-15
2^49 =	5.63E+14	2^-49 =	1.78E-15
2^50 =	1.13E+15	2^-50 =	8.88E-16
2^51 =	2.25E+15	2^-51 =	4.44E-16
2^52 =	4.5E+15	2^-52 =	2.22E-16
2^53 =	9.01E+15	2^-53 =	1.11E-16
2^54 =	1.8E+16	2^-54 =	5.55E-17
2^55 =	3.6E+16	2^-55 =	2.78E-17
2^56 =	7.21E+16	2^-56 =	1.39E-17
2^57 =	1.44E+17	2^-57 =	6.94E-18

2^58 =	2.88E+17	2^-58 = 3.47E-18
2^59 =	5.76E+17	2^-59 = 1.73E-18
2^60 =	1.15E+18	2^-60 = 8.67E-19
2^61 =	2.31E+18	2^-61 = 4.34E-19
2^62 =	4.61E+18	2^-62 = 2.17E-19
2^63 =	9.22E+18	2^-63 = 1.08E-19
2^64 =	1.84E+19	2^-64 = 5.42E-20
2^65 =	3.69E+19	2^-65 = 2.71E-20
2^66 =	7.38E+19	2^-66 = 1.36E-20
2^67 =	1.48E+20	2^-67 = 6.78E-21
2^68 =	2.95E+20	2^-68 = 3.39E-21
2^69 =	5.9E+20	2^-69 = 1.69E-21
2^70 =	1.18E+21	2^-70 = 8.47E-22
2^71 =	2.36E+21	2^-71 = 4.24E-22
2^72 =	4.72E+21	2^-72 = 2.12E-22
2^73 =	9.44E+21	2^-73 = 1.06E-22
2^74 =	1.89E+22	2^-74 = 5.29E-23
2^75 =	3.78E+22	2^-75 = 2.65E-23
2^76 =	7.56E+22	2^-76 = 1.32E-23
2^77 =	1.51E+23	2^-77 = 6.62E-24
2^78 =	3.02E+23	2^-78 = 3.31E-24
2^79 =	6.04E+23	2^-79 = 1.65E-24
2^80 =	1.21E+24	2^-80 = 8.27E-25

จำนวนเต็ม 16 บิท ค่าสูงสุดคือ 0111 1111 1111 1111

มีค่าเท่ากับ 2^16 - 1 เท่ากับ 32767

จำนวนจริง 32 บิท ค่าสูงสุดคือ 0 011 1111 11 11 1111 1111 1111 1111 1111

มีค่าเท่ากับ 0.99999994 x 2^7-1.70141E + 38

งานที่มอบหมาย จงพิสูจน์ว่า

1. ค่าเลขจำนวนเต็ม (int) ในคอมพิวเตอร์ หรือภาษาซี มีค่าอยู่ระหว่าง -32768 ถึง 32767
2. ค่าเลขจำนวนเต็มชนิดยาว (long) ในคอมพิวเตอร์ หรือภาษาซี มีค่าอยู่ระหว่าง –2147483648 ถึง 2147483648
3. ค่าเลขจำนวนจริง (float) ในคอมพิวเตอร์ หรือภาษาซี มีค่าอยู่ระหว่าง -3.4×10^{38} ถึง 3.4×10^{38}
4. ค่าตัวเลขจำนวนจริงแบบเท่าตัว (double) ในคอมพิวเตอร์ หรือภาษาซี มีค่าอยู่ระหว่าง $-1.79\text{x}10^{308}$ ถึง $1.79\text{x}10^{308}$
5. ค่าเลขจำนวนจริงแบบเท่าตัวชนิดยาว (long double) ในคอมพิวเตอร์ หรือภาษาซี มีค่าอยู่ระหว่าง $-3.4\text{x}10^{4932}$ ถึง $3.4\text{x}10^{4932}$