

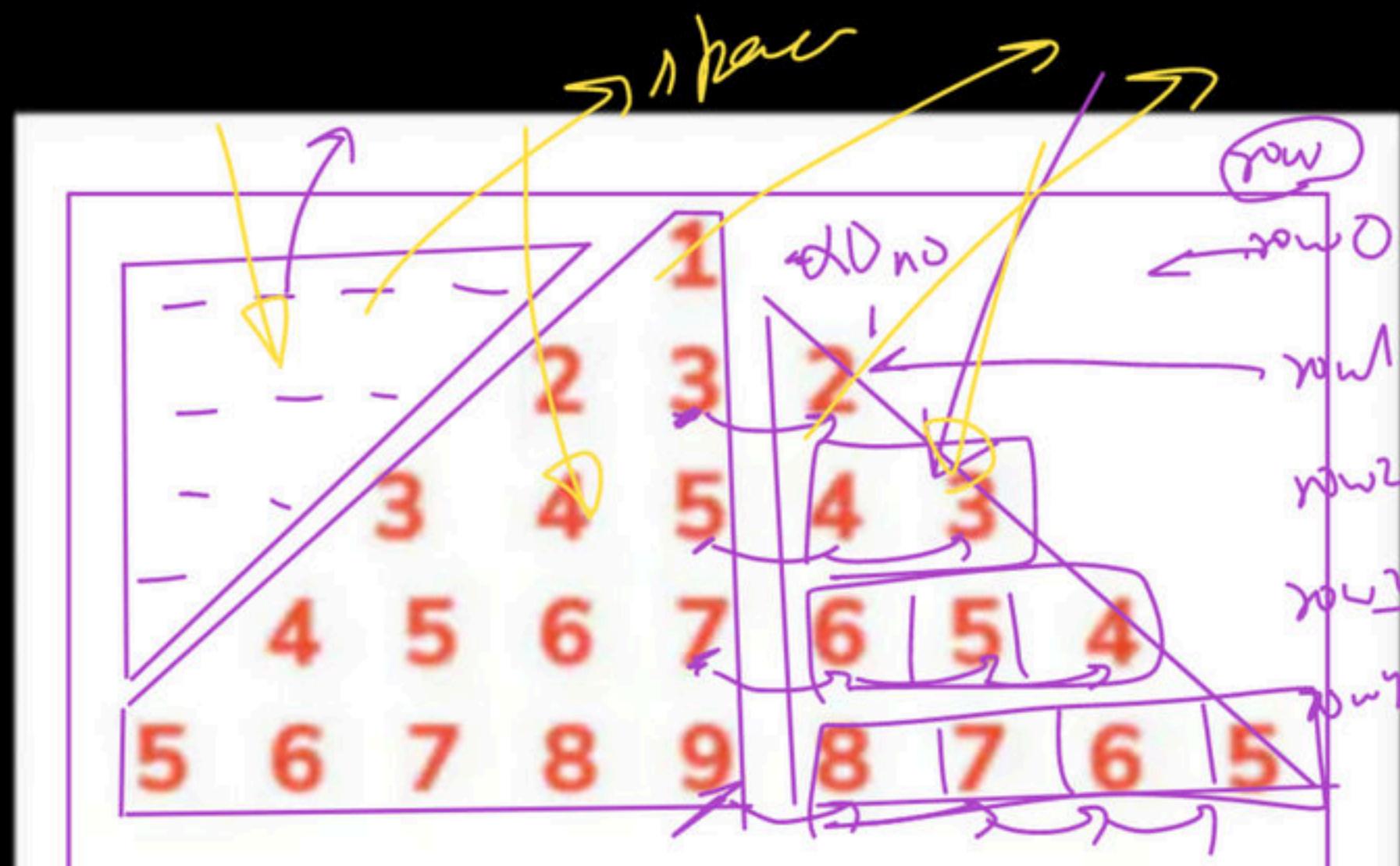
Operators, Loops & Conditionals

Special class

Operators, Functions & lot more

Instructor: Love Babbar

Numeric Full Pyramid



$$n=5$$

$$n-1=4$$

$$n-1=3$$

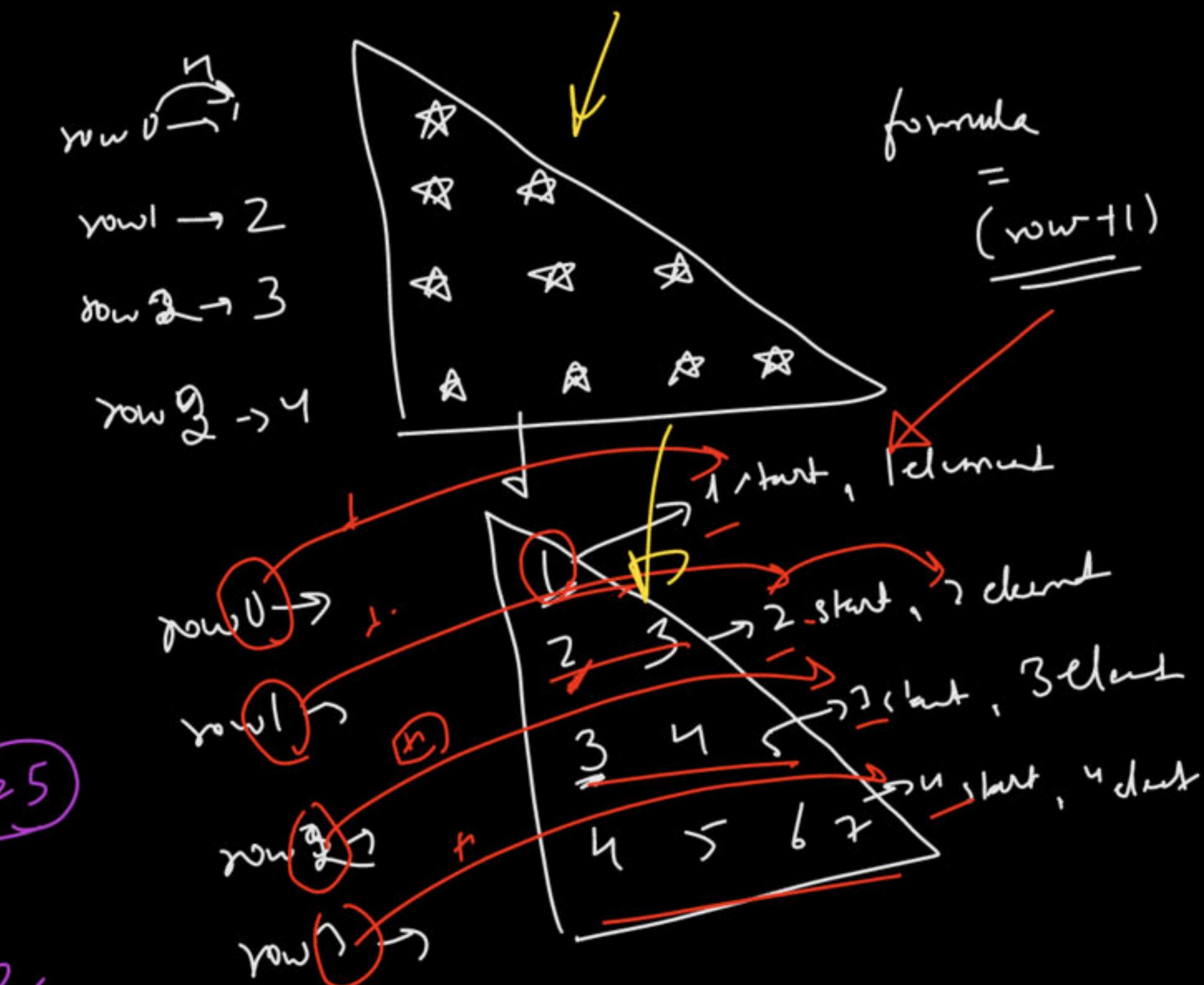
$$n=1$$

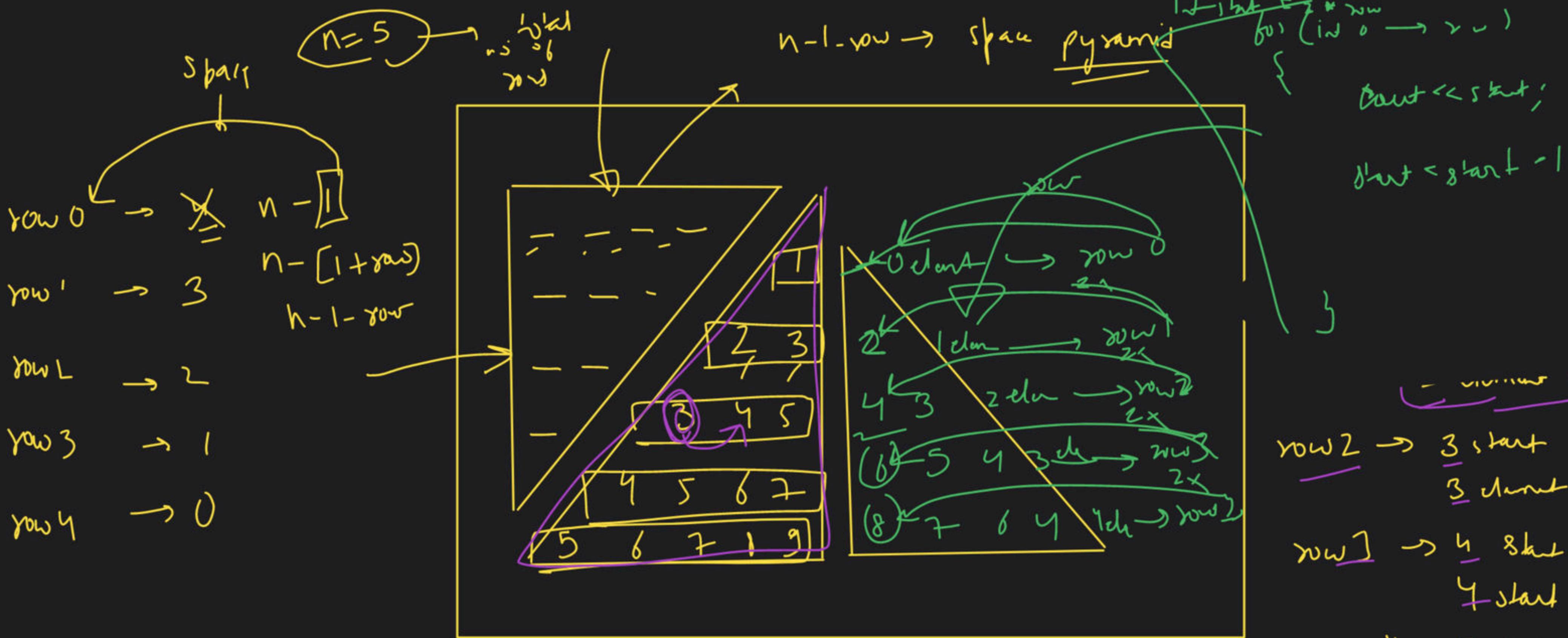
$$n - [1 \text{ row}]$$

$$n-1-\text{row}$$

$$\begin{aligned} n-1-\text{row} &= 1 \\ \text{row} &= 0 \end{aligned}$$

$$\begin{aligned} \text{row} 0 &\rightarrow 4 \\ \text{row} 1 &\rightarrow 3 \\ \text{row} 2 &\rightarrow 2 \\ \text{row} 3 &\rightarrow 1 \\ \text{row} 4 &\rightarrow 0 \end{aligned}$$





$n = 5$

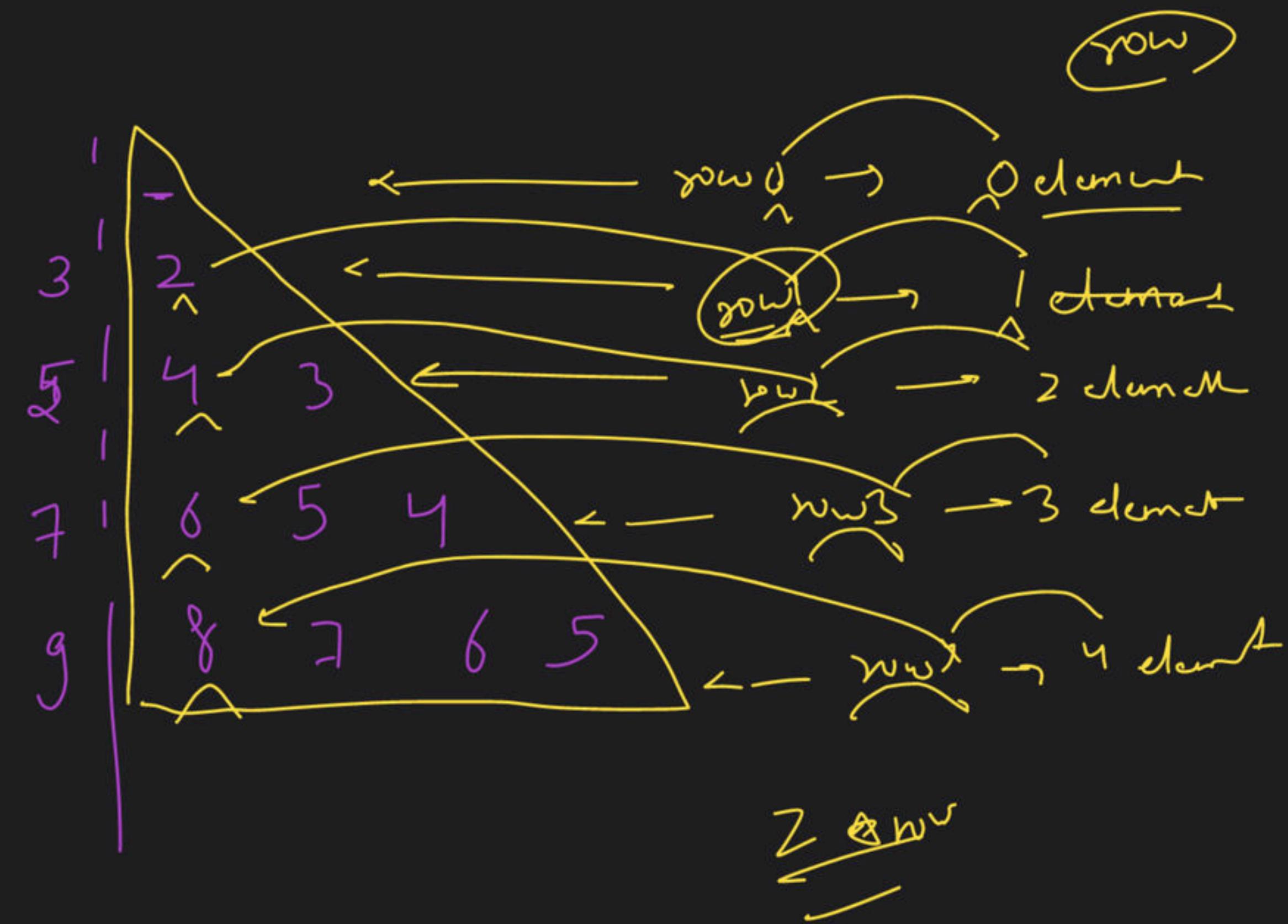
$$n - 1 = 5 - 1$$

$$n - 1 = 4$$

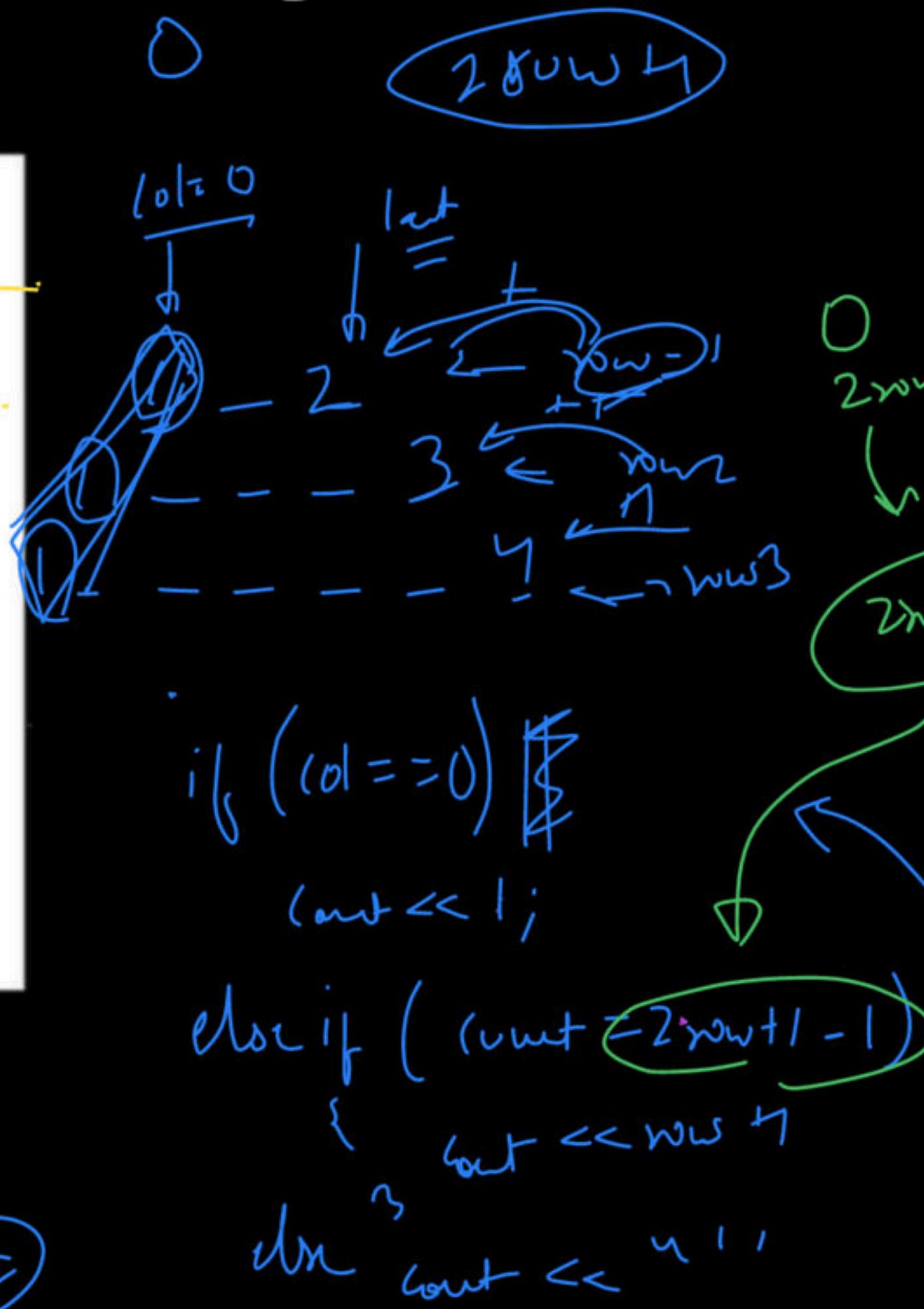
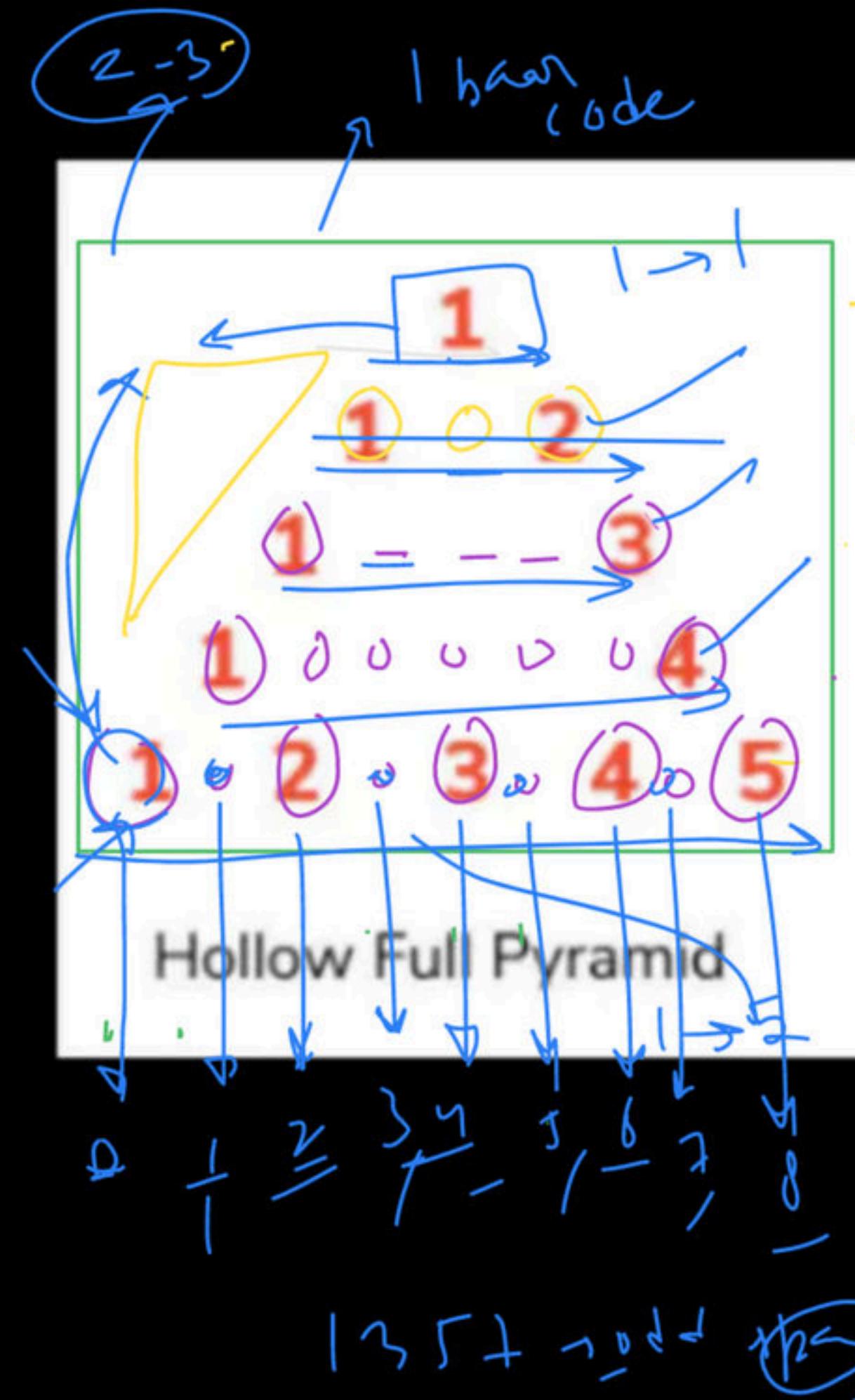
4

$1 + \text{row} = 0 + 1$
 $\text{start} \rightarrow \text{start}$

$\text{int stat} = \text{row}++$
 $\text{for } (0 \rightarrow \text{row}+1)$
 $\{$
 $\text{row} < \text{start}$
 $\text{start} \rightarrow \text{start} + 1$



Numeric Hollow Pyramid



```

    int start = 1;
    for (int i = 1; i <= 2 * n - 1; i++) {
        if (row == 0 || row == n - 1) {
            if (col == 0 || col == 2 * n - 2) {
                cout << " ";
            } else {
                cout << "1";
            }
        } else if (count == 2 * row + 1 - 1) {
            cout << " ";
        } else {
            cout << "1";
        }
    }
}
  
```

Inner ↓

0 → m → m-1

if (row == 0 || row == n - 1)

if (col == 0 || col == 2 * n - 2)

cout << " "

else

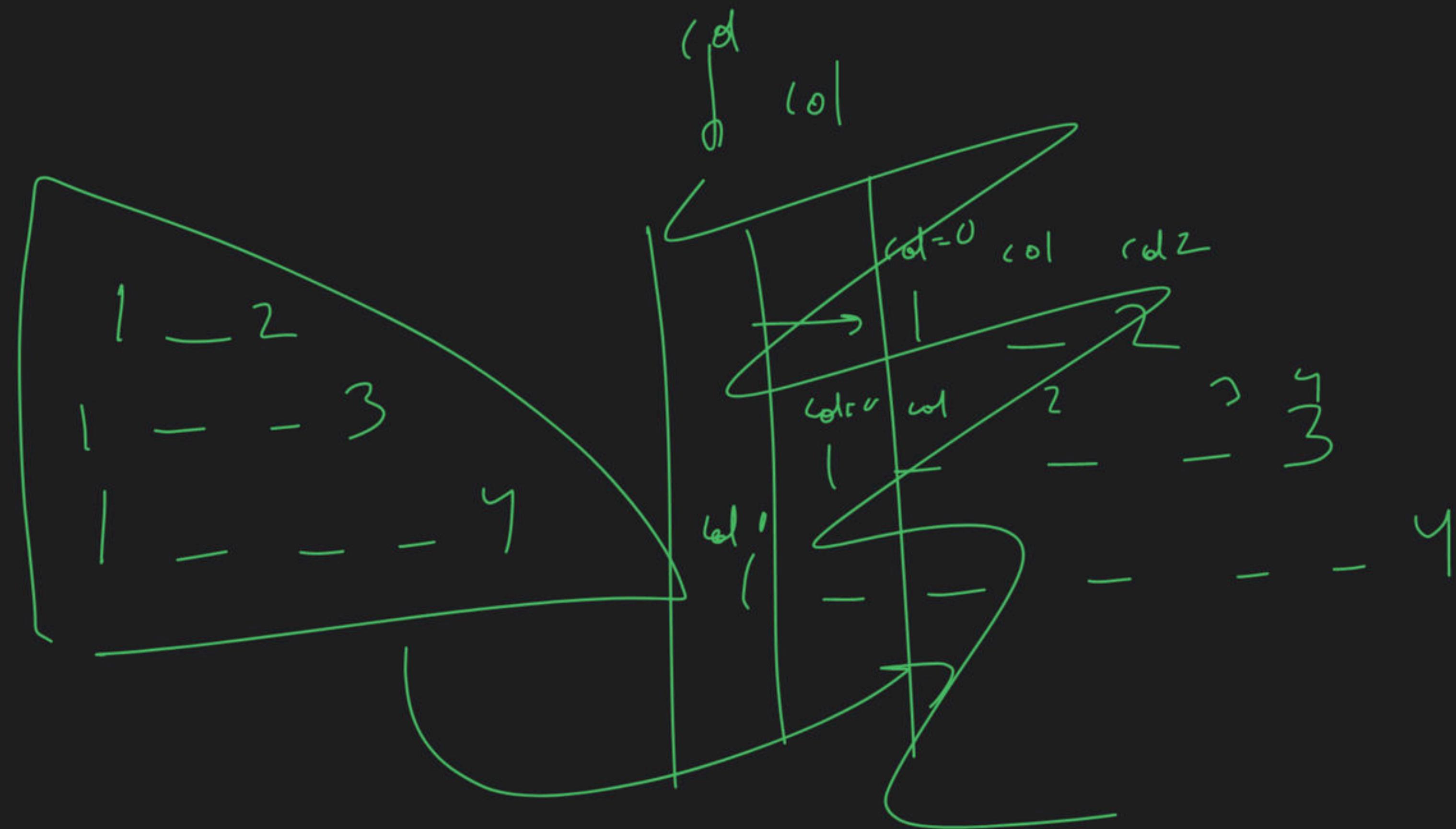
cout << "1"

else if (count == 2 * row + 1 - 1)

cout << " "

else

cout << "1"



Bitwise Operators

- AND
- OR
- NOT
- XOR

&

a	b	O/P
0	0	0
0	1	0
1	0	0
1	1	1

$a \& b$

a	b	XOR
0	0	0
0	1	1
1	0	1
1	1	0

$a \wedge b$

same value

↓

0

diff value

↓

.. ↗ ↘

↑ ↓

↑ ↓

Logic

↑

a	b	a b
0	0	0
0	1	1
1	0	1
1	1	1

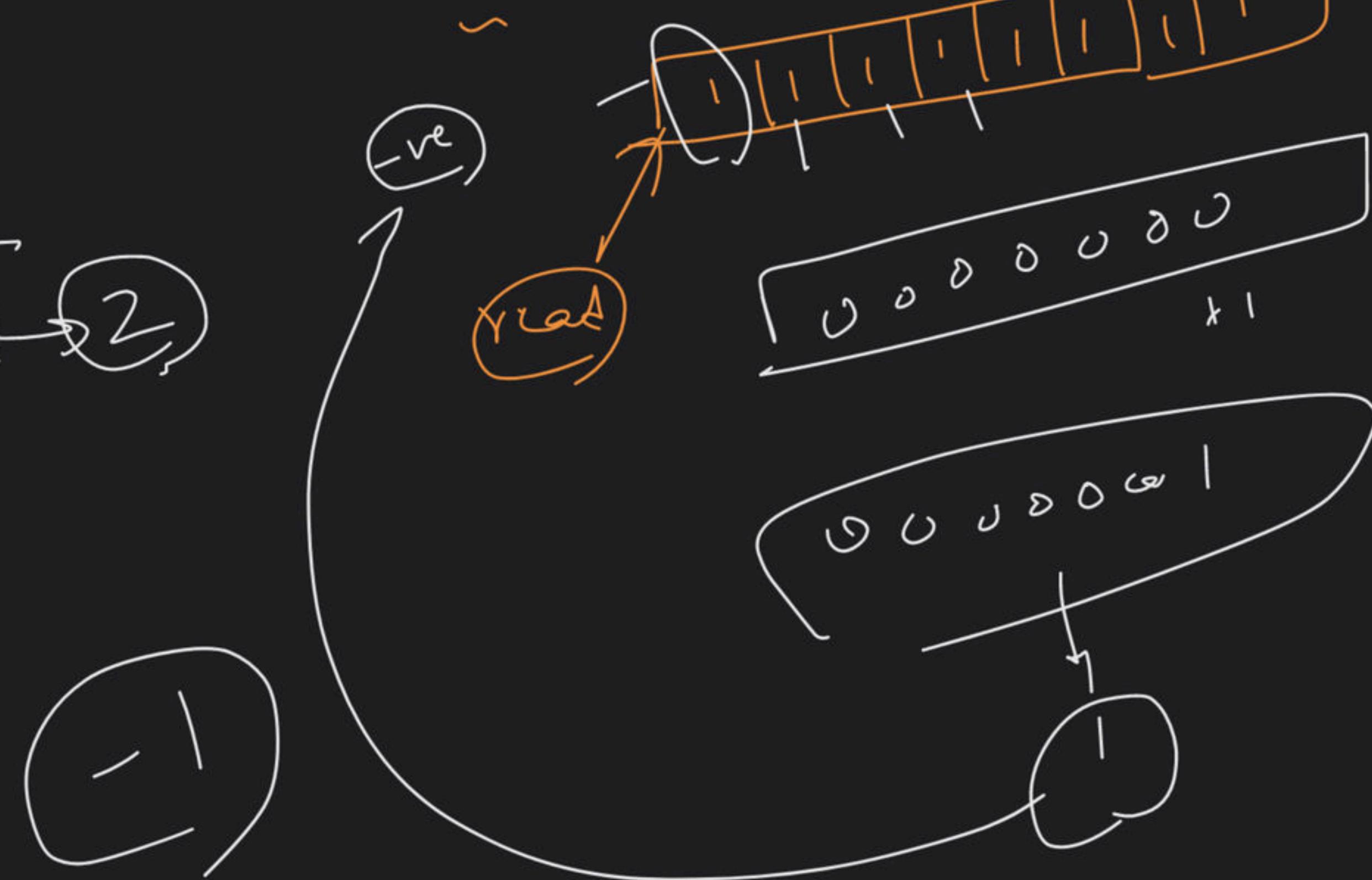
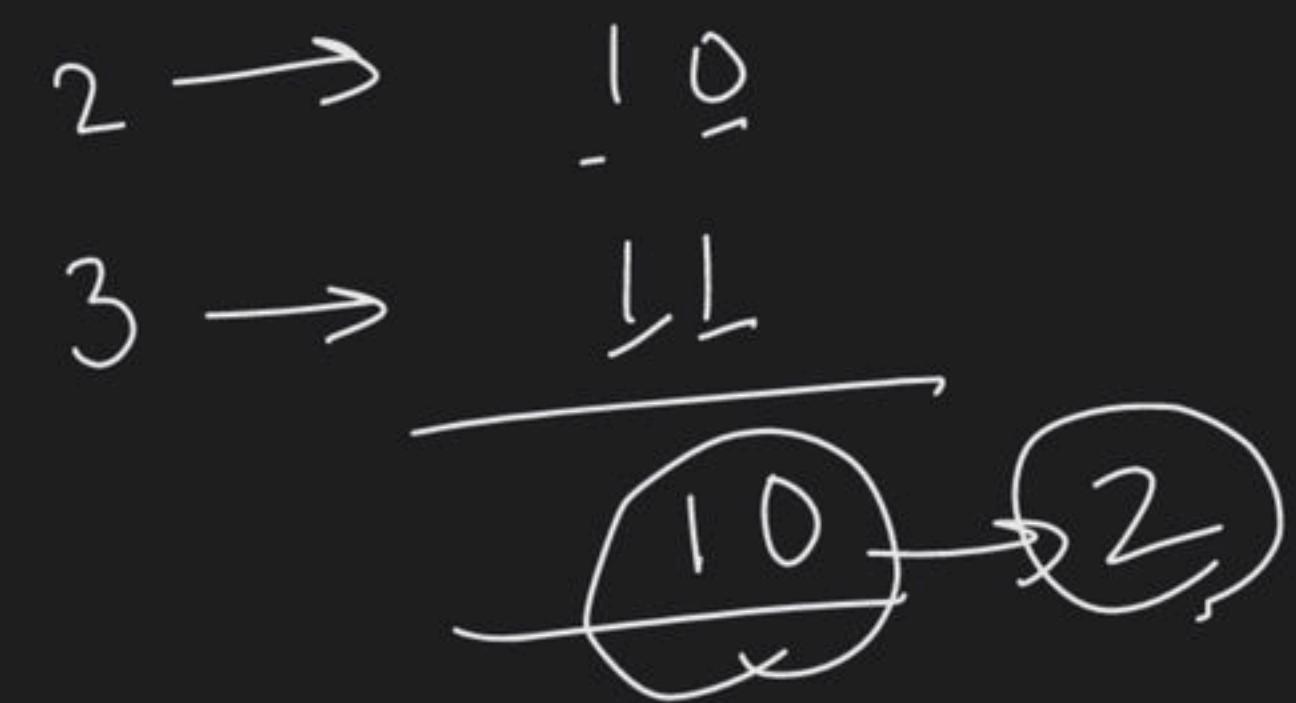
$a | b$

Logic

↑

a	$\neg a \leftarrow \neg \neg a$
0	1
1	0

min break



$a = \sim a$

$a = \sim(\sim a)$

$a = (\sim a)$

$\exists y$ \rightarrow 

$\exists y$ \rightarrow 

$5 \rightarrow 00101$

10

$\overline{1010}$



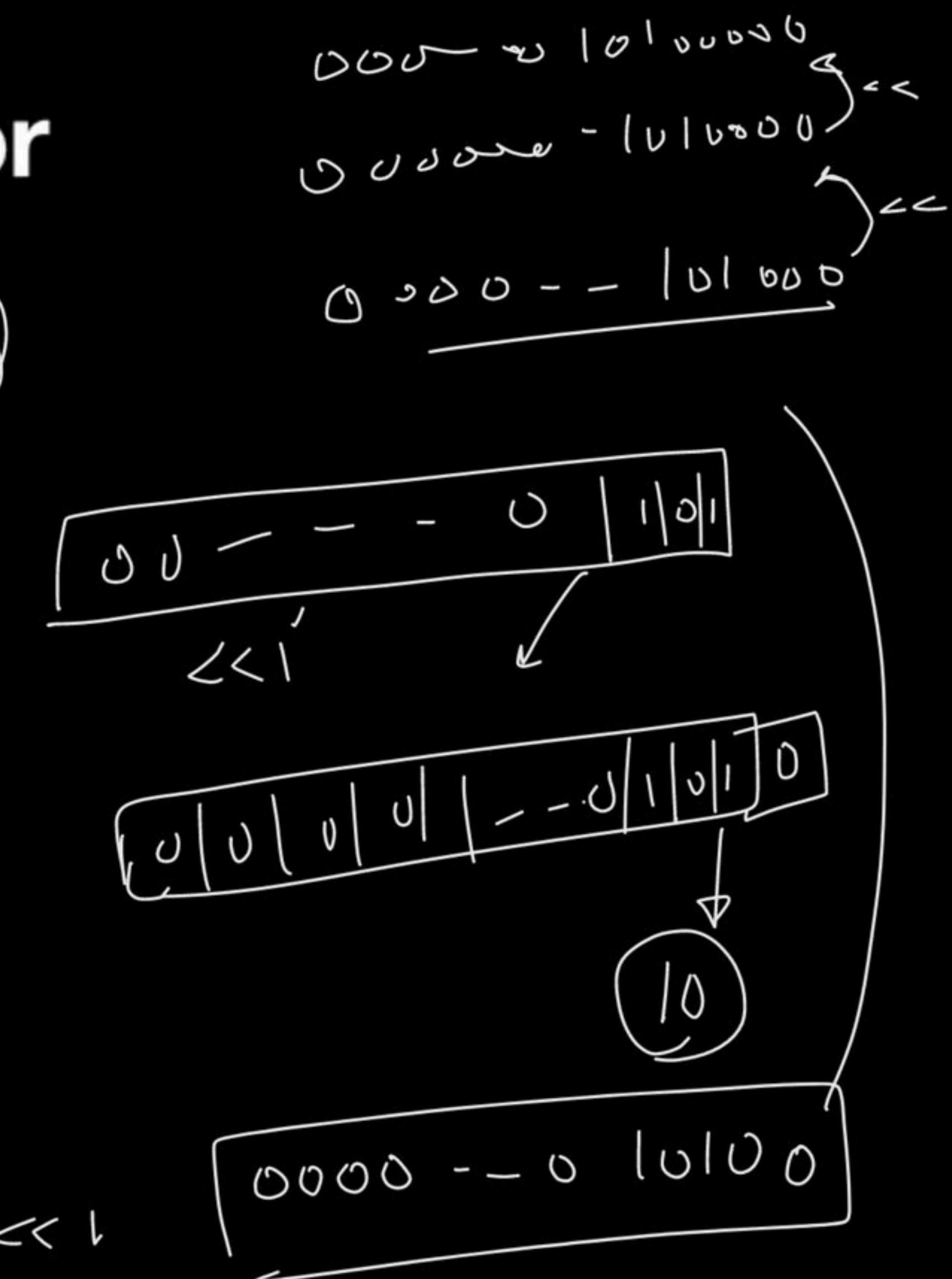
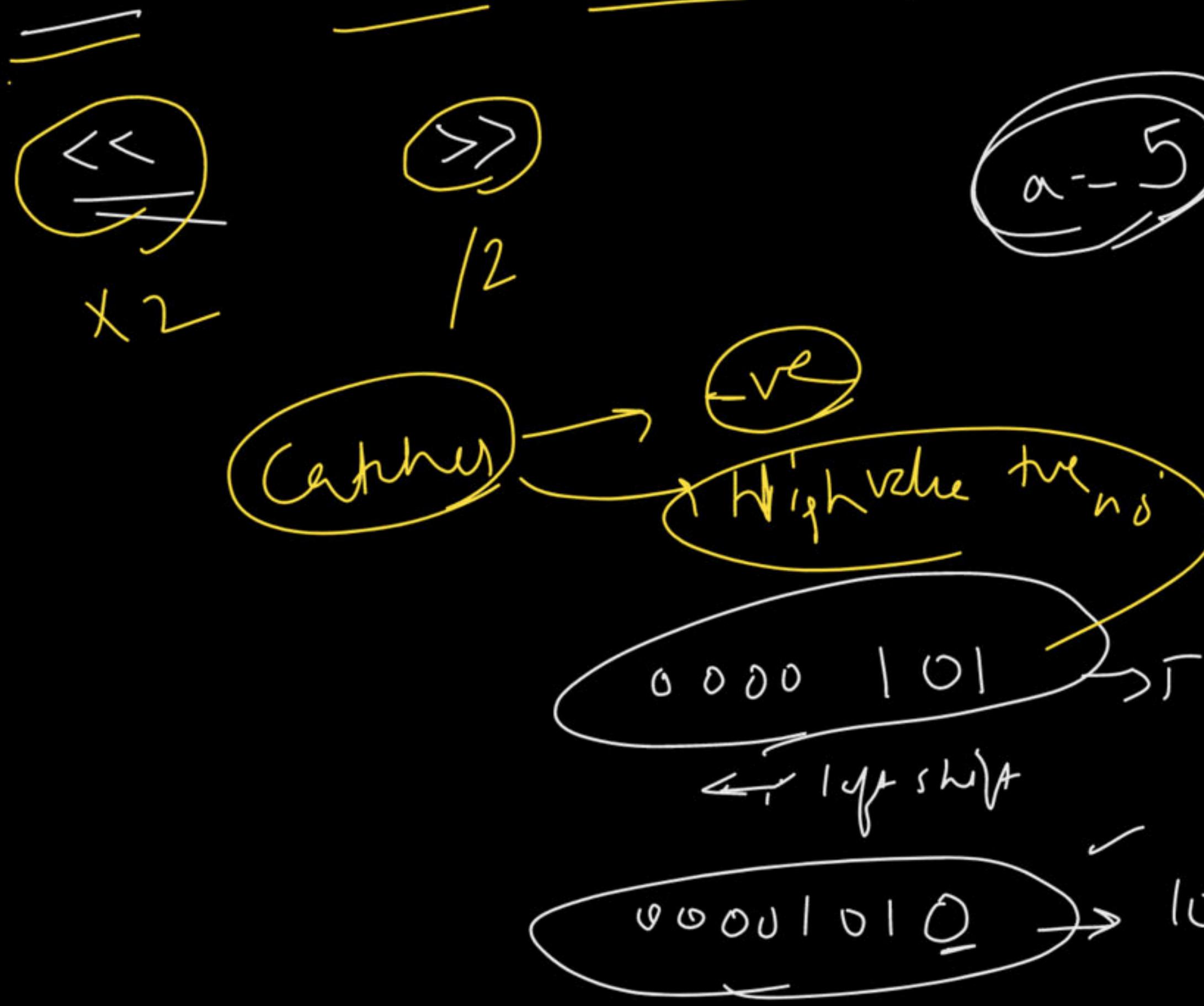
AND Operator

OR Operator

NOT Operator

XOR Operator

Left & Right Shift Operator



<<
27

$$0000 - - 00 \mid \longrightarrow a=1$$

<<1 $\downarrow \times 2$

$$0000 \longrightarrow 0010 \rightarrow \boxed{2}$$

<<1 $\downarrow \times 2$

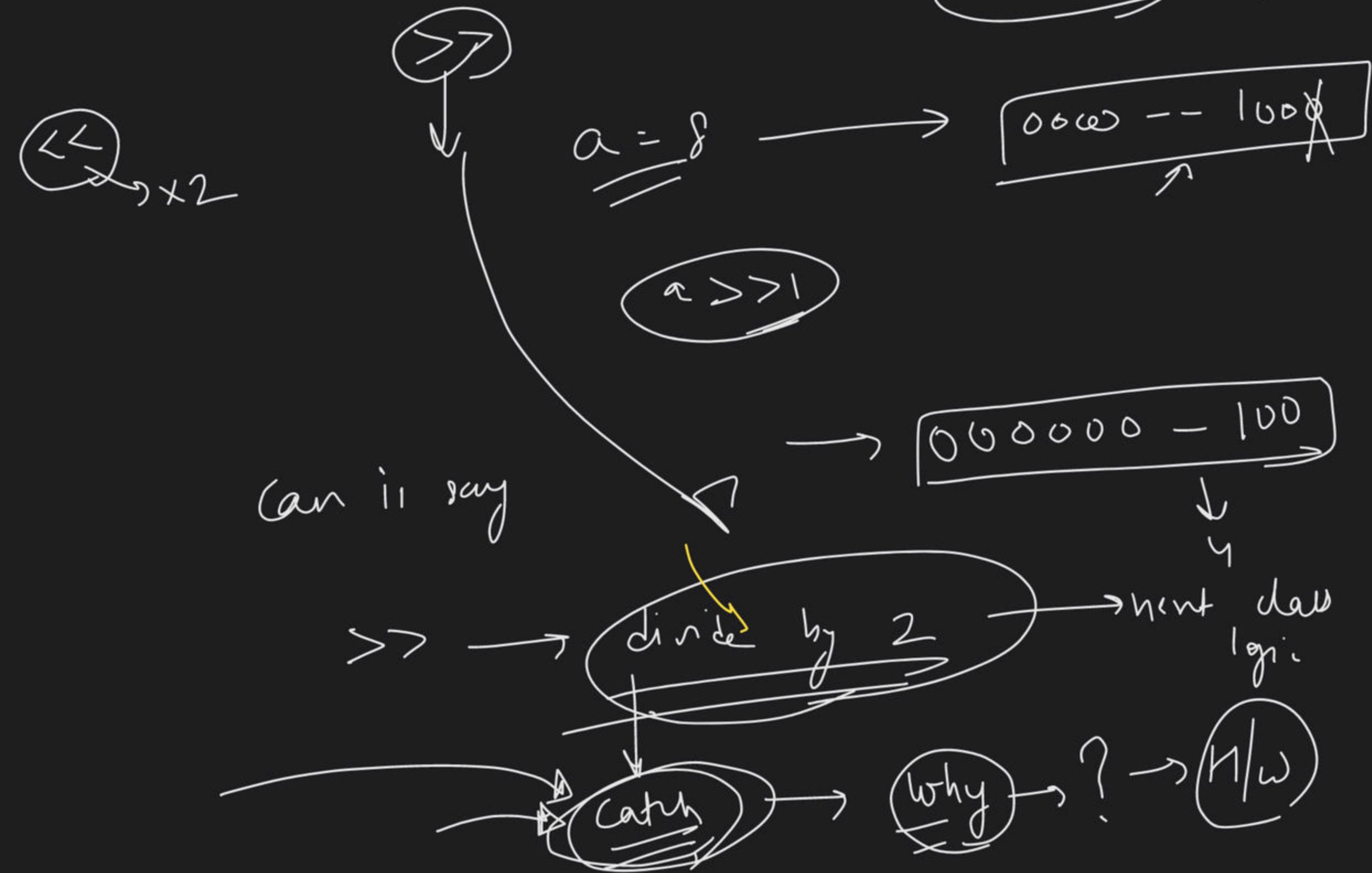
$$000 \longrightarrow 001\cancel{0}0 \rightarrow 4$$

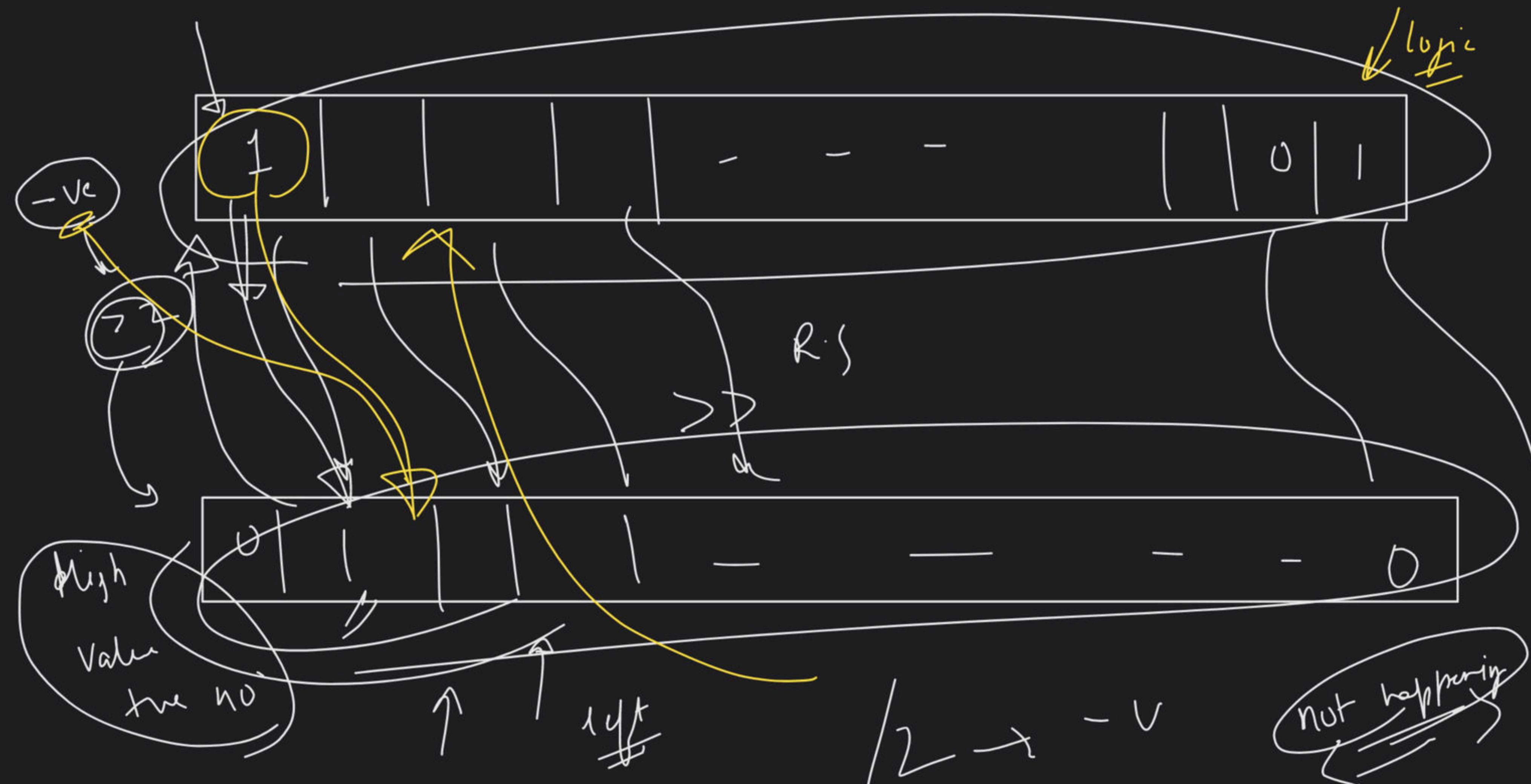
<<1 $\downarrow \times 2$

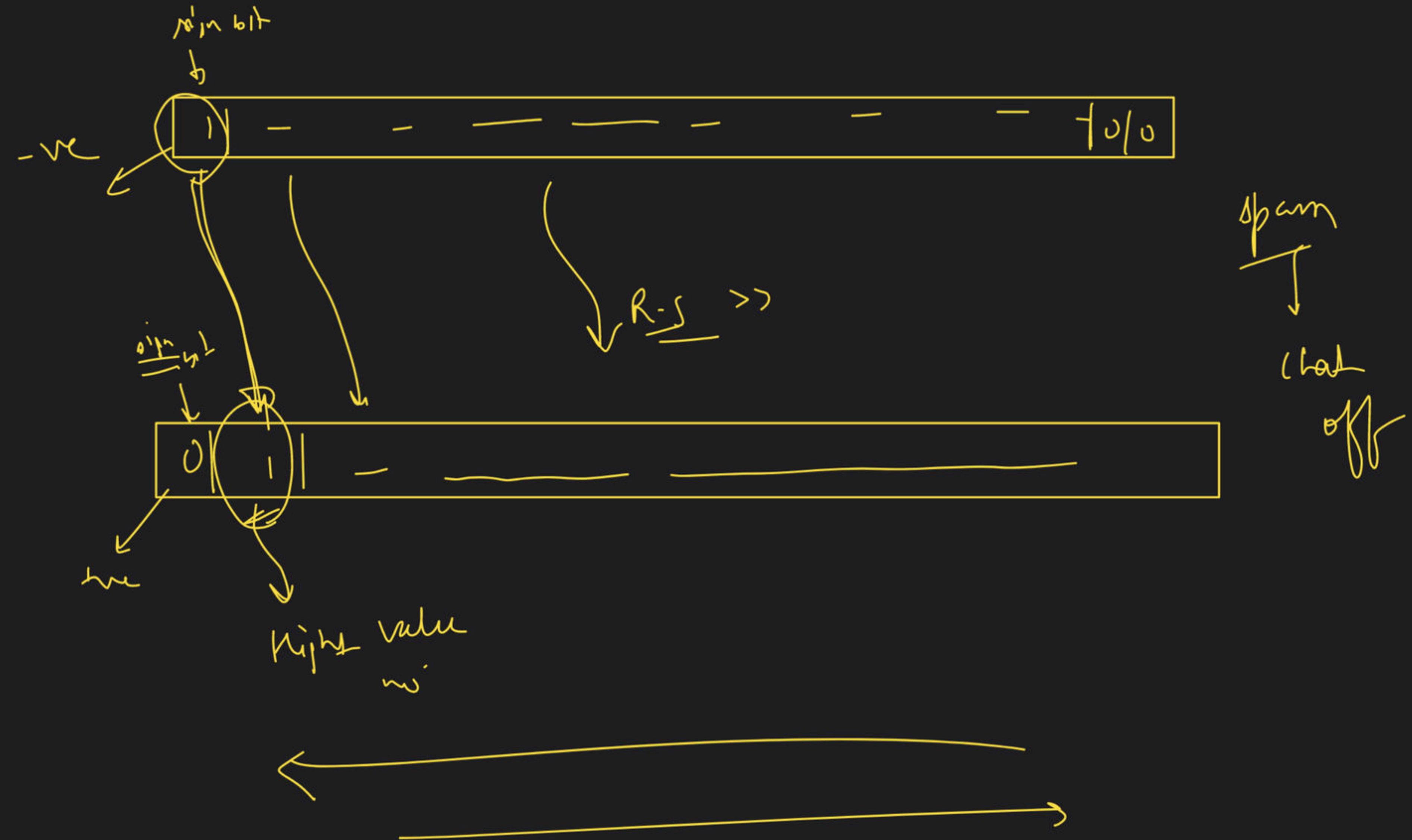
$$00000 \overset{0\cancel{0}00}{\longrightarrow} 8 \rightarrow$$

<<1 $\downarrow \times 2$

$$00000 \overset{(0000)}{\longrightarrow} 10$$

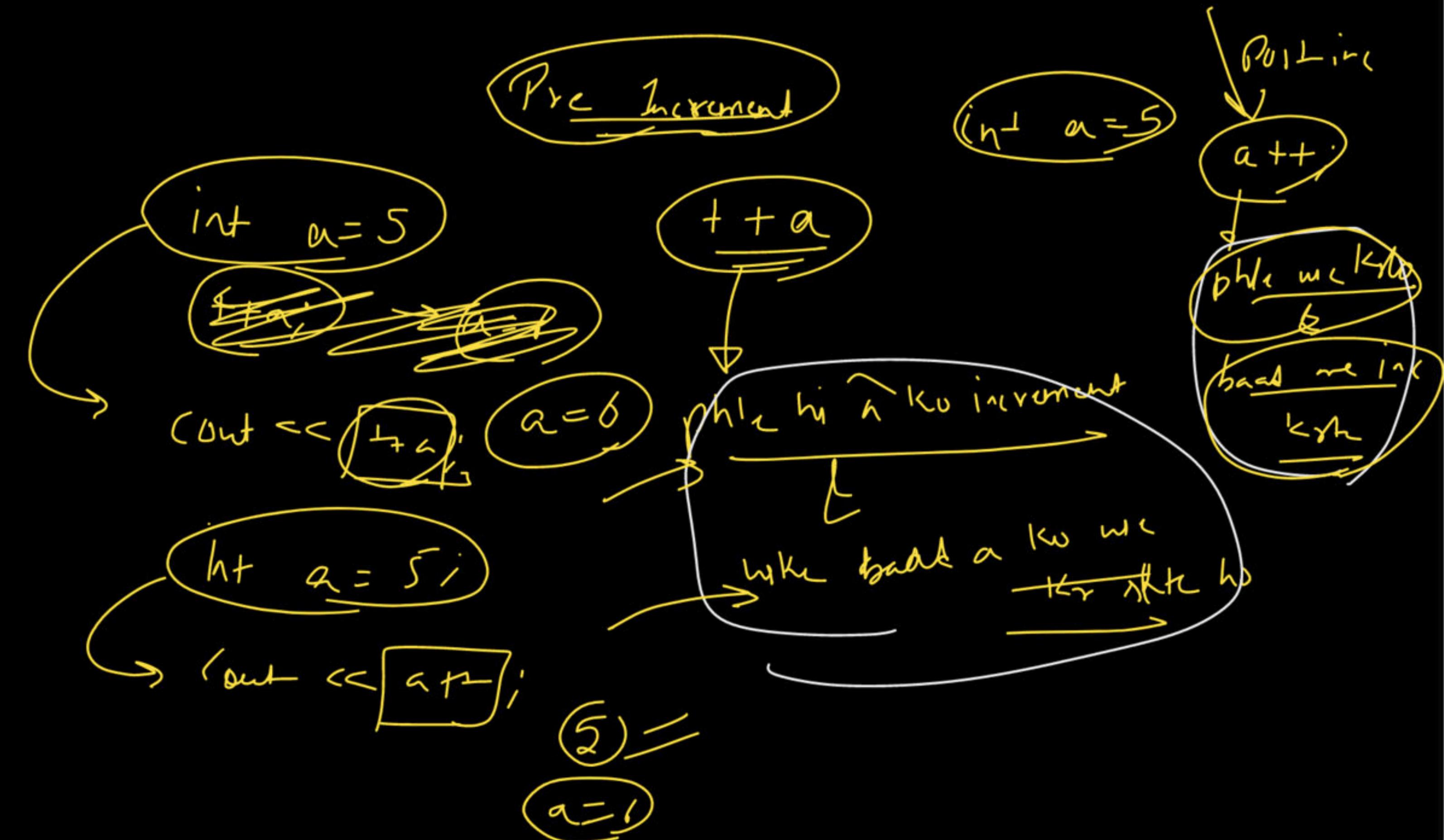






Pre/Post -> Increment/Decrement Operator

- $a++$
- $++a$
- $A--$
- $--A$



main()

{

int a = 5;



cout << a;

}


print

main()

{

int a = 5

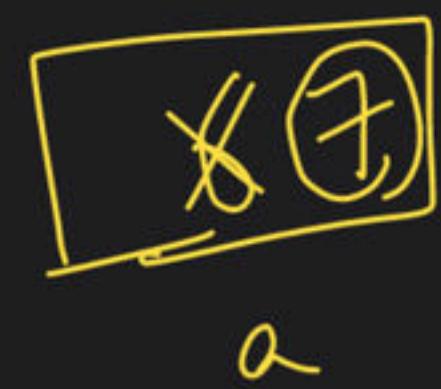
cout << a++ ;

cout << a;

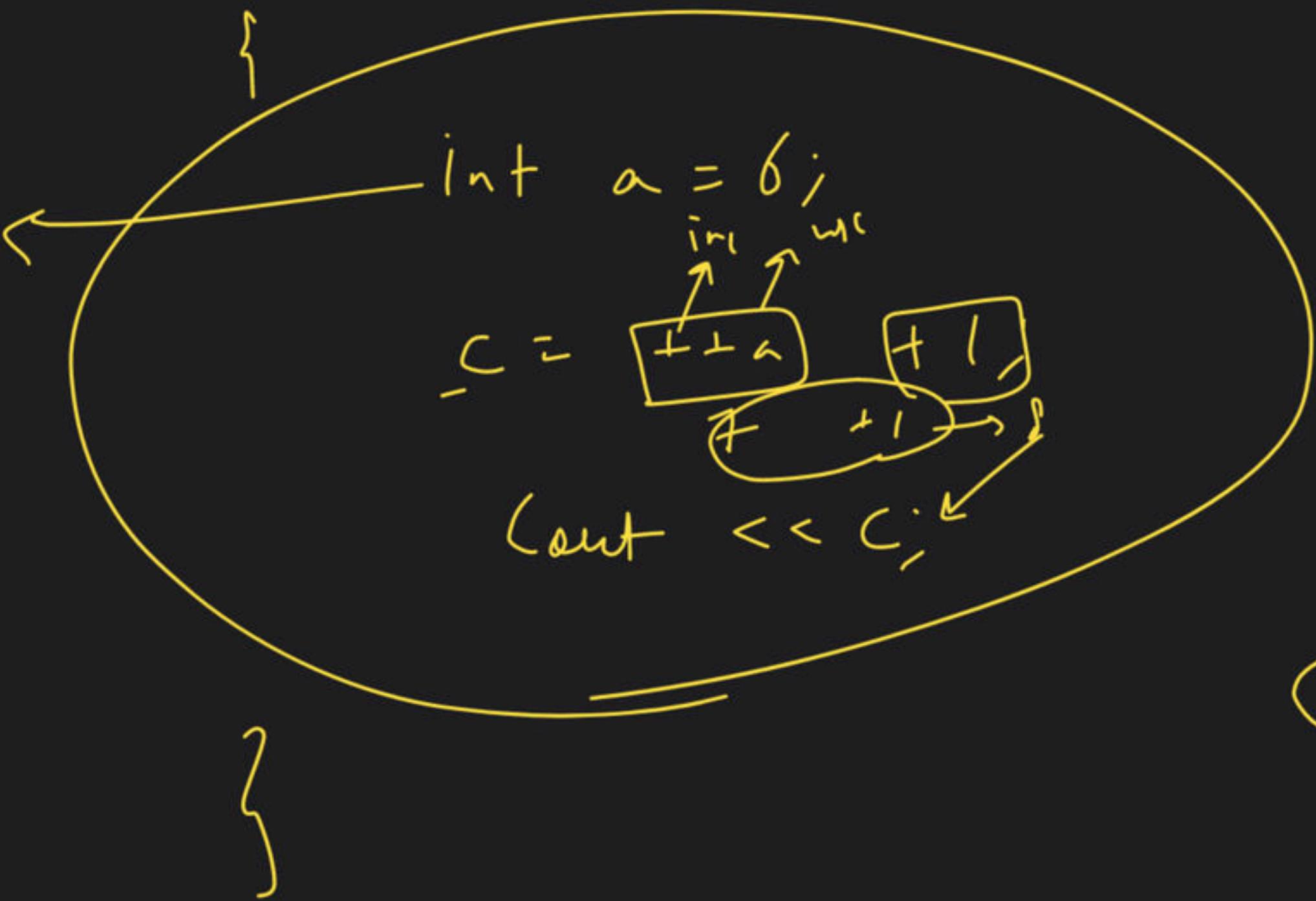


wilkuw
increment

main()



a



1.1.
g → T
T → F

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

~~i = i - 1~~

main()

{

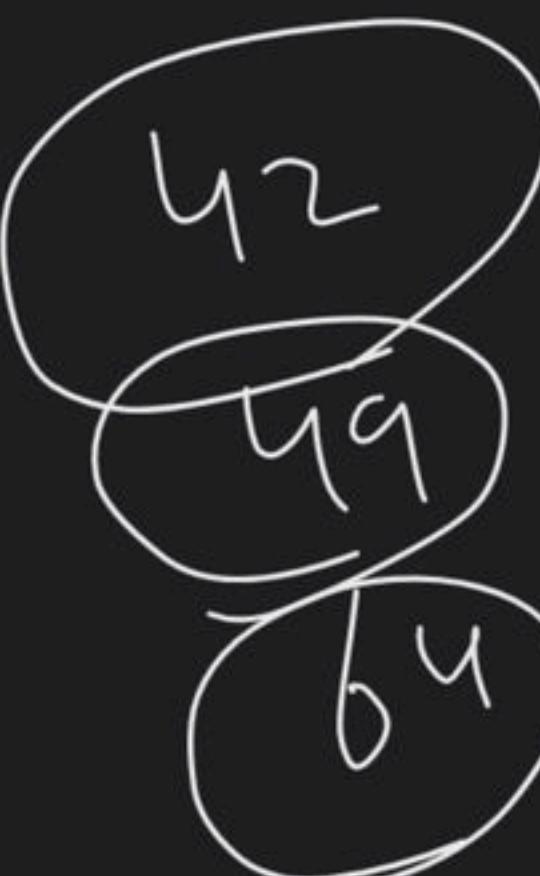
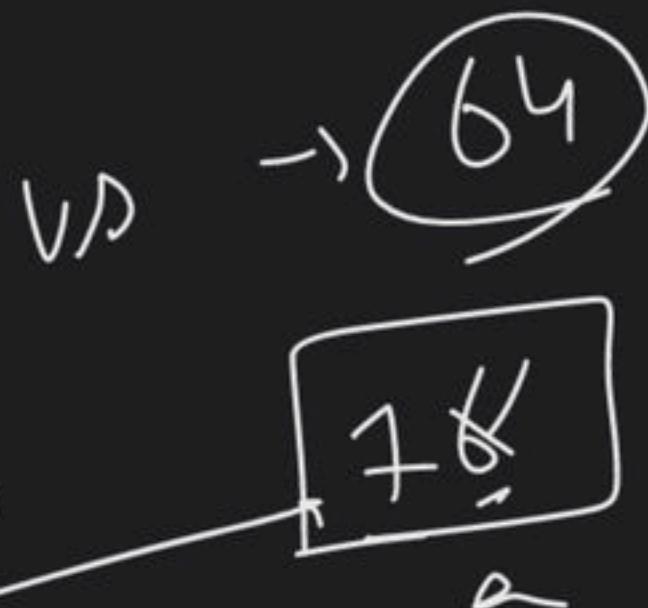
int a = 6;

int c = (a++ + 1);

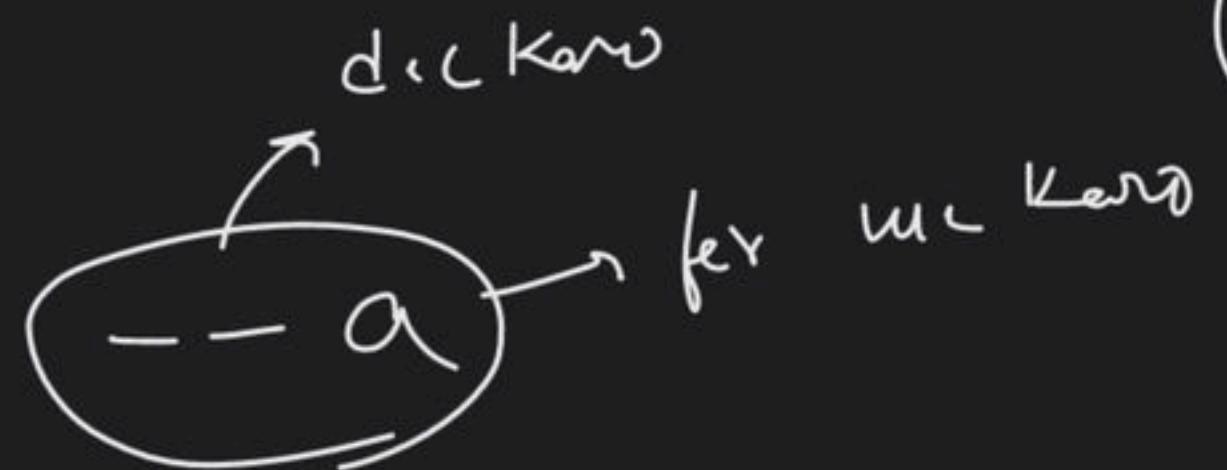
int b = (a++ + 1);

cout << c;

}



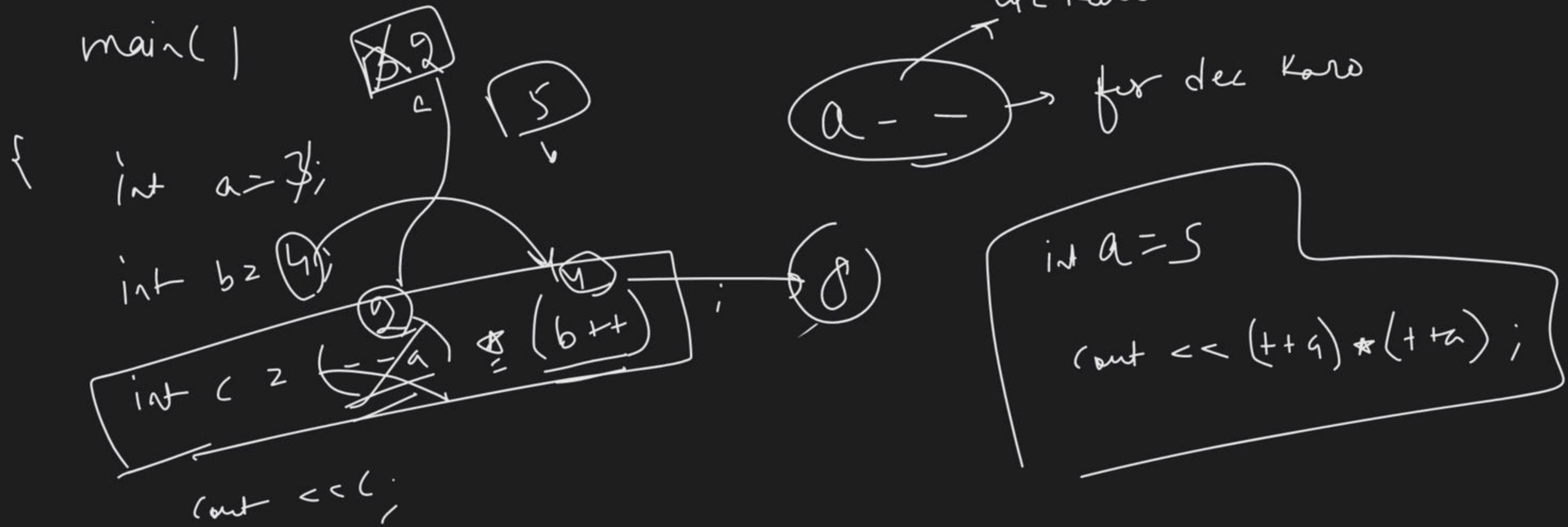
Pre-dec



Operator
Präzedenz

41

main()



Break and Continue Keyword

$n=5$

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

{ ↓

```
cout << "Babbar";
```

break;

for j

more barat take

chart

```
cout << "Love"
```

!

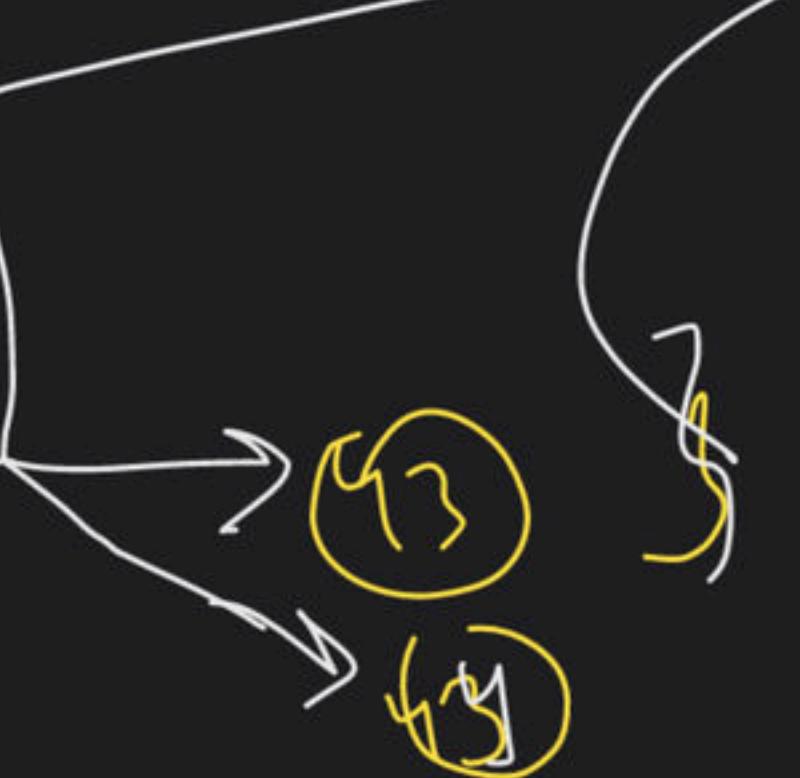
Babbar → String

Love → 1 time

for i

for j

break;



X

for ($i = 0; i < r; i++$)

{

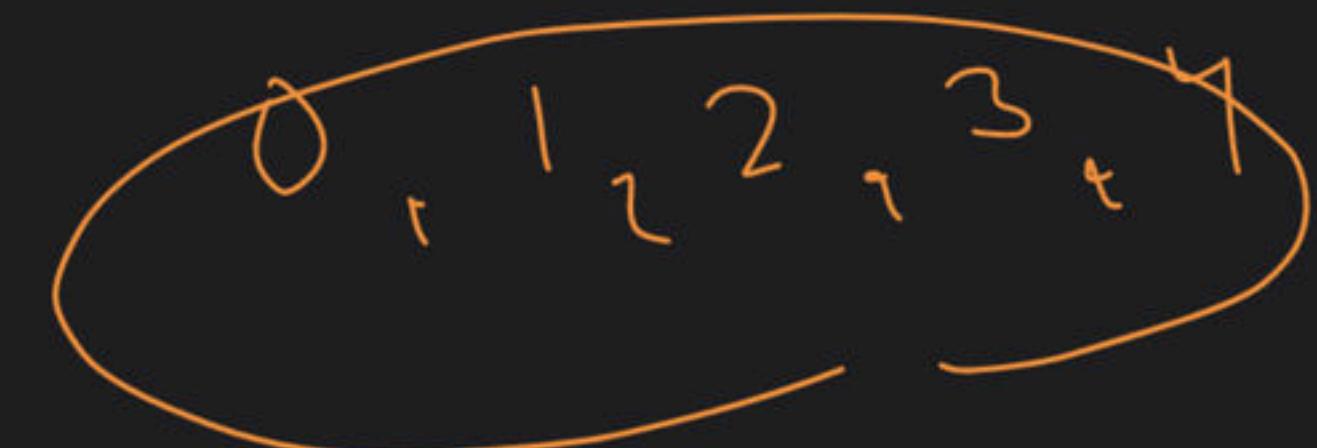
 continue;

 cout << *i;

}

char agli' iteration pr

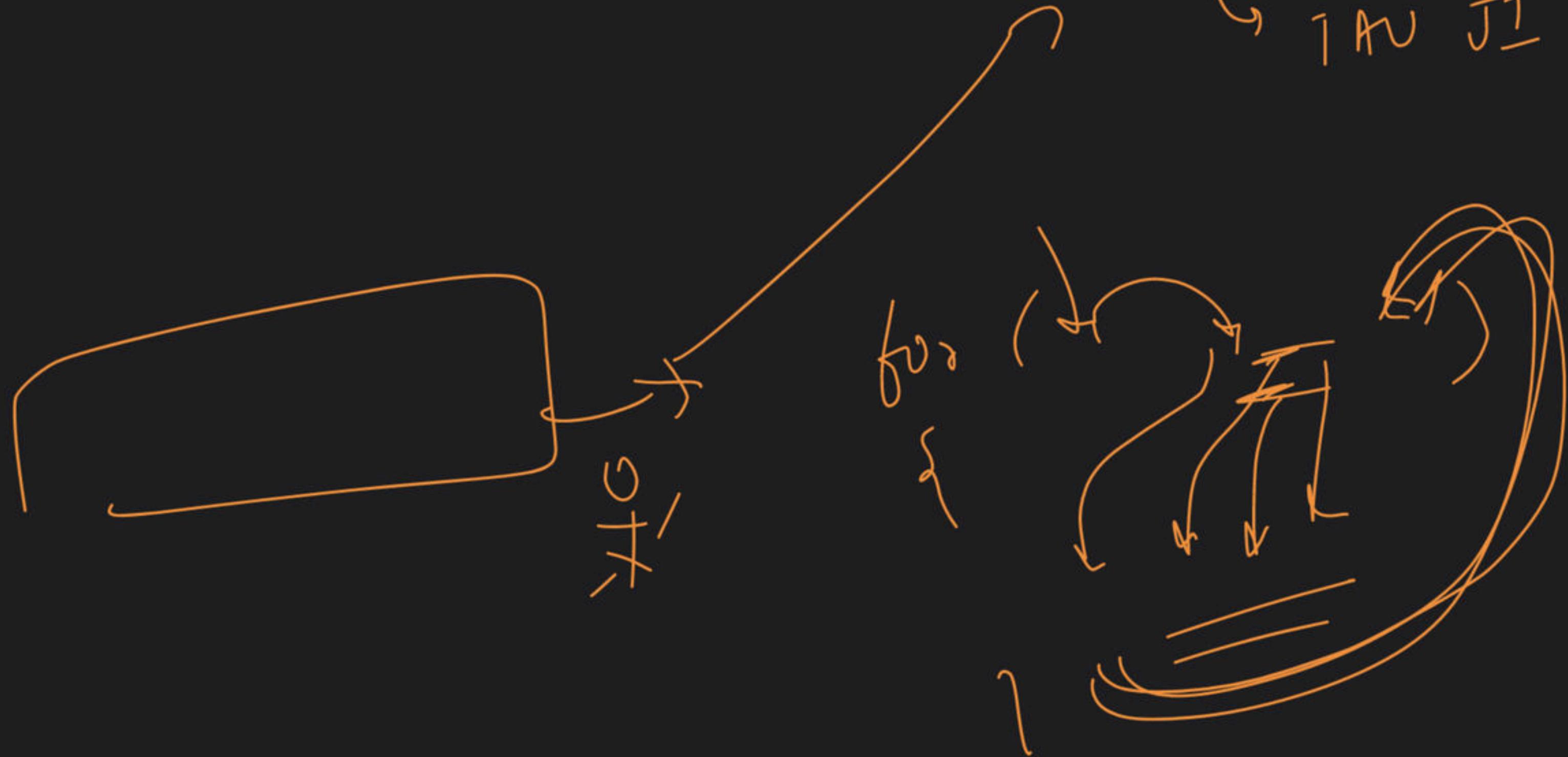
(continue)

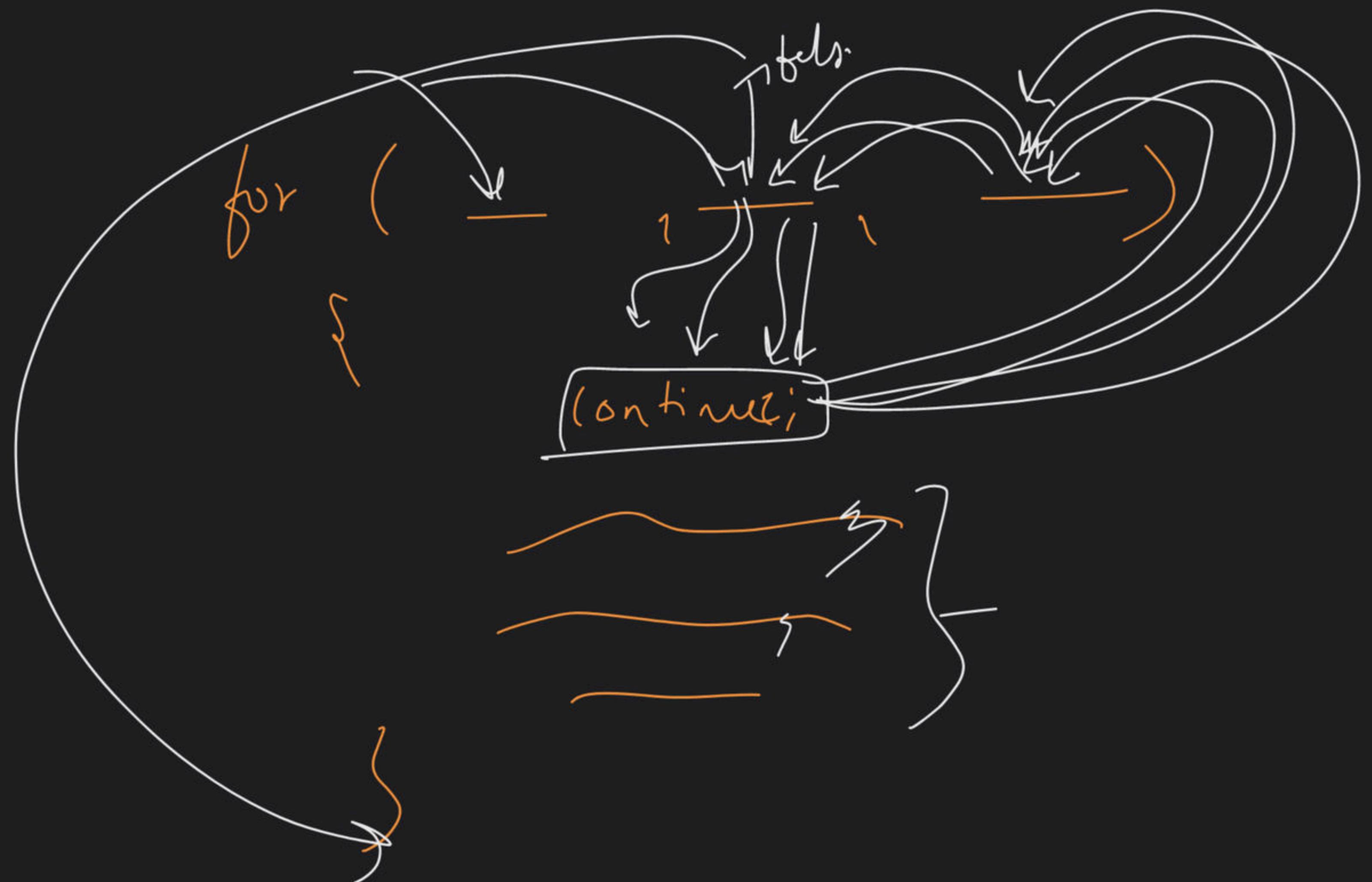


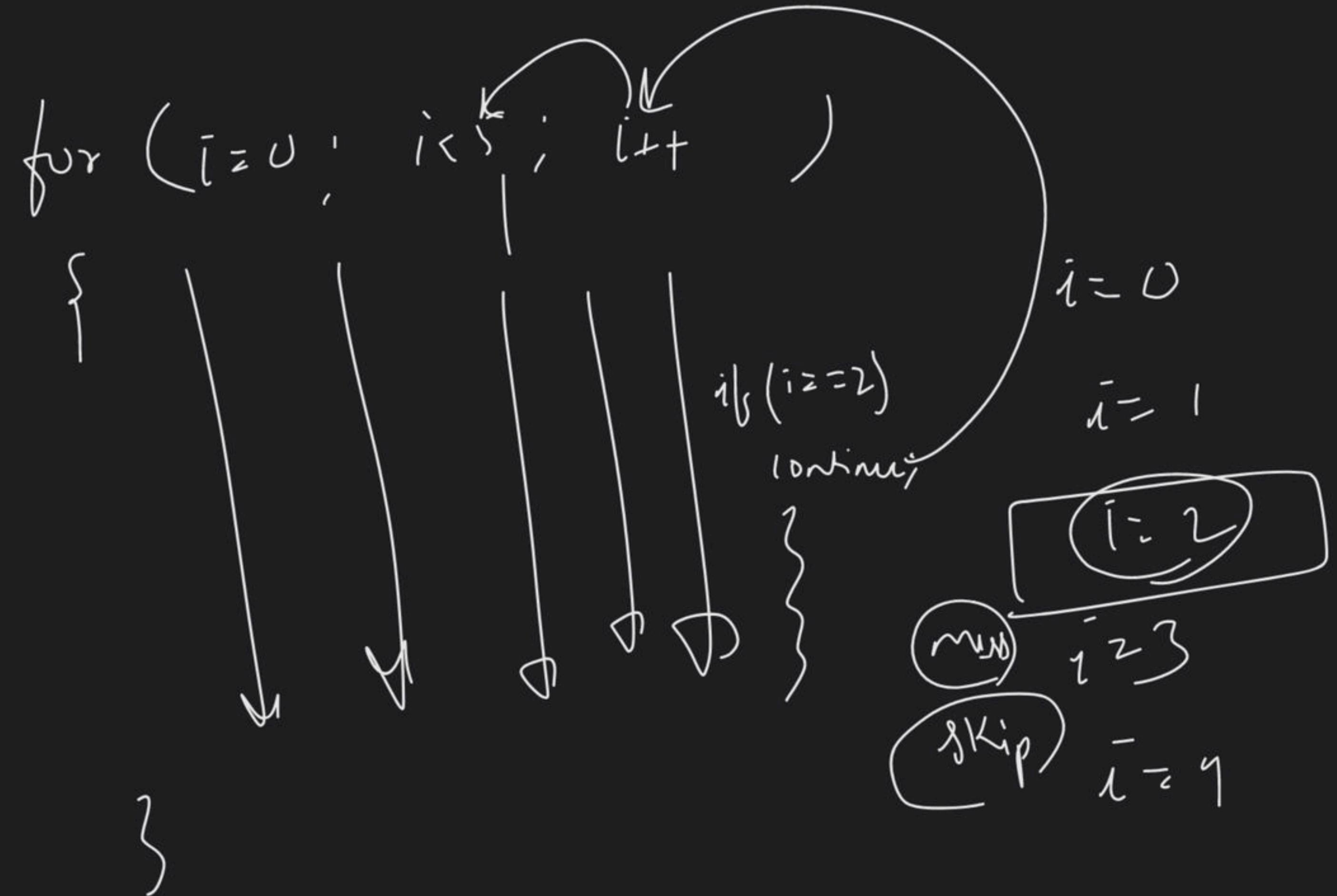
halo halo halo halo

continues

tan J1



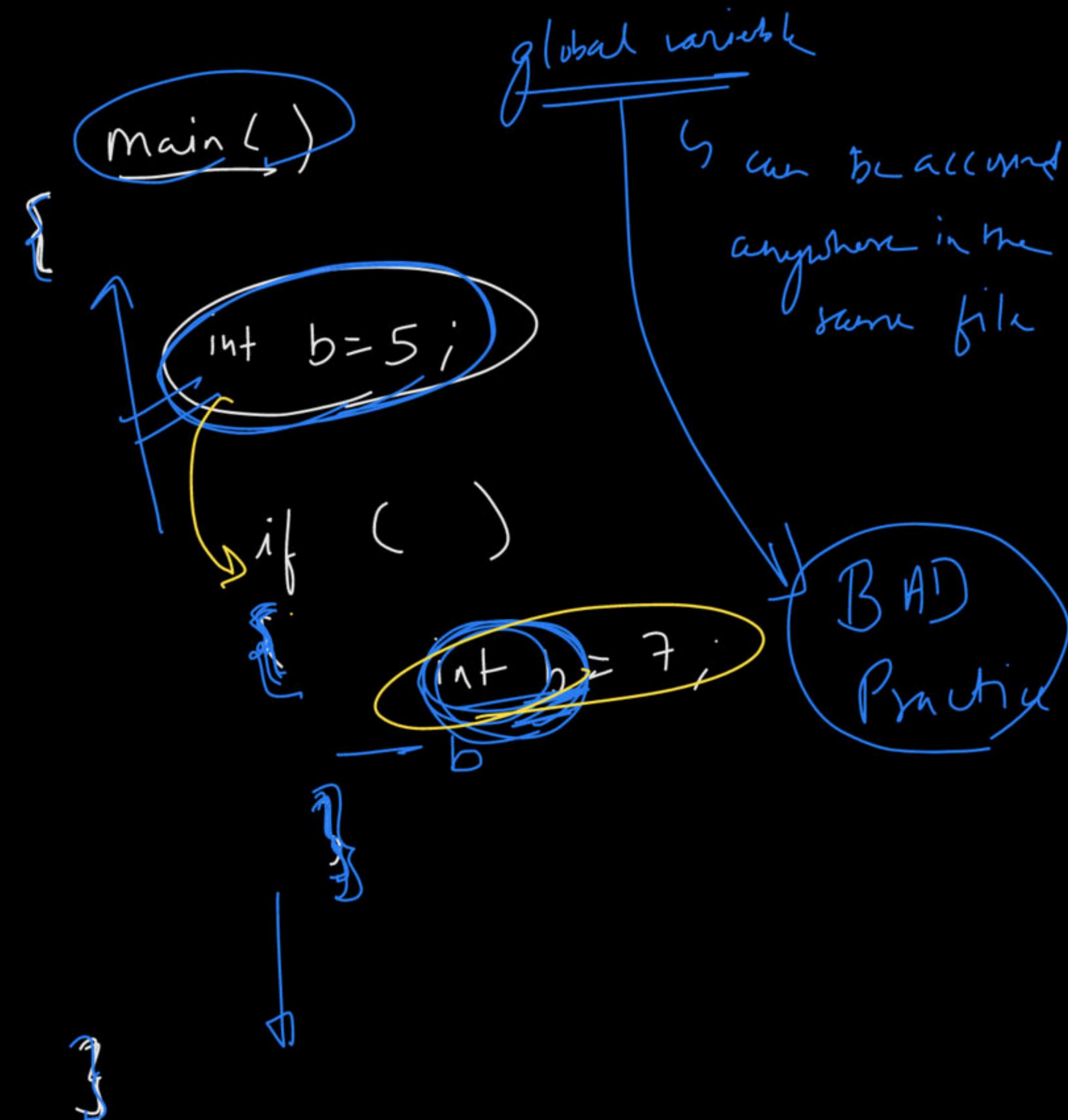




Variable Scoping

{
local variable
weak → open

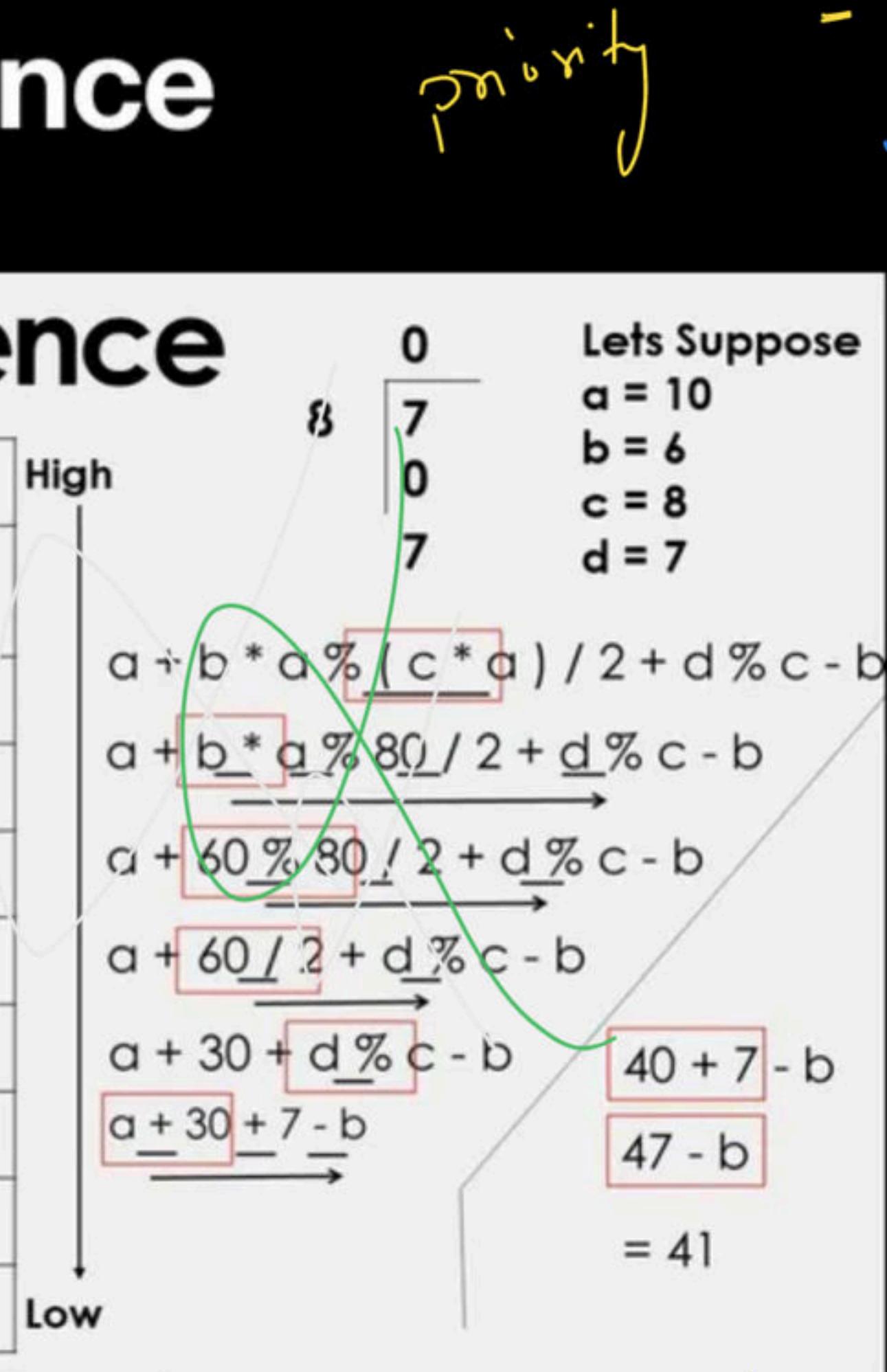
→ ↗ min type



Operator Precedence

Operators precedence

Category	Operators	Associativity
Brackets / Parenthesis	(), [], ->, ., ++, --	Left to Right
Unary	!, ~, ++, --, &, +, -	Right to Left
Multiplicative	*	Left to Right
Additive	+, -	Left to Right
Bitwise Shift	>>, <<	Left to Right
Relational	<, >, <=, >=	Left to Right
Equality	==, !=	Left to Right
Bitwise AND	&	Left to Right
Bitwise XOR	^	Left to Right



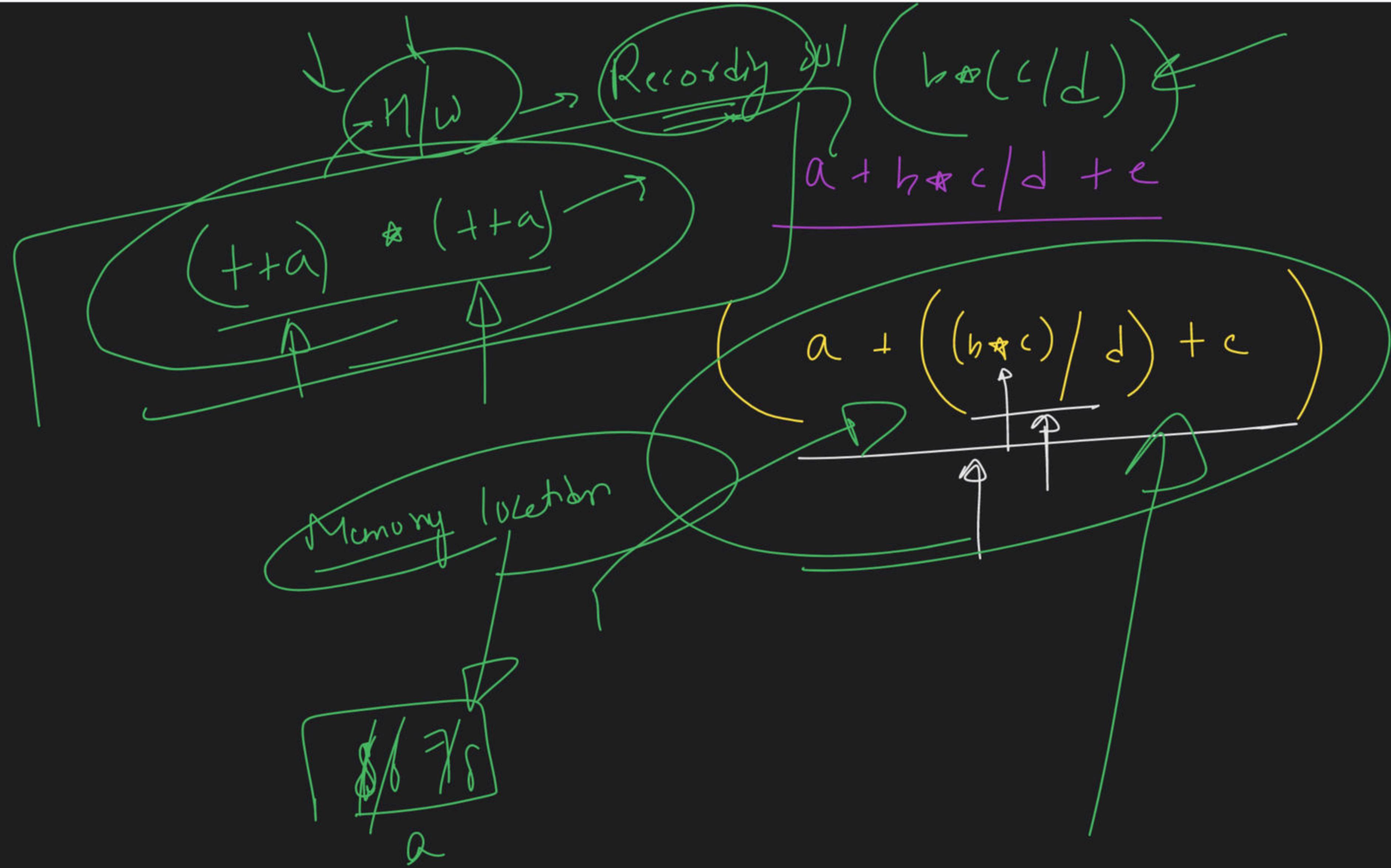
- + \star int

$a = (a + b * c / d) - a$

$2 + 3 / 2 - 1$
 $2 + 1 - 1 = 2$

$2 \pm (3 * 1) / 3 - 1$
 $5 * 1 / 3 - 1$
 $5 / 3 - 1$
 $1 - 1 = 0$

NO



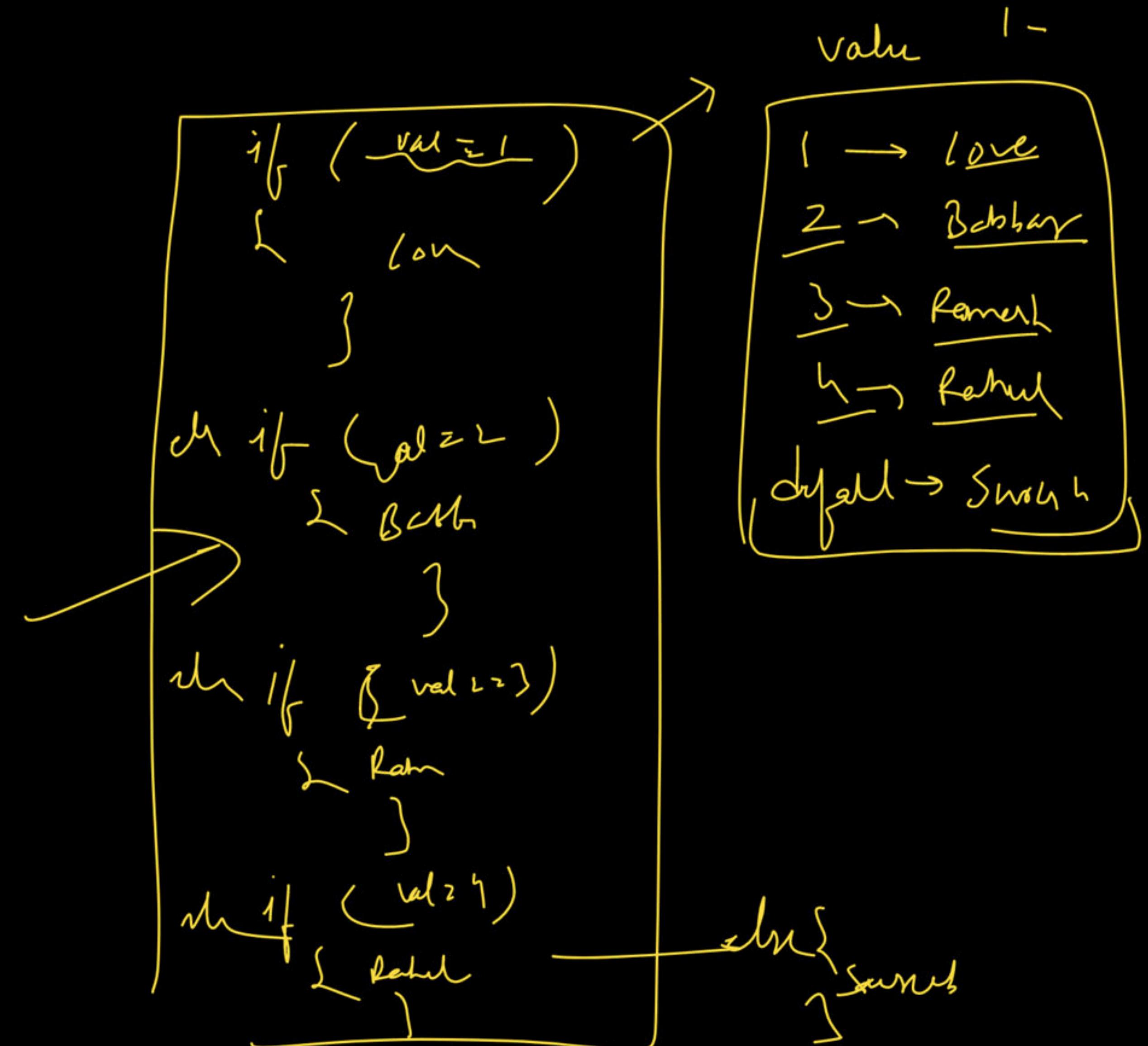
Can we use “continue” in switch case ?

You or No ?

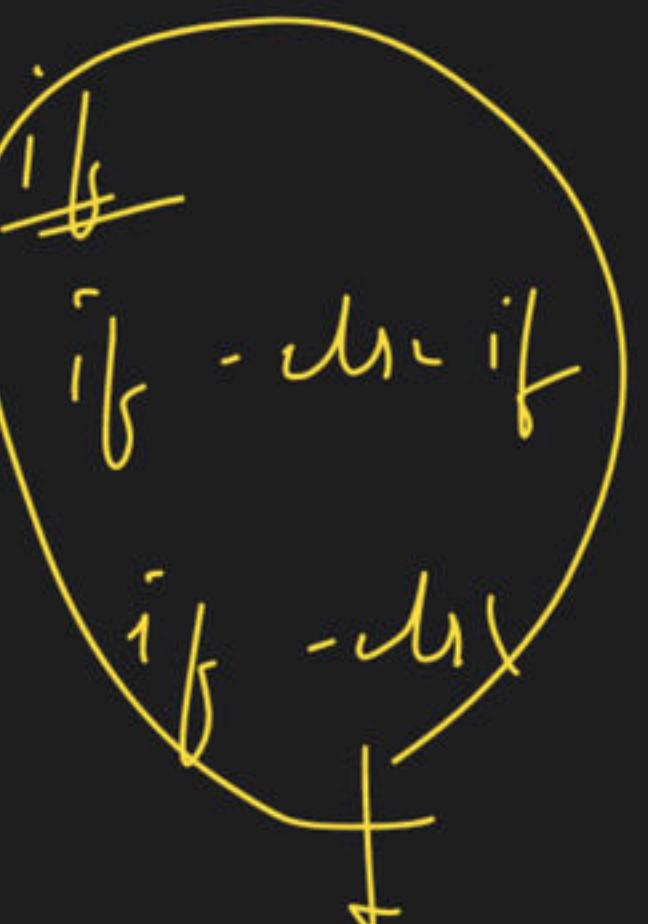
Switch Case:

```
switch (expr)
{
    case 1: -----
        break;

    case 2: -----
        break;
    ...
    default: -----
        -----
}
```

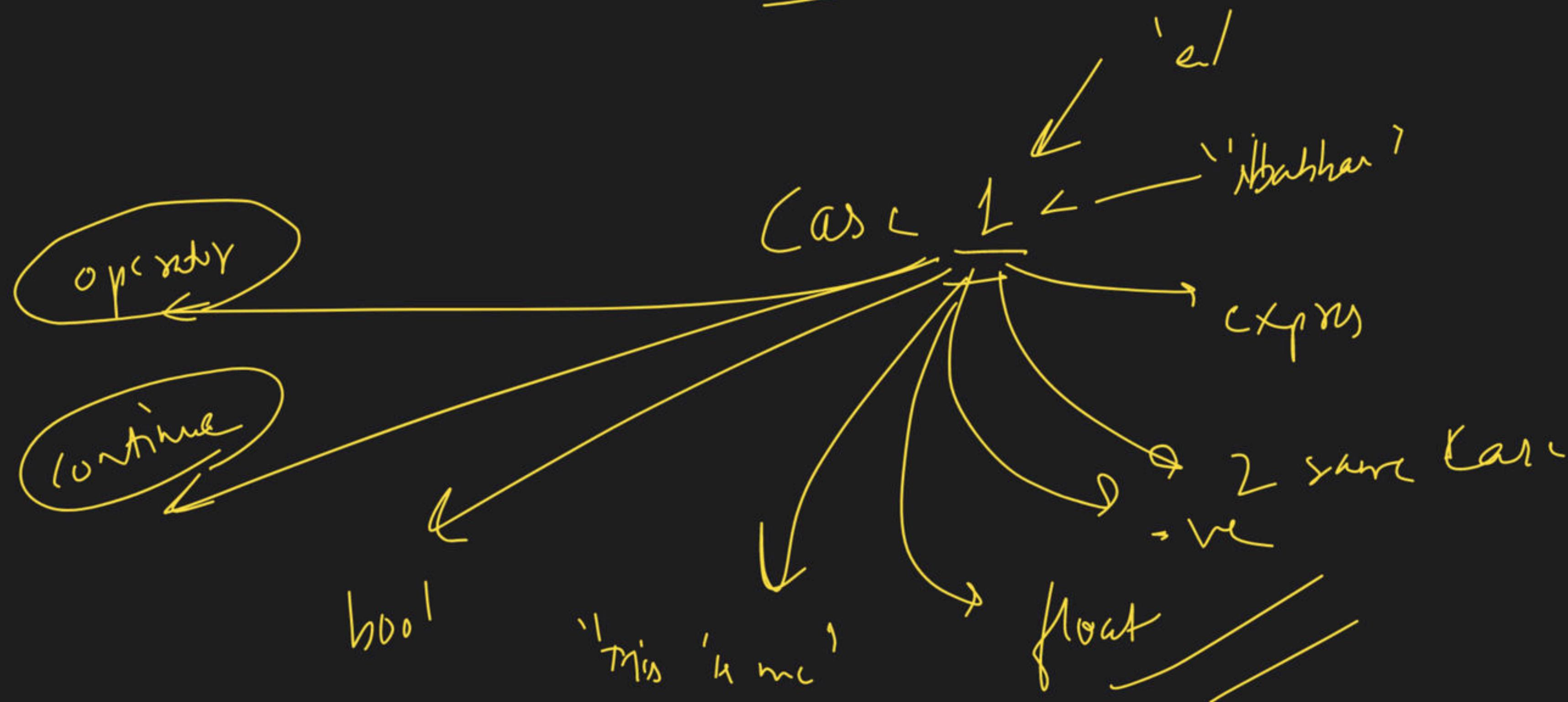


```
switch (val)
{
    case 1: if (colt << "love";
                break;
    case 2: cout << "BABAHR";
              break; continue
    case 3: cout << Ramah;
              break;
    case 4: cout << "fahul";
              break;
    default: cout << "suruh"
}
```



Variable

Switch case



12 - 2 pm

for (int i= 0; i<n

{

}

i++
or
+ + i

- m'

hi

hi+