Negative No.

```
byte b = -42;
```

byte ---------

8 bit

-128 to +127

① +42 →

0	0	1	0	1	0	1	0
128	64	32	16	8	4	2	1

00101010 = (42)

2	42	
2	21	0
2	10	1
2	5	0
2	2	1
	(1)	0

```
byte b = 42;
System.out.println(b);
```

→ 00101010

→ 00101010 (b)

= (42) ✓

---

byte b = -42;

2's Complement

① -42 = 00101010

= 11010101

1's t Complement → 11010101

↓

2nd Complement = 1's t C + 1

11010101

\+ 1

11010110

byte b = -42;

↳ 11010110

↳ 11010110

b

System.out.println(b);

MSB → 1 1 0 1 0 1 1 0 ← LSB

MSB = 1 ⇒ -ve No.

MSB = 0 ⇒ +ve No.

byte = 0 1 1 1 1 1 1 1

128 64 32 16 8 4 2 1

-128 to +127

-127 ≠ 127

-128 to 0 to 1

10000000

(0-9) +1

(1)9 + 1 = 10

MSB (2) LSB (1)

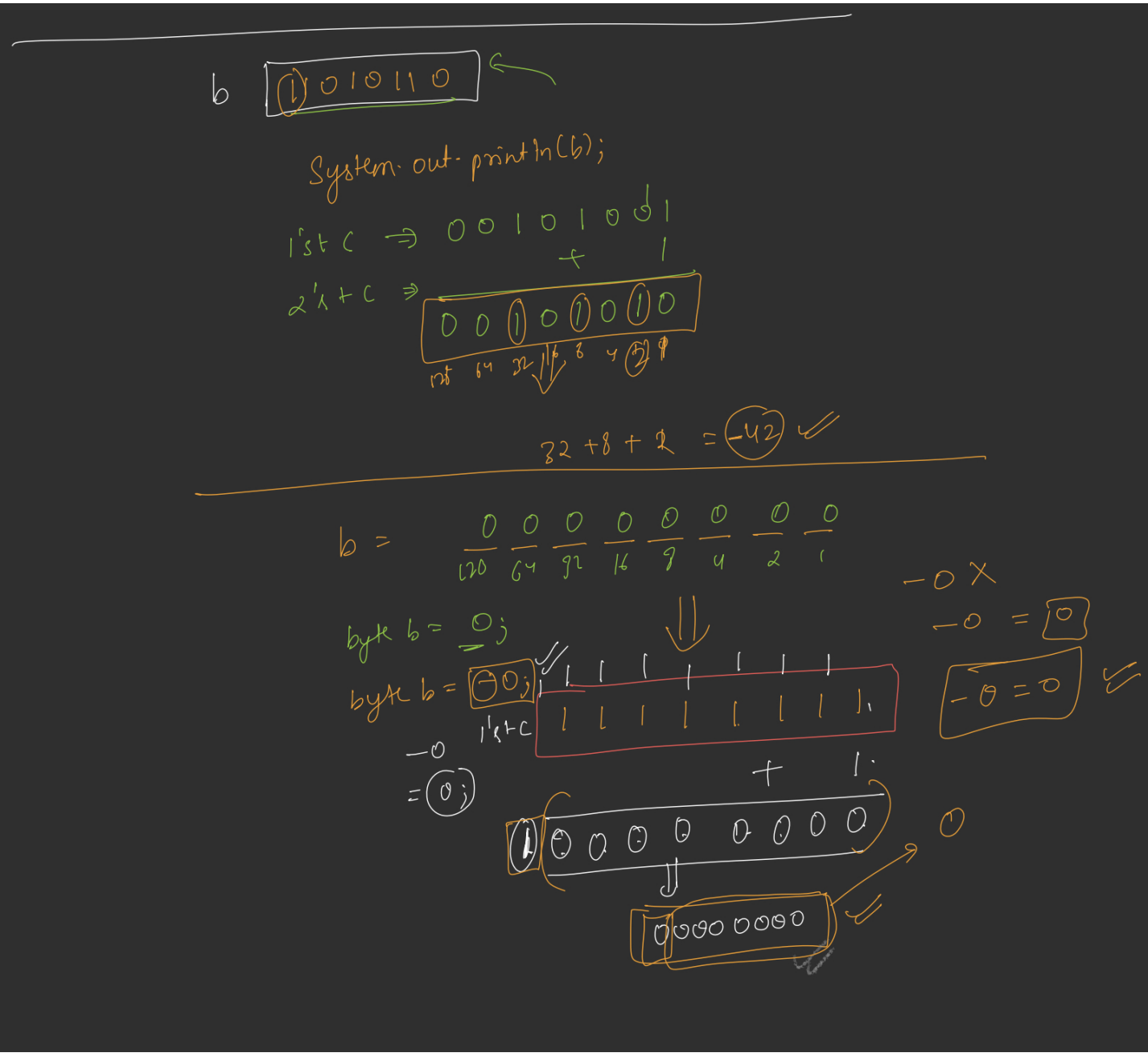
(2)3 2 (1) +1

2 3 2 2

(2) 3 2 1 +1

1000

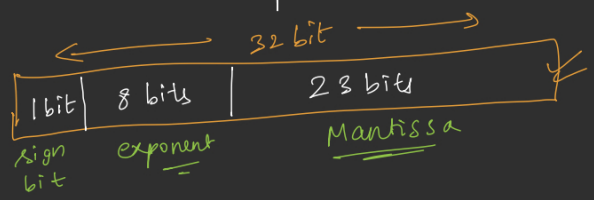
3 3 2 1


b 1 0 1 0 1 1 0
System.out.println(b);
1's t c ⇒ 0 0 1 0 1 0 0 1
+ 1
2's t c ⇒ 0 0 1 0 1 0 1 0
128 64 32 16 8 4 2 1
32 + 8 + 2 = -42
b = 0 0 0 0 0 0 0 0
120 64 32 16 8 4 2 1
byte b = 0;
byte b = 00;
1's t c
-0
= 0;
-0 X
-0 = 0
-0 = 0
+ 1
1 0 0 0 0 0 0 0
0
0000 0000

Floating No.

float f = 8.125 f;

float f = 0.7 f;



f = 8.125

① 1000.001 (Find Binary of above No.)

0.125 × 2 = 0.25 → 0
0.25 × 2 = 0.5 → 0
0.5 × 2 = 1.0 → 1

② Make it in the form of —

$(1.xx) * 2^{exp}$

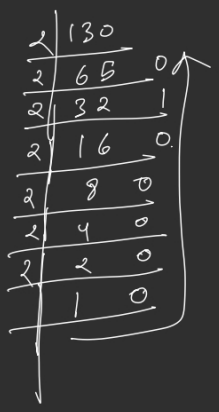
⇒ $1.000001 * 2^{3}$

③ Add Bias to the exponent.

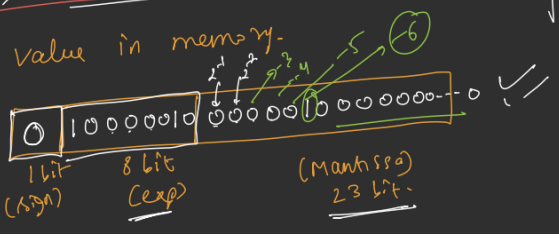
for float bias = 127

⇒ 3 + 127 = 130

exp ⇒ 10000010



④ Place value in memory.



System.out.println(f);

$$(-1)^{sign} * (1 + Mantissa) * 2^{exp-Bias}$$

$$\Rightarrow (-1)^0 * \left(1 + \frac{\quad}{\quad}\right) * 2^3$$

$$\Rightarrow \left(1 + \frac{}{2^6}\right) * 8$$

$$= \left(1 + \frac{1}{26}\right) * 8$$

$$= \left(1 + \frac{1}{64}\right) * 8 \Rightarrow$$

$$\Rightarrow (1 + 0.015625)$$

$$= (1.015625) * 8$$

$$= 8.125$$

$$(130)_{10} = (10000010)_2$$

$$\Rightarrow 130 - 127 = 3$$

float b = 0.7 f;

① Express to Binary.

0.7

0 . 10110011 0110 ----

0.7 × 2 = 1.4 → ①
0.4 × 2 = 0.8 → 0
0.8 × 2 = 1.6 → 1
0.6 × 2 = 1.2 → 1
0.2 × 2 = 0.4 → 0
0.4 × 2 = 0.8 → 0
0.8 · 2 = 1.6 → 1
0.6 × 2 = 1.2 → 1
0.2
!
!

② $(1.xx * 2^{exp})$

$(1.0110\ 0110\text{---}) * 2^{-1}$

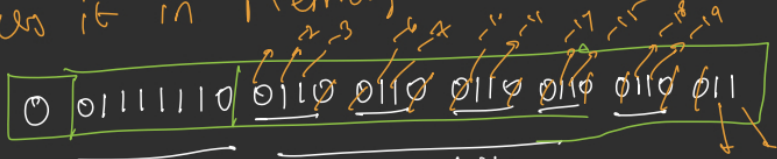
③ Add Bias ⇒ 127

(-1 + 127) = 126

= 0 1 1 1 1 1 1 0

2	126
2	63 0
2	31 1
2	15 1
2	7 1
2	3 1
	1 1

④ Express it in Memory.



23 bit

System.out.println(f);

$$(-1)^S * (1 + M) * 2^{e-B}$$

$$= (-1)^0 * \left(1 + \frac{\quad}{\quad}\right) * 2^{126-127}$$

$$= \left(1 + \frac{\quad}{\quad}\right) * \frac{1}{2}$$

$$= \left(1 + \frac{1}{2^2} + \frac{1}{2^3} \; \frac{1}{2^6} + \frac{1}{2^7} + \frac{1}{2^{10}} + \frac{1}{2^{11}} + \frac{1}{2^{14}} + \frac{1}{2^{15}} + \frac{1}{2^{18}} + \frac{1}{2^{19}} + \frac{1}{2^{22}} + \frac{1}{2^{23}}\right) * \frac{1}{2}$$

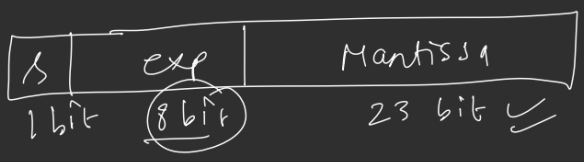
$$\Rightarrow \frac{1.3999999761581420898437 5}{2}$$

$$0.69999998807907104421875$$

$$\neq 0.7 \approx \boxed{0.7} \lesssim \boxed{0.7b}$$

$20 \ldots$

(1) Bias ?? Bias = 127



32 bit → IEEE

0.7f = $(2)^{(-1)}$ = $(2)^{(+127)}$

exp = 8 bit

-127 + 127 = 0

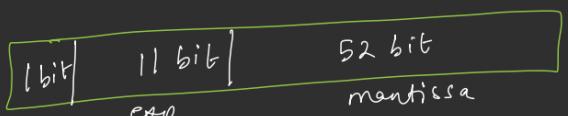
0 ---- 255 = 2^8 = 256

+127

$(2^{8-1}-1) = (2^7-1) : (128-1) = 127$

$(2^{exp-1}-1) = ??$

double d = 32.4186;



$(2^{11-1}-1) = (2^{10}-1) = 1023$

Float → 0.7f

double

BigDecimal

↳ 0.7 → 0.7