CS & IT ENGINEERING

Programming in C

Chapter-1

Data types and Operators

Lec- 06

By- Pankaj Sharma sir

Operators-I

Number System

1) Decimal number system (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

2) Binary number system (0, 1)

3) Octal number system (0-7)

4) Hexadecimal number system (0–9, A, B, C, D, E, F)

Decimal number System

$$2\ 6 = 2\times10^{1} + 6\times10^{0}$$

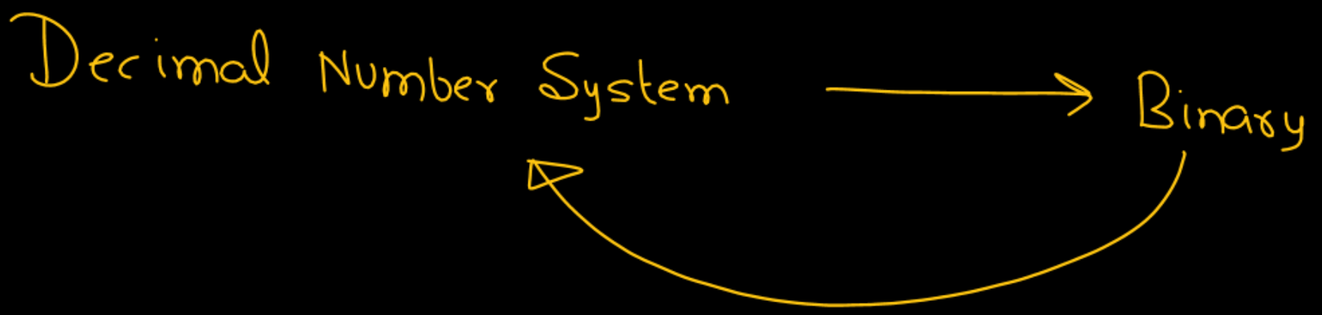
$$2\ 6\ 4 = 2\times10^{2} + 6\times10^{1} + 4\times10^{0}$$

$$= (2\times10^{1} + 6\times10^{0})\times10 + 4$$

New value = old value × 10 + 4

10 → Decimal

Binary Number System (0,1)



Decimal ⟶ Binary

$(67)_{10} \longrightarrow (\quad)_2$

$(1000011)_2$

2	67	Rem
2	33	1
2	16	1
2	8	0
2	4	0
2	2	0
2	1	0
	0	1

Stop

$2\overline{)67}\ 33$ ← Q

66

1 ← rem

$2\overline{)1}\ 0$

0

1

$(121)_{10} = (\quad)_2$

$= (1111001)_2$

2	121	Rem
2	60	1
2	30	0
2	15	0
2	7	1
2	3	1
2	1	1
	0	1

stop

Binary $\longrightarrow$ Decimal

(2) (10)

$1\ 0\ 0\ 0\ 0\ 1\ 1$

$2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

$= 1\times2^6 + 0\times2^5 + 0\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$

$= 64 + 0 + 0 + 0 + 0 + 2 + 1$

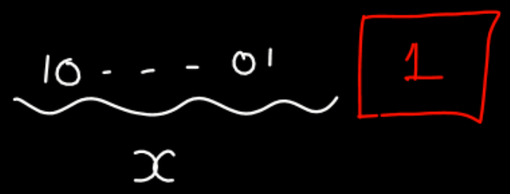
$= 67$

$$
\begin{array}{c}
1\,1\,1\,1\,0\,0\,1 \\
2^6\,2^5\,2^4\,2^3\,2^2\,2^1\,2^0
\end{array}
= 1\times2^6 + 1\times2^5 + 1\times2^4 + 1\times2^3 + 0 + 0 + 1\times2^0
$$

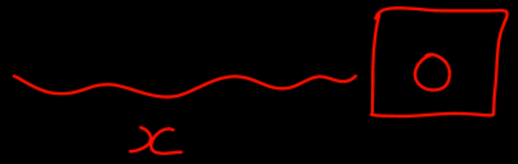
$$= 64 + 32 + 16 + 8 + 0 + 0 + 1$$

$$= 96 + 24 + 1$$

$$= 121$$



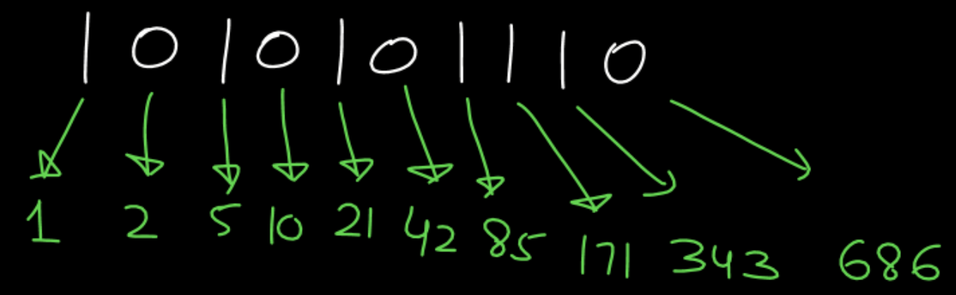
$= 2 \times x + 1 \Rightarrow$ odd



$= 2 \times x + 0 = 2x$

(i) odd $\Rightarrow$ 

(ii) Even $\Rightarrow$ 


1 0 1 0 1 0 1 1 1 0
1
2
5
10
21
42
85
171
343
686

Bitwise Operator

→ Binary values

```
int a = 7, b = 11, c;
c = a & b;
printf("%d", c);
```

③

0 & 0 = 0
0 & 1 = 0
1 & 0 = 0
1 & 1 = 1

(i) Bitwise AND (&)

```
a    0 0 0 0  0 0 0 0  0 0 0 0  0 1 1 1
b    0 0 0 0  0 0 0 0  0 0 0 0  1 0 1 1
-------------------------------------------
     0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 1
```

$2^1 2^0$ → 3

$= 3$

(ii) Bitwise OR (|)

```
int a = 9, b = 17, c;
c = a | b;
printf("%d", c);
```

0 | 0 = 0

0 | 1 = 1

1 | 0 = 1

1 | 1 = 1

```
  0000 0000 0000 1001
  0000 0000 0001 0001
-------------------------
  0000 0000 0001 1001
-------------------------
```

25

(iii) Bitwise XOR (^)

```
int a = 5, b = 9, c;
c = a ^ b;
printf("%d", c);
```

0 ^ 0 = 0
0 ^ 1 = 1
1 ^ 0 = 1
1 ^ 1 = 0

both operands/bits are same

```
  0000 0000 0000 0101
  0000 0000 0000 1001
  ---------------------
  0000 0000 0000 1100
  ---------------------
```

12

$0 \wedge a = a$

$a \wedge a = 0$

$a \wedge a \wedge a = a$

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

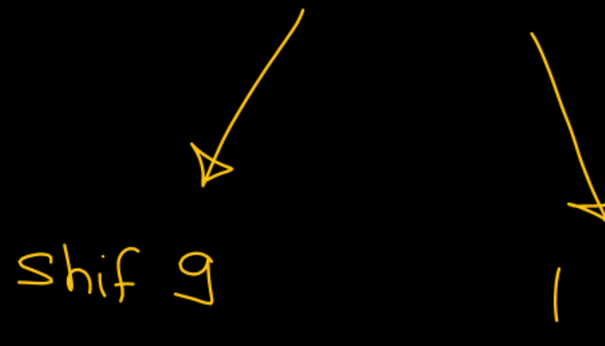
Even no. of times XOR the same no. $\Rightarrow 0$

Odd no. of times $\Rightarrow a$

Bitwise Left-shift (<<)

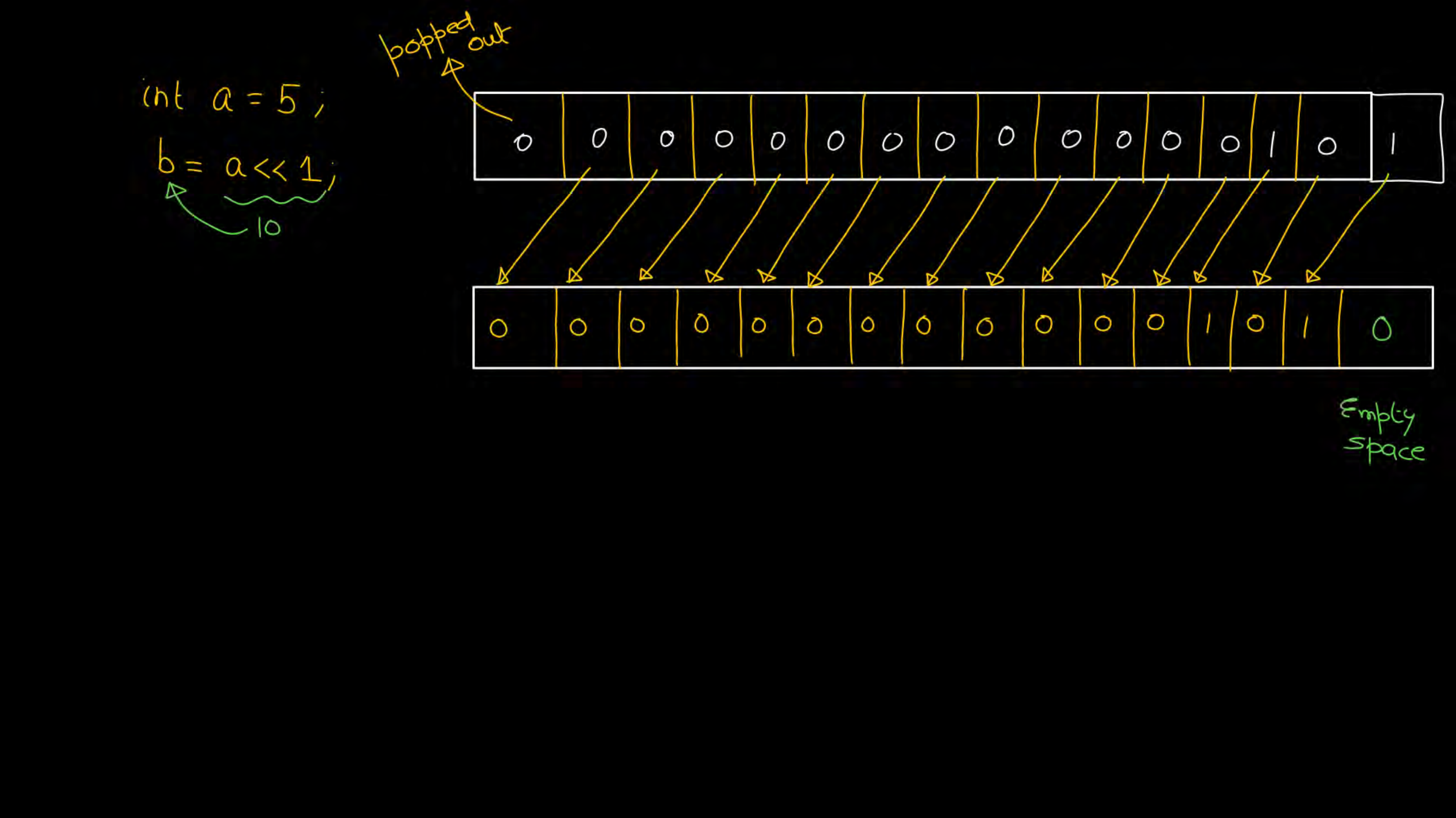
* Binary operator

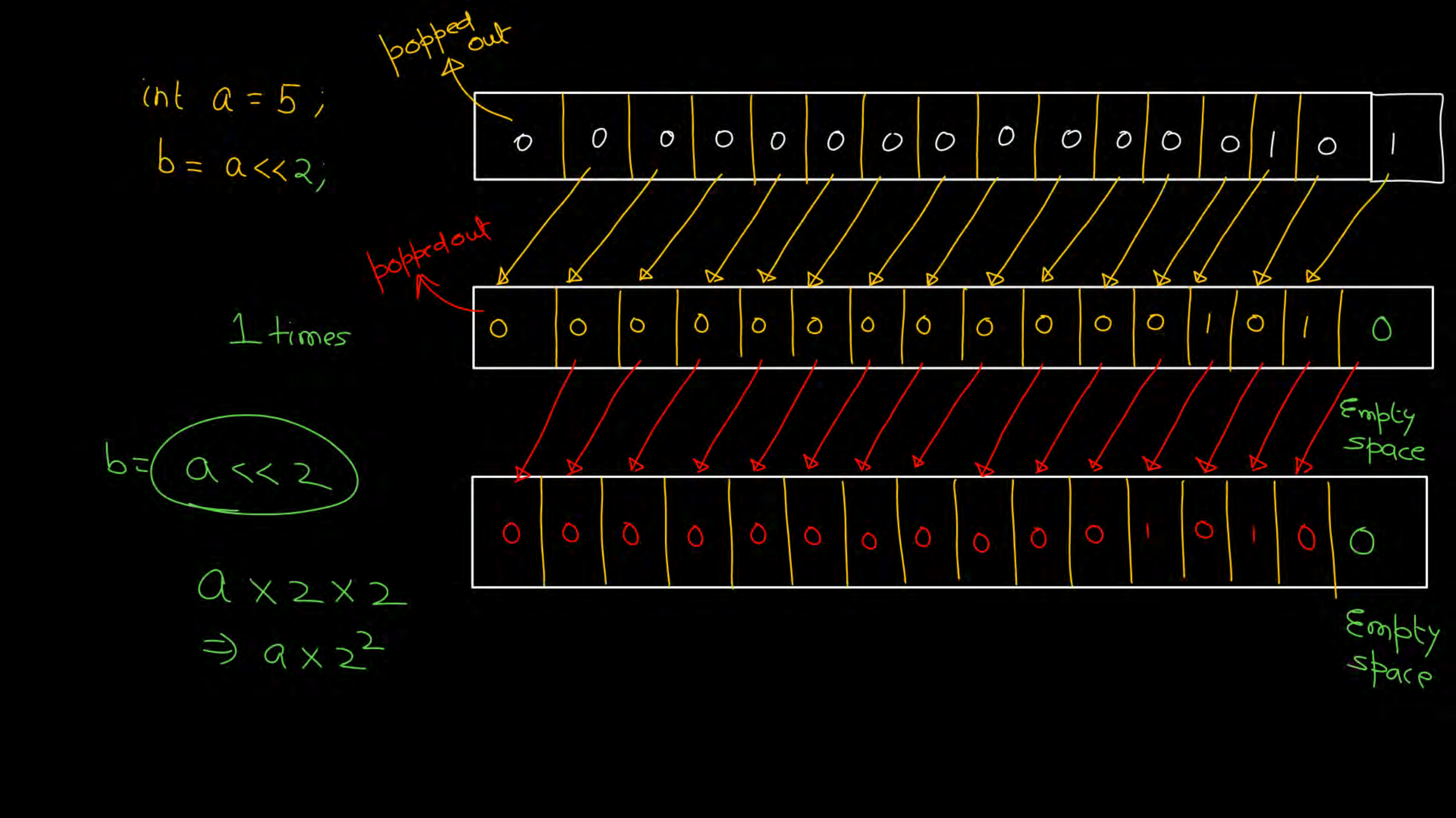
9 << 1



17 << 2

left Shift 17 ⇒ 2 times


int a = 5;
b = a << 1;
10
popped out
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
Empty space


int a = 5;
b = a<<2;
popped out
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
popped out
0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
1 times
Empty space
b = a<<2
0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0
a x 2 x 2
⇒ a x 2^2
Empty space

int a = 10;

In general

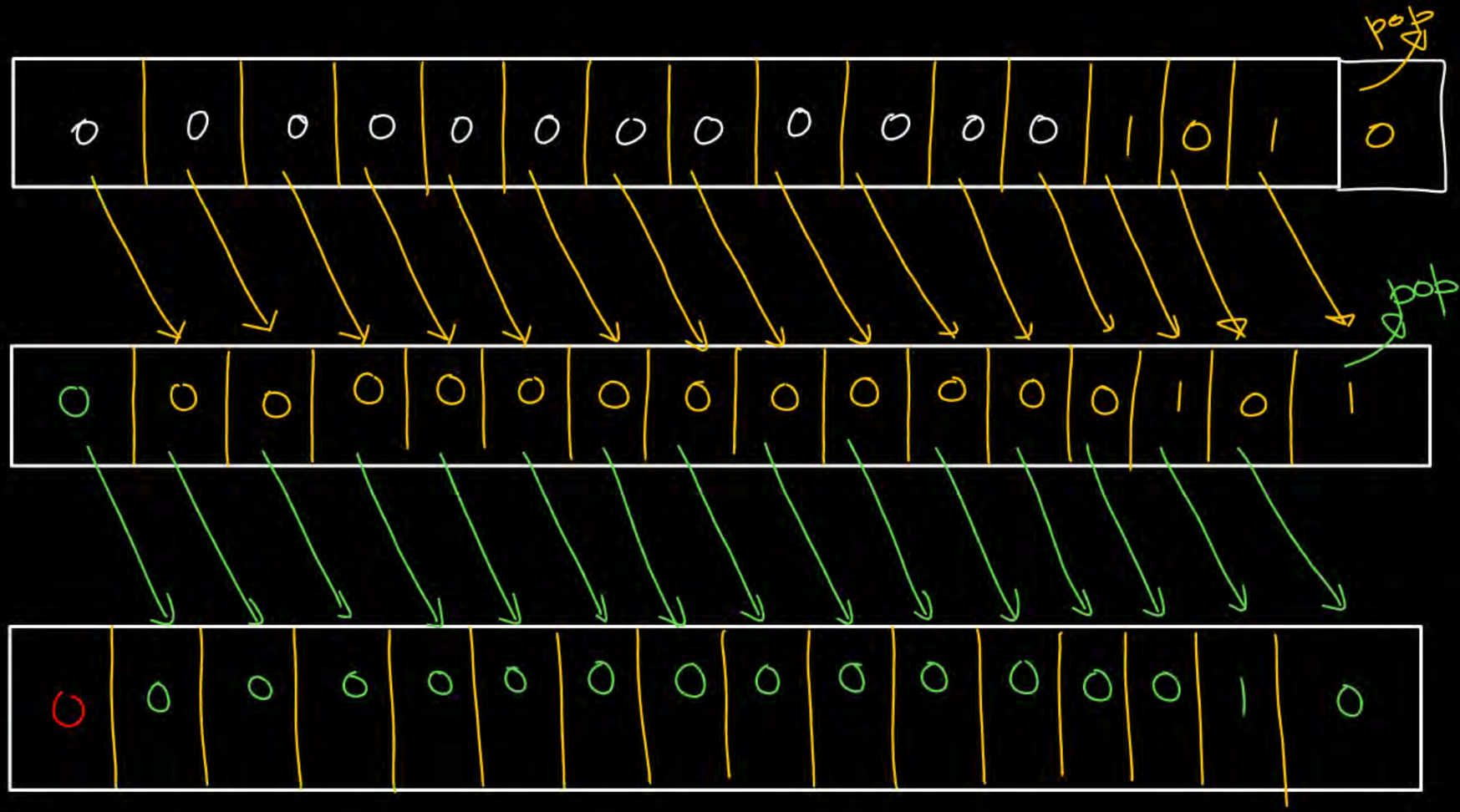
a >> 2

⇒ a / 2^2

10

5

2



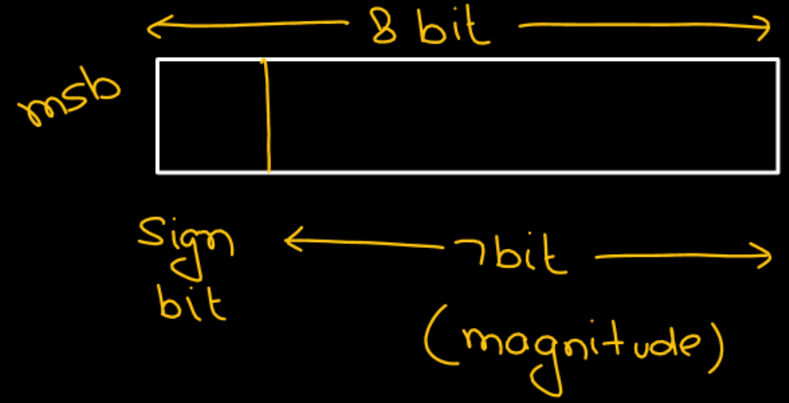
-ve Number

Sign

\+ → 0

\- → 1

① Sign magnitude



+11 ⇒ 0 | 0001011

-11 ⇒ 1 | 0001011

Problem:

0 0 0 0 0 0 0 0

1 0 0 0 0 0 0 0

+0

-0

Confusion

2's complementation

+ve ⇒ As it is

-ve ⇒ 2's complementation.

1's complement ⇒ flip 0,1

0 1 0 0 1 0 1

1's comp 1 0 1 1 0 1 0

2's comp : 1's comp + 1

0 1 0 1 0 0 1

1's comp: 1 0 1 0 1 1 0

2's com +1

1 0 1 0 1 1 1

direct

0 1 0 1 0 0 1

1 0 1 0 1 1 1

1+0 = 1

0+1 = 1

0+0 = 0

1+1 = 10 0

Carry

+ve ⇒ binary

-ve ⇒ 2's comp.

-11 ⇒ a) +11 ⇒ 00001011

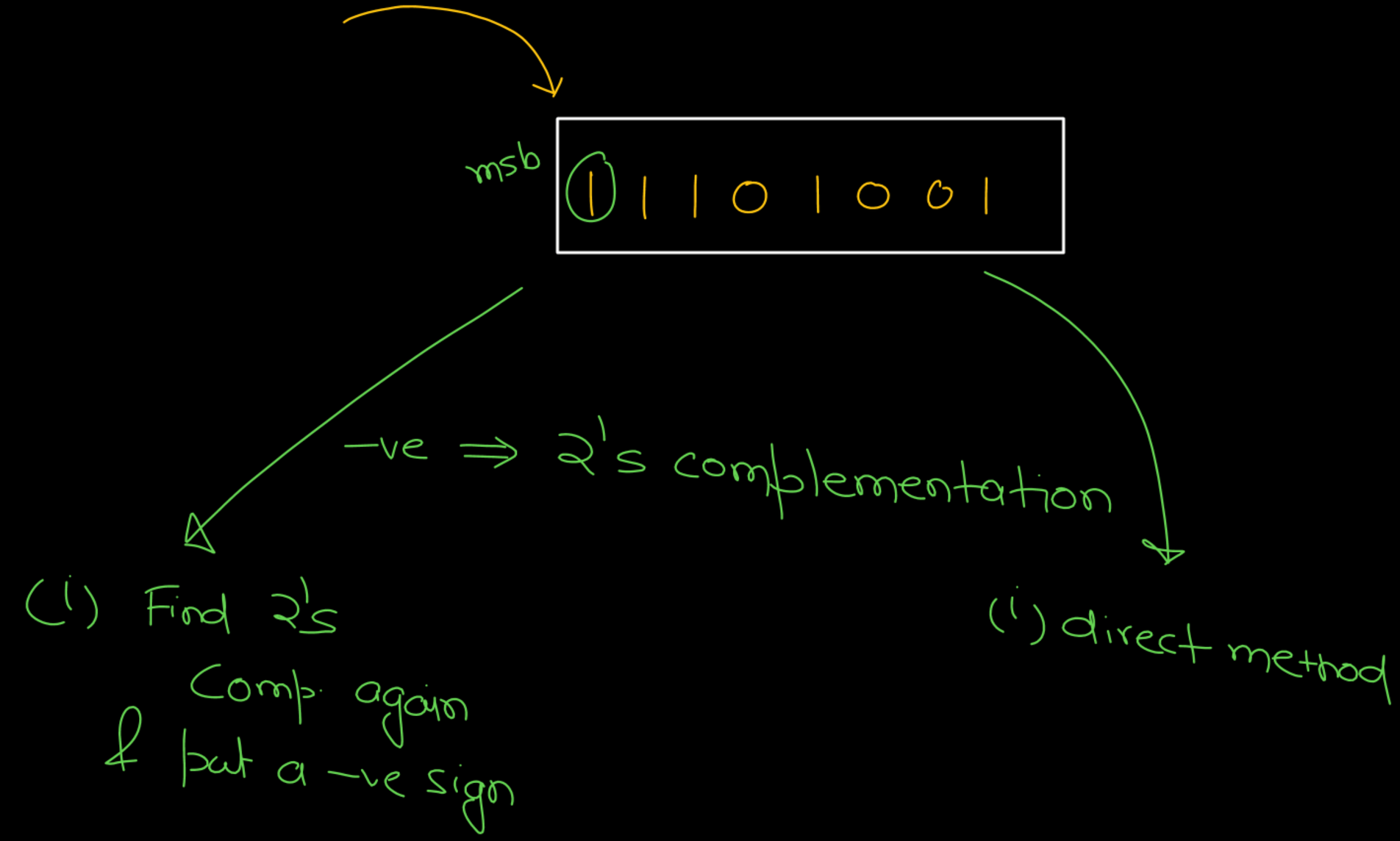
b) 2's comp ⇒ 11110101

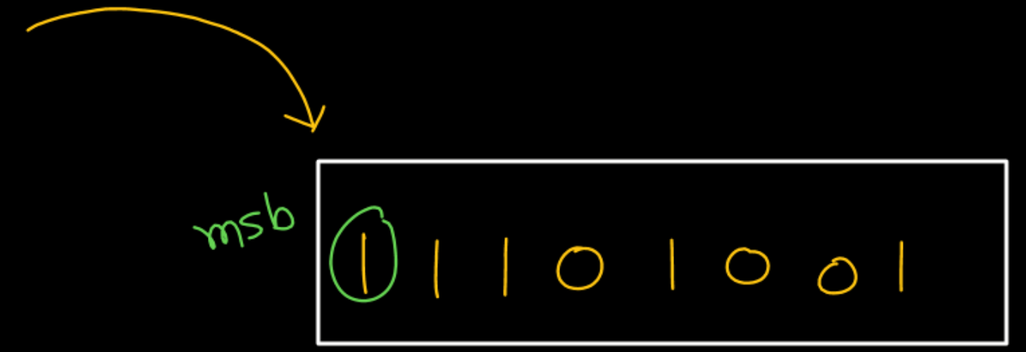
$-23 \Rightarrow$ 00010111

11101001

-23

11101001

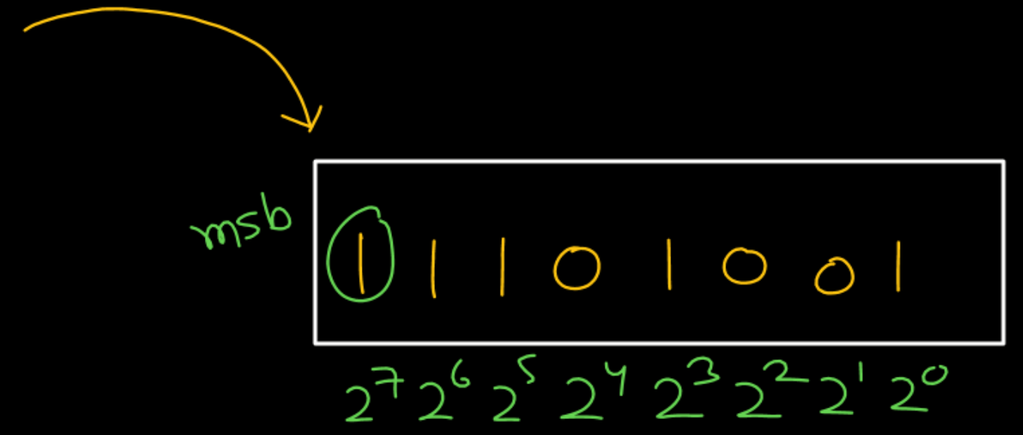

msb
1 1 1 0 1 0 0 1
-ve ⇒ 2's complementation
(i) Find 2's Comp. again & put a -ve sign
(i) direct method



(i) 1 1 1 0 1 0 0 1

0 0 0 1 0 1 1 1

-23



(i) 1 1 1 0 1 0 0 1

0 0 0 1 0 1 1 1

-23

$-2^4 - 2^2 - 2^1 - 1$

$-16 - 4 - 2 - 1$

$= -23$

2's comp

$$1\,0\,0\,1\,0\,0\,0\,1$$
$$2^7\,2^6\,2^5\,2^4\,2^3\,2^2\,2^1\,2^0$$

$\Rightarrow$

$$-2^6 - 2^5 - 2^3 - 2^2 - 2^1 - 1$$

$$= -64 - 32 - 8 - 4 - 2 - 1$$

$$= -96 - 15$$

$$= -111$$

$$\sim x = -(x+1)$$

bitwise not ($\sim$)

$$x = 2^5 + 2^4$$

$3^2 16$

$5\ 4 \qquad 3\ 2\ 1\ 0$

$2^5\ 2^4 \qquad 2^3\ 2^2\ 2^1\ 2^0$

$\sim$: bits flip

a

0000 0000 0011 0000

```
int a = 48;
b = ~a;
pf("%d", b);
```

$\sim a$

$2^5\ 2^4$

1111 1111 1100 1111

2's comp

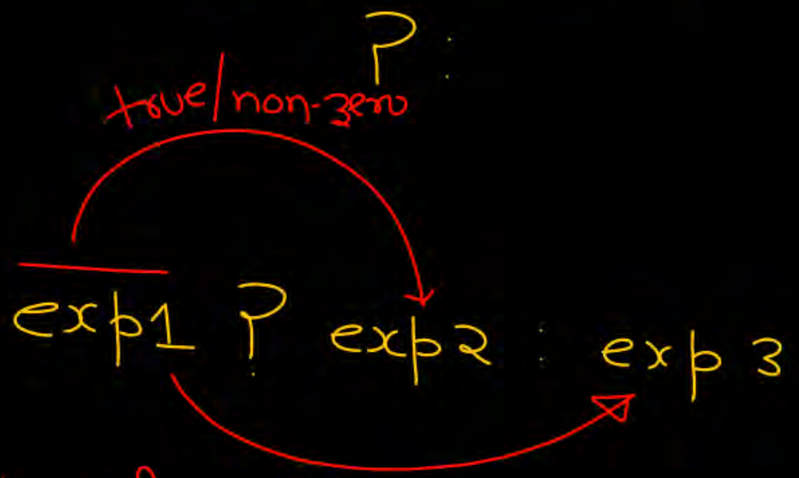
$$-2^5 - 2^4 - 1$$

$$= -(2^5 + 2^4 + 1)$$

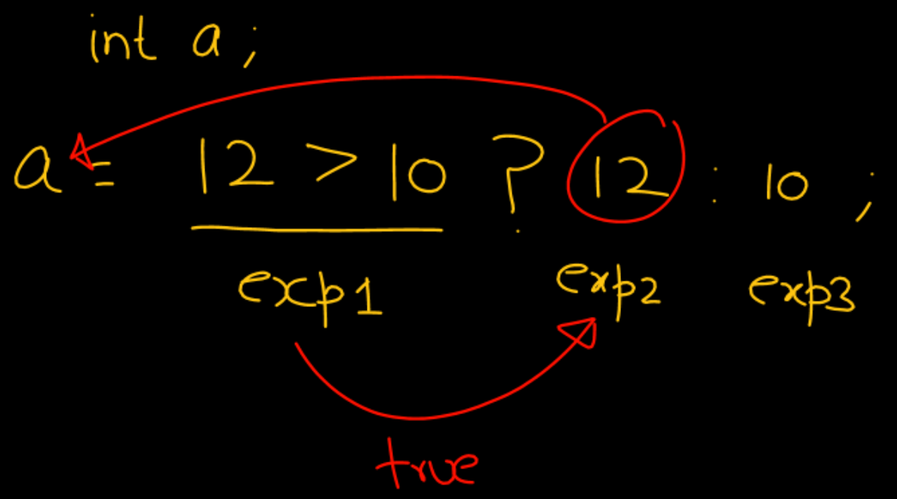
$$= -(x+1)$$

$$-(48+1) = -49$$

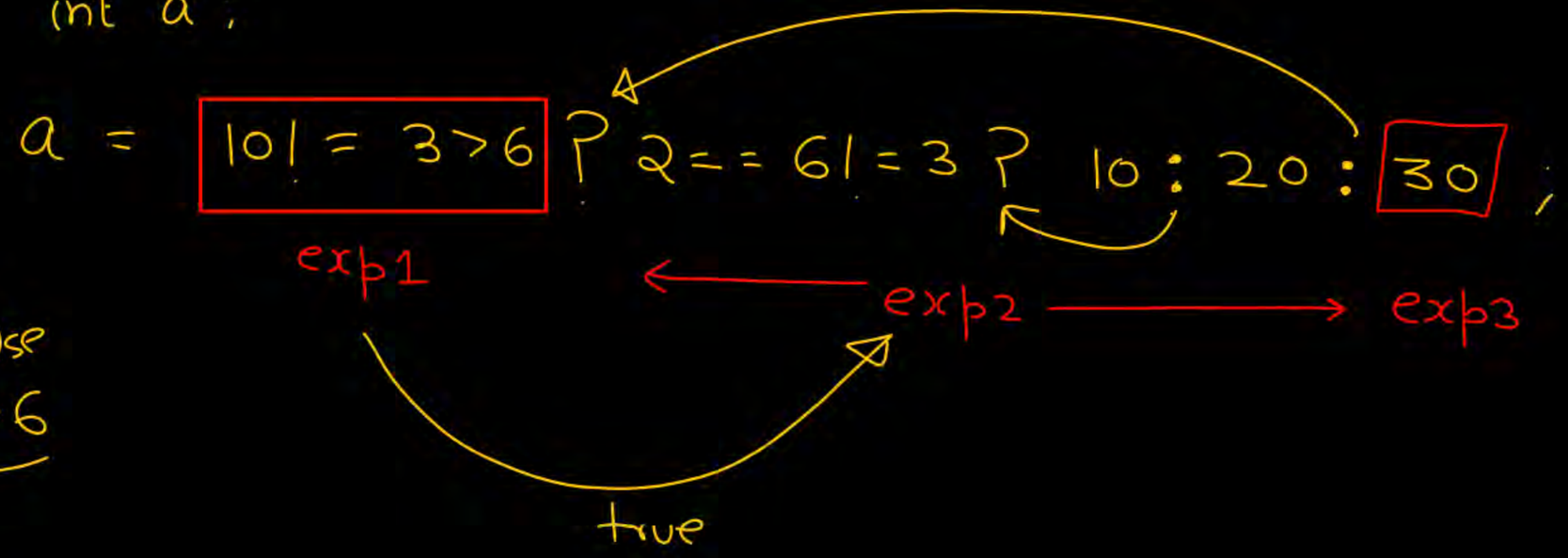
Ternary operator



Exp1 is eval first.
If it is non-zero (true), then value of entire exp is exp2
Otherwise, value of entire exp is exp3


int a ;
a = 12 > 10 ? 12 : 10 ;
exp1
exp2
exp3
true

>, <, <=, >= int a;

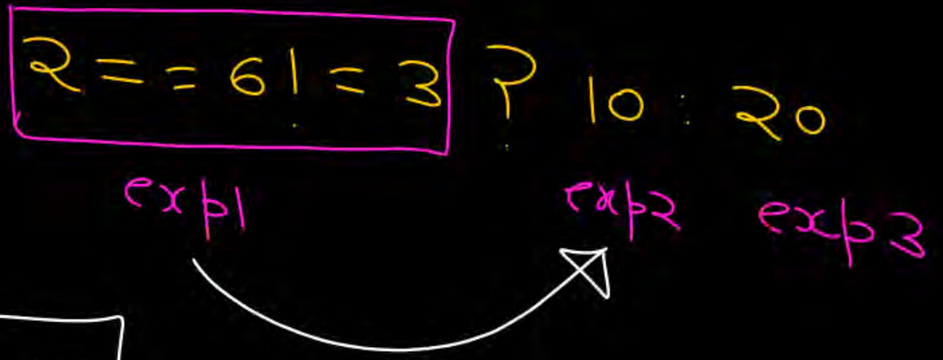


10 != 3>6

false

10 != 0

true

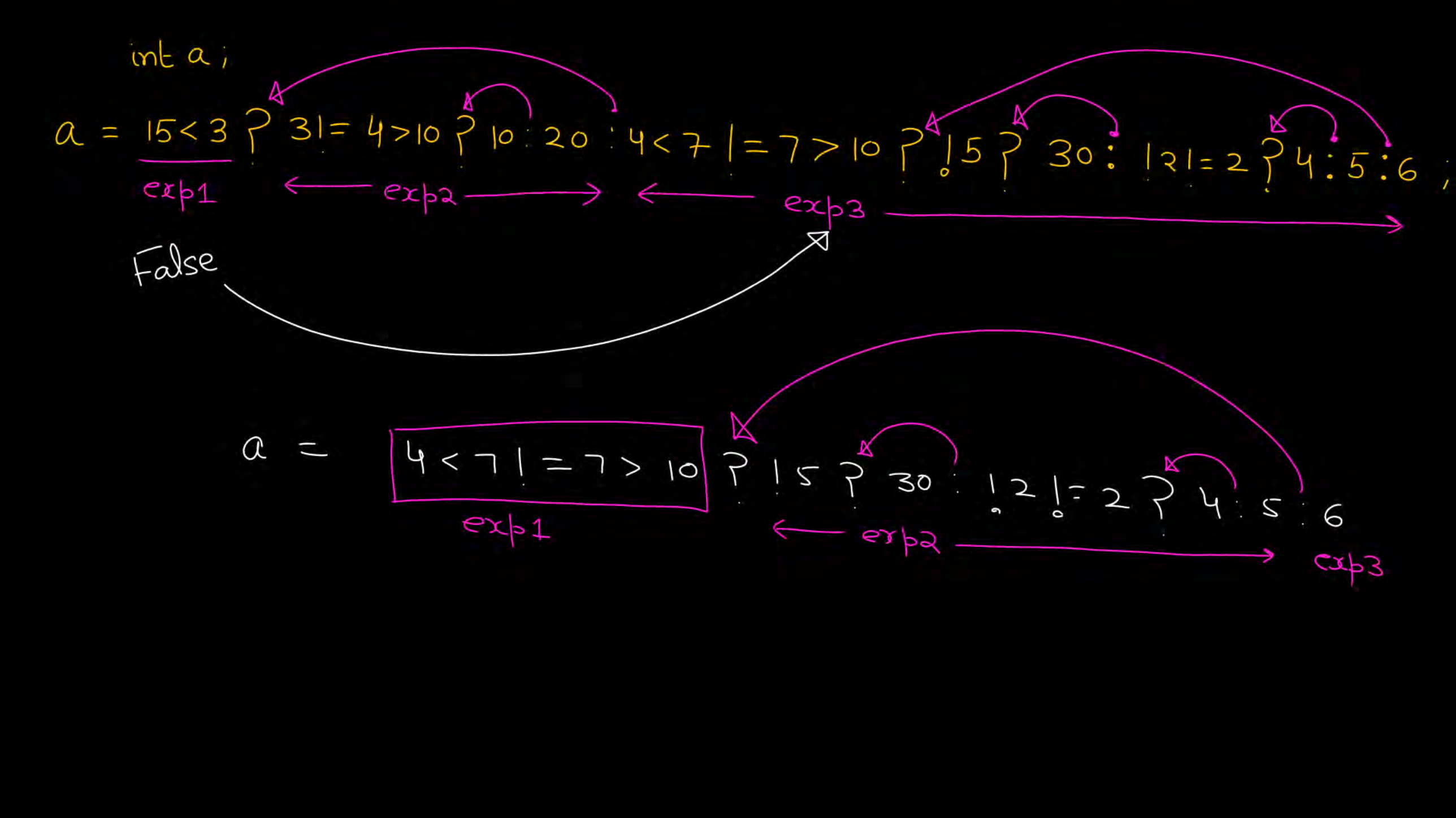


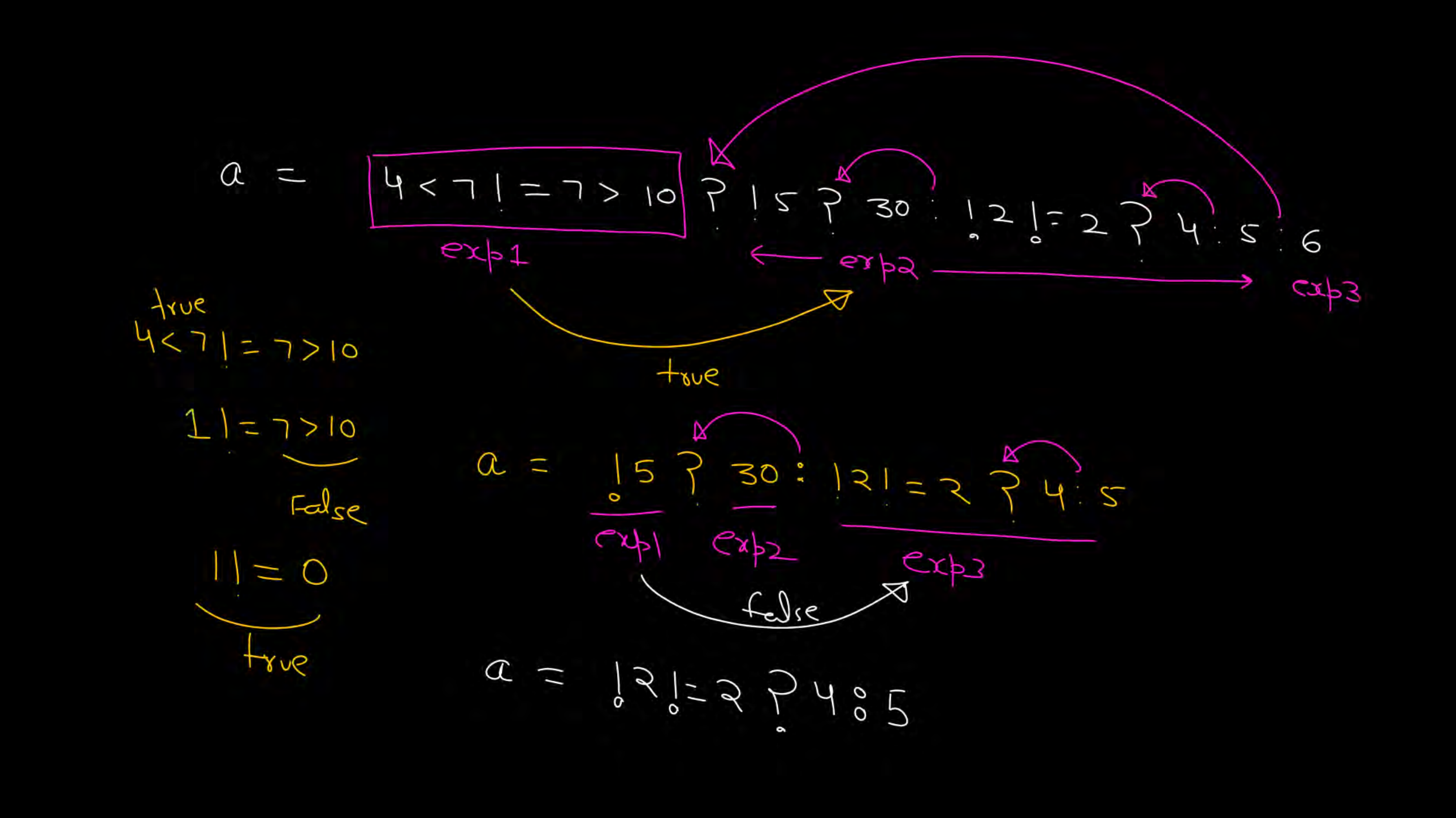
2==6!=3

0!=3

true

a = 10


int a;
a = 15<3 ? 31!=4>10 ? 10 : 20 : 4<7 !=7>10 ? !5 ? 30 : !21!=2 ? 4 : 5 : 6 ;
exp1
exp2
exp3
False
a = 4<7 !=7>10 ? !5 ? 30 : !21!=2 ? 4 : 5 : 6
exp1
exp2
exp3


a = 4<7!=7>10 ? !5 ? 30 : !2!=2 ? 4 : 5 : 6
exp1
exp2
exp3
true
true
4<7!=7>10
1!=7>10
false
!!=0
true
a = !5 ? 30 : !2!=2 ? 4 : 5
exp1
exp2
exp3
false
a = !2!=2 ? 4 : 5

$a = !2 != 2 \; ? \; 4 : 5$

$!$ non-zero $= 0$

$!2 = 2$

$0 != 2$

true

$a = 4$

$x = x + 10 \implies x += 10$

$x = x - 10 \implies x -= 10$

$x = x / 10 \implies x /= 10$

$x = x * 10 \implies x *= 10$

$x = x \% 10 \implies x \%= 10$

$x = x \& 10 \implies x \&= 10$

$x = x | 10 \implies x |= 10$

$x = x \hat{} 10 \implies x \hat{}= 10$

$x = x << 10 \implies x <<= 10$

$x = x >> 10 \implies x >>= 10$

sizeof
comma
scoping

] last class

1.

```
int a;
a = 12>2 ? printf("Gate") && printf("Wallah") || printf("2023") : pf("sir");
printf("%d",a);
```

2.

```
int a;
a = 20>110 ? 100 : !2! = 3>50 ? 300 : 400;
bf("%d",a);
```

3.

```
int a;
a = 10! = 12>50 ? !4! = 4 ? 8>8! = 0 ? 10 : 20 : 30 : 40;
bf("%d",a);
```


THANK YOU GW
SOLDIERS !