



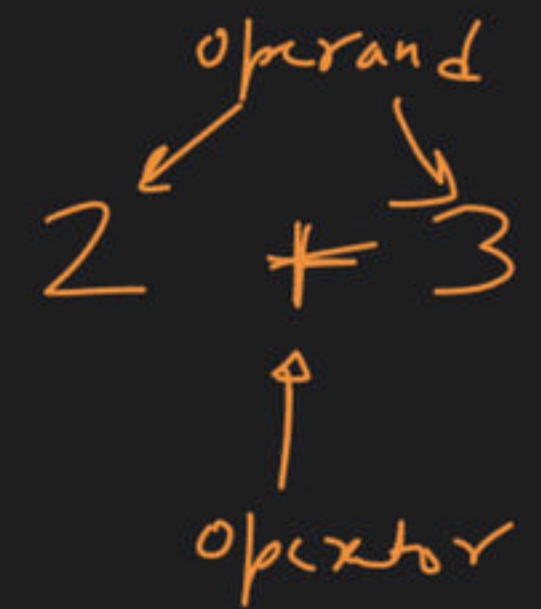
Bitwise Operators & Loops

Special class

→ Discord link



→ Bitwise \rightarrow (bit level)



truth table

- AND $\rightarrow \&$
- OR $\rightarrow |$
- NOT $\rightarrow \sim$
- XOR $\rightarrow \wedge$

&

i/p

a	b	o/p
0	0	0
0	1	0
1	0	0
1	1	1

\wedge

i/p

a	b	o/p
0	0	0
0	1	1
1	0	1
1	1	1

Not \sim

~	o/p
0	1
1	0

AND $\rightarrow \&$
 OR $\rightarrow ||$
NOT $\rightarrow \sim$
 $T \rightarrow F$
 $F \rightarrow T$

XOR

/

i/o		
a	b	o/p
0	0	0
0	1	1
1	0	1
1	1	0

same \rightarrow 0

diff \rightarrow 1

~~5~~ [^] ~~5~~ \rightarrow 0

~~5~~ [^] ~~5~~ \rightarrow

$\xrightarrow{i/p}$ ~~2~~ [^] ~~3~~ [^] ~~4~~ [^] 5 [^] ~~2~~ [^] ~~4~~ [^] ~~3~~ \rightarrow 5

2 & 3

a	b	o/p
0	0	0
0	1	0
1	0	0
1	1	1

2 → 000000010

3 → 000000011

000000010



2 & 3 → 2

5 2 10

5 | 10

5 → 00000101
10 → 00001010

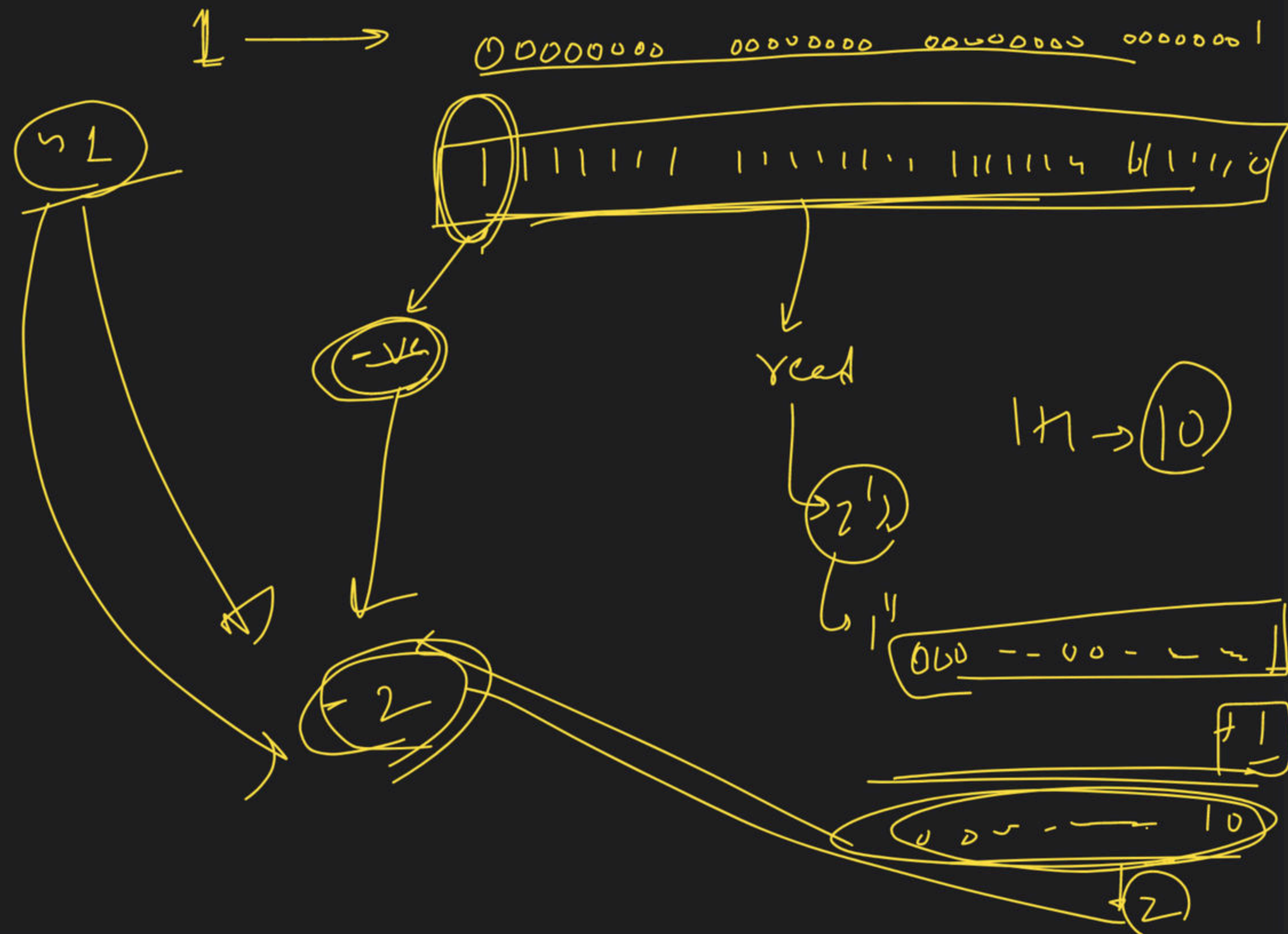
00001111
→ 15

5 → 0000101

10 → 00001010

00000000

5 2 10 →



(1)

5 → 0000 0101

5 → 0000 0101

00000000 — 0

5⁻¹ 5⁻ → 0

~~5⁻ 5⁻~~

5[^]10 →

5 → 0000 0101

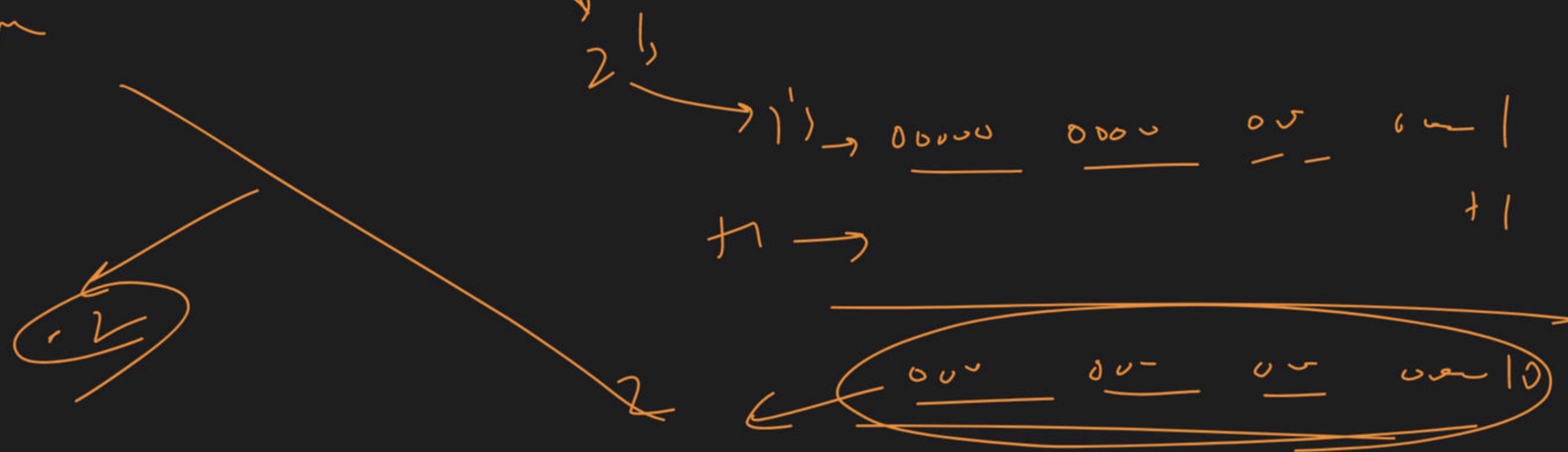
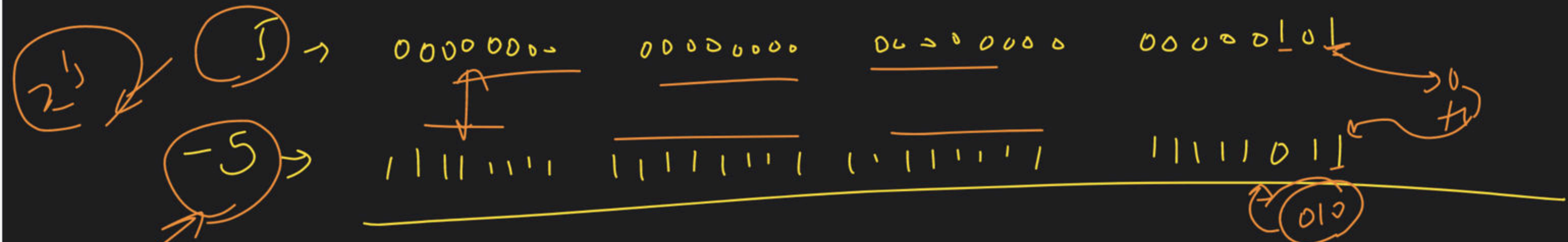
10 → 0000 1010

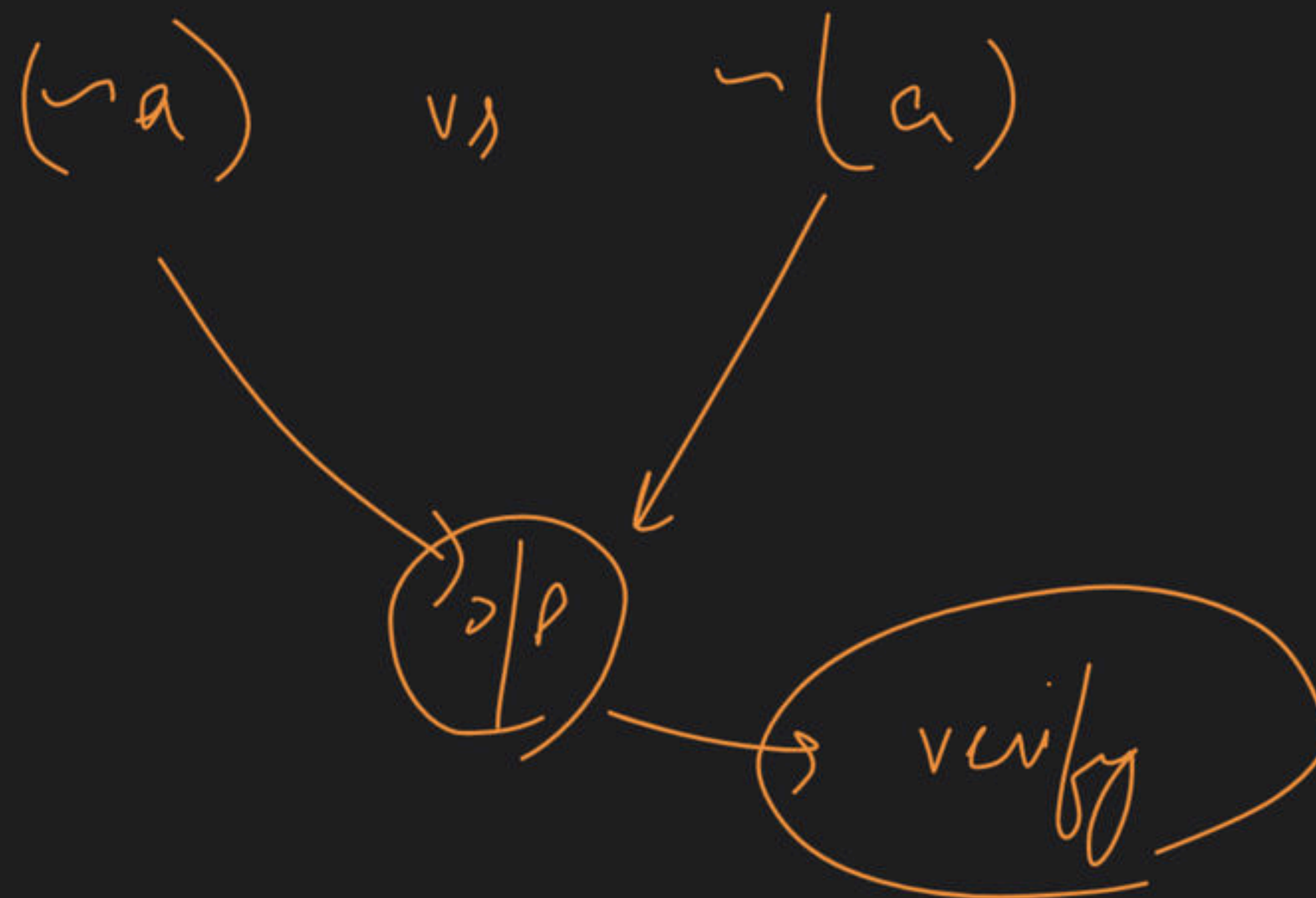
0000 1111

5[^]10 → 11



~~5 - 10 = 5~~





Left & Right Shift

Operators (2)

5 → 0000 - - 00101 → 5

← (<< 1)

0000 - - 001010 → 10

<<

>>

int a = 2

0000 - - 000010 → 2

a << 1 → a is left shift by 1 bit

0000 - - 0000100 → 4

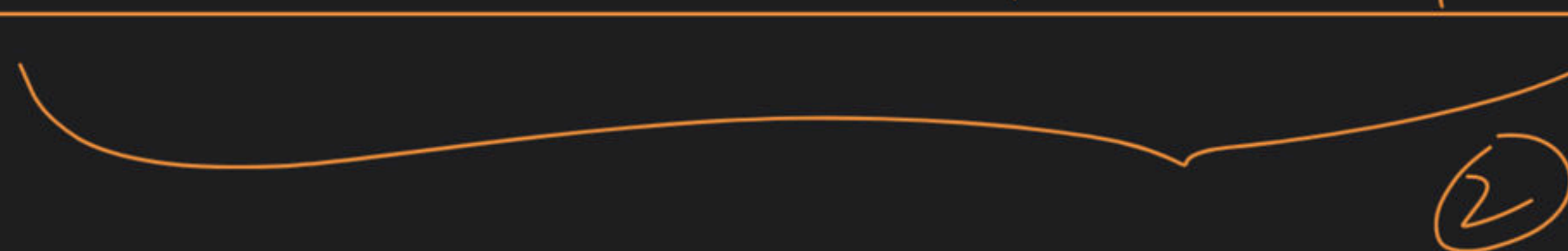
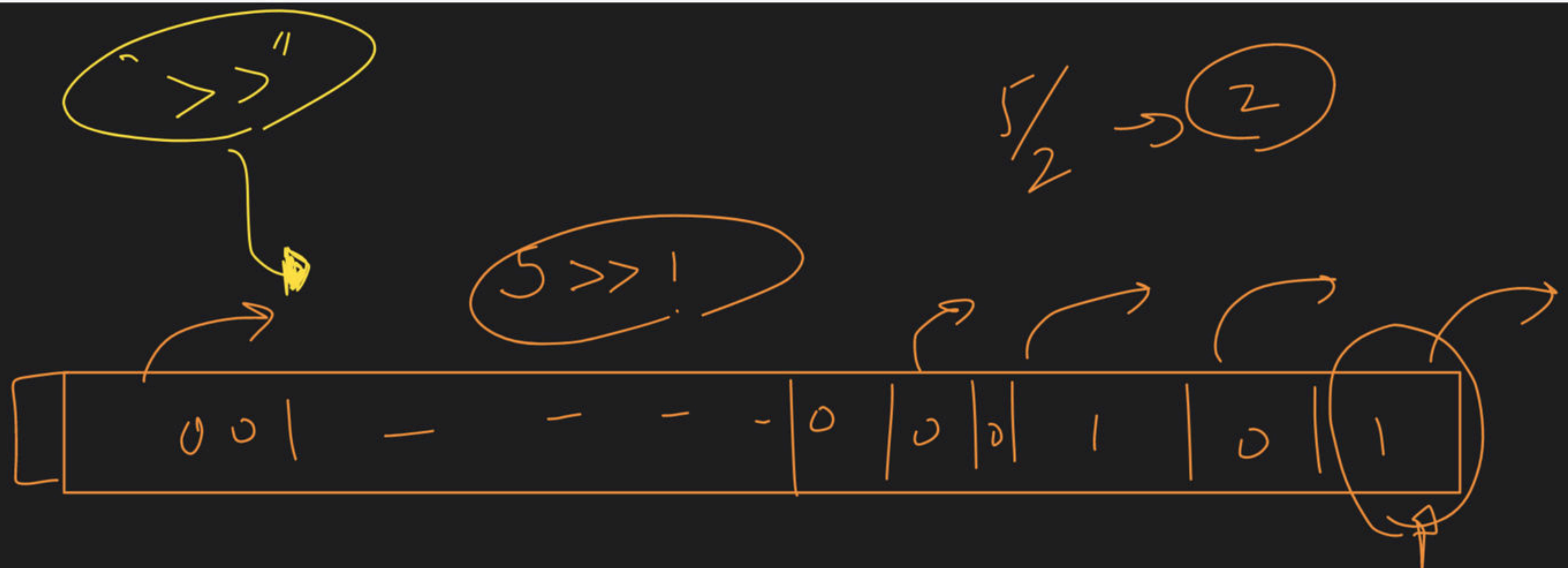


<<1

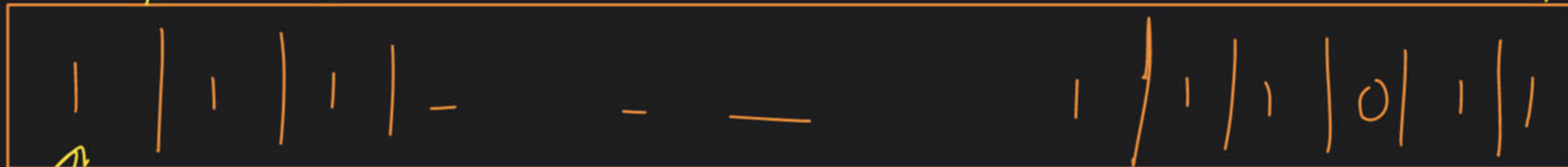


<<1





→ 5

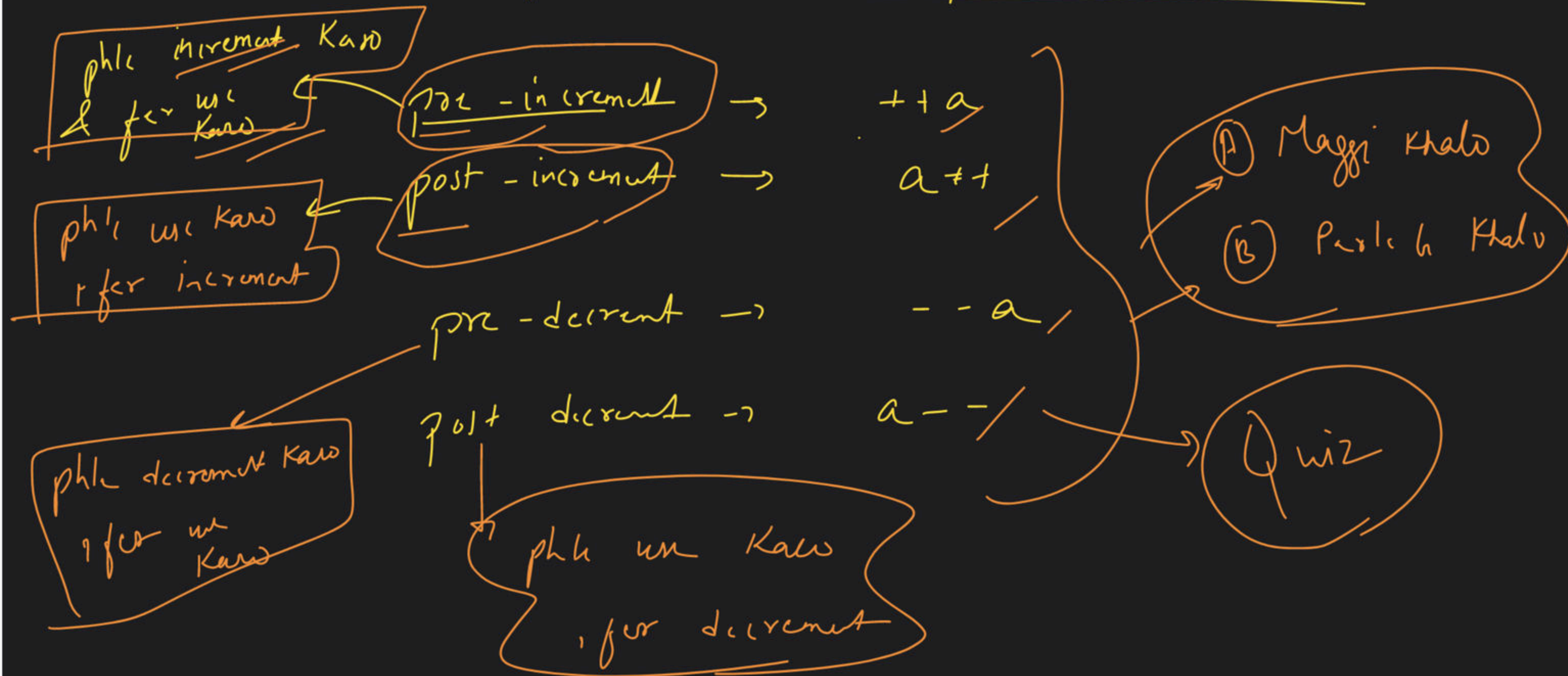


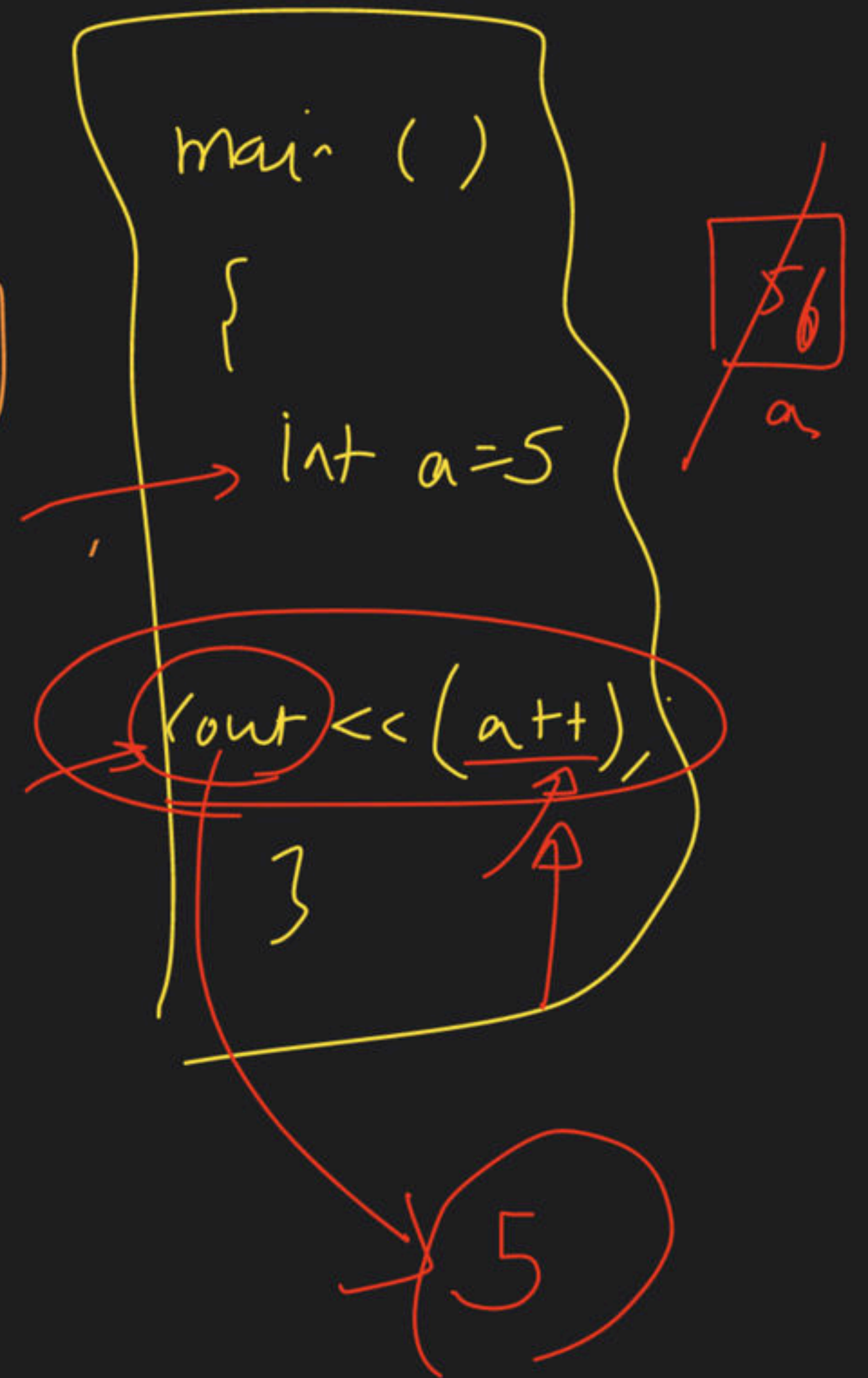
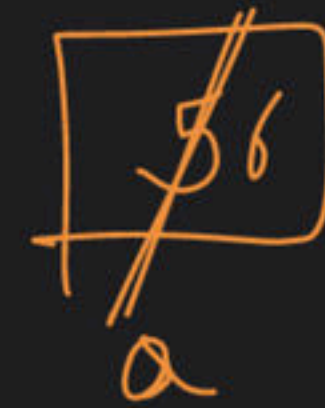
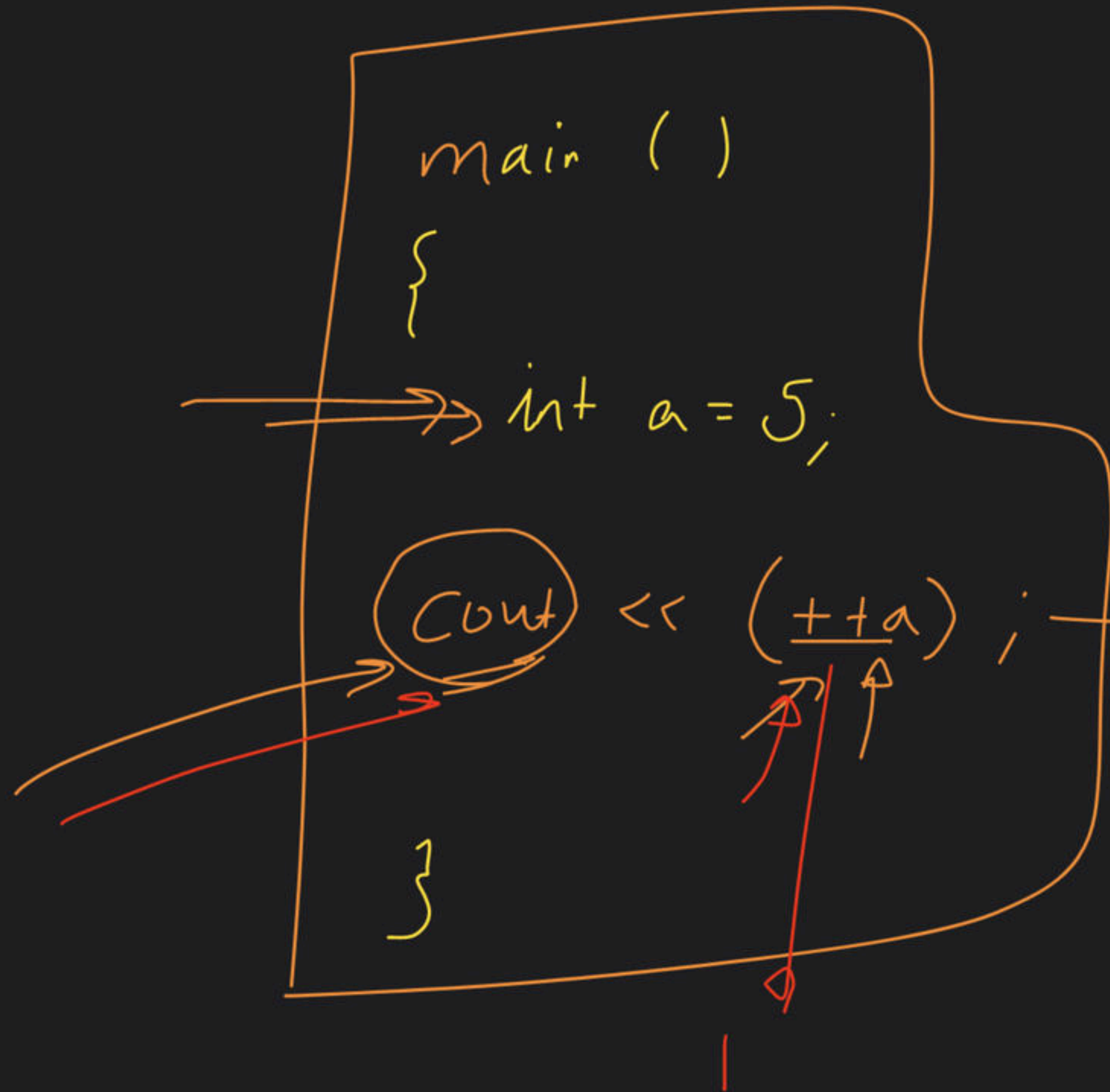
>>1



five large no'

Pre / Post Inc / decrement operator





main()

{

int a = 10

~~109~~
a

cout << (~~109~~ → a) * 10;

90

}

main()

{

int a = 10;

~~910~~
a

cout << (a → 10) * 10;

100

}

for (int i = 0; i < n; ~~i++~~)

~~i++~~ ↙
++i ↘

↖ A
↗ C

↖
B ++ --

main ()

{

int a = 21 ;

a ~~21~~ ~~22~~ 23

22 print ← (out << ++a ;) → (A) inc ✓
→ (B) mc → 22 print

(out << a++ ;) → (A) un → 22 print
→ (B) inc → ✓

→ (out << a ;) → 23

}

main()

{

int a = 10;

(11)
cout << (++a) * 10;

(A) → inc

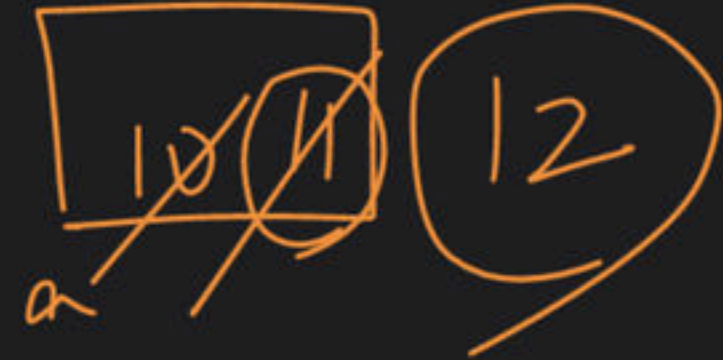
(B) mul

cout << (a++) * 10;

(A) mul (B) inc

cout << a;

}



110

110

12

main()

{

int a = 10;

cout << (++a) * (a++),

}

110
121
144
120
90
111
132

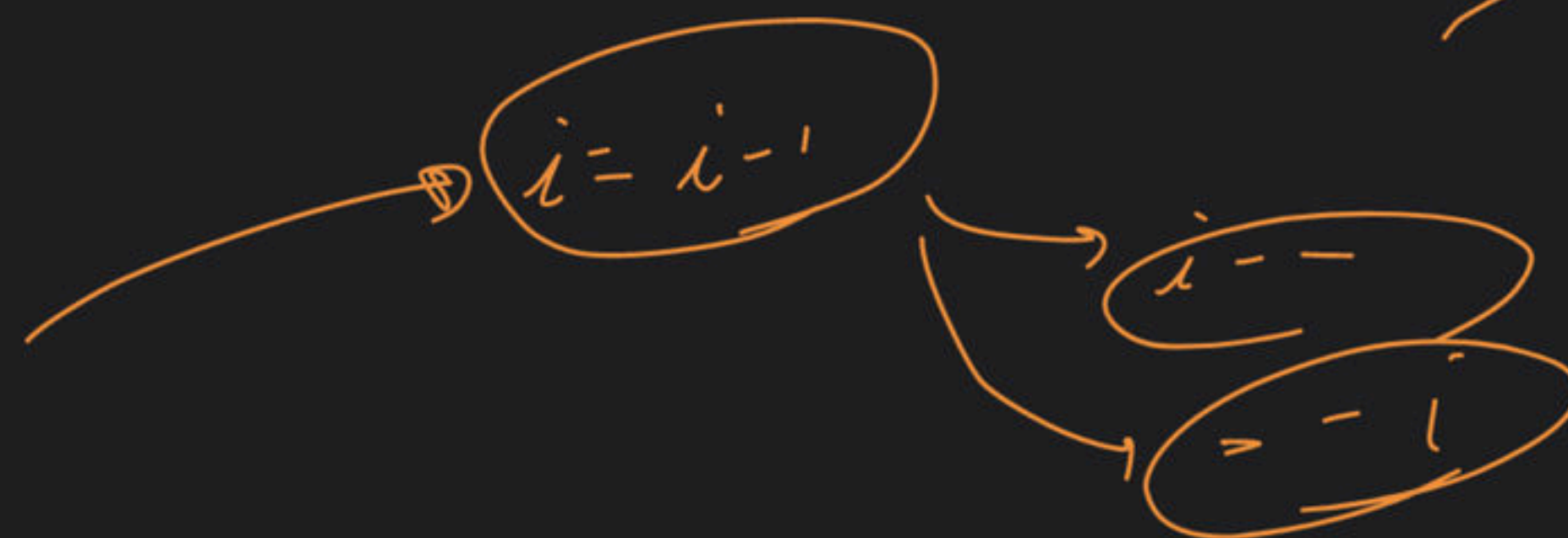
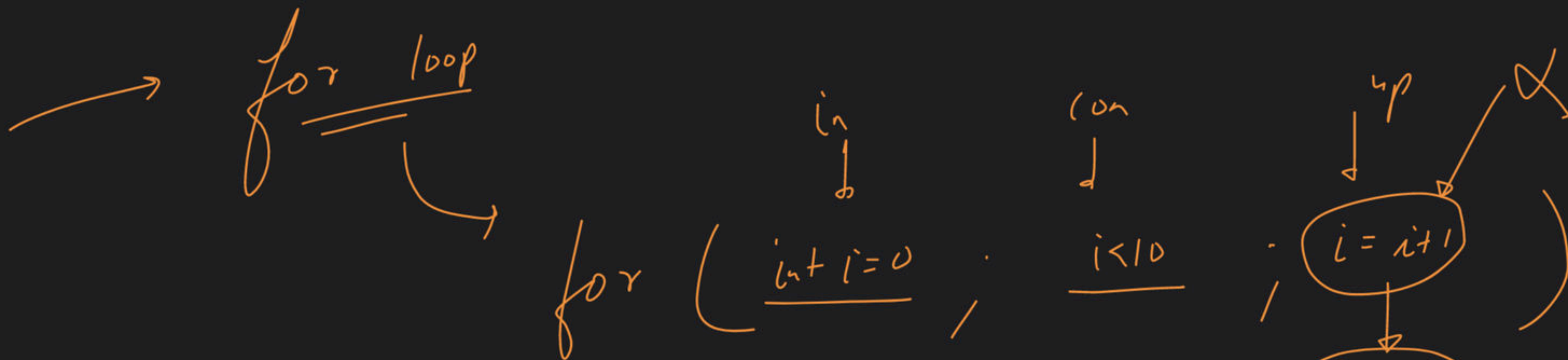
(++a) * (a++) → 121/132
(a++) * (++a) → 120

why

Lakshay Bhatnagar

121

132



2 min Break



```
for (int i = 0; i <= 5; i++)
```

```
if (i == 2) {
```

```
break;
```

loop behavior

for $i = 0$
 $0 <= 5 \rightarrow T$

for $i = 1$
 $1 <= 5 \rightarrow T$

for $i = 2$
 $2 <= 5 \rightarrow T$

break

$i = 3$ ✗
 $i = 4$ ✗
 $i = 5$ ✗

"0" "2" "3" "4" "5"

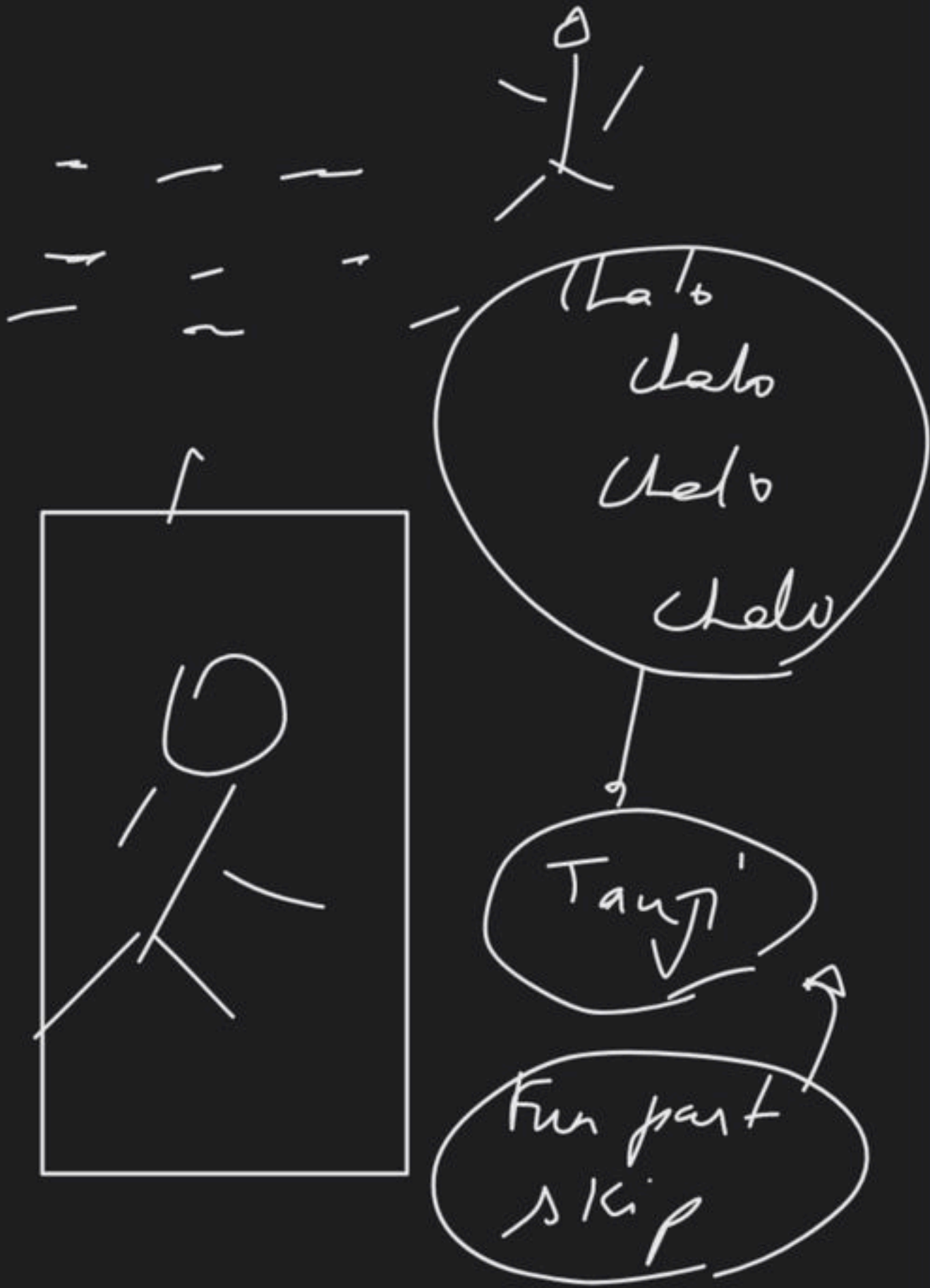
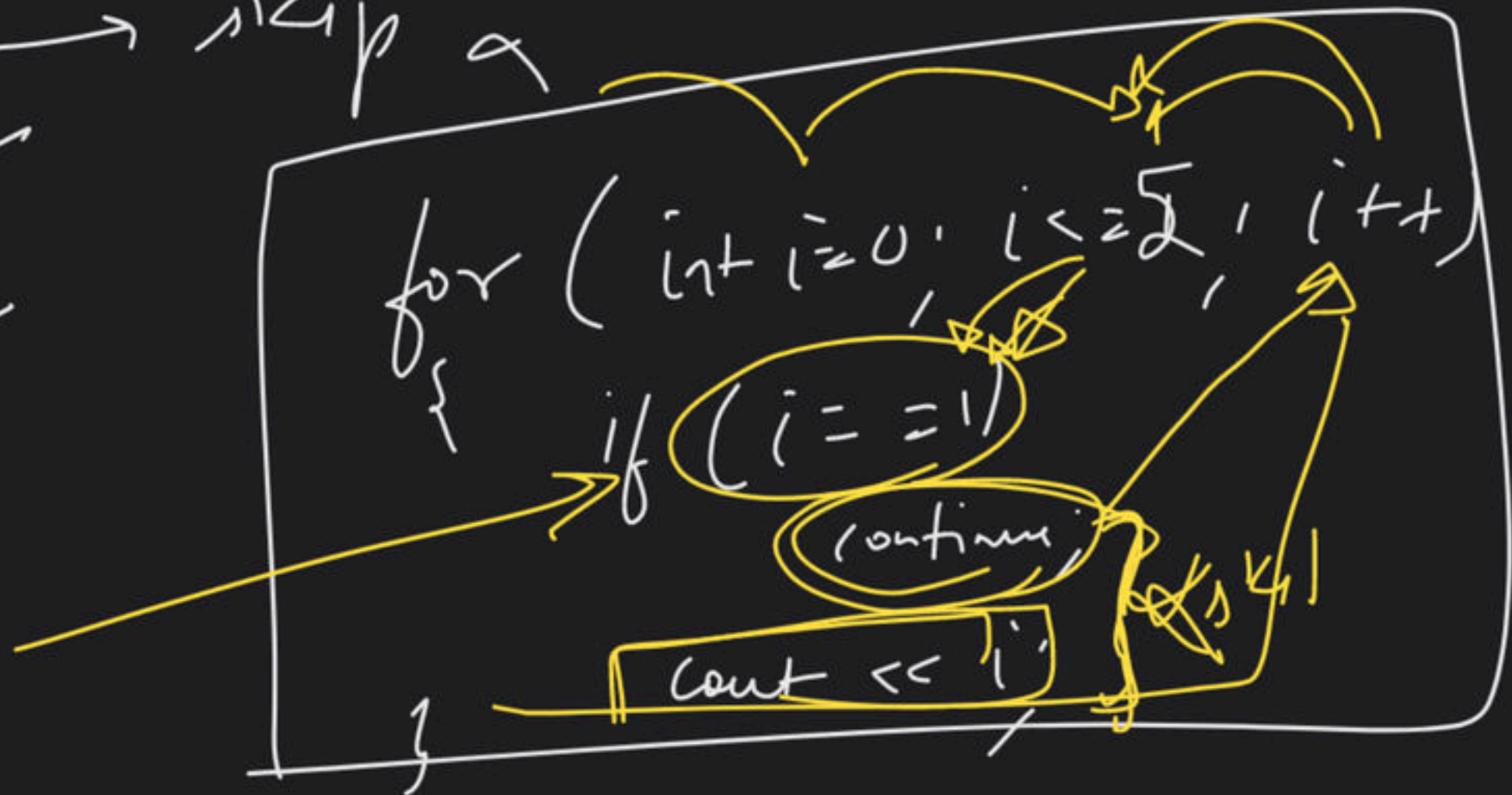


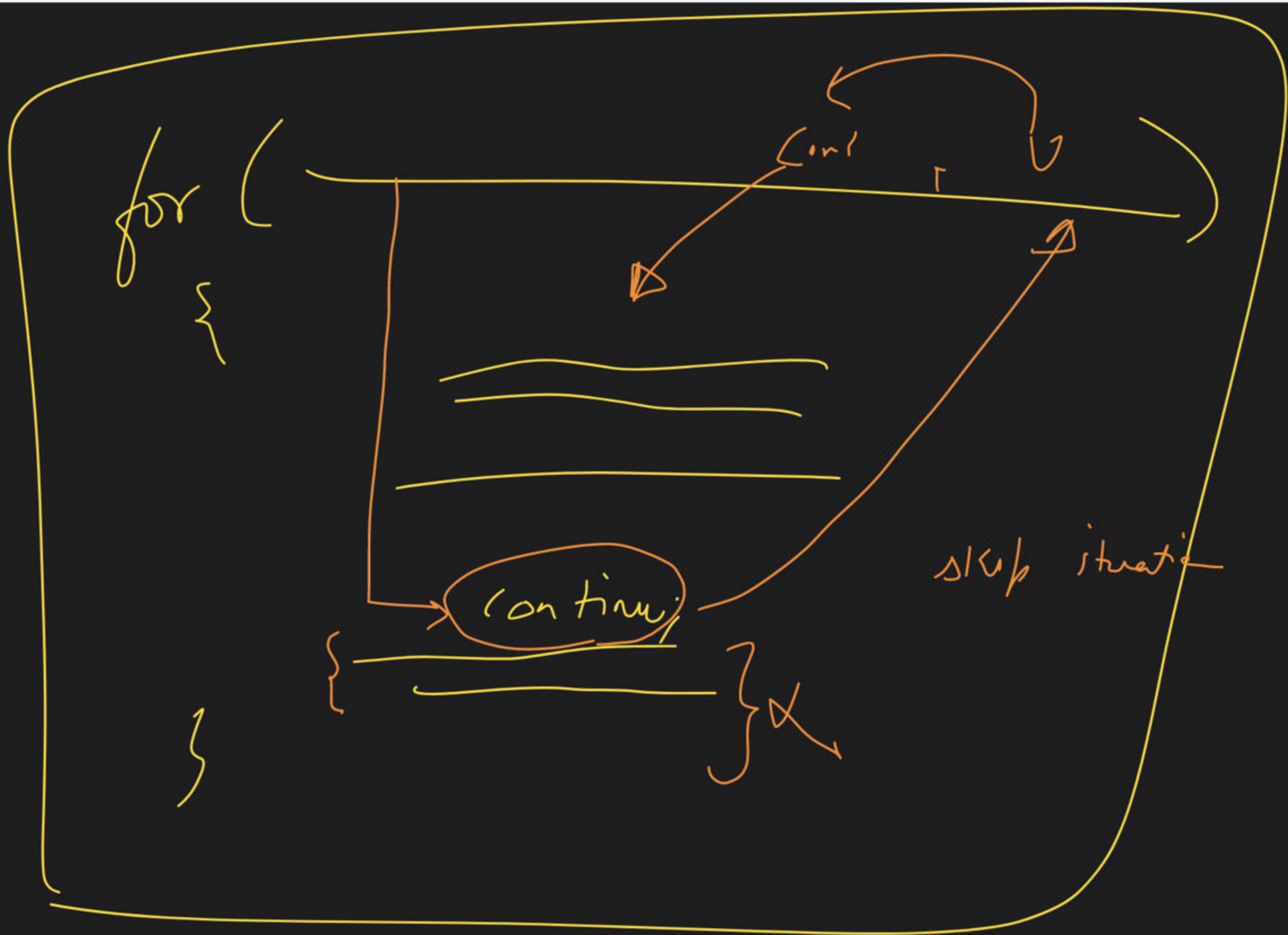
Continue

target → iteration skip

i=0 ✓
i=1 → skip α
i=2 → skip α
i=3 ✓
i=4 ✓

i=0
i=1 → skip
i=2
i=3





→ Variable Scoping

→ Local Variable

→ Global Variable

main ()

{

for (int i = 0; i < 5; i++)

{

cout << i;

}

}

cout << i;

X

1 min

~~goto~~

→ Express

15 min

$$2 * 3 + 5/10 - 2$$

$$6 + 0.5 - 2$$

$$6.5 - 2$$

$$4.5$$

$$\left(\left(\frac{2 * 3}{\downarrow} \right) + \left(\frac{5/10}{\downarrow} \right) \right) - 2$$

$$\left(\frac{6 + 0.5}{\downarrow} \right) - 2$$

$$6.5 - 2$$

$$4.5$$

→ Operator Precedence table

Use Brackets