bitAnd

0 ~ 255
↳ $2^8 - 1$

&	0	1
0	0	0
1	0	1

|	0	1
0	0	1
1	1	1

&~	0	1
0	1	1
1	1	0

|~	0	1
0	1	0
1	0	0

~

	1	0
1	1	0
0	1	0

---

getByte

0x 12 34 56 78

3 2 1 0

MSB LSB

1 byte = 8 bit

n = 1 (1 byte) <<
3 byte >>

ff ff ff 56 & 0 0 0 1 = 0 0 0 56

11 11 11 11 11 11 01 01 01 10 & 00 00 00 00 00 00 11 11 11 11 = 0 0 0 0 0 0 10 01 10 10

n

n	0	1	2	3
0b	③11	③11	③11	③11
- 0b	⓪00	①01	②10	③11
	③11	②10	①01	⓪00

---

logical shift

0x 8 7 6 5 4 3 2 1

10 00 01 11 01 10 01 01 01 00 00 11 00 10 00 01

n bit shift

n1 → 0b11111 → 00000 / 11111

n0 → 0b11110 → 00001 / 11110

n9 → 0b11101 → 00010 / 11101

⋮

0xFFFFFFFF

31-15 → 11111 / 01111 / 10000

-16 8 4 1

X / 2^n

round to zero

		-16 8 4 1
-10	10110	
-5	11011	
-3	11101	
-2	11110	
-1	11111	

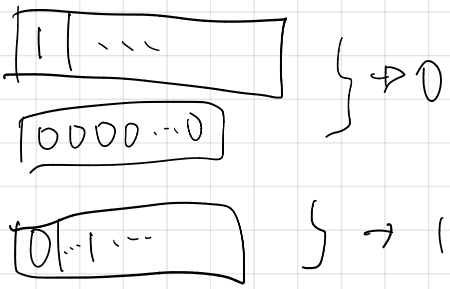
음수 나머지 0 → 오차 -16+8

~ 0001 ~

나머지 있으면 +1?
음수에서

-15	10001	-101 (-3)
-8	11000	-110 (-2)
-4	11100	-6 -1010
-2	11110	-3 -1101
-1	1111	-2 1110

isPositive



음수 거르기

0 거르기

or

양수 찾기

zero = 1
sign = 1

0001

0010 >> 3 ... 0000 !0000 = 1

1010 >> ... 1111 !1111 = 0

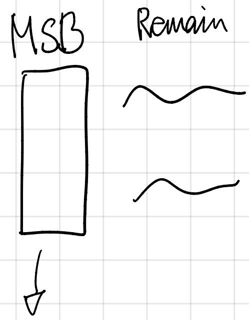
---

isLessOrEqual

X + (-Y)

X−Y > 0 → 0

X−Y ≤ 0 → 1

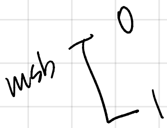


1
1 Remain ↑ big!

1 or 0
0 or 1 MSB 0 big!

0
0 Remain ↑ big!

MSB → XOR



n 01100

11101
10001

& 10001
| 11101

isLessOrEqual

4bit

x (-7) 1001

x ≤ y

y (-6) 1010

float_neg

1 ~

1 0000

∧ → 같은가?

~

1 0

1 1 1

0 1 0

1101

∧ 1000

0101

0000 0001

0111 1111 1000 ...

7 F 8

0x7F800000

0 0 0111 1111 1111

0 0 7 F F F F

nan → full exp

non zero frac

0000 0000

∧ 0 | 0000000

0 | 0000

divpwr2

4 bits

positive → 0b0110 >> 2 = 0b0010

6 >> 2 = 1

6 / 2^n = 1

negative → 0b11001001 >> n = 0b 111 11001

-55 >> n = -7

positive → x >> n

zero → x >> n

negative → 4.의 | x >> n

4.의 x | x >> n+1

1 1 1 1 1 1 1 1

-1

zero → 0b0000 >> n = 0b0000

0 >> n = 0

0 / 2^n = 0

$0 \leq y - x$

$0 > y - x$

xsign 0 -2^31 --- INT_MIN

ysign 1 2^31 - 1 --- INTMAX

diffsign 1

0 | 0&1

0 | (1 & 0)