1-1

+ , −

1 , 0 - binary bits

3 2 1 0
1, 2 9 9
↑ smallest
1×10^3

0 1 0 1 = B : index ⇒ bit
3 2 1 0
$B_2 = 1$

$U2(B) = 2^2 + 2^0 = 4 + 1 = 5$
$= \sum_{i=0}^{n} B_i \times 2^i$

0 111 = 7
sign 3bits of quantity

1111 = -7?
negtive

$N2(B) = (-1)^{B_n} \times \left(\sum_{i=0}^{n-1} B_i \times 2^i \right)$
sign-magnitude

0000 1000
+0 -0

Two's Complement

$TC2(B) = -B_n \times 2^n + \left(\sum_{i=0}^{n-1} B_i \times 2^i \right)$

0111 = 7 1111
$2^2 + 2^1 + 2^0 - 2^3$
7 - 8 = -1

0111
+1111
0110 = $2^2 + 2^1 = 6$

Two's : $-B_n \times 2^n$
Ones' : $-B_n \times (2^n - 1)$

1111 >> 1 = 0111
(n & (1 << x) > 0)
:= n's x bit is 1

C : 5 + x (4-11)

-2/ Bit-width is the length of B vector
(byte n = 8) (or octet)

				min size of	x86-32	x86-64
char	8	4	1	8	8	8
short	8			16	16	16
int	16			16	32	32
long	32			32	32	64
long long	64			64	64	64

uint8_t, uint16_t, sint32_t

addb	8
addw	16
addd	32
addq	64

sizeof (x) x = var, or type $x-1 < x < x+1$

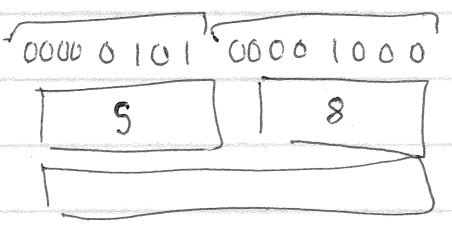
$a+b \bmod z^n$

```
for (int i=0 ; ; i++) {
  if (i < i+1) continue;
  printf ;
  exit
}
```

54, 389, 399, 127, 133 (32-bit) (32-bit)

\+ 10,000,000

Big Int



char, char

short

LSB is highest in memory => Big Endian

LSB is lowest => Little-endian