# Operators in Java

1. Arithmetic operators

   (+, −, *, /, %
   +=, −=, *=, /=, %=
   ++, −−)

2. Relational operators :

   ==, !=, <, >, <=, >=

```
int a = 5;
int b = 10;
```

Relational

`a == b;` Is a is equal to b?

`a = b;` assign value of b to a.

↳ Assignment

```
boolean c = (a==b);
System.out.println(c);
```

→ True

`a != b` ⇒ Is a not equal to b?

→ false

`c = (a != b);` ⇒ expressions

```
int c = (a < b);
```

↳ is a < b? → T / F

```
c = (a > b);
c = (a <= b);
```

↳ (is a < or = b);

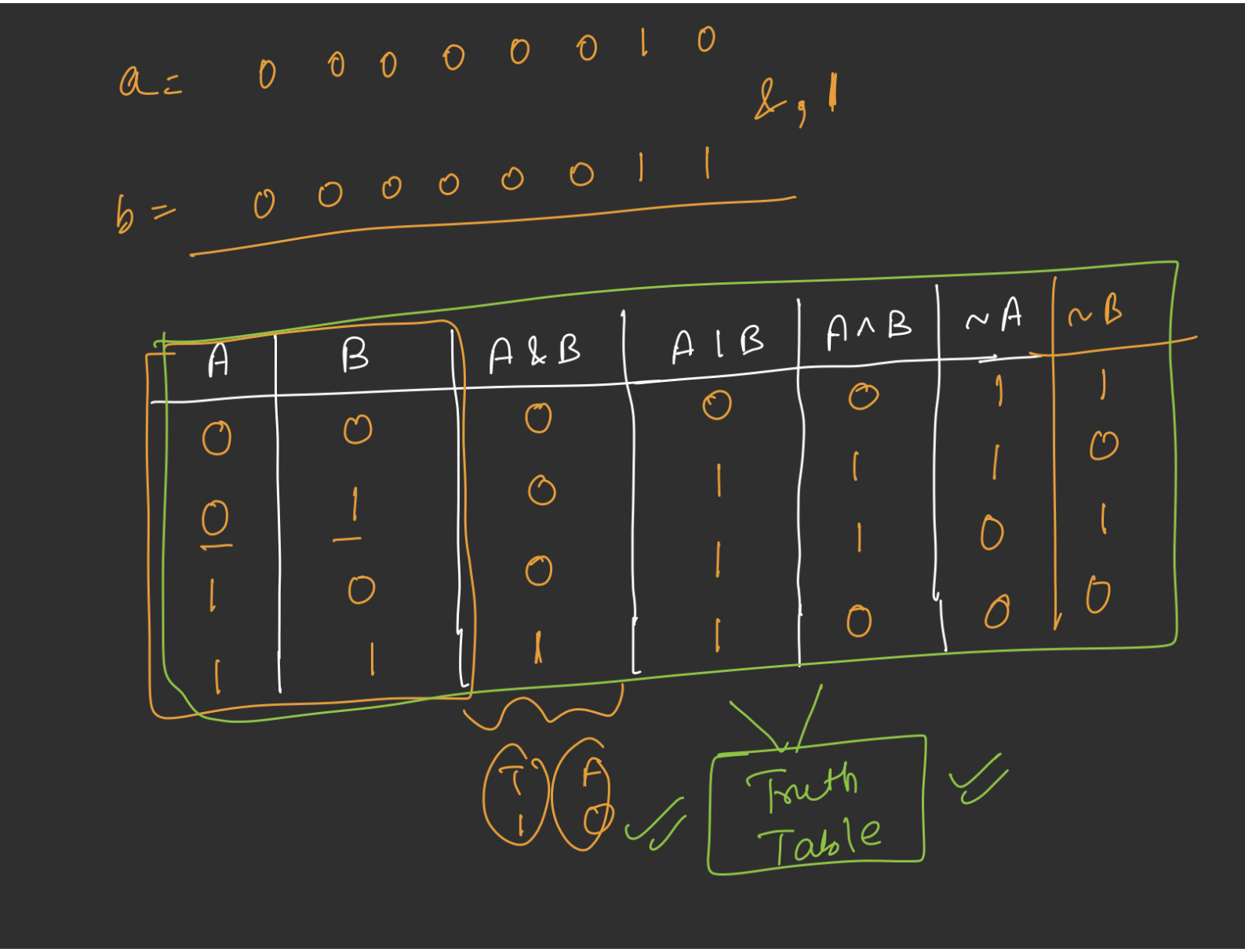
≥=

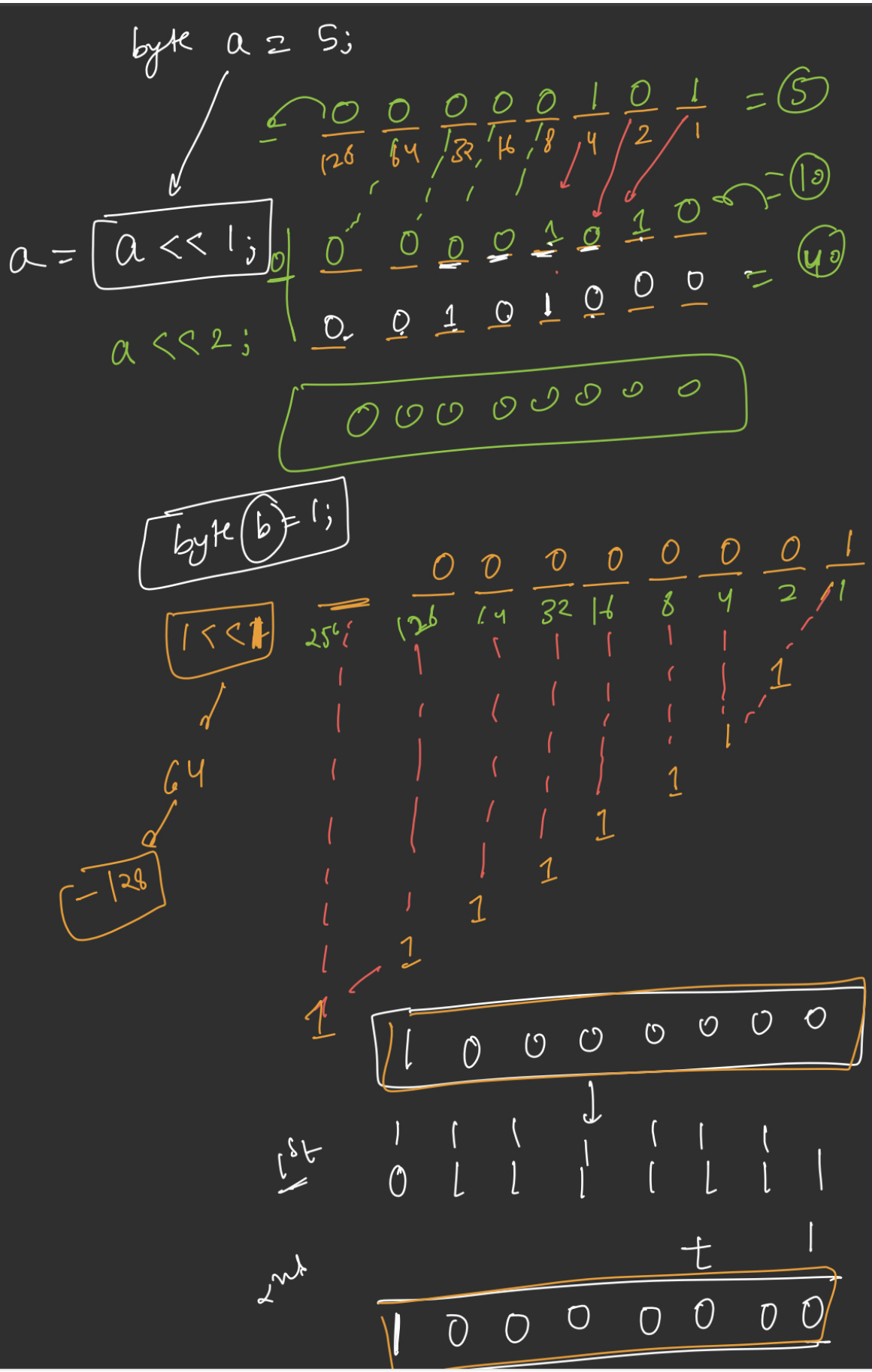
## Bitwise Operators

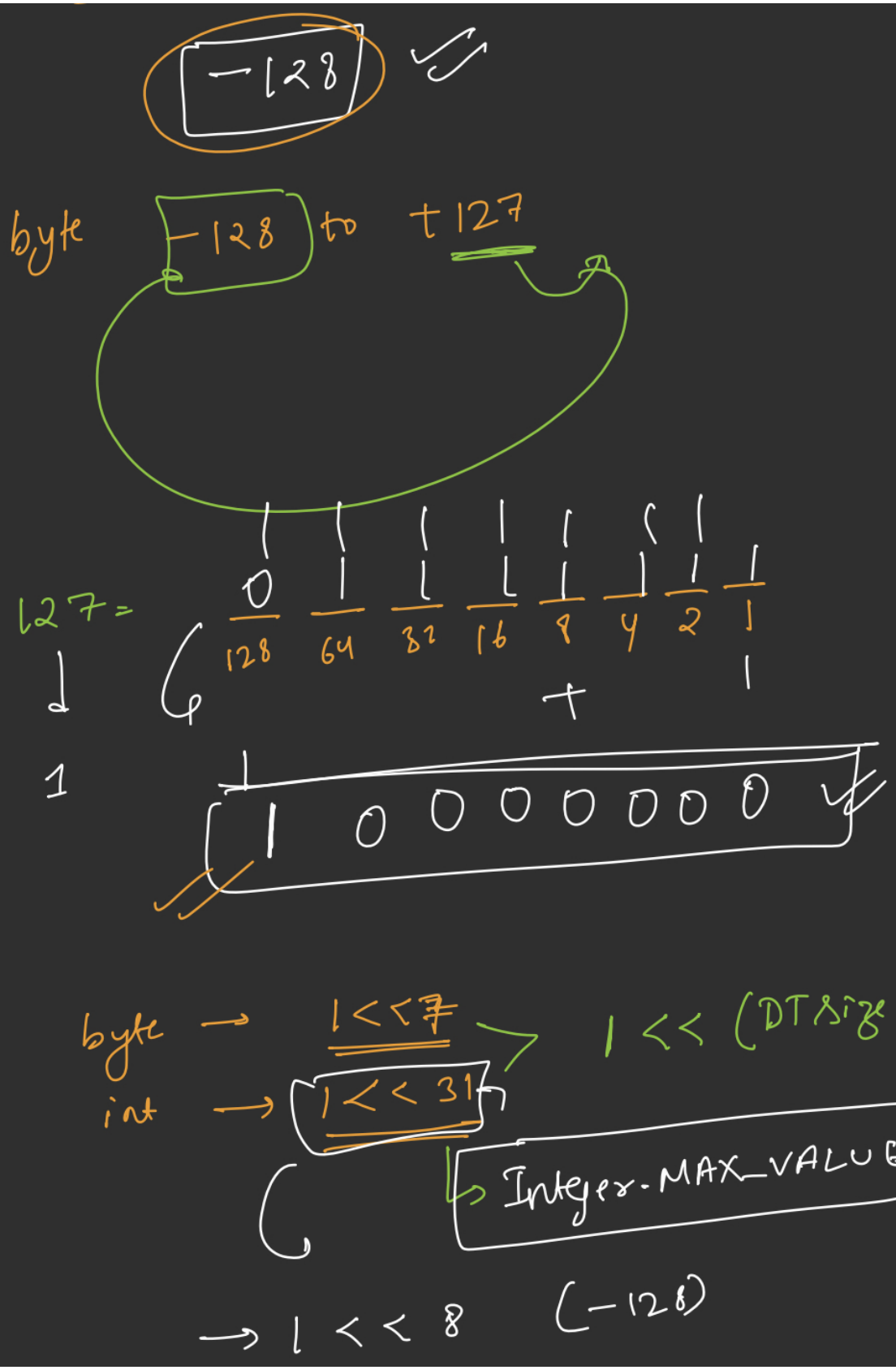
↳ Bit Manipulation

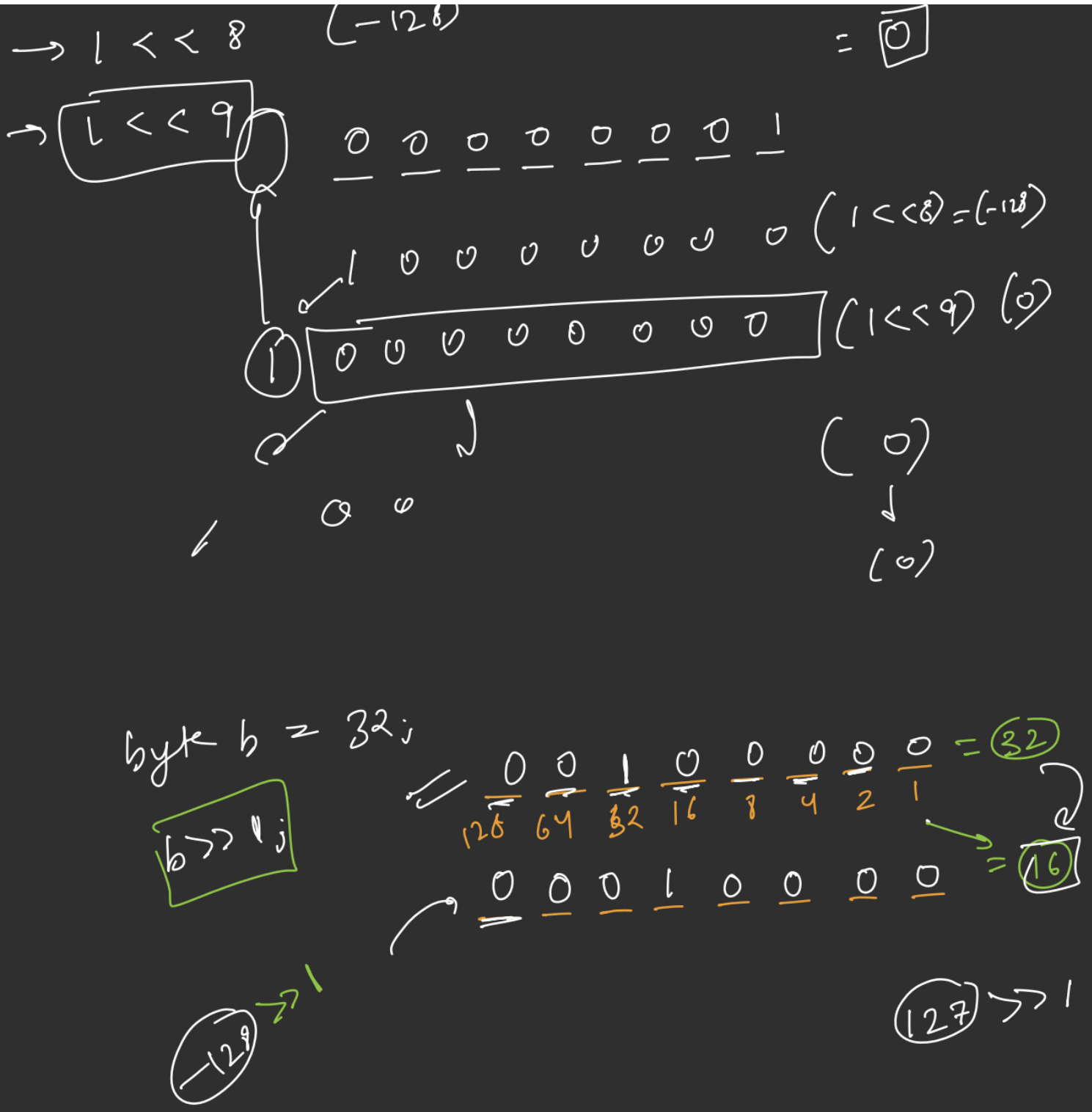
&, |, ^, ~, >>, <<,
>>>, &=, |=, ^=,
>>=, <<=, >>>=

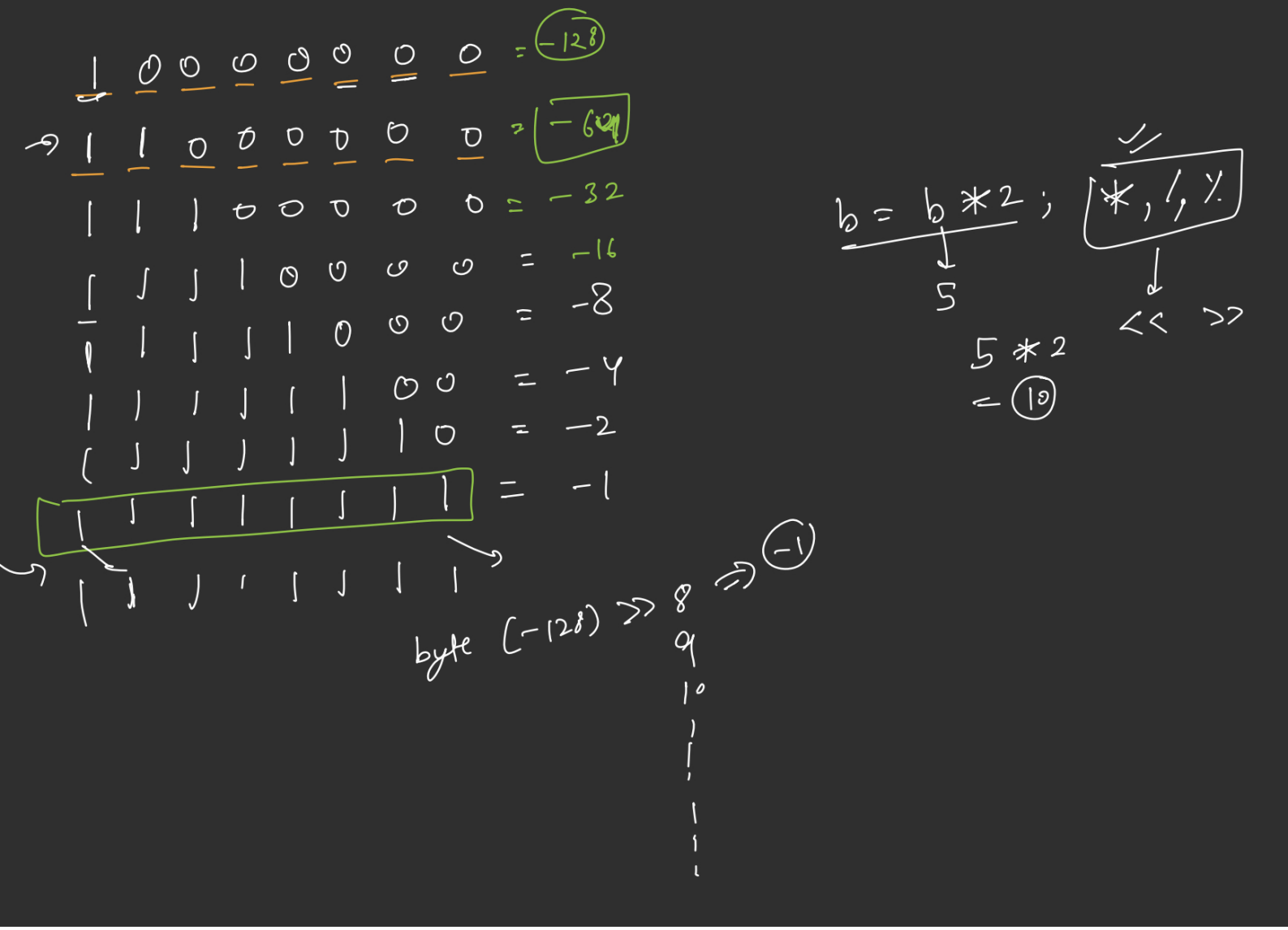
```
byte a = 2;
byte b = 3;
int c = a & b;
```

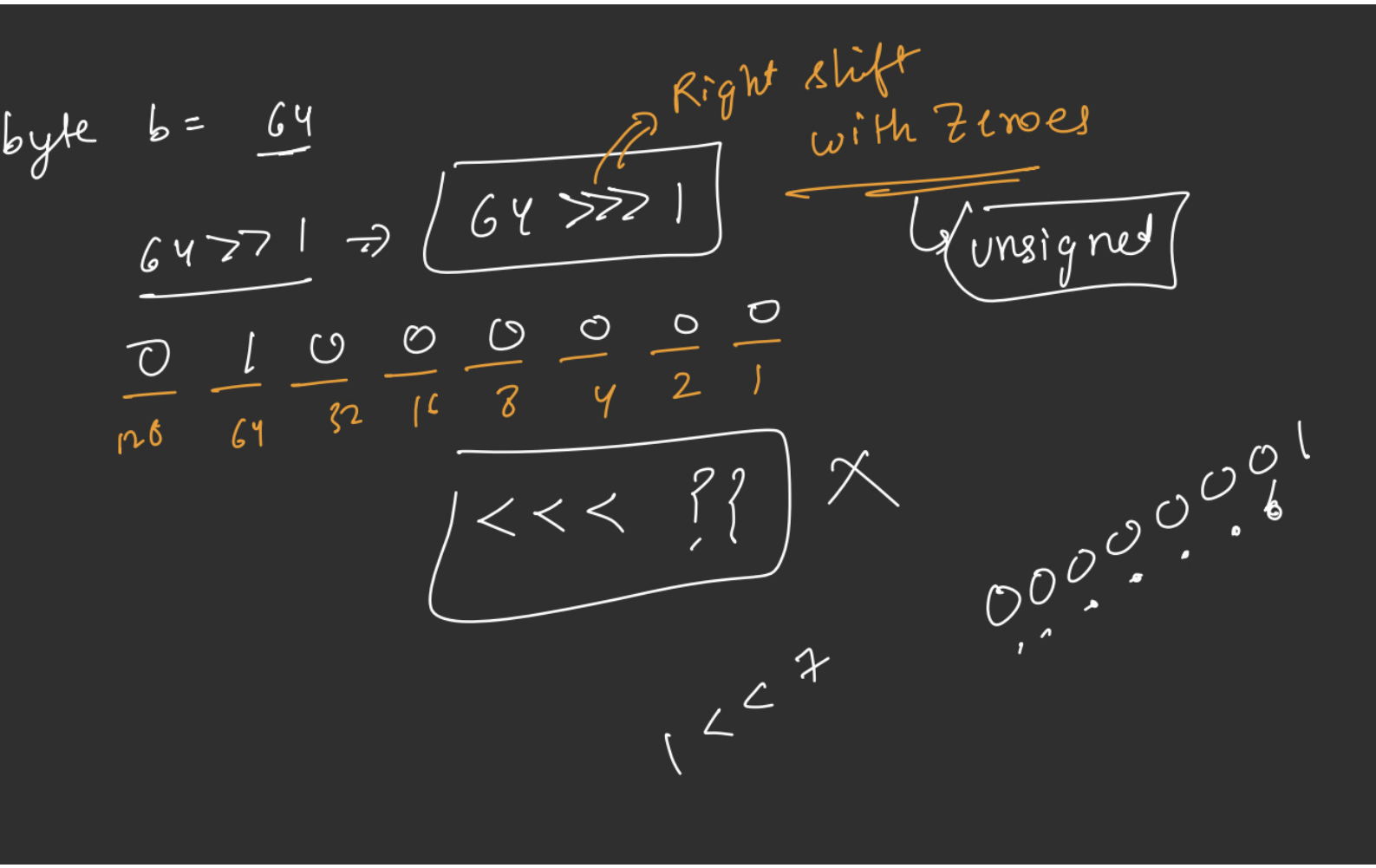

a = 0 0 0 0 0 0 1 0
& , |
b = 0 0 0 0 0 0 1 1
A | B | A&B | A|B | A^B | ~A | ~B
0 0 0 0 0 1 1
0 1 0 1 1 0 1
1 0 0 1 1 0 0
1 1 1 1 0 0 0
T 1
F 0
Truth Table

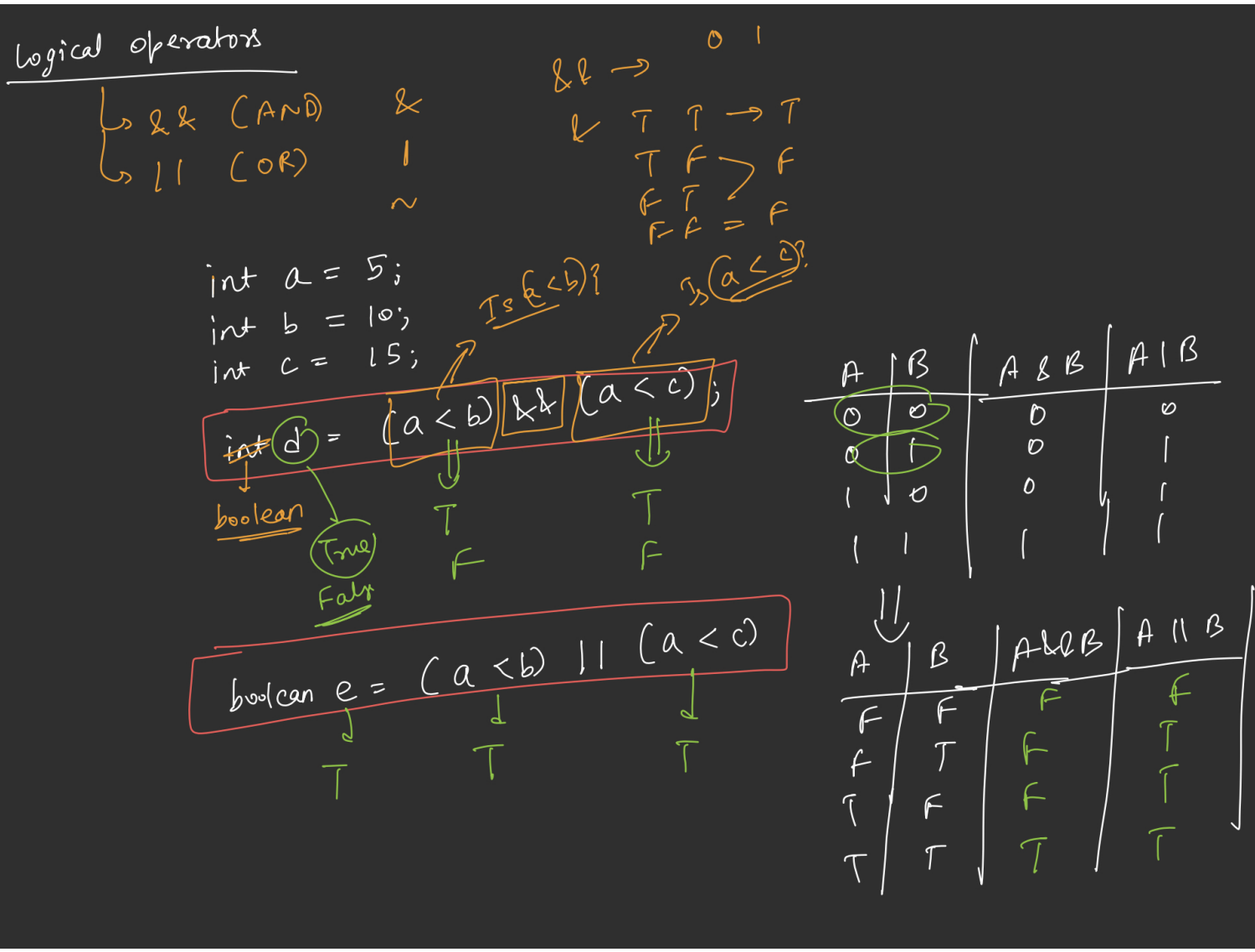

byte a = 5;
a = a << 1;
a << 2;
byte b = 1;
1 << 1
64
-128
256
1st
2nd


-128
byte -128 to +127
127 =
0 1 1 1 1 1 1 1
128 64 32 16 8 4 2 1
1
1 0 0 0 0 0 0 0
byte → 1<<7
int → 1<<31
1 << (DTSize -1)
Integer.MAX_VALUE ;
→ 1 << 8 (-128)


1 << 8
(-128)
= 0
1 << 9
0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0
(1<<8) = (-128)
1
0 0 0 0 0 0 0 0
(1<<9) (0)
(0)
(0)
byte b = 32;
b >> 1;
0 0 1 0 0 0 0 0 = 32
128 64 32 16 8 4 2 1
0 0 0 1 0 0 0 0 = 16
-128 >> 1
127 >> 1


1 0 0 0 0 0 0 0 = -128
1 1 0 0 0 0 0 0 = -64
1 1 1 0 0 0 0 0 = -32
1 1 1 1 0 0 0 0 = -16
1 1 1 1 1 0 0 0 = -8
1 1 1 1 1 1 0 0 = -4
1 1 1 1 1 1 1 0 = -2
1 1 1 1 1 1 1 1 = -1
byte (-128) >> 8 => (-1)
b = b * 2;
5
*, /, %
<< >>
5 * 2
= 10


byte b = 64
64 >> 1 ⇒
64 >>> 1
Right shift with Zeroes
unsigned
0 1 0 0 0 0 0 0
128 64 32 16 8 4 2 1
<<< ??
X
1 << 7
0 0 0 0 0 0 0 1
b




| A | B | A & B | A \| B |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 |

| A | B | A&&B | A \|\| B |
|---|---|---|---|
| F | F | F | F |
| F | T | F | T |
| T | F | F | T |
| T | T | T | T |

## Short-Circuit

↳ &&

| A | B | A && B | A \|\| B |
|---|---|---|---|
| F | F | F | F |
| F | T | F | T |
| T | F | F | T |
| T | T | | T |

A && B

A || B → True

T

## Bitwise op$^{n}$

↳ & |

```
int a = 5;
    b = 10;
    c = 15;
d = (a<b) & (a<c)
```

Does not do short-Circuiting

## Assignment operator

```
int a = b = c = 10;
```

BODMAS

## Operator precedence.

```
int c = a * b + d - b / g << 2;
```

| Highest | | | | | | |
|---|---|---|---|---|---|---|
| ++ (postfix) | – – (postfix) | | | | | |
| ++ (prefix) | – – (prefix) | ~ | ! | + (unary) | – (unary) | (*type-cast*) |
| * | / | % | | | | |
| + | – | | | | | |
| >> | >>> | << | | | | |
| > | >= | < | <= | instanceof | | |
| == | != | | | | | |
| & | | | | | | |
| ^ | | | | | | |
| \| | | | | | | |
| && | | | | | | |
| \|\| | | | | | | |
| ?: | | | | | | |
| -> | | | | | | |
| = | op= | | | | | |
| Lowest | | | | | | |

( ) → highest precedence.

```
int a = b + c * d;
```

```
int a = (b + c) * d;
```