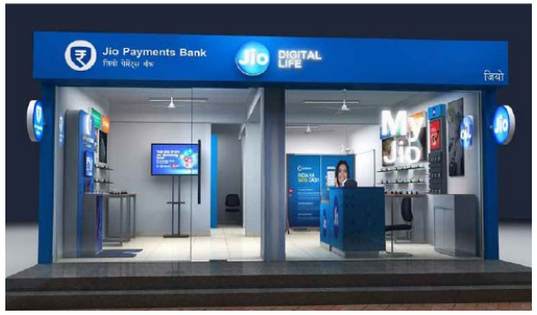
• 10301222 โครงสร้างข้อมูลและอัลกอริทึม

• Chapter 2

- **โครงสร้างข้อมูล**

ผศ.ดร. ปวีณ เขื่อนแก้ว
สาขาวิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยแม่โจ้

Data type and Data structure



Data type

- ใช้สำหรับเก็บข้อมูลลงสู่หน่วยความจำ
- มีใช้ในทุกภาษา
- มี 2 แบบคือ Primitive และ Composite

Data type: Primitive

- ข้อมูลเชิงเดี่ยว
- หน่วยเก็บข้อมูลที่มีขนาดเล็กที่สุด
- เก็บข้อมูลได้เพียงตัวเดียว (atomic)
- แบ่งตามประเภทของข้อมูล
- มีตัวกระทำ ที่สามารถกระทำต่อ ชนิดของข้อมูลนั้นเท่านั้น

ชนิดของข้อมูล	ข้อมูลที่บันทึกได้	การกระทำ
Boolean	0 หรือ 1	And Or Not
Integer	จำนวนเต็ม	การกระทำทางคณิตศาสตร์ เช่น + - x /

Data type: Primitive

Data Type	Range
Integer (32-bit)	-2,147,483,648 to +2,147,483,647
Integer (16-bit)	-32,768 to 32,767
ASCII Characters (8-bit)	0 to 255
Unicode Characters (16-bit)	65,535
Signed Characters (8-bit)	-128 to 127
Boolean	0(false), 1(true)

Data type: Primitive

C	Java	C++	Python	Assembly
char	byte	bool	bool	Byte
short	short	char	int , long	Word
int	int	int	float	DoubleWord
long	long	float	str	
float	float	double	none	
long double	double	void		
	boolean	wchar_t		
	char			

- เป็นการเก็บข้อมูลที่ยึดติดกับ hardware หรือวงจรไฟฟ้า
- ยึดกับ instruction set
- อาจเป็น hardware instruction set หรือ software instruction set
- ความหมายอาจเปลี่ยนไปตาม platform (เช่น int ของภาษา C)

Data type: Primitive

เครื่องคอมพิวเตอร์บันทึกข้อมูลตัวเลขลงสู่หน่วยความในรูปแบบใด ?

a = 123

- จำนวนเต็ม
- เลขทศนิยม
- ค่าบวก / ติดลบ

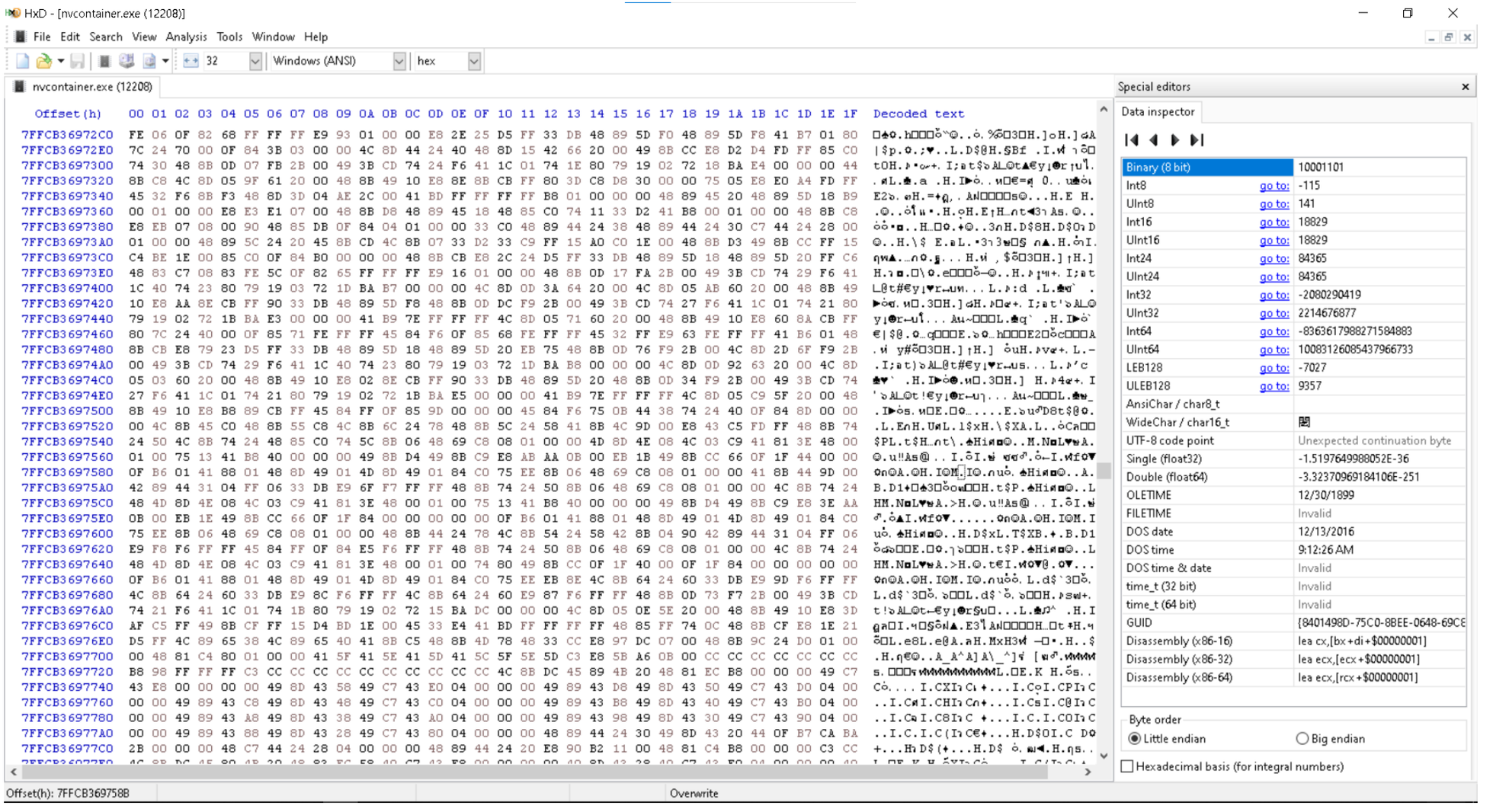
Data type: Primitive

ตัวอย่างการบันทึกและแปลความหมายข้อมูล

จำนวน	ข้อมูลบนหน่วยความจำ ระบบ Big-endian	ข้อมูลบนหน่วยความจำ ระบบ Little-endian (Intel X86)
1	00000000 00000000 00000000 00000001	00000001 00000000 00000000 00000000
-1	11111111 11111111 11111111 11111111	11111111 11111111 11111111 11111111
1.0	00111111 10000000 00000000 00000000	00000000 00000000 10000000 00111111
-1.0	10111111 10000000 00000000 00000000	00000000 00000000 10000000 10111111
1.000000001	00111111 11110000 00000000 00000000 00000000 00000110 11011111 00111000	00111000 11011111 00000110 00000000 00000000 00000000 11110000 00111111
-1.000000001	10111111 11110000 00000000 00000000 00000000 00000110 11011111 00111000	00111000 11011111 00000110 00000000 00000000 00000000 11110000 10111111

Data type: Primitive

ตัวอย่างการบันทึกและแปลความหมายข้อมูล



Data type: Primitive

การบันทึกและแปลความหมายข้อมูล: จำนวนเต็มบวก

จำนวนเต็มบวกที่มีค่าน้อยที่สุดคือ 0

จำนวนเต็มบวกค่าสูงสุดที่แทนได้ขึ้นอยู่กับ จำนวนบิตที่ใช้

ค่าสูงสุด = 2^n -1

8 บิต

7	6	5	4	3	2	1	0

16 บิต

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

32 บิต

Data type: Primitive

การบันทึกและแปลความหมายข้อมูล: จำนวนเต็มบวก

การแทนจำนวนเต็มบวก ใช้วิธีการแปลงเลขฐาน 10 เป็นเลขฐาน 2
แล้วแทนลงหน่วยความจำได้เลย

ตัวอย่างเช่น 1234 แปลเป็นเลขฐาน 2 ขนาด 16 บิตได้เป็น 0000 0100 1101 0010

นักศึกษายังจำวิธีแปลงเลขฐาน 10 เป็นฐาน 2 ได้อยู่หรือไม่ ?

การเปลี่ยนเลขจากระบบเลขฐานสิบให้เป็นฐานสอง

- กรณีของเลขจำนวนเต็ม
  - ถ้า N เป็นเลขจำนวนเต็มฐาน 10 ใดๆ ให้นำ 2 ไปหารสั้น N ได้ผลหารเป็น N_0 และได้เศษเป็น d_0 (ค่าที่เป็นไปได้คือ 0 หรือ 1)
  - นำ 2 มาหาร N_0 ต่อไปอีก ได้ผลหารเป็น N_1 และเศษเป็น d_1
  - นำ N_1 มาหารต่อไปอีกได้ N_2 และเศษเป็น d_2
  - ดำเนินการเช่นนี้เรื่อยไป ผลที่ได้คือ $N > N_0 > N_1 > N_2 > \ldots\ldots$

    กระทั่ง 2 มีค่ามากกว่า N_p ได้ p ครั้ง
  - นั่นคือเลขฐานสองที่มีค่าเท่ากับ N ในเลขฐานสิบ คือ $d_p \ldots d_2 d_1 d_0$

การแปลงเลขจากฐาน 10 เป็นฐาน 2

ฐาน 10		ฐาน		ได้		เศษ
1 2 3	÷	2	=	61		1
61	÷	2	=	30		1
30	÷	2	=	15		0
15	÷	2	=	7		1
7	÷	2	=	3		1
3	÷	2	=	1	→	1

$$123_{10} = 1111011_2$$

การแปลงเลขจากฐาน 2 เป็นฐาน 10

$$N = \sum d \times base^p$$

base=2

$$N = 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$N = 1 \times 64 + 1 \times 32 + 1 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1$$

$$N = 64 + 32 + 16 + 8 + 0 + 2 + 1$$

$$N = 123$$

$$1111011_2 = 123_{10}$$

ลองฝึกแปลงเลขฐานสิบเป็นเลขฐานสอง

ฐาน 10	ฐาน 2
10	1010
100	0110 0100
1023	0011 1111 1111
5555	0001 0101 1011 0011
13333	0011 0100 0001 0101
654321	1001 1111 1011 1111 0001

Data type: Primitive

การบันทึกและแปลความหมายข้อมูล: จำนวนเต็มลบ

การแทนจำนวนเต็มลบ ใช้วิธีการแปลงเลขฐาน 10 เป็นเลขฐาน 2
แล้วแทนลงหน่วยความจำในรูปแบบเลขฐานสองแบบ 2’s complement

ตัวอย่างเช่น 1234 แปลเป็นเลขฐาน 2 ขนาด 16 บิตได้เป็น 0000 0100 1101 0010

-1234 แปลเป็นเลขฐาน 2 ขนาด 16 บิตได้เป็น 1111 1011 0010 1110

นักศึกษายังจำวิธีแปลงเลขฐาน 2 เป็นแบบ 2’s complement
ได้อยู่หรือไม่ ?

การแปลงเลขเป็น 2's complement

1) กลับบิตให้เป็นตรงกันข้าม
2) นำค่าที่ได้ + 1

การกลับบิตคือการทำให้บิตให้เป็นค่าตรงข้าม เช่น 1 –> 0 หรือ 0 –> 1

เช่น 1001 กลับบิตจะได้เป็น 0110
0111 กลับบิตจะได้เป็น 1000

นักศึกษายังจำวิธีแปลงบวกเลขฐาน 2 ได้อยู่หรือไม่ ?

การบวกเลขฐาน 10 ยากกว่าฐาน 2 ถึง 25 เท่า

กฎการบวกเลขฐาน 10

carry in	0	0 + 0	=	0 carry out 0	carry in	0	3 + 0	=	3 carry out 0	carry in	0	6 + 0	=	6 carry out 0	carry in 0 9 + 0 = 9 carry out 0
carry in	0	0 + 1	=	1 carry out 0	carry in	0	3 + 1	=	4 carry out 0	carry in	0	6 + 1	=	7 carry out 0	carry in 0 9 + 1 = 0 carry out 1
carry in	0	0 + 2	=	2 carry out 0	carry in	0	3 + 2	=	5 carry out 0	carry in	0	6 + 2	=	8 carry out 0	carry in 0 9 + 2 = 1 carry out 1
carry in	0	0 + 3	=	3 carry out 0	carry in	0	3 + 3	=	6 carry out 0	carry in	0	6 + 3	=	9 carry out 0	carry in 0 9 + 3 = 2 carry out 1
carry in	0	0 + 4	=	4 carry out 0	carry in	0	3 + 4	=	7 carry out 0	carry in	0	6 + 4	=	0 carry out 1	carry in 0 9 + 4 = 3 carry out 1
carry in	0	0 + 5	=	5 carry out 0	carry in	0	3 + 5	=	8 carry out 0	carry in	0	6 + 5	=	1 carry out 1	carry in 0 9 + 5 = 4 carry out 1
carry in	0	0 + 6	=	6 carry out 0	carry in	0	3 + 6	=	9 carry out 0	carry in	0	6 + 6	=	2 carry out 1	carry in 0 9 + 6 = 5 carry out 1
carry in	0	0 + 7	=	7 carry out 0	carry in	0	3 + 7	=	0 carry out 1	carry in	0	6 + 7	=	3 carry out 1	carry in 0 9 + 7 = 6 carry out 1
carry in	0	0 + 8	=	8 carry out 0	carry in	0	3 + 8	=	1 carry out 1	carry in	0	6 + 8	=	4 carry out 1	carry in 0 9 + 8 = 7 carry out 1
carry in	0	0 + 9	=	9 carry out 0	carry in	0	3 + 9	=	2 carry out 1	carry in	0	6 + 9	=	5 carry out 1	carry in 0 9 + 9 = 8 carry out 1
carry in	0	1 + 0	=	1 carry out 0	carry in	0	4 + 0	=	4 carry out 0	carry in	0	7 + 0	=	7 carry out 0	carry in 1 0 + 0 = 1 carry out 0
carry in	0	1 + 1	=	2 carry out 0	carry in	0	4 + 1	=	5 carry out 0	carry in	0	7 + 1	=	8 carry out 0	carry in 1 0 + 1 = 2 carry out 0
carry in	0	1 + 2	=	3 carry out 0	carry in	0	4 + 2	=	6 carry out 0	carry in	0	7 + 2	=	9 carry out 0	carry in 1 0 + 2 = 3 carry out 0
carry in	0	1 + 3	=	4 carry out 0	carry in	0	4 + 3	=	7 carry out 0	carry in	0	7 + 3	=	0 carry out 1	carry in 1 0 + 3 = 4 carry out 0
carry in	0	1 + 4	=	5 carry out 0	carry in	0	4 + 4	=	8 carry out 0	carry in	0	7 + 4	=	1 carry out 1	carry in 1 0 + 4 = 5 carry out 0
carry in	0	1 + 5	=	6 carry out 0	carry in	0	4 + 5	=	9 carry out 0	carry in	0	7 + 5	=	2 carry out 1	carry in 1 0 + 5 = 6 carry out 0
carry in	0	1 + 6	=	7 carry out 0	carry in	0	4 + 6	=	0 carry out 1	carry in	0	7 + 6	=	3 carry out 1	carry in 1 0 + 6 = 7 carry out 0
carry in	0	1 + 7	=	8 carry out 0	carry in	0	4 + 7	=	1 carry out 1	carry in	0	7 + 7	=	4 carry out 1	carry in 1 0 + 7 = 8 carry out 0
carry in	0	1 + 8	=	9 carry out 0	carry in	0	4 + 8	=	2 carry out 1	carry in	0	7 + 8	=	5 carry out 1	carry in 1 0 + 8 = 9 carry out 0
carry in	0	1 + 9	=	0 carry out 1	carry in	0	4 + 9	=	3 carry out 1	carry in	0	7 + 9	=	6 carry out 1	carry in 1 0 + 9 = 0 carry out 1
carry in	0	2 + 0	=	2 carry out 0	carry in	0	5 + 0	=	5 carry out 0	carry in	0	8 + 0	=	8 carry out 0	carry in 1 1 + 0 = 2 carry out 0
carry in	0	2 + 1	=	3 carry out 0	carry in	0	5 + 1	=	6 carry out 0	carry in	0	8 + 1	=	9 carry out 0	carry in 1 1 + 1 = 3 carry out 0
carry in	0	2 + 2	=	4 carry out 0	carry in	0	5 + 2	=	7 carry out 0	carry in	0	8 + 2	=	0 carry out 1	carry in 1 1 + 2 = 4 carry out 0
carry in	0	2 + 3	=	5 carry out 0	carry in	0	5 + 3	=	8 carry out 0	carry in	0	8 + 3	=	1 carry out 1	carry in 1 1 + 3 = 5 carry out 0
carry in	0	2 + 4	=	6 carry out 0	carry in	0	5 + 4	=	9 carry out 0	carry in	0	8 + 4	=	2 carry out 1	carry in 1 1 + 4 = 6 carry out 0
carry in	0	2 + 5	=	7 carry out 0	carry in	0	5 + 5	=	0 carry out 1	carry in	0	8 + 5	=	3 carry out 1	carry in 1 1 + 5 = 7 carry out 0
carry in	0	2 + 6	=	8 carry out 0	carry in	0	5 + 6	=	1 carry out 1	carry in	0	8 + 6	=	4 carry out 1	carry in 1 1 + 6 = 8 carry out 0
carry in	0	2 + 7	=	9 carry out 0	carry in	0	5 + 7	=	2 carry out 1	carry in	0	8 + 7	=	5 carry out 1	carry in 1 1 + 7 = 9 carry out 0
carry in	0	2 + 8	=	0 carry out 1	carry in	0	5 + 8	=	3 carry out 1	carry in	0	8 + 8	=	6 carry out 1	carry in 1 1 + 8 = 0 carry out 1
carry in	0	2 + 9	=	1 carry out 1	carry in	0	5 + 9	=	4 carry out 1	carry in	0	8 + 9	=	7 carry out 1	carry in 1 1 + 9 = 1 carry out 1

กฎการบวกเลขฐาน 10

carry in 1 2 + 0 = 3 carry out 0
carry in 1 2 + 1 = 4 carry out 0
carry in 1 2 + 2 = 5 carry out 0
carry in 1 2 + 3 = 6 carry out 0
carry in 1 2 + 4 = 7 carry out 0
carry in 1 2 + 5 = 8 carry out 0
carry in 1 2 + 6 = 9 carry out 0
carry in 1 2 + 7 = 0 carry out 1
carry in 1 2 + 8 = 1 carry out 1
carry in 1 2 + 9 = 2 carry out 1
carry in 1 3 + 0 = 4 carry out 0
carry in 1 3 + 1 = 5 carry out 0
carry in 1 3 + 2 = 6 carry out 0
carry in 1 3 + 3 = 7 carry out 0
carry in 1 3 + 4 = 8 carry out 0
carry in 1 3 + 5 = 9 carry out 0
carry in 1 3 + 6 = 0 carry out 1
carry in 1 3 + 7 = 1 carry out 1
carry in 1 3 + 8 = 2 carry out 1
carry in 1 3 + 9 = 3 carry out 1
carry in 1 4 + 0 = 5 carry out 0
carry in 1 4 + 1 = 6 carry out 0
carry in 1 4 + 2 = 7 carry out 0
carry in 1 4 + 3 = 8 carry out 0
carry in 1 4 + 4 = 9 carry out 0
carry in 1 4 + 5 = 0 carry out 1
carry in 1 4 + 6 = 1 carry out 1
carry in 1 4 + 7 = 2 carry out 1
carry in 1 4 + 8 = 3 carry out 1
carry in 1 4 + 9 = 4 carry out 1

carry in 1 5 + 0 = 6 carry out 0
carry in 1 5 + 1 = 7 carry out 0
carry in 1 5 + 2 = 8 carry out 0
carry in 1 5 + 3 = 9 carry out 0
carry in 1 5 + 4 = 0 carry out 1
carry in 1 5 + 5 = 1 carry out 1
carry in 1 5 + 6 = 2 carry out 1
carry in 1 5 + 7 = 3 carry out 1
carry in 1 5 + 8 = 4 carry out 1
carry in 1 5 + 9 = 5 carry out 1
carry in 1 6 + 0 = 7 carry out 0
carry in 1 6 + 1 = 8 carry out 0
carry in 1 6 + 2 = 9 carry out 0
carry in 1 6 + 3 = 0 carry out 1
carry in 1 6 + 4 = 1 carry out 1
carry in 1 6 + 5 = 2 carry out 1
carry in 1 6 + 6 = 3 carry out 1
carry in 1 6 + 7 = 4 carry out 1
carry in 1 6 + 8 = 5 carry out 1
carry in 1 6 + 9 = 6 carry out 1
carry in 1 7 + 0 = 8 carry out 0
carry in 1 7 + 1 = 9 carry out 0
carry in 1 7 + 2 = 0 carry out 1
carry in 1 7 + 3 = 1 carry out 1
carry in 1 7 + 4 = 2 carry out 1
carry in 1 7 + 5 = 3 carry out 1
carry in 1 7 + 6 = 4 carry out 1
carry in 1 7 + 7 = 5 carry out 1
carry in 1 7 + 8 = 6 carry out 1
carry in 1 7 + 9 = 7 carry out 1

carry in 1 8 + 0 = 9 carry out 0
carry in 1 8 + 1 = 0 carry out 1
carry in 1 8 + 2 = 1 carry out 1
carry in 1 8 + 3 = 2 carry out 1
carry in 1 8 + 4 = 3 carry out 1
carry in 1 8 + 5 = 4 carry out 1
carry in 1 8 + 6 = 5 carry out 1
carry in 1 8 + 7 = 6 carry out 1
carry in 1 8 + 8 = 7 carry out 1
carry in 1 8 + 9 = 8 carry out 1
carry in 1 9 + 0 = 0 carry out 1
carry in 1 9 + 1 = 1 carry out 1
carry in 1 9 + 2 = 2 carry out 1
carry in 1 9 + 3 = 3 carry out 1
carry in 1 9 + 4 = 4 carry out 1
carry in 1 9 + 5 = 5 carry out 1
carry in 1 9 + 6 = 6 carry out 1
carry in 1 9 + 7 = 7 carry out 1
carry in 1 9 + 8 = 8 carry out 1
carry in 1 9 + 9 = 9 carry out 1

กฎการบวกเลขฐาน 2

- กฎการบวก (ไม่มีตัวทด)

0. 1 + 1 = 0 ทด 1
1. 1 + 0 = 1
2. 0 + 1 = 1
3. 0 + 0 = 0

- กฎการบวก (มีตัวทด)

0. 1 + 1 = 1 ทด 1
1. 1 + 0 = 0 ทด 1
2. 0 + 1 = 0 ทด 1
3. 0 + 0 = 1

Carry in	a	b		Result	Carry out
0	0	0	=	0	0
0	0	1	=	1	0
0	1	0	=	1	0
0	1	1	=	0	1
1	0	0	=	1	0
1	0	1	=	0	1
1	1	0	=	0	1
1	1	1	=	1	1

การบวกเลข

- กฎการบวก (ไม่มีตัวทด)

0. 1 + 1 = 0 ทด 1
1. 1 + 0 = 1
2. 0 + 1 = 1
3. 0 + 0 = 0

- กฎการบวก (มีตัวทด)

0. 1 + 1 = 1 ทด 1
1. 1 + 0 = 0 ทด 1
2. 0 + 1 = 0 ทด 1
3. 0 + 0 = 1

หลัก	3	2	1	0
ตัวทด	0	0	0	0
	1	1	0	1
	1	1	1	0
				1

+

ตัวทด = 0
ตัวตั้ง = 1
ตัวบวก= 0

การบวกเลข

- กฎการบวก (ไม่มีตัวทด)

0. 1 + 1 = 0 ทด 1
1. 1 + 0 = 1
2. 0 + 1 = 1
3. 0 + 0 = 0

- กฎการบวก (มีตัวทด)

0. 1 + 1 = 1 ทด 1
1. 1 + 0 = 0 ทด 1
2. 0 + 1 = 0 ทด 1
3. 0 + 0 = 1

หลัก	3	2	1	0
ตัวทด	0	0	0	0
	1	1	0	1
+	1	1	1	0
			1	1

ตัวทด = 0
ตัวตั้ง = 0
ตัวบวก= 1

การบวกเลข

- กฎการบวก (ไม่มีตัวทด)

0. 1 + 1 = 0 ทด 1
1. 1 + 0 = 1
2. 0 + 1 = 1
3. 0 + 0 = 0

- กฎการบวก (มีตัวทด)

0. 1 + 1 = 1 ทด 1
1. 1 + 0 = 0 ทด 1
2. 0 + 1 = 0 ทด 1
3. 0 + 0 = 1

หลัก	3	2	1	0
ตัวทด	0	0	0	0
	1	1	0	1
+	1	1	1	0
		0	1	1

ตัวทด = 0
ตัวตั้ง = 1
ตัวบวก= 1

การบวกเลข

- กฎการบวก (ไม่มีตัวทด)

0. 1 + 1 = 0 ทด 1
1. 1 + 0 = 1
2. 0 + 1 = 1
3. 0 + 0 = 0

- กฎการบวก (มีตัวทด)

0. 1 + 1 = 1 ทด 1
1. 1 + 0 = 0 ทด 1
2. 0 + 1 = 0 ทด 1
3. 0 + 0 = 1

หลัก	3	2	1	0
ตัวทด	1	0	0	0
	1	1	0	1
+	1	1	1	0
ตอบ	1	1	0	1	1

ตัวทด = 1
ตัวตั้ง = 1
ตัวบวก= 1

การบวกเลข

- กฎการบวก (ไม่มีตัวทด)

0. 1 + 1 = 0 ทด 1
1. 1 + 0 = 1
2. 0 + 1 = 1
3. 0 + 0 = 0

- กฎการบวก (มีตัวทด)

0. 1 + 1 = 1 ทด 1
1. 1 + 0 = 0 ทด 1
2. 0 + 1 = 0 ทด 1
3. 0 + 0 = 1

หลัก	5	4	3	2	1	0
ตัวทด	0	0	0	0	0	0
	1	1	0	1	1	1
+	0	1	1	0	1	1
						0

ตัวทด = 0
ตัวตั้ง = 1
ตัวบวก= 1

การบวกเลข

- กฎการบวก (ไม่มีตัวทด)

0. 1 + 1 = 0 ทด 1
1. 1 + 0 = 1
2. 0 + 1 = 1
3. 0 + 0 = 0

- กฎการบวก (มีตัวทด)

0. 1 + 1 = 1 ทด 1
1. 1 + 0 = 0 ทด 1
2. 0 + 1 = 0 ทด 1
3. 0 + 0 = 1

หลัก	5	4	3	2	1	0
ตัวทด	0	0	0	0	1	0
	1	1	0	1	1	1
+	0	1	1	0	1	1
					1	0

ตัวทด = 1
ตัวตั้ง = 1
ตัวบวก= 1

การบวกเลข

- กฎการบวก (ไม่มีตัวทด)

0. 1 + 1 = 0 ทด 1
1. 1 + 0 = 1
2. 0 + 1 = 1
3. 0 + 0 = 0

- กฎการบวก (มีตัวทด)

0. 1 + 1 = 1 ทด 1
1. 1 + 0 = 0 ทด 1
2. 0 + 1 = 0 ทด 1
3. 0 + 0 = 1

หลัก	5	4	3	2	1	0
ตัวทด	0	0	0	1	1	0
	1	1	0	1	1	1
	0	1	1	0	1	1
				0	1	0

+

ตัวทด = 1
ตัวตั้ง = 1
ตัวบวก= 0

การบวกเลข

- กฎการบวก (ไม่มีตัวทด)

0. 1 + 1 = 0 ทด 1
1. 1 + 0 = 1
2. 0 + 1 = 1
3. 0 + 0 = 0

- กฎการบวก (มีตัวทด)

0. 1 + 1 = 1 ทด 1
1. 1 + 0 = 0 ทด 1
2. 0 + 1 = 0 ทด 1
3. 0 + 0 = 1

- ตัวอย่าง

```
  1001
  1110   +
 ------
 10111
```

การบวกเลข

- กฎการบวก (ไม่มีตัวทด)

0. 1 + 1 = 0 ทด 1
1. 1 + 0 = 1
2. 0 + 1 = 1
3. 0 + 0 = 0

- กฎการบวก (มีตัวทด)

0. 1 + 1 = 1 ทด 1
1. 1 + 0 = 0 ทด 1
2. 0 + 1 = 0 ทด 1
3. 0 + 0 = 1

หลัก	5	4	3	2	1	0	
ตัวทด	0	0	1	1	1	0	
	1	1	0	1	1	1	+
	0	1	1	0	1	1	
			0	0	1	0	

ตัวทด = 1
ตัวตั้ง = 0
ตัวบวก= 1

การบวกเลข

- กฎการบวก (ไม่มีตัวทด)

0. 1 + 1 = 0 ทด 1
1. 1 + 0 = 1
2. 0 + 1 = 1
3. 0 + 0 = 0

- กฎการบวก (มีตัวทด)

0. 1 + 1 = 1 ทด 1
1. 1 + 0 = 0 ทด 1
2. 0 + 1 = 0 ทด 1
3. 0 + 0 = 1

หลัก	5	4	3	2	1	0
ตัวทด	0	1	1	1	1	0
	1	1	0	1	1	1
+	0	1	1	0	1	1
	1	0	0	1	0	

ตัวทด = 1
ตัวตั้ง = 1
ตัวบวก= 1

การบวกเลข

- กฎการบวก (ไม่มีตัวทด)

0. 1 + 1 = 0 ทด 1
1. 1 + 0 = 1
2. 0 + 1 = 1
3. 0 + 0 = 0

- กฎการบวก (มีตัวทด)

0. 1 + 1 = 1 ทด 1
1. 1 + 0 = 0 ทด 1
2. 0 + 1 = 0 ทด 1
3. 0 + 0 = 1

หลัก	5	4	3	2	1	0
ตัวทด	1	1	1	1	1	0
	1	1	0	1	1	1
	0	1	1	0	1	1

+

ตอบ 1 0 1 0 0 1 0

ตัวทด = 1
ตัวตั้ง = 1
ตัวบวก= 0

การบวกเลข

ทดลองบวกเลขฐานสองดังต่อไปนี้

1101 + 1111 = 1 1100

1011 + 100000 = 10 1011

111 + 11111 = 10 0110

การแทนจำนวนติดลบ

ค่าติดลบจากการคำนวณเลขฐาน 2 จะทำให้เกิดเลข 1 อยู่ด้านหน้าไปจนถึงหลัก ∞

แต่ไม่โครโพรเซสเซอร์มีหน่วยความจำอย่างจำกัน ดังนั้นการแทนจำนวนในระบบคอมพิวเตอร์ต้องมีการกำหนด ขนาด หรือนิยามค่ามากที่สุด โดยใช้จำนวนหลัก (Digit) ที่ใช้ในการแทนจำนวน

1 หลัก แทนได้ 2 จำนวน คือ 0 , 1

2 หลัก แทนได้ 4 จำนวน คือ 00,01,10,11

3 หลัก แทนได้ 8 จำนวน คือ 000,001,010,011,100,101,110,111

จำนวนหลักของเลขฐานสอง เรียกว่า Binary Digit หรือ Bit

จำนวนสูงสุดที่แทนได้ = 2^{Bit}

ขนาดของจำนวนที่นิยมใช้ในระบบคอมพิวเตอร์คือ

1 bit, 4 bits, 8 bits, 16 bits, 32 bits, 64 bits, 128 bits

การแทนจำนวนติดลบ

วิธีการสร้างเลขฐานสองแบบติดลบในระบบคอมพิวเตอร์

1 ต้องนิยาม หรือกำหนดขนาดของข้อมูล หรือจำนวนบิต

2 ทำการแปลง 2’s complement

2.1 ทำการกลับบิต ให้เป็นค่าตรงข้าม

2.2 นำค่าที่ได้ + 1

การแทนจำนวนติดลบ

1 ต้องนิยาม หรือกำหนดขนาดของข้อมูล หรือจำนวนบิต

2 ทำการแปลง 2's complement

2.1 ทำการกลับบิต ให้เป็นค่าตรงข้าม

2.2 นำค่าที่ได้ + 1

ตัวอย่าง จงแปลง 10_2 ให้เป็น 2's complement โดยกำหนดให้มีขนาด 8 บิต

ทำให้มีขนาด 8 บิตด้วยการเติม 0 ด้านซ้ายมือ	0 0 0 0 0 0 1 0	
กลับบิตให้เป็นค่าตรงข้าม	1 1 1 1 1 1 0 1	+
ค่าที่ได้ + 1	1	
ตอบ	1 1 1 1 1 1 1 0	

ทดลองตรวจคำตอบได้ง่าย ๆ เพราะการทำ 2's complement ครั้งที่ 2 คำตอบจะกลับมาเป็นค่าเดิม

ค่าเริ่มต้นมีขนาด 8 บิตอยู่แล้ว	1 1 1 1 1 1 1 0	
กลับบิตให้เป็นค่าตรงข้าม	0 0 0 0 0 0 0 1	+
ค่าที่ได้ + 1	1	
ตอบ	0 0 0 0 0 0 1 0	

ได้เท่าเดิม

จงแปลงเลขฐาน10 ค่า -123 ให้อยู่ในเป็นเลขฐาน 2 ขนาด 8 บิต

1) แปลงเลข 123 เป็นเลขฐาน 2 โดยยังไม่คิดเครื่องหมาย

123 / 2 ได้ 61 เศษ 1

61 / 2 ได้ 30 เศษ 1

30 / 2 ได้ 15 เศษ 0

15 / 2 ได้ 7 เศษ 1

7 / 2 ได้ 3 เศษ 1

3 / 2 ได้ 1 เศษ 1

ได้เป็น 1 1 1 1 0 1 1

2) เนื่องจากมีค่าติดลบ จึงต้องแปลงเป็น 2's complement

เติม 0 ด้านซ้ายให้ครบ 8 บิต	**0 1 1 1 1 0 1 1**
กลับบิตเป็นตรงข้าม	**1 0 0 0 0 1 0 0**
บวก 1	**0 0 0 0 0 0 0 1** +
ตอบ	**1 0 0 0 0 1 0 1**

8 บิต

7	6	5	4	3	2	1	0

16 บิต

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

32 บิต

จำนวนเต็มในระบบคอมพิวเตอร์มีอยู่ 2 แบบคือ

- **คิดเครื่องหมาย (Signed)**
- **ไม่คิดเครื่องหมาย (Unsigned)**

ตัวอย่างเช่นเลขฐาน 2 ขนาด 8 บิตค่า **1 0 0 0 0 1 0 1** หากเป็นแบบ**คิดเครื่องหมาย** ให้เติม 0 ด้านซ้ายจนครบทุกบิต แล้วดูบิตซ้ายสุดว่าเป็น 1 หรือไม่

1 0 0 0 0 1 0 1 กรณีนี้บิตซ้ายสุดเป็น 1 แสดงว่าเป็นเลขติดลบ ให้ทำการแปลง 2's complement อีก 1 ครั้งจะได้ค่าจริง

กลับบิต +1

1 0 0 0 0 1 0 1 -> 0 1 1 1 1 0 1 0 -> 0 1 1 1 1 0 1 1

แปลงเป็นฐาน 10 ได้เป็น $(0x2^7)+ (1x2^6) + (1x2^5) + (1x2^4) + (1x2^3) + (0x2^2) + (1x2^1) + (1x2^0)$

64 + 32 + 16 + 8 +2 + 1 = 123

ตอบ -123

จำนวนเต็มในระบบคอมพิวเตอร์มีอยู่ 2 แบบคือ

- **คิดเครื่องหมาย (Signed)**
- **ไม่คิดเครื่องหมาย (Unsigned)**

ตัวอย่างเช่นเลขฐาน 2 ขนาด 8 บิตค่า **1 0 0 0 0 1 0 1** หากเป็นแบบ**ไม่คิดเครื่องหมาย** สามารถแปลงเป็นเลขฐาน 10 ได้ทันที

แปลงเป็นฐาน 10 ได้เป็น $(1x2^7)+ (0x2^6) + (0x2^5) + (0x2^4) + (0x2^3) + (1x2^2) + (0x2^1) + (1x2^0)$

128 +4 + 1 = 133

ตอบ 133

จำนวนเต็มในระบบคอมพิวเตอร์มีอยู่ 2 แบบคือ

- **คิดเครื่องหมาย (Signed)**
- **ไม่คิดเครื่องหมาย (Unsigned)**

ตัวอย่างเช่นเลขฐาน 2 ขนาด **16** บิตค่า **1 0 0 0 0 1 0 1** หากเป็นแบบ**คิดเครื่องหมาย** แปลงเป็นเลขฐาน 10 ได้เท่าใด ?

ให้เติม 0 จนครบ 16 บิต **0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1**

บิตซ้ายสุดมีค่าเป็น 0 แสดงว่าไม่ติดลบ จึงไม่ต้องทำ 2’ complement อีก และสามารถแปลงเป็นฐาน 10 ได้เลย

แปลงเป็นฐาน 10 ได้เป็น $(1x2^7)+ (0x2^6) + (0x2^5) + (0x2^4) + (0x2^3) + (1x2^2) + (0x2^1) + (1x2^0)$

128 **+4** **+ 1 = 133**

ตอบ 133

ไมโครโพรเซสเซอร์ไม่จำเป็นต้องมีวงจรลบเลข หากโปรแกรมเมอร์ต้องการคำนวณการลบ สามารถทำได้โดยการนำค่าที่ต้องการลบไปหา 2’s complement ก่อน จากนั้นจึงนำไปบวก

ตัวอย่างเช่นจงคำนวณ 9 - 3

หากใช้วงจรบวกเลขขนาด 8 บิต จะมีวิธีการทำดังนี้

ตัวตั้ง 9 แปลงเป็นเลขฐาน 2 ขนาด 8 บิตได้เป็น 0 0 0 0 1 0 0 1

ตัวลบ 3 แปลงเป็นเลขฐาน 2 ขนาด 8 บิตได้เป็น 0 0 0 0 0 0 1 1

แปลงตัวลบเป็น 2’s complement เพื่อให้มีค่าเป็น -3 ได้เป็น 1 1 1 1 1 1 0 1

+

นำไปบวกกับตัวตั้ง 0 0 0 0 1 0 0 1

ได้ผลบวกเป็น 1 1 1 1 1 0 1 0

ทำ 2’s complement อีกครั้งเพื่อทำให้เครื่องหมายติดลบ หายไป 0 0 0 0 0 1 1 0

ตอบ 0 0 0 0 0 1 1 0 แปลงเป็น ฐาน 10 คือ 6

จะพบว่าวงจรลบเลขนั้นไม่จำเป็น มีเพียงวงจรบวกเลข (ADD) และวงจรกลับบิต (NOT) ก็เพียงพอแล้ว

เส้นจำนวนของจำนวนเต็มแบบคิดเครื่องหมาย

0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	a	10
11	b	11
12	c	12
13	d	13
14	e	14
15	f	15
16	10	16
17	11	17
18	12	18
19	13	19
20	14	20
21	15	21
22	16	22
23	17	23
24	18	24
25	19	25
26	1a	26
27	1b	27
28	1c	28
29	1d	29
30	1e	30
31	1f	31
32	20	32
33	21	33
34	22	34
35	23	35
36	24	36
37	25	37
38	26	38
39	27	39
40	28	40
41	29	41
42	2a	42
43	2b	43
44	2c	44
45	2d	45
46	2e	46
47	2f	47
48	30	48
49	31	49
50	32	50
51	33	51
52	34	52
53	35	53
54	36	54
55	37	55
56	38	56
57	39	57
58	3a	58
59	3b	59
60	3c	60
61	3d	61
62	3e	62
63	3f	63

64	40	64
65	41	65
66	42	66
67	43	67
68	44	68
69	45	69
70	46	70
71	47	71
72	48	72
73	49	73
74	4a	74
75	4b	75
76	4c	76
77	4d	77
78	4e	78
79	4f	79
80	50	80
81	51	81
82	52	82
83	53	83
84	54	84
85	55	85
86	56	86
87	57	87
88	58	88
89	59	89
90	5a	90
91	5b	91
92	5c	92
93	5d	93
94	5e	94
95	5f	95
96	60	96
97	61	97
98	62	98
99	63	99
100	64	100
101	65	101
102	66	102
103	67	103
104	68	104
105	69	105
106	6a	106
107	6b	107
108	6c	108
109	6d	109
110	6e	110
111	6f	111
112	70	112
113	71	113
114	72	114
115	73	115
116	74	116
117	75	117
118	76	118
119	77	119
120	78	120
121	79	121
122	7a	122
123	7b	123
124	7c	124
125	7d	125
126	7e	126
127	7f	127

128	80	-128
129	81	-127
130	82	-126
131	83	-125
132	84	-124
133	85	-123
134	86	-122
135	87	-121
136	88	-120
137	89	-119
138	8a	-118
139	8b	-117
140	8c	-116
141	8d	-115
142	8e	-114
143	8f	-113
144	90	-112
145	91	-111
146	92	-110
147	93	-109
148	94	-108
149	95	-107
150	96	-106
151	97	-105
152	98	-104
153	99	-103
154	9a	-102
155	9b	-101
156	9c	-100
157	9d	-99
158	9e	-98
159	9f	-97
160	a0	-96
161	a1	-95
162	a2	-94
163	a3	-93
164	a4	-92
165	a5	-91
166	a6	-90
167	a7	-89
168	a8	-88
169	a9	-87
170	aa	-86
171	ab	-85
172	ac	-84
173	ad	-83
174	ae	-82
175	af	-81
176	b0	-80
177	b1	-79
178	b2	-78
179	b3	-77
180	b4	-76
181	b5	-75
182	b6	-74
183	b7	-73
184	b8	-72
185	b9	-71
186	ba	-70
187	bb	-69
188	bc	-68
189	bd	-67
190	be	-66
191	bf	-65

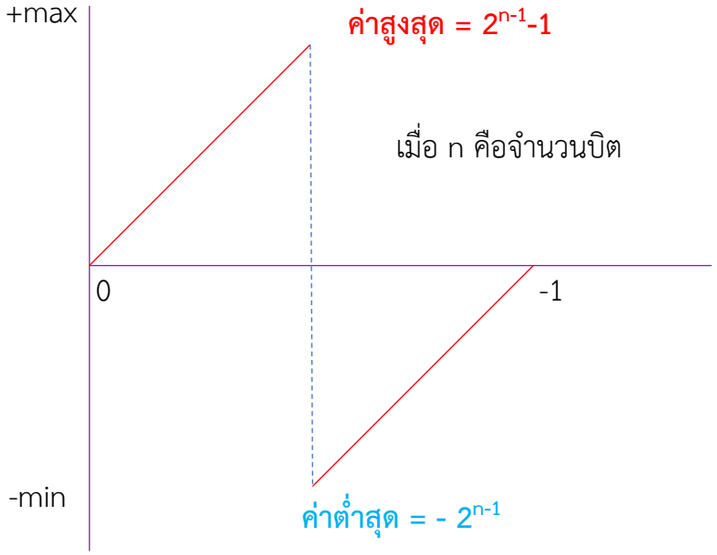
192	c0	-64
193	c1	-63
194	c2	-62
195	c3	-61
196	c4	-60
197	c5	-59
198	c6	-58
199	c7	-57
200	c8	-56
201	c9	-55
202	ca	-54
203	cb	-53
204	cc	-52
205	cd	-51
206	ce	-50
207	cf	-49
208	d0	-48
209	d1	-47
210	d2	-46
211	d3	-45
212	d4	-44
213	d5	-43
214	d6	-42
215	d7	-41
216	d8	-40
217	d9	-39
218	da	-38
219	db	-37
220	dc	-36
221	dd	-35
222	de	-34
223	df	-33
224	e0	-32
225	e1	-31
226	e2	-30
227	e3	-29
228	e4	-28
229	e5	-27
230	e6	-26
231	e7	-25
232	e8	-24
233	e9	-23
234	ea	-22
235	eb	-21
236	ec	-20
237	ed	-19
238	ee	-18
239	ef	-17
240	f0	-16
241	f1	-15
242	f2	-14
243	f3	-13
244	f4	-12
245	f5	-11
246	f6	-10
247	f7	-9
248	f8	-8
249	f9	-7
250	fa	-6
251	fb	-5
252	fc	-4
253	fd	-3
254	fe	-2
255	ff	-1

ตัวอย่างเส้นจำนวนของเลข 2's complement ขนาด 8 บิต

จำนวนตัวเลขทั้งหมด 256

ค่าสูงสุด 127

ค่าน้อยสุด -128



การป้อนข้อมูลตรวจสอบให้อยู่ในช่วง เพราะถ้าเกิดการล้น หรือ Overflow

ข้อมูลอาจเปลี่ยนจาก 127 เป็น -128 ได้

Data type: Primitive

การบันทึกและแปลความหมายข้อมูล: เลขทศนิยม

ใช้วิธีการแปลงเลขฐาน 10 เป็นเลขฐาน 2
แล้วแทนลงหน่วยความจำได้เลย

นักศึกษายังจำวิธีแปลงเลขฐาน 10 เป็นฐาน 2
กรณีที่เป็นเลขทศนิยม ได้อยู่หรือไม่ ?

ใช้วิธีการแปลงทศนิยมฐาน 10 เป็นเลขฐาน 2

ใช้วิธีคูณด้วย 2 แล้วยกเศษเหลือมาตั้งแล้วคูณไปเรื่อย ๆ จนกว่าจะได้ 0

จงแปลง 0.654 ให้เป็นเลขฐานสอง

0.654 X 2 = 1 . 308

.308 X 2 = 0 . 616

.616 X 2 = 1 . 232

.232 X 2 = 0 . 464

.464 X 2 = 0 . 928

ตอบ 0.10100

ใช้วิธีการแปลงทศนิยมฐาน 2 เป็นเลขฐาน 10

$$N = \sum d \times base^{-p}$$

base=2

$$N = 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5} + \cdots \times 2^{p}$$

0. 1 0 1 0 0

$$N = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 0 \times 2^{-5}$$

$$N = 1 \times 0.5 + 0 \times 0.25 + 1 \times 0.125 + 0 \times 0.0625 + 0 \times 0.03125$$

$$N = 0.5 + .125$$

$$N = .625$$

การแปลงเลขทศนิยมไปกลับ อาจทำให้เกิดความคลาดเคลื่อนจากการปัดเลขได้

ใช้วิธีการแปลงทศนิยมฐาน 10 เป็นเลขฐาน 2

หากมีทั้งจำนวนเต็มและทศนิยม ให้แยกคิดส่วนที่เป็นจำนวนเต็ม และทศนิยม

16 . 52

16 / 2	= 8	เศษ	0	
8 / 2	= 4	เศษ	0	
4 / 2	= 2	เศษ	0	
2 / 2	= 1	เศษ	0	

1 0 0 0 0

.52 X 2	= 1	เศษ	0.04	
.04 X 2	= 0	เศษ	0.08	
.08 X 2	= 0	เศษ	0.16	
.16 X 2	= 0	เศษ	0.32	
.32 X 2	= 0	เศษ	0.64	
.64 X 2	= 1	เศษ	0.28	
.28 X 2	= 1	เศษ	0.56	
.56 X 2	= 1	เศษ	0.12	
0.12 X 2	= 0	เศษ	0.24	

.1 0 0 0 0 1 1 1 0

ตอบ 10000.100001110

ใช้วิธีการแปลงทศนิยมฐาน 2 เป็นเลขฐาน 10

หากมีทั้งจำนวนเต็มและทศนิยม ให้แยกคิดส่วนที่เป็นจำนวนเต็ม และทศนิยม

10000 . 100001110

$$N = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

$$N = 1 \times 16 = 16$$

$$N = 1 \times 2^{-1} + 0 \times 2^{-6} + 1 \times 2^{-7} + 0 \times 2^{-8}$$

$$N = 0.5 + 0.015625 + 0.0078125 + 0.00390625$$

$$N = 0.5 + 0.015625 + 0.0078125 + 0.00390625$$

$$N = 0.52734375$$

ตอบ 16.52734375

Data type: Primitive

เลขทศนิยม คำนวณซับซ้อน ใช้พื้นที่มากในการจัดเก็บ และมีการสูญเสียความแม่นยำจากการปัดเศษ

เดิมไมโครโพรเซสเซอร์ 8086 ไม่รองรับการคำนวณแบบเลขทศนิยม
ดังนั้นการคำนวณจึงเกิดขึ้นที่ฝั่ง software โดยโปรแกรมเมอร์ต้องวนลูปคำนวณแปลงเลขเอง
ไม่มีมาตรฐานแน่นอนในการจัดเก็บในหน่วยความจำ

หากต้องการคำนวณเลขทศนิยมด้วย Hardware ต้องใช้หน่วยประมวลผลช่วยหรือ Coprocessor
ปัจจุบันเรียกว่า Floating Point Processing Unit : FPU
ทุกวันนี้ FPU ถูกรวมเอาไว้ใน CPU แล้ว
และได้มีการกำหนดมาตรฐานในการจัดเก็บข้อมูลเลขทศนิยมในหน่วยความจำ

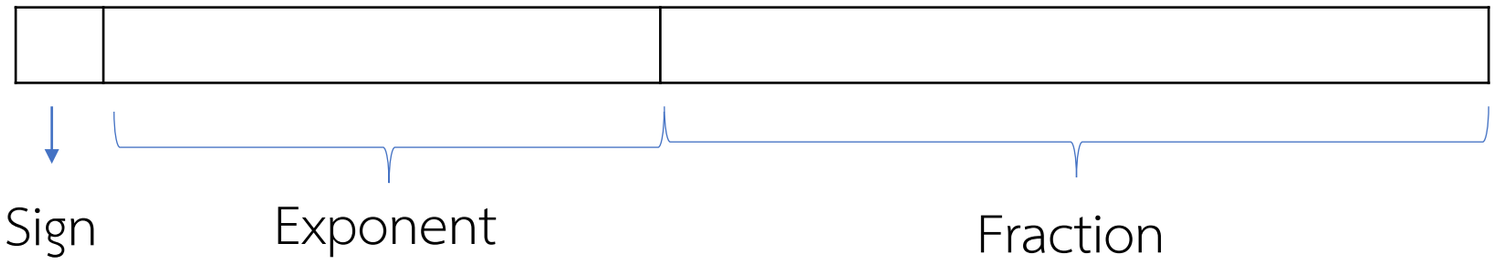
IEEE 754

มาตรฐานนี้คลอบคลุม การแทนบิต การคำนวณ และการปัดเลข
เพื่อให้คอมพิวเตอร์ทุกเครื่อง โปรแกรมทุกภาษา คำนวณแล้วได้คำตอบเดียวกัน

IEEE

IEEE 754

โครงสร้างของข้อมูลจำนวนทศนิยมแบบ IEEE754



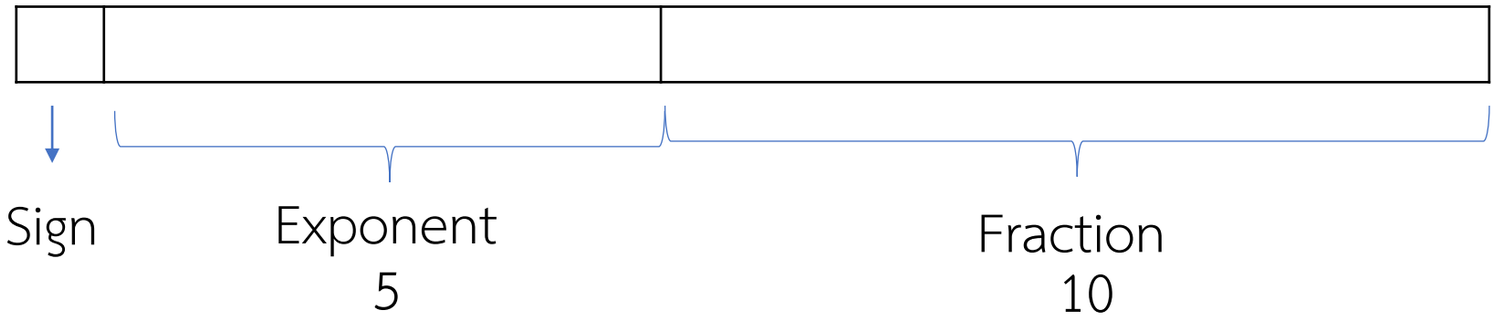
	Bits	Exponent	Fraction	C/C++
Half precision	16	5	10	ไม่รองรับ
Single Precision	32	8	23	float
Double Precision	64	11	52	double

IEEE 754

การแปลงจำนวนเลขฐาน 10 เป็นเลขฐาน 2 แบบ IEEE754 มีขั้นตอนดังตัวอย่าง

ให้แปลงจำนวน 63.72 เป็นเลขฐานสองด้วยความละเอียดแบบ Half Precision

	Bits	Exponent	Fraction	C/C++
Half precision	16	5	10	ไม่รองรับ
Single Precision	32	8	23	float
Double Precision	64	11	52	double



ให้แปลงจำนวน 63.72 เป็นเลขฐานสองด้วยความละเอียดแบบ Half Precision

1 ให้แปลงเป็นเลขฐานสอง

63 . 72

63 / 2 = 31 เศษ 1

31 / 2 = 15 เศษ 1

15 / 2 = 7 เศษ 1

7 / 2 = 3 เศษ 1

3 / 2 = 1 เศษ 1

.72 X 2 = 1 เศษ .44

.44 X 2 = 0 เศษ .88

.88 X 2 = 1 เศษ .76

.76 X 2 = 1 เศษ .52

.52 X 2 = 1 เศษ .04

.04 X 2 = 0 เศษ .08

.04 X 2 = 0 เศษ .16

.16 X 2 = 0 เศษ .32

1 1 1 1 1 1 . 1 0 1 1 1 0 0 0

ให้แปลงจำนวน 63.72 เป็นเลขฐานสองด้วยความละเอียดแบบ Half Precision

1 1 1 1 1 1 . 1 0 1 1 1 0 0 0

1 ทำการ Normalization ให้มีค่าเป็น 1 หรือ 1 x 2^s

1 1 1 1 1 1.1 0 1 1 1 0 0 0 x 2^0

1 1 1 1 1.1 1 0 1 1 1 0 0 0 x 2^1

1 1 1 1.1 1 1 0 1 1 1 0 0 0 x 2^2

1 1 1.1 1 1 1 0 1 1 1 0 0 0 x 2^3

1 1.1 1 1 1 1 0 1 1 1 0 0 0 x 2^4

1.1 1 1 1 1 1 0 1 1 1 0 0 0 x 2^5

Fraction

S = 5

Fraction = 1 1 1 1 1 1 0 1 1 1 0 0 0

ให้แปลงจำนวน 63.72 เป็นเลขฐานสองด้วยความละเอียดแบบ Half Precision

1 1 1 1 1 1 . 1 0 1 1 1 0 0 0

2 คำนวณค่า Base จาก Base = 2^{k-1} -1 โดย k คือ จำนวนบิตของ Exponent

	Bits	Exponent	Fraction	C/C++
Half precision	16	5	10	ไม่รองรับ
Single Precision	32	8	23	float
Double Precision	64	11	52	double

Base= $2^{5-1}-1 = 2^4-1 = 15$

** ค่า Base จะเปลี่ยนแปลงตามจำนวนบิต เพื่อความรวดเร็ว นักศึกษาสามารถคำนวณค่านี้แล้วจดไว้ได้เลย โจทย์ข้ออื่นที่ใช้ความละเอียดเท่ากันจะได้นำมาใช้ได้ทันทีโดยไม่ต้องคำนวณใหม่ **

ให้แปลงจำนวน 63.72 เป็นเลขฐานสองด้วยความละเอียดแบบ Half Precision

3 คำนวณส่วน Exponent จาก Base + s

Base= 15

S = 5

Exponent = 15 + 5 = 20

ให้แปลงจำนวน 63.72 เป็นเลขฐานสองด้วยความละเอียดแบบ Half Precision

3 แปลงส่วน Exponent เป็นเลขฐานสอง

Exponent = 20

20 / 2 = 10 เศษ 0

10 / 2 = 5 เศษ 0

5 / 2 = 2 เศษ 1

2 / 2 = 1 เศษ 0

1 0 1 0 0

ให้แปลงจำนวน 63.72 เป็นเลขฐานสองด้วยความละเอียดแบบ Half Precision

4 นำส่วน Sign , Exponent และ Fraction มารวมกัน

Exponent = 1 0 1 0 0

Fraction = 1 1 1 1 1 1 0 1 1 1 0 0 0

Sign = 1 ถ้าค่าติดลบ

Sign = 0 ถ้าเป็นค่าบวก

	Bits	Exponent	Fraction	C/C++
Half precision	16	5	10	ไม่รองรับ
Single Precision	32	8	23	float
Double Precision	64	11	52	double

0	1	0	1	0	0	1	1	1	1	1	1	0	1	1	1

5

10

ให้แปลงจำนวน 63.72 เป็นเลขฐานสองด้วยความละเอียดแบบ Half Precision

0	1	0	1	0	0	1	1	1	1	1	1	0	1	1	1

5 10

ตอบ 0101 0011 1111 0111 หรือ 53F7 ในระบบเลขฐาน16

แปลงกลับเป็นเลขฐาน10แล้วค่าเท่าเดิมหรือไม่ ?

จะแปลงกลับได้อย่างไร?

ให้แปลงค่าต่อไปนี้เป็นเลขฐานสิบแบบ Half precision

0	1	0	1	0	0	1	1	1	1	1	1	0	1	1	1

Sign — 5 Exponent — 10 Fraction

Sign = 0

Exponent= 10100

Fraction=1111110111

Bias= 15

ให้แปลงค่าต่อไปนี้เป็นเลขฐานสิบแบบ Half precision

Sign = 0 , Exponent= 10100 , Fraction=1111110111, Bias= 15

1 แปลง Exponent และ Fraction ให้เป็นเลขฐานสิบ

Exponent $=1x2^4+0x2^3+1x2^2+0x1^1+0x2^0$

Exponent =16+0+4+0+0= 20

Fraction $=1x2^{-1}+1x2^{-2}+1x2^{-3}+1x1^{-4}+1x2^{-5}+1x2^{-6}+0x2^{-7}+1x2^{-8}+1x1^{-9}+1x2^{-10}$

Fraction =0.5+0.25+0.125+0.0625+0.03125+0.015625+0+0.00390625

+ 0.001953125 + 0.0009765625 = 0.9912109375

ให้แปลงค่าต่อไปนี้เป็นเลขฐานสิบแบบ Half precision

Sign = 0 , Exponent= 10100 , Fraction=1111110111, Bias= 15

20 0.9912109375

2 คำนวณ Bias_Exponent จาก Exponent - Bias

Bias_Exponent = 20 – 15 = 5

3 คำนวณเครื่องหมายเป็น + เมื่อ Sign = 0

เป็น - เมื่อ Sign =1

4 คำนวณค่าจริงจาก $2^{Bias_Exponent}$ x (1 + Fraction)

= 2^5 x (1 + 0.9912109375)

= 32 x 1.9912109375 = 63.71875 ตอบ

Data type: Composite

- ข้อมูลเชิงประกอบ
- สร้างจากการรวมข้อมูลชนิดเดียวกัน หรือต่างชนิด เข้าด้วยกัน

Array	Vector	String
สร้างจาก data type ที่เป็นชนิดเดียวกัน (Homogenous)	เหมือน Array แต่สามารถสร้างจาก data type ที่เป็นชนิดเดียวกัน หรือ ต่างชนิดกัน	สร้างจากการนำเอา char มารวมกัน เพื่อเก็บข้อความ

- แต่ละภาษาจะจัดการข้อมูลไม่เหมือนกัน
- ข้อมูล String ของแต่ละภาษาก็ไม่เหมือนกัน
- ภาษา C ไม่มี String และ Vector
- สามารถ กำหนดค่า(assign) รวม(join) เปรียบเทียบ(compare)

ตัวอย่าง String ในภาษา Java

กำหนดค่า	รวม	เปรียบเทียบ
A=“Hello” B=“World”	C=A + B	c= A == B

Data type:

ข้อมูลเชิงเดี่ยว และเชิงประกอบ

- เหมาะสำหรับการแก้ปัญหาที่ไม่ซับซ้อน
- หากใช้แก้ปัญหาที่ซับซ้อนจะเขียนโปรแกรมได้ยาก
- Data type มักไม่ตรงตามความต้องการ

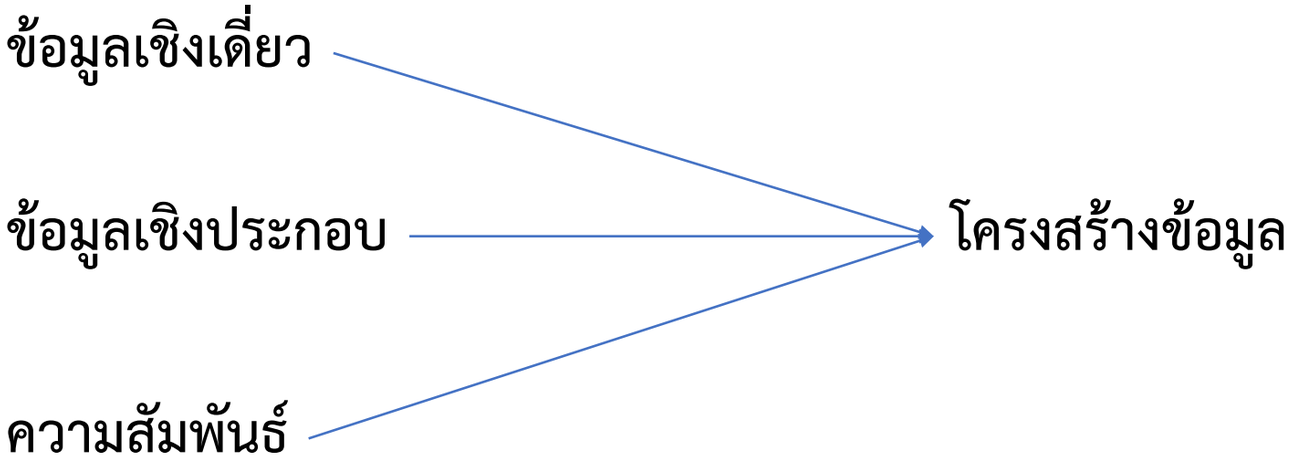
การสร้าง Data type ชนิดใหม่ให้ตรงตามความต้องการ

คือ การสร้าง โครงสร้างของข้อมูล (data structure)

คือ วิชานี้

โครงสร้างข้อมูล

- โปรแกรมสำหรับใช้ในการจัดเก็บข้อมูล
- เป็นส่วนขยายของ ข้อมูลเชิงประกอบ
- ครอบคลุมถึงการมีคุณสมบัติเชิงความสัมพันธ์



ตัวอย่างความสัมพันธ์ ของข้อมูล

“Hello World”

H	e	l	l	o		W	o	r	l	d

String คือข้อมูลเชิงประกอบ ที่มีความสัมพันธ์เชิงลำดับ

หากลำดับเปลี่ยนไป ความหมายของข้อมูลจะเปลี่ยน

W	o	r	l	d		H	e	l	l	o

d	l	r	o	W		o	l	l	e	H

ข้อมูลเหมือนเดิม แต่ลำดับเปลี่ยน ความหมายจะเปลี่ยนไป

Data Structure: Relation

โครงสร้างข้อมูลแบ่งตามความสัมพันธ์ได้ 2 ประเภท

- แบบเป็นเชิงเส้น (Linear)
    - มีความสัมพันธ์ในเชิงลำดับ ก่อนหลัง (Sequence)
    - ความสัมพันธ์ไม่ซับซ้อน ตัวอย่างเช่น Array
- แบบไม่เป็นเชิงเส้น (Non-Linear)
    - มีความสัมพันธ์ในเชิงลำดับชั้น (Hierarchical)
    - ความสัมพันธ์เป็นแบบซับซ้อน ตัวอย่างเช่น ต้นไม้ Mind map

ฟังก์ชันย่อยของโครงสร้างข้อมูล ส่วนใหญ่ประกอบไปด้วย

เพิ่มข้อมูล()

เรียกใช้ข้อมูล()

ค้นหาข้อมูล()

เรียงข้อมูล()

เรียกข้อมูลทั้งหมด()

แทรกข้อมูล()

ลบข้อมูล()

จองพื้นที่หน่วยความจำ()

บันทึกข้อมูล()

อ่านข้อมูล()

ค้นหา()

เรียง()

คืนพื้นที่หน่วยความจำ()

หน่วยความจำ

Data Structure: Data Operation

ฟังก์ชันย่อยของโครงสร้างข้อมูล ส่วนใหญ่ประกอบไปด้วย

เพิ่มข้อมูล()
เรียกใช้ข้อมูล()
ค้นหาข้อมูล()
เรียงข้อมูล()
เรียกข้อมูลทั้งหมด()
แทรกข้อมูล()
ลบข้อมูล()
จองพื้นที่หน่วยความจำ()
บันทึกข้อมูล()
อ่านข้อมูล()
ค้นหา()
เรียง()
คืนพื้นที่หน่วยความจำ()

C

Class : Data structure

- หน่วยความจำ

+เพิ่มข้อมูล()
+เรียกใช้ข้อมูล()
+ค้นหาข้อมูล()
+เรียงข้อมูล()
+เรียกข้อมูลทั้งหมด()
+แทรกข้อมูล()
+ลบข้อมูล()

-จองพื้นที่หน่วยความจำ()
-บันทึกข้อมูล()
-อ่านข้อมูล()
-ค้นหา()
-เรียง()
-คืนพื้นที่หน่วยความจำ()

Java (ADT)

วิธีการสร้างโครงสร้างข้อมูล

- เขียนโครงสร้างข้อมูล รวมกับโปรแกรม
  - วิธีเก่า
  - ไม่ปลอดภัย
  - นำ code กลับมาใช้ใหม่ได้ยาก

- แบบชนิดข้อมูลนามธรรม (Abstract Data Type)
  - วิธีใหม่ แยก code การจัดการข้อมูลออกจากโปรแกรม
  - ปลอดภัย
  - นำ code กลับมาใช้ใหม่ได้ง่าย
  - เชิงวัตถุ

แบบชนิดข้อมูลนามธรรม

Class : Data structure

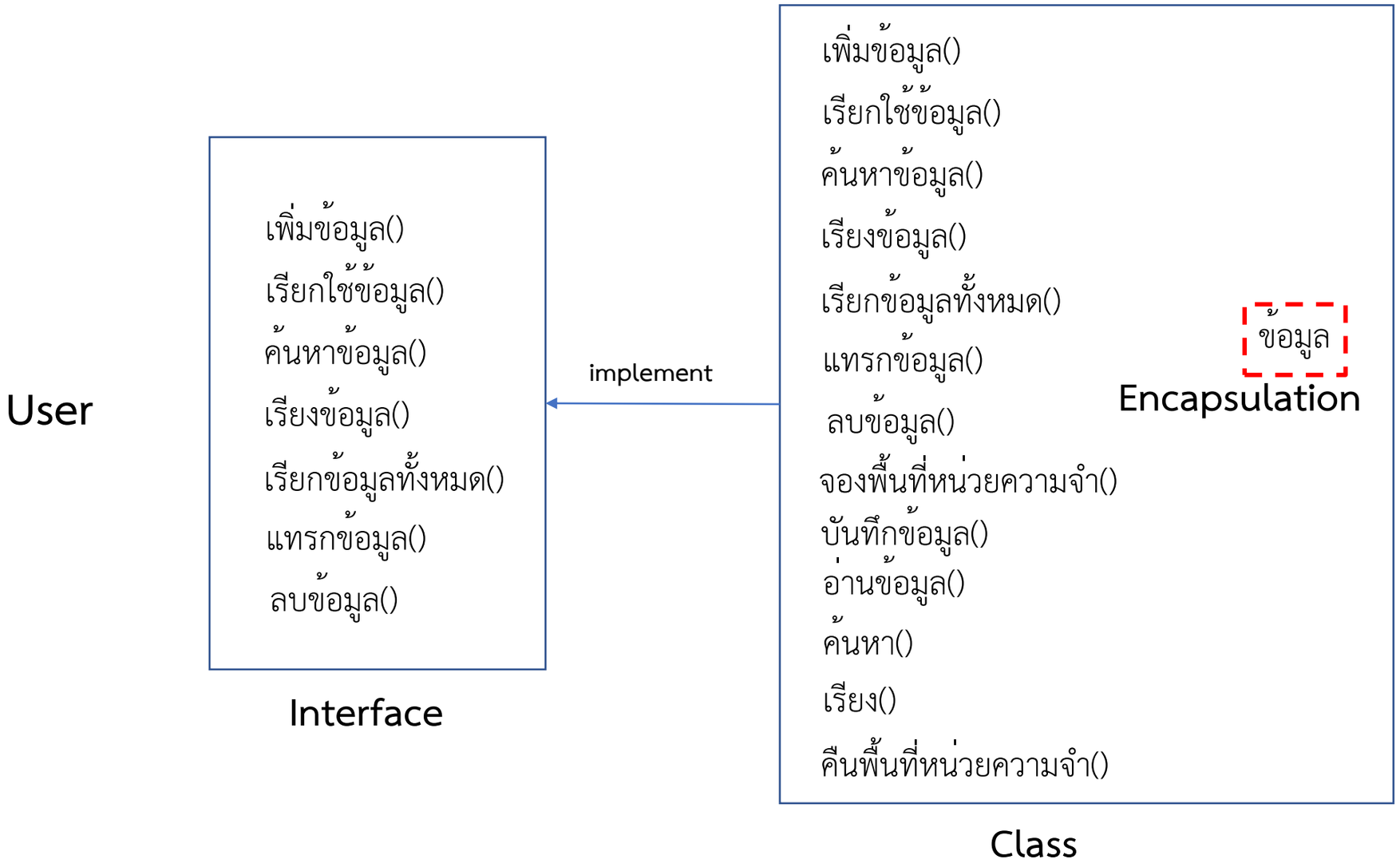
- หน่วยความจำ

+เพิ่มข้อมูล()
+เรียกใช้ข้อมูล()
+ค้นหาข้อมูล()
+เรียงข้อมูล()
+เรียกข้อมูลทั้งหมด()
+แทรกข้อมูล()
+ลบข้อมูล()

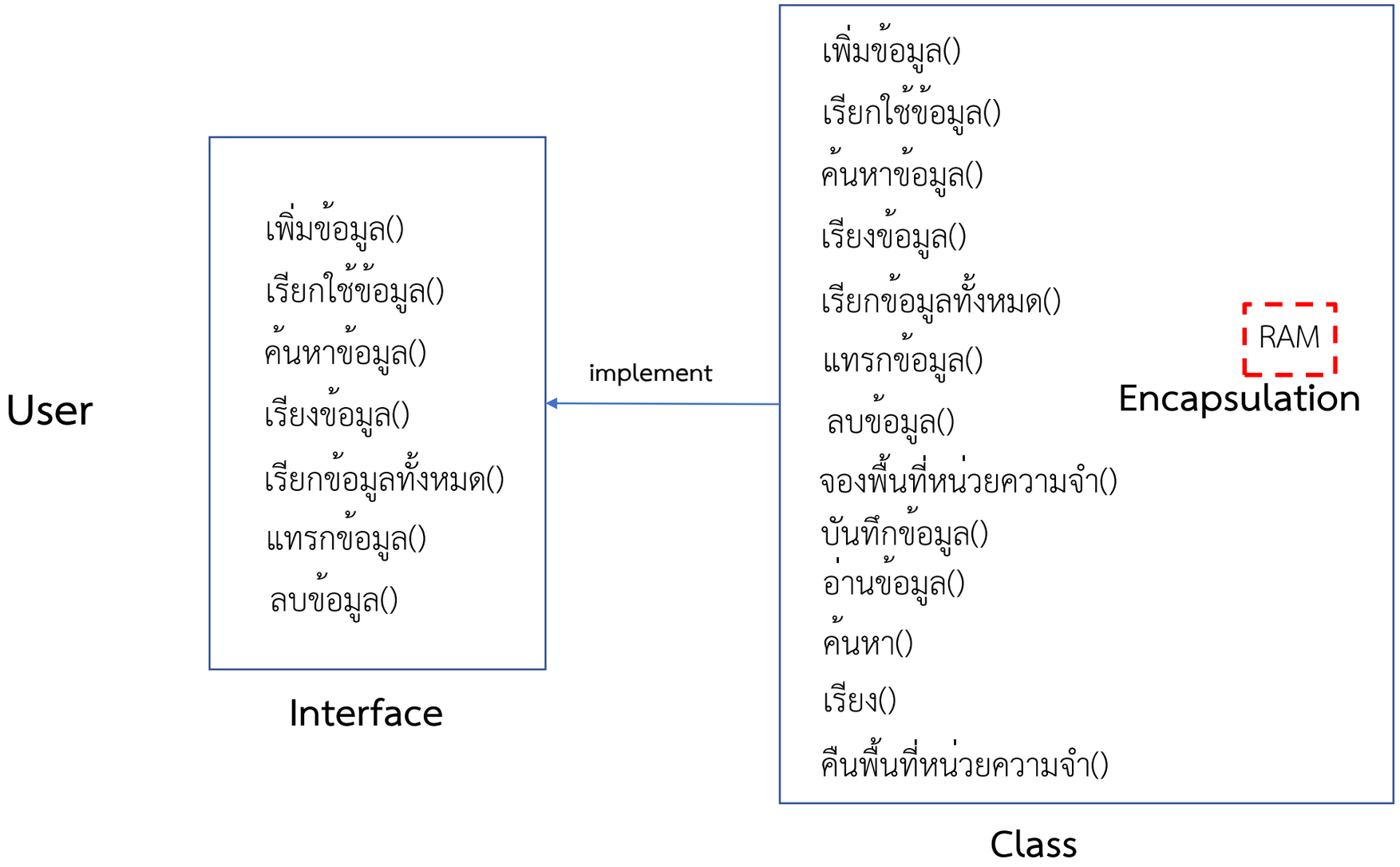
-จองพื้นที่หน่วยความจำ()
-บันทึกข้อมูล()
-อ่านข้อมูล()
-ค้นหา()
-เรียง()
-คืนพื้นที่หน่วยความจำ()

โครงสร้างข้อมูลนี้เป็นนามธรรม
หรือไม่ ?

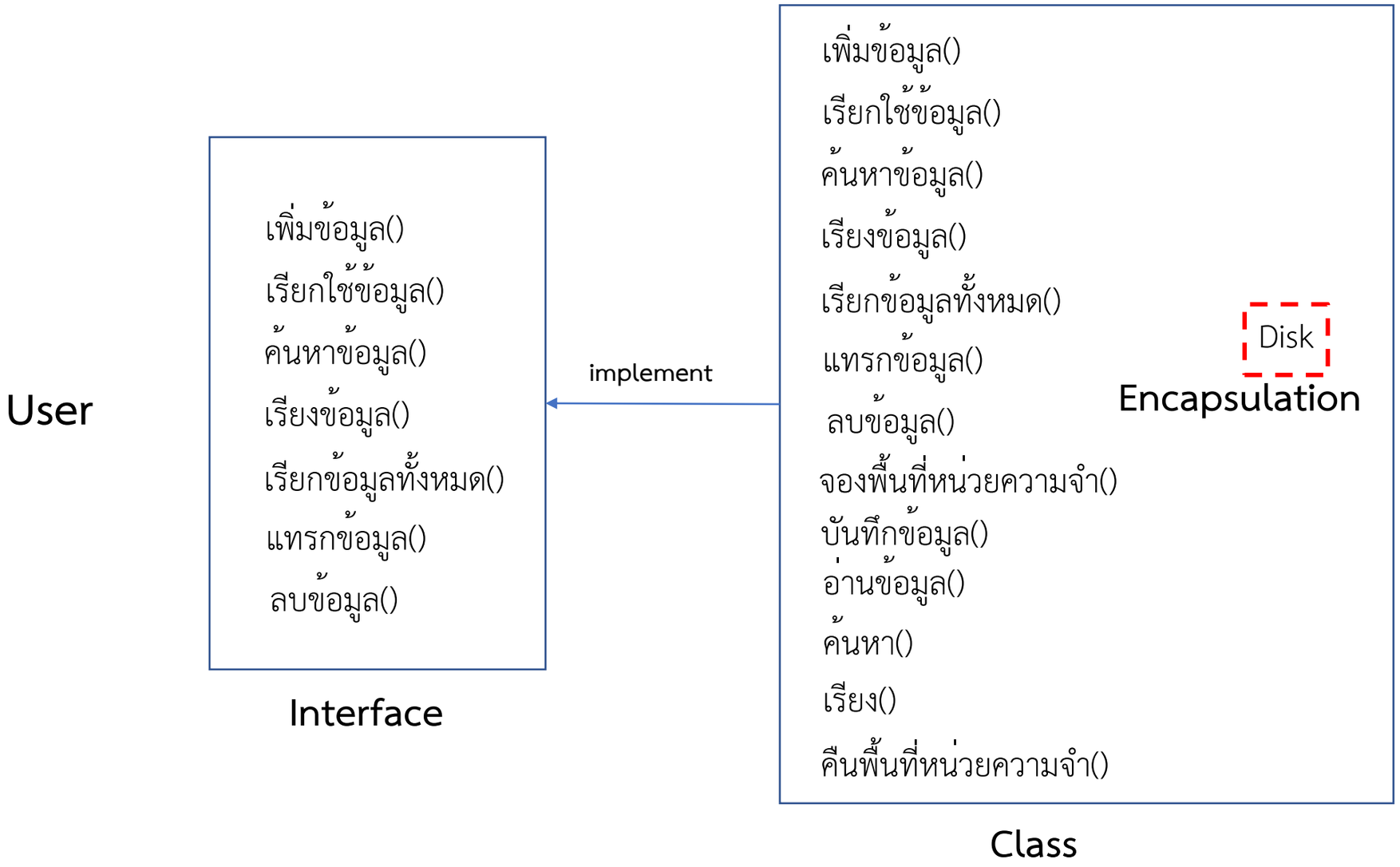
แบบชนิดข้อมูลนามธรรม



แบบชนิดข้อมูลนามธรรม



แบบชนิดข้อมูลนามธรรม



แบบชนิดข้อมูลนามธรรม

char A[]="Hello World"

	H	e	l	l	o		W	o	r
l	d	\0							

Primitive Type String

(char array)

Char A[]= new Char("Hello World")

	H	e	l		l	d
l	o					
			W	o	r	

Abstract Data Type String

(Class type String)

แบบชนิดข้อมูลนามธรรมใน Java (Wrapper class)

Primitive Data Type	Wrapper Class
char	Character
byte	Byte
short	Short
long	Integer
float	Float
double	Double
boolean	Boolean