

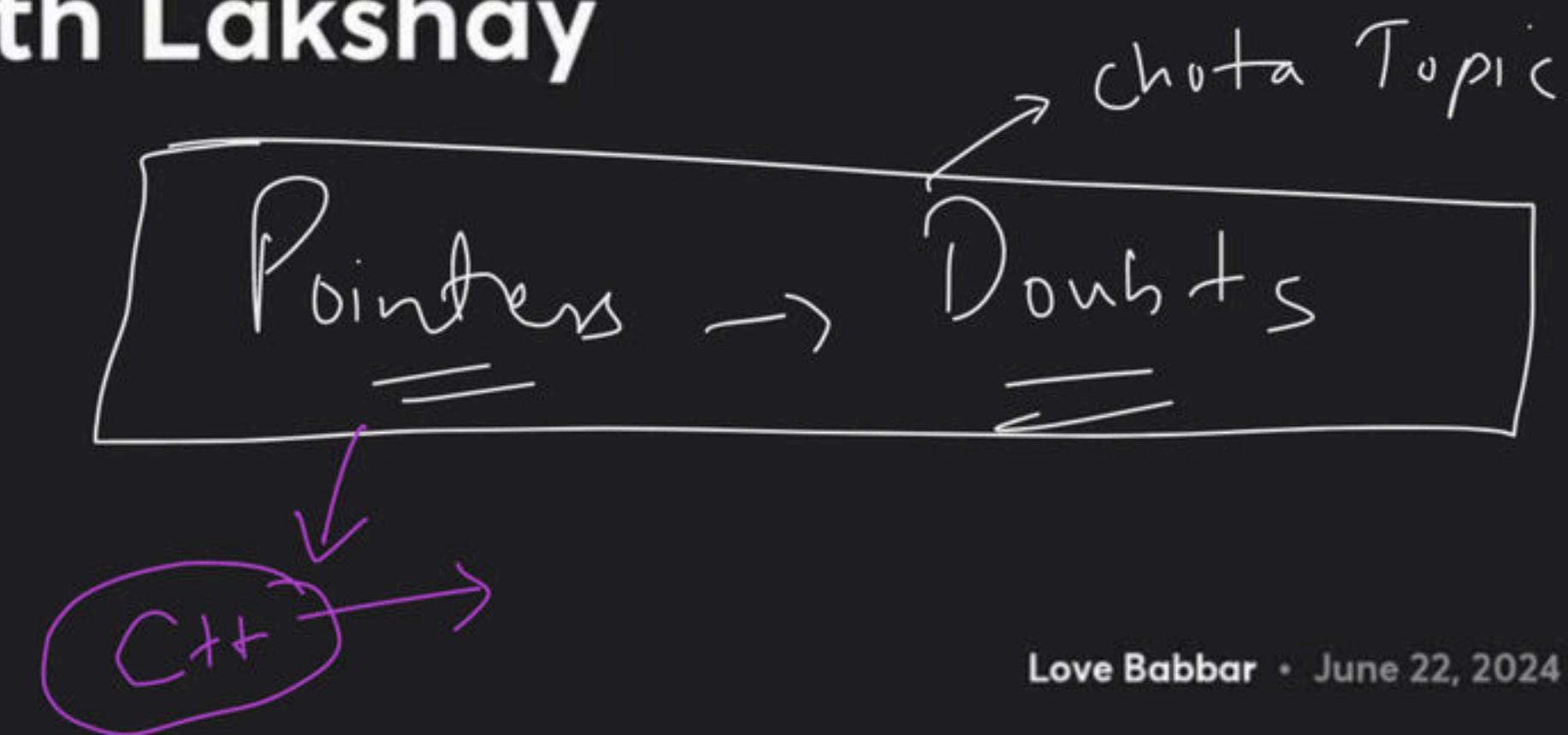
String / char array Mega Class

23 June

2 PM

Doubt Class With Lakshay

Special class





Pointers - Class 1

Special class

Maths

=

Prime ??

no.

DSA Maths

① Naive

factors $1, N$

$[2 \rightarrow N-1]$

↳ Non-prime

② Sqrt

$[2 \rightarrow \sqrt{N}]$

③ Sieve

of Eratosthenes

12 Sieve

F	F	T	T	F	T	F	T	F	F	F	T	F
0	1	2	3	4	5	6	7	8	9	10	11	12

2 → 4, 6, 8, 10, 12, ~~14~~

3 → 6, 9, 12,

3×2

Optimising Sieve = Sieve of E



inner loop

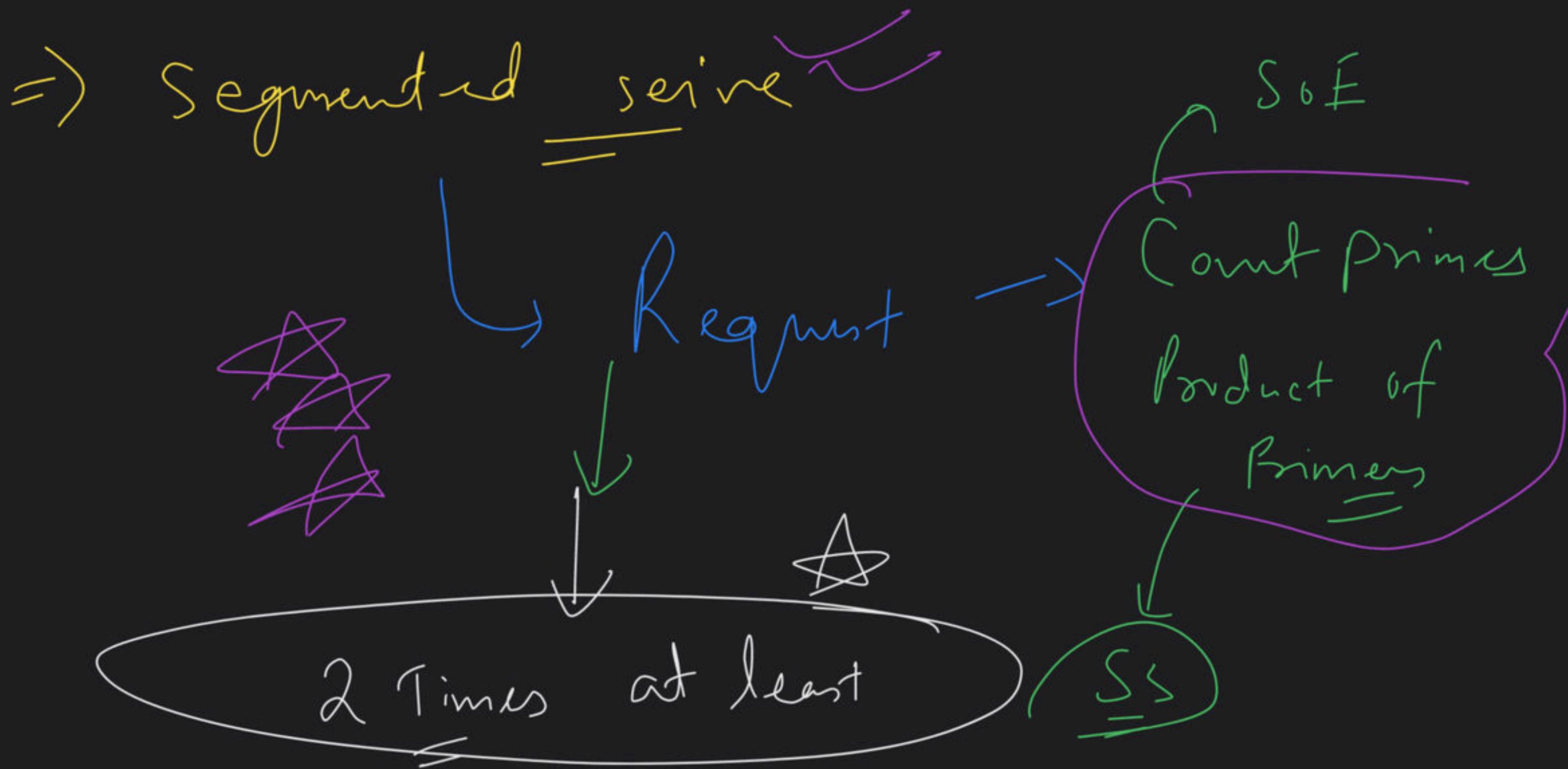
$$\Rightarrow j = i \times 2 \times \cancel{\times}$$

$$j = i \times i$$

②

outer loop \Rightarrow

$$i \leq \sqrt{n}$$



Basic Maths

=

GCD / HCF

=

=

LCM

\Rightarrow

HCF

$$a \times b =$$

LCM

\times HCF

\Rightarrow Fast exponent
=====

$$a^b$$

\Rightarrow



$$a \times a \cdots \times a$$

b times

$$2 \Rightarrow 2 \times 2 \times 2 \times 2 = 32$$

$$O(b)$$

$$\Rightarrow a^b \Rightarrow O(l_s b)$$

$$\Rightarrow a^b \Rightarrow$$

$b \text{ even} =$

$$a^b = a^{b/2} \cdot a^{b/2}$$

$$b \text{ odd} \Rightarrow a^b = (a^{b/2} \cdot a^{b/2}) \cdot a$$

$$2^{10} \Rightarrow 2^5 \cdot 2^5, \quad 2^{11} \Rightarrow \overline{(2^5 \cdot 2^5) 2^1}$$

$$\begin{array}{c} 2^5 \\ \downarrow \\ (2^2 \cdot 2^2) \cdot 2 \\ \text{even} \quad \text{even} \\ (2 \cdot 2) \cdot (2 \cdot 2) \cdot 2 \end{array}$$

odd

\Rightarrow

$$\text{ans} = 1$$

$$/ \quad a = 5, \quad b = 4$$

① $b \rightarrow \text{even} \rightarrow$

$$a = \underbrace{a \times a}_{2 \times 2} \Rightarrow$$

$$(5 \times 5) \Rightarrow a = 25$$

② $b = 2 \rightarrow$

$$a = a \times a \Rightarrow$$

$$(25 \times 25) \Rightarrow 625$$

=

③ $b = 1 \Rightarrow$

$$\text{ans} = \text{ans} \times a \Rightarrow$$

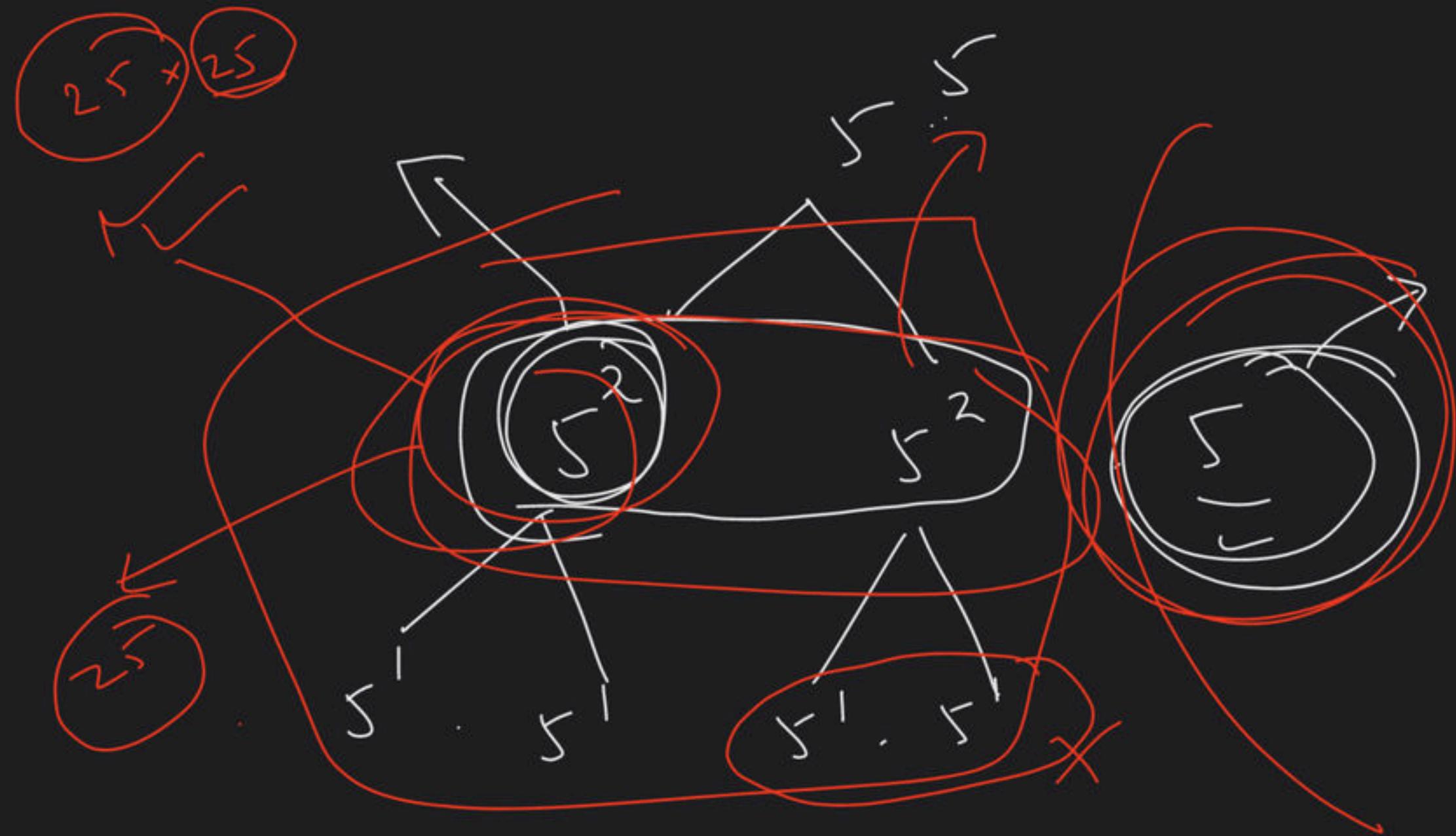
$$1 \times 625 \rightarrow$$

$$625$$

$$\omega_m = 1$$

$$3\sqrt{25}$$

$$a = 5, b = 5$$



$$(x^2)^2 \cdot 5 =$$

$$am = 1$$

\Rightarrow
① $b = 5$, odd \Rightarrow

$$am = rm * a \Rightarrow$$

$$a = 5, b = 5$$



$$a = a \times a \Rightarrow$$

$$5 \times 5 \Rightarrow 25$$

② $b = 2$, even \Rightarrow

$$a = a \times a \Rightarrow$$

$$25 \times 25 = 625$$

③ $b = 1$, odd \Rightarrow

$$am = rm * a \Rightarrow$$

$$am = 1 * 625 \Rightarrow 625$$

$\sim s \times \sim s$



$\nwarrow \nwarrow \rightarrow$



a^b



\sim

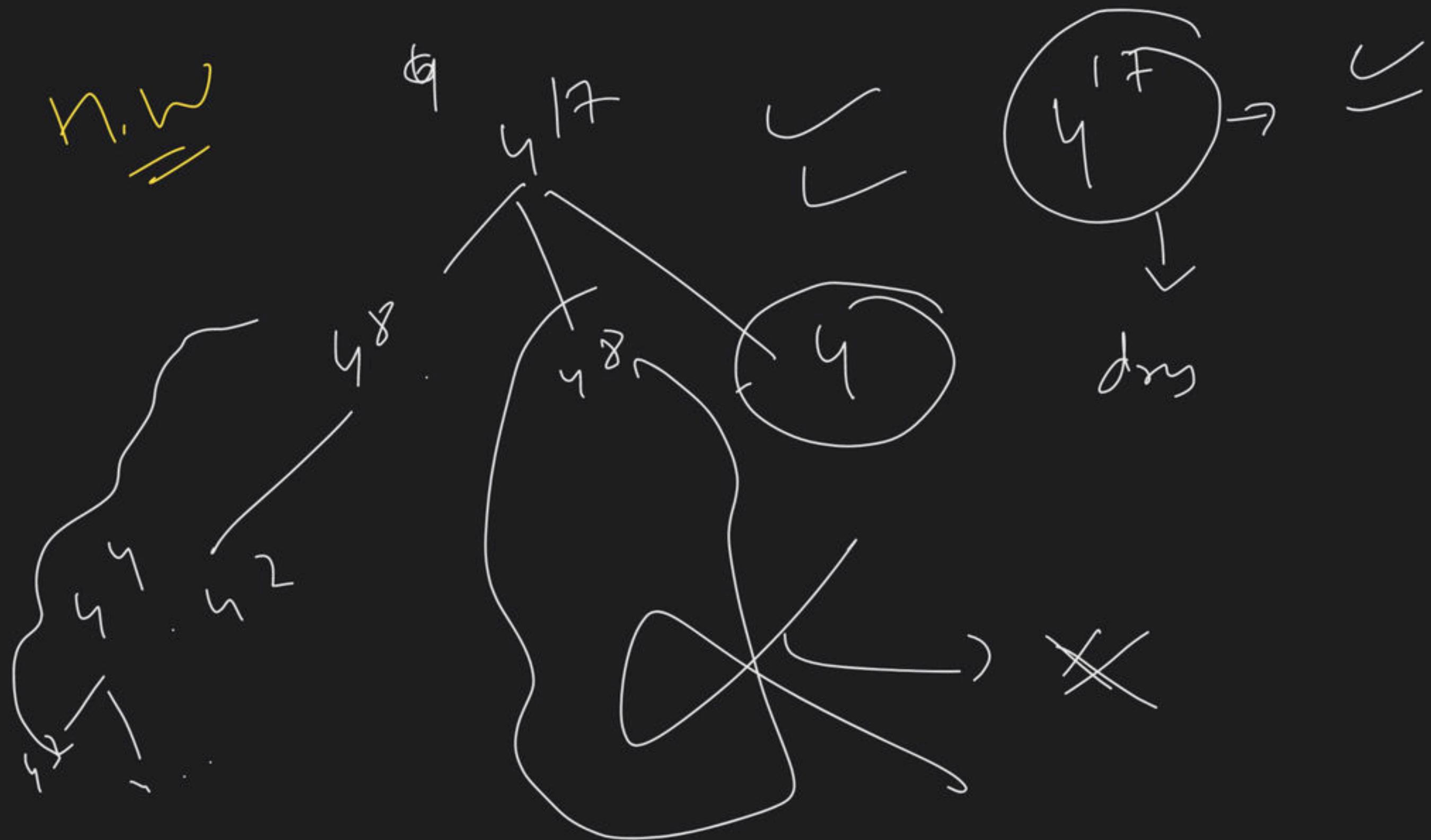
$\sim \sim$

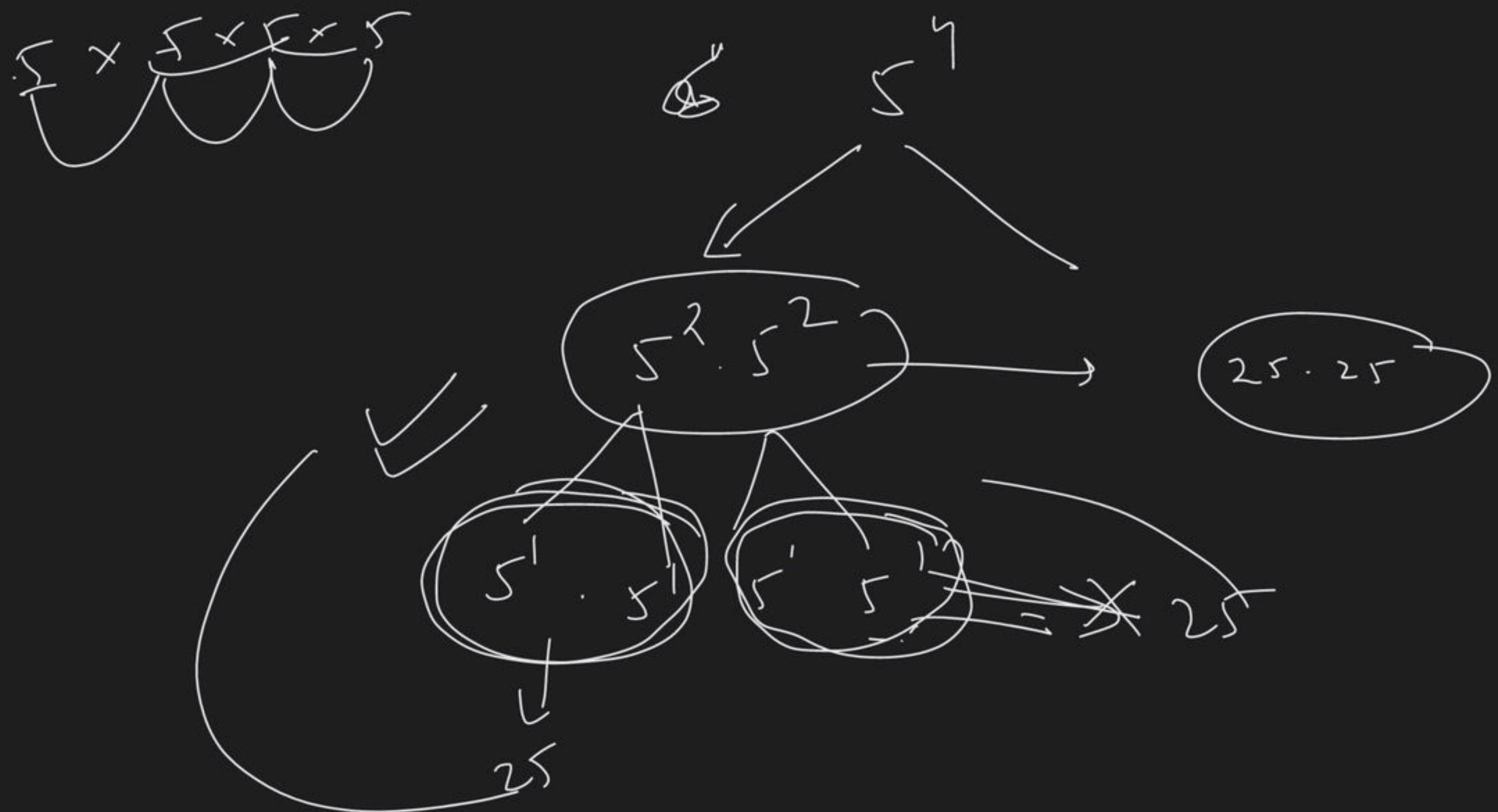
$\sim \sim$

$\sim \sim$

$\sim \sim$

$\sim \sim$





Variable

int a = 5;

lwood

Vegg

Milk

COLD STOCK

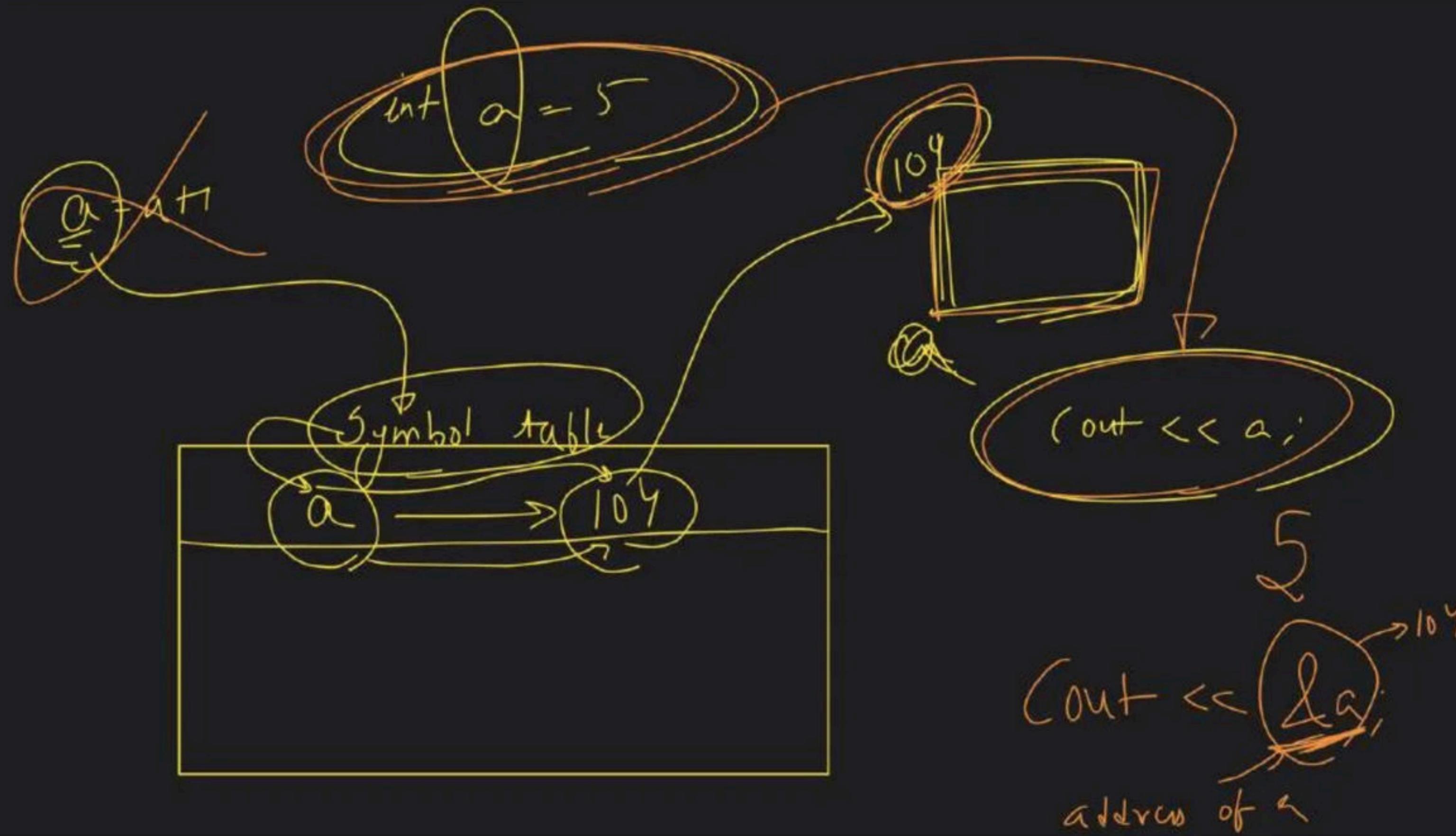
pointer



Storage address

Pointers

address



`int a = 5;`

`cout << a;`

`int b = a;`

~~Int~~

`c = a`

~~la~~

~~var~~

~~pointer~~

~~Pointer~~

~~=~~

`int b = 5`

`int *a = &b;`

~~Int~~

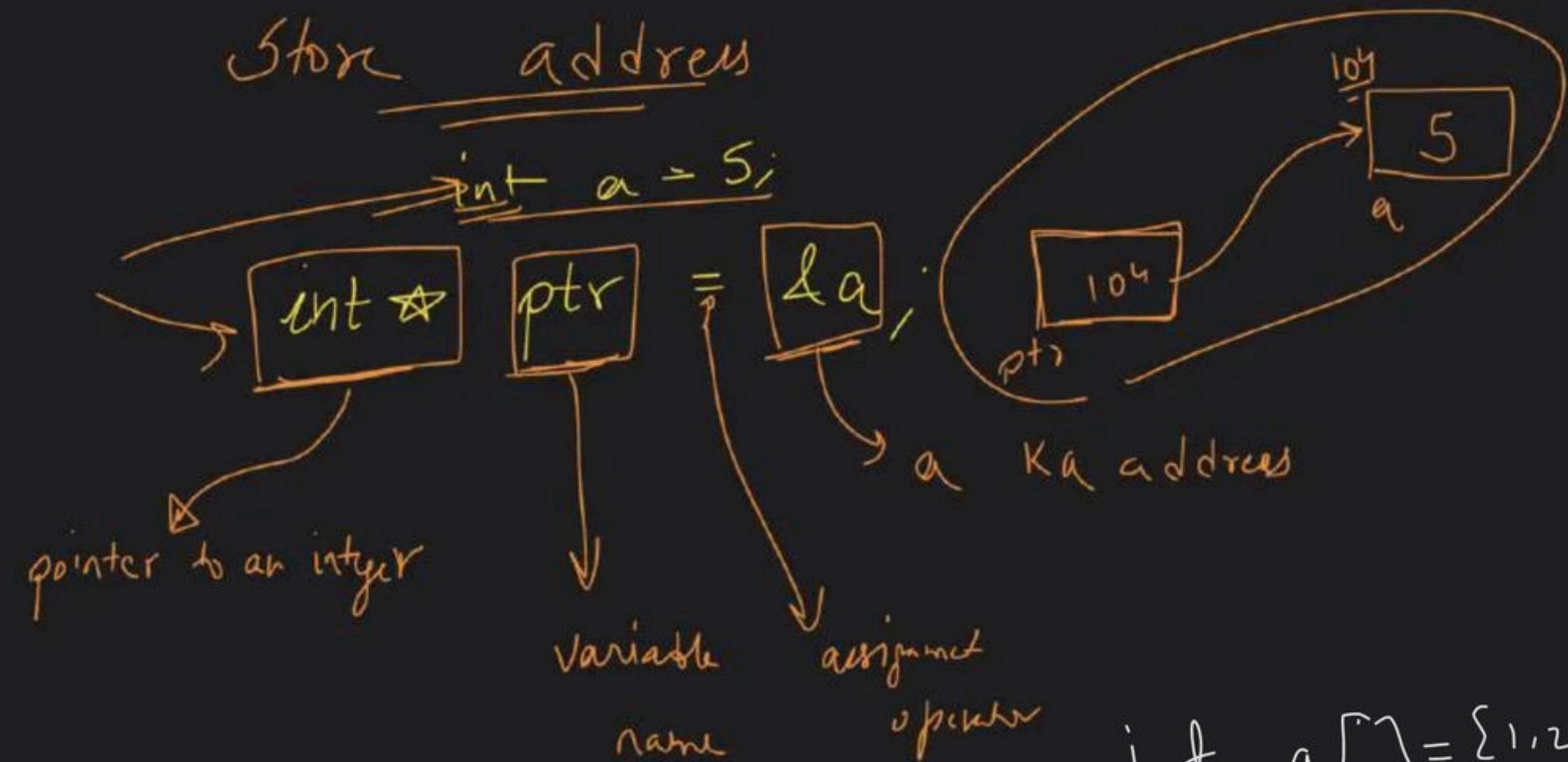
5

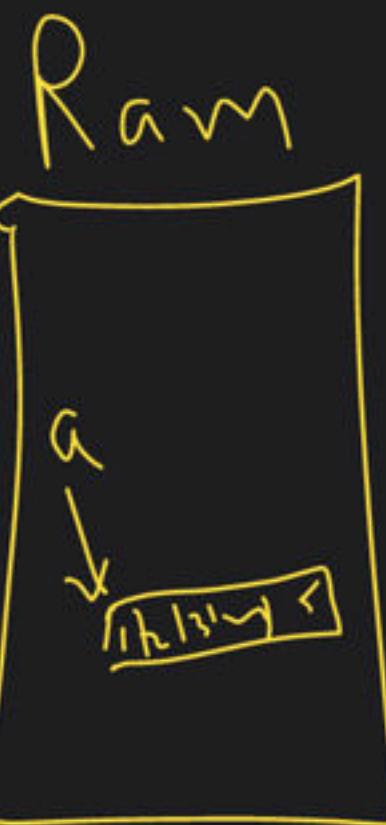
b

Int

105

a
205

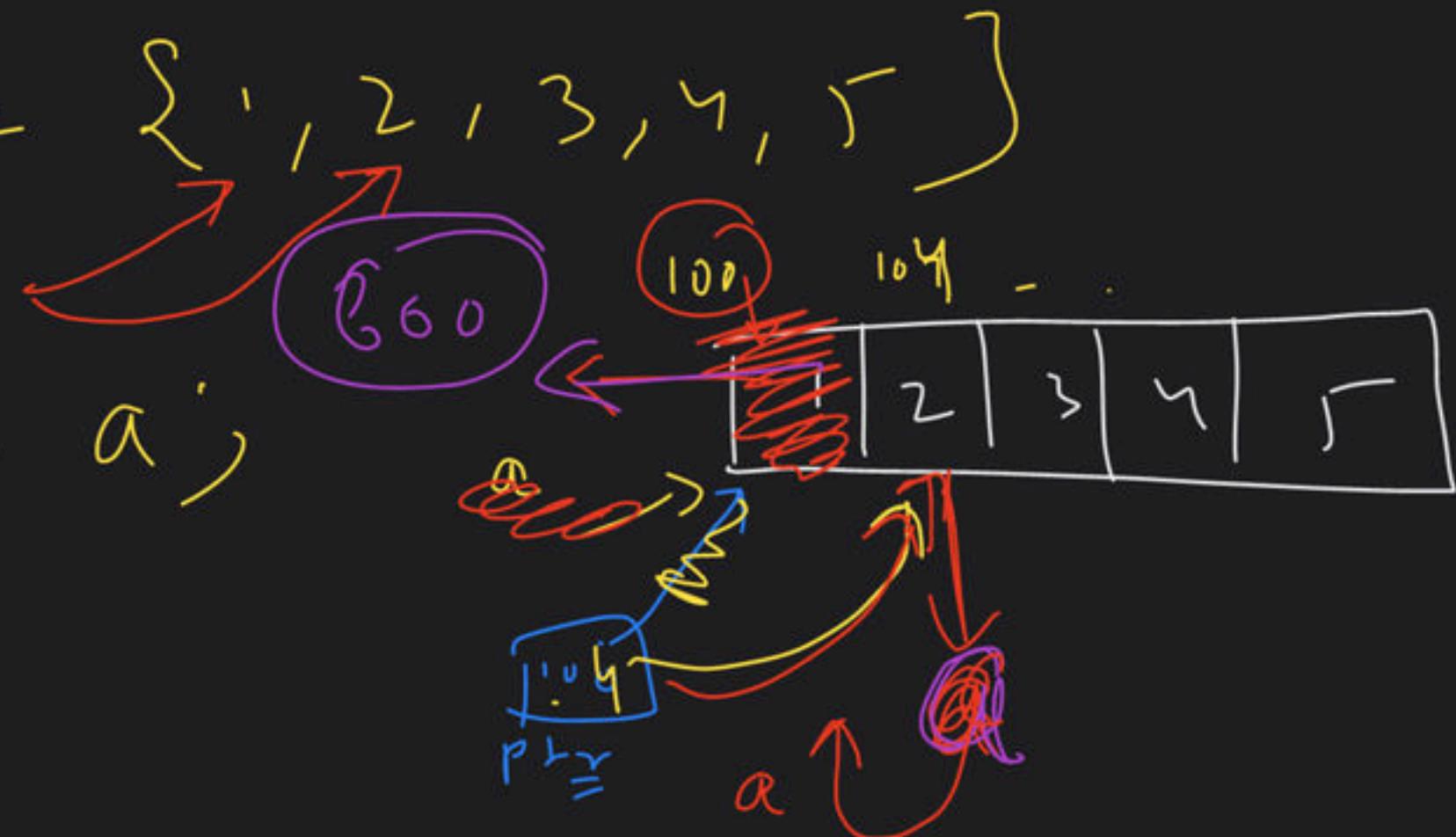




`int a[] = {1, 2, 3, 4, 5}`

`int *ptr = a;`

`ptr++;`



`cout << *ptr << endl;`

`Count = arr[3]`

2

`Count << arr[3] << endl;`

Ref variable

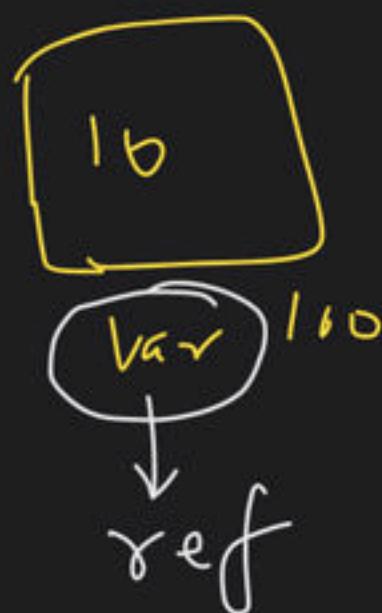
Symbol

100	Var, ref
-----	----------

Pointer (old)

①

int var = 10;



=> has no. memory block

=> Ref variable has no.

Memory space assigned

to it.

① holds address

Bucket location
↳ Memory Block

② Size + 8 byte

int var = 10;

int *ptr = &var;

concept

ref.

~~int & ref;~~ X

error

int b = 10

int & ref = b;

int * ptr;

unassigned ~

grab

ptr 100



```
int var = 10;  
int &ref = var;
```

```
int var2 = 5;
```

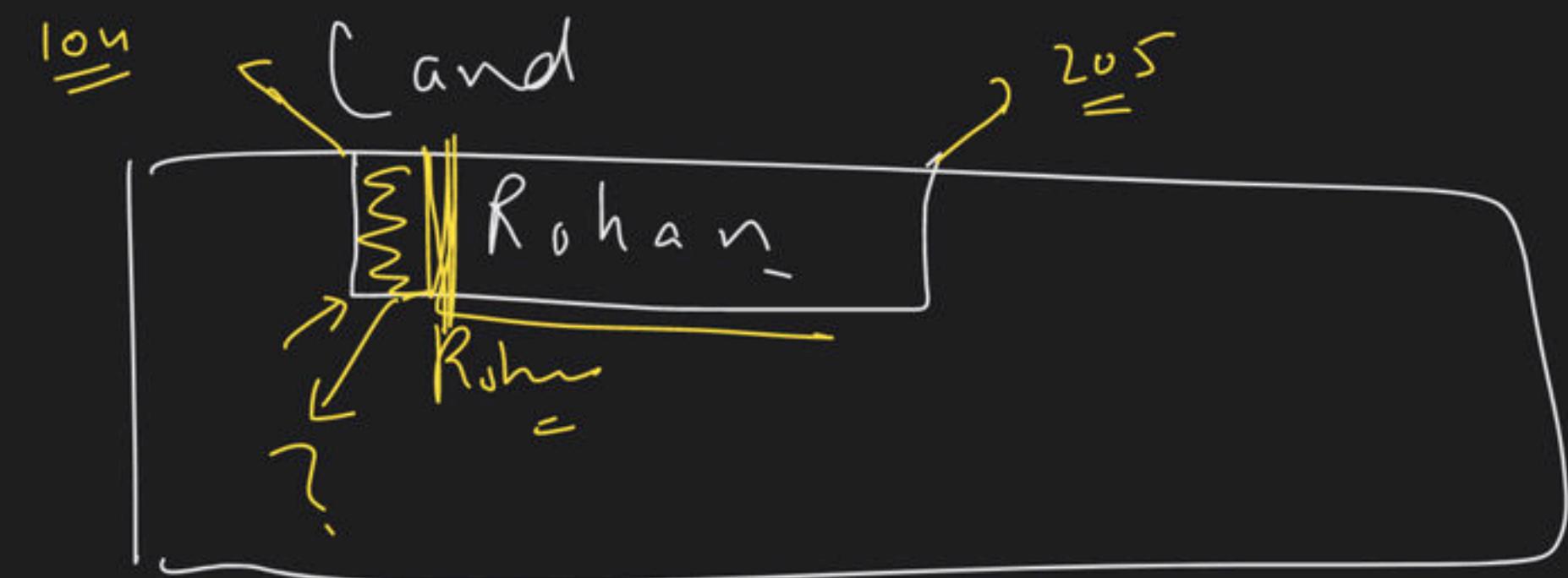
ref = var2

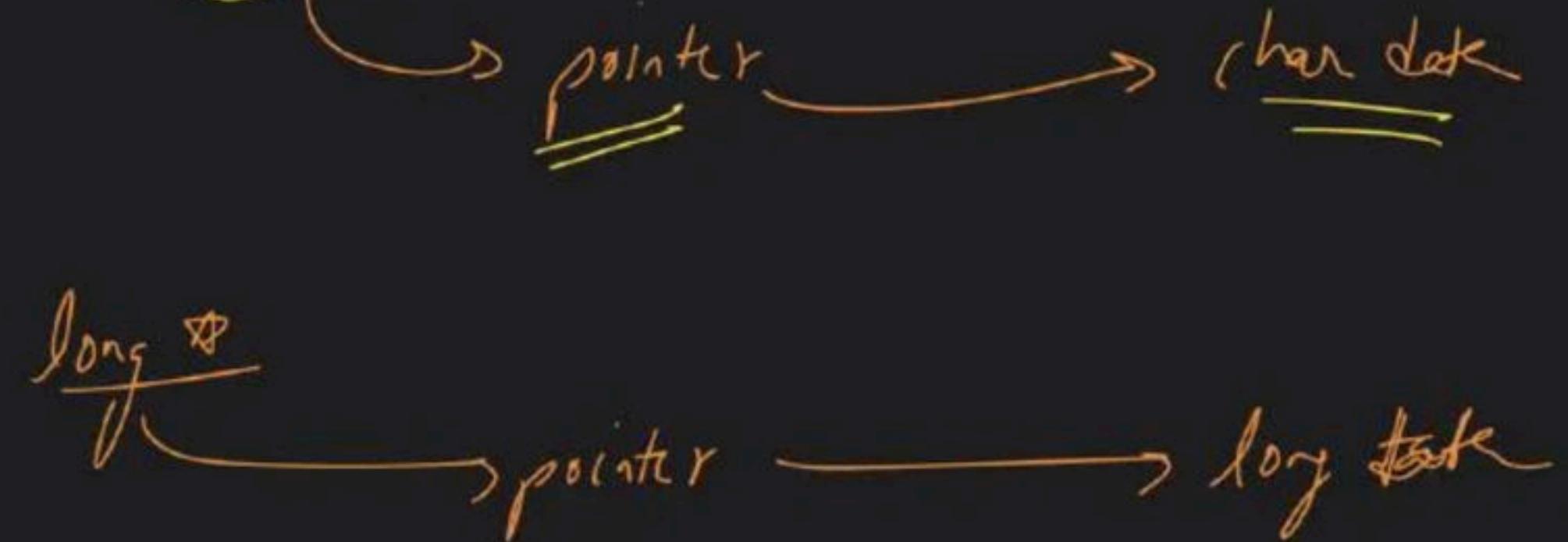
```
int b = 10;  
int *ptr = &b;
```

```
int a = 5;
```

```
ptr = &a;
```

a ↴





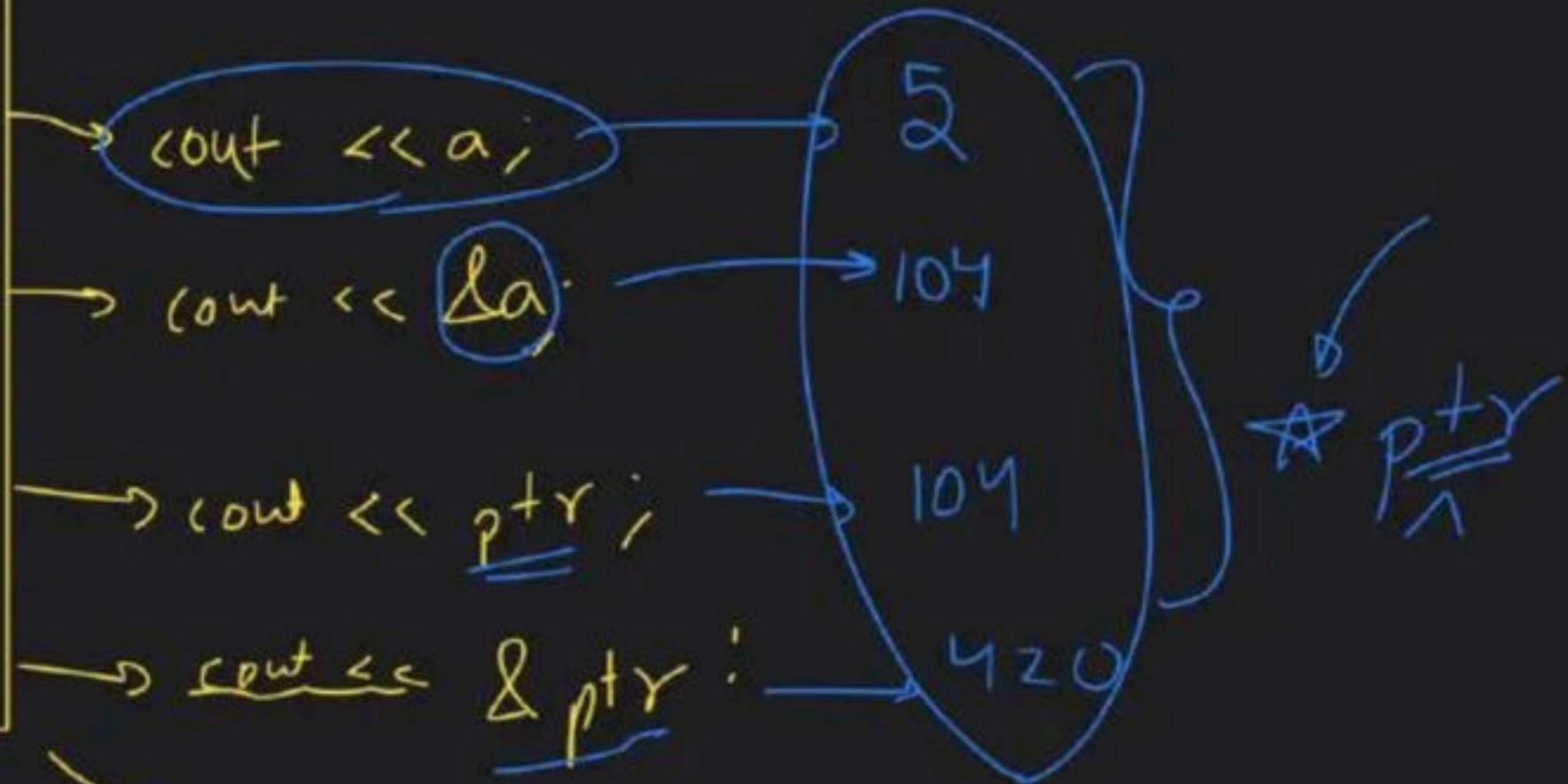
Custom
Datatype → Babbar

Babbar * ptr

ptr is a pointer
& pointer to Babbar type
data

int a = 5

int *ptr = &a;



dereference
operator



value at

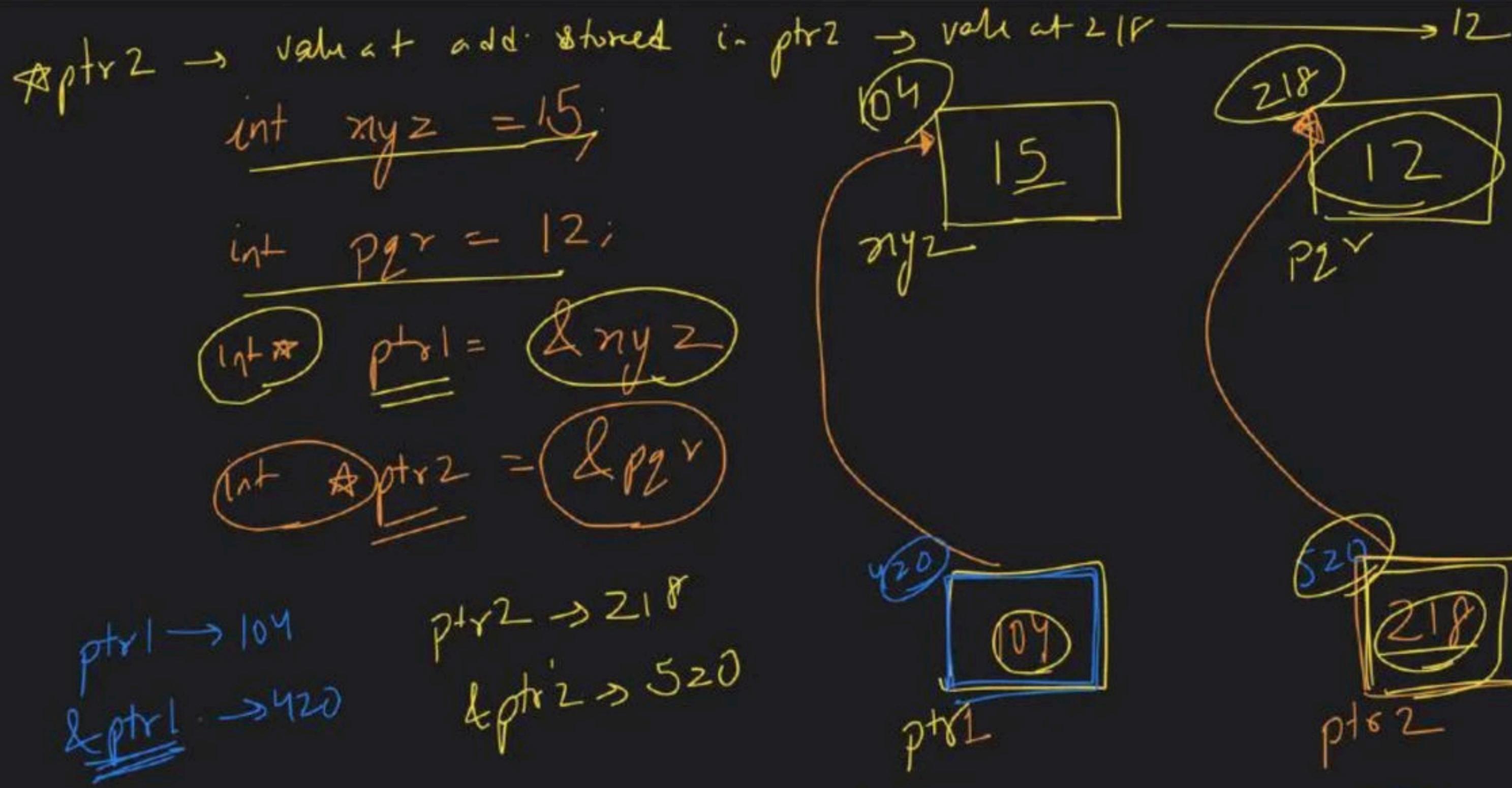


value at $\frac{ptr}{or}$

value at address stored in ptr

value at $10y$

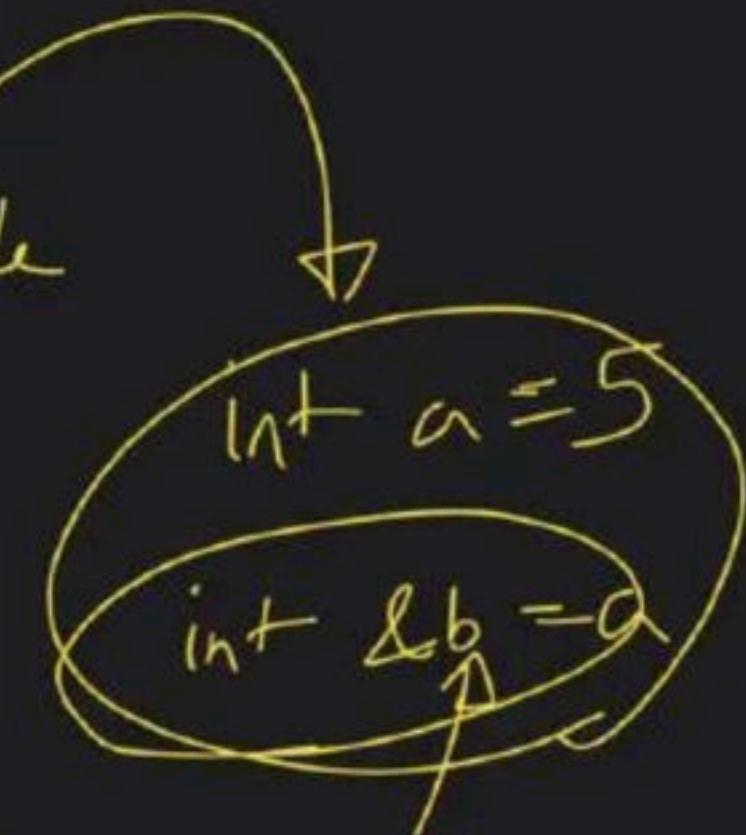




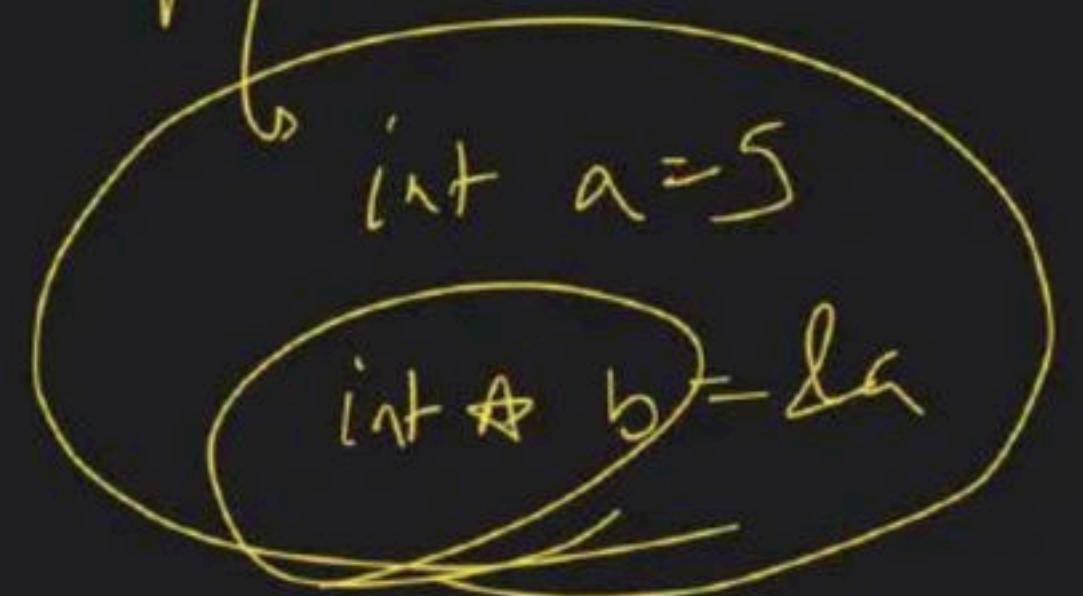
$\star \text{ptr1} \rightarrow$ Value at address stored in $\text{ptr1} \rightarrow$ value at 104 \rightarrow 15

Objt (ampw)

Reference variable



pointers



difference



$\text{int } a = 5;$

$\text{int } * \underline{\text{ptr}} = \underline{\&a}$

size of (ptr)

↓

8

why size is coming of?

$\text{char } ch = 'a';$

$\text{char } * \underline{\text{cptr}} = \underline{\&ch}$

size of (ptr)

↓

8

$\text{long } l = 1040;$

$\text{long } * \underline{\text{lptr}} = \underline{\&l}$

size of (lptr)

↓

8

H/w



declaration

int *ptr;

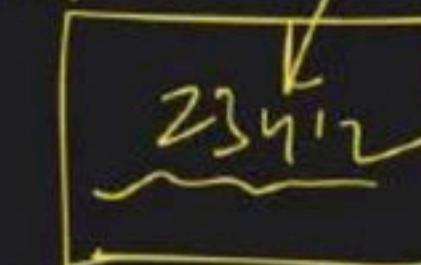
syn

BAD
Practice

p/q

int *ptr;
cout << ptr

104



a

420



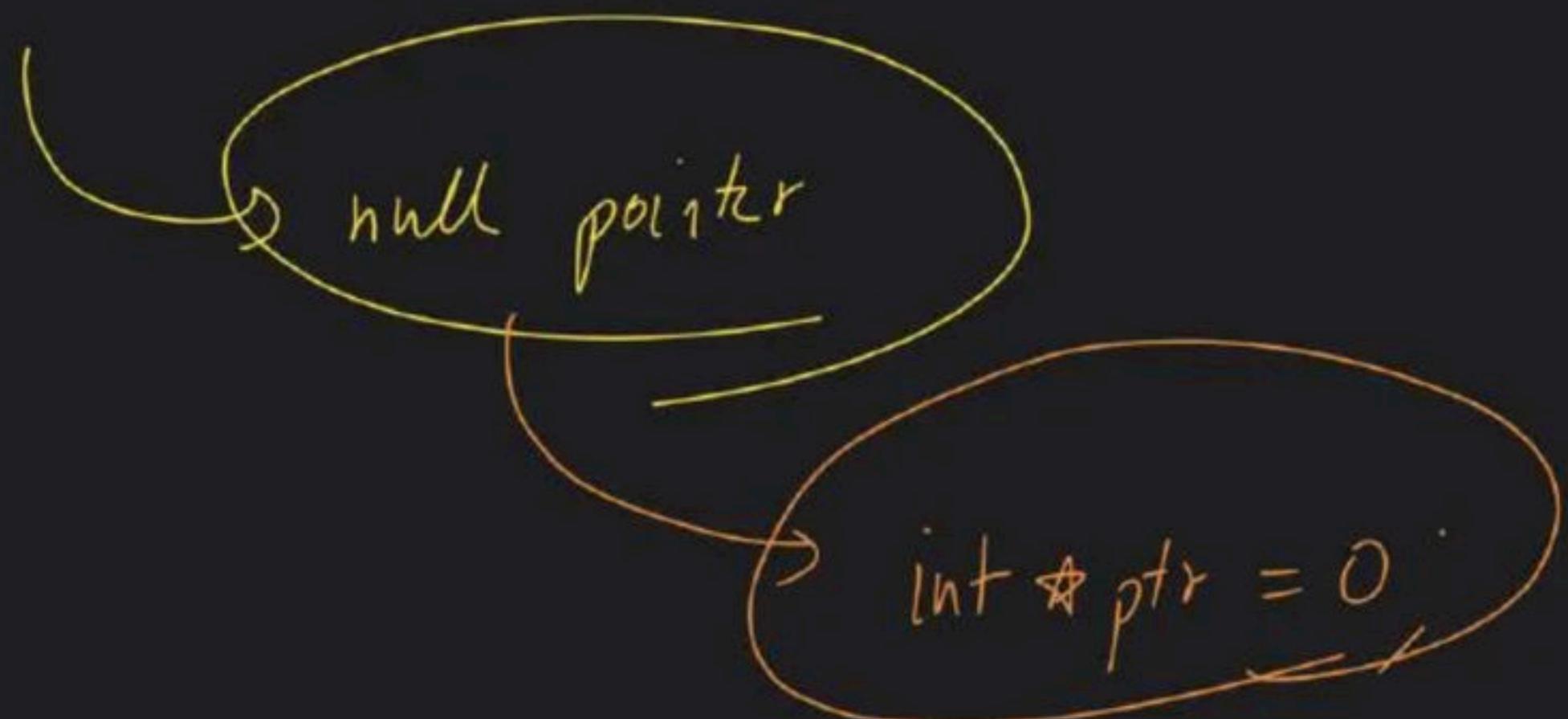
ptr

cout << *ptr

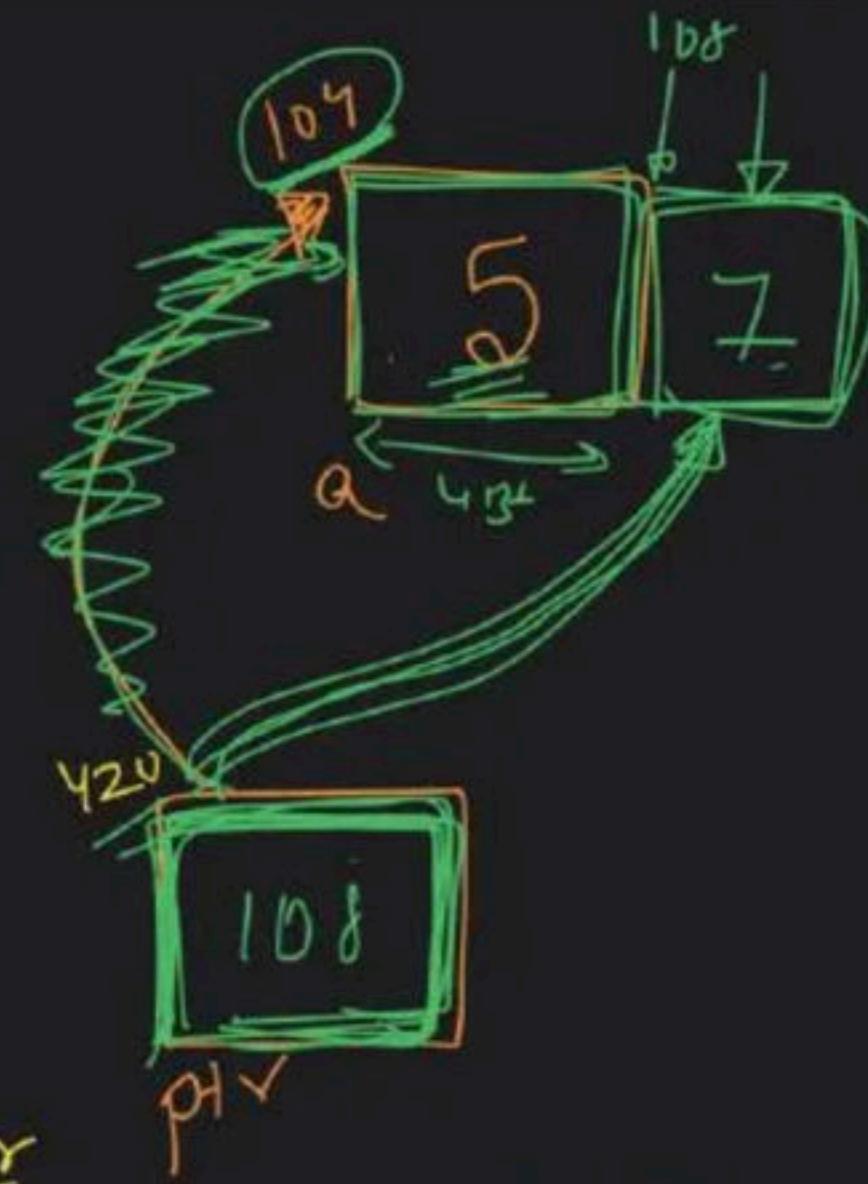
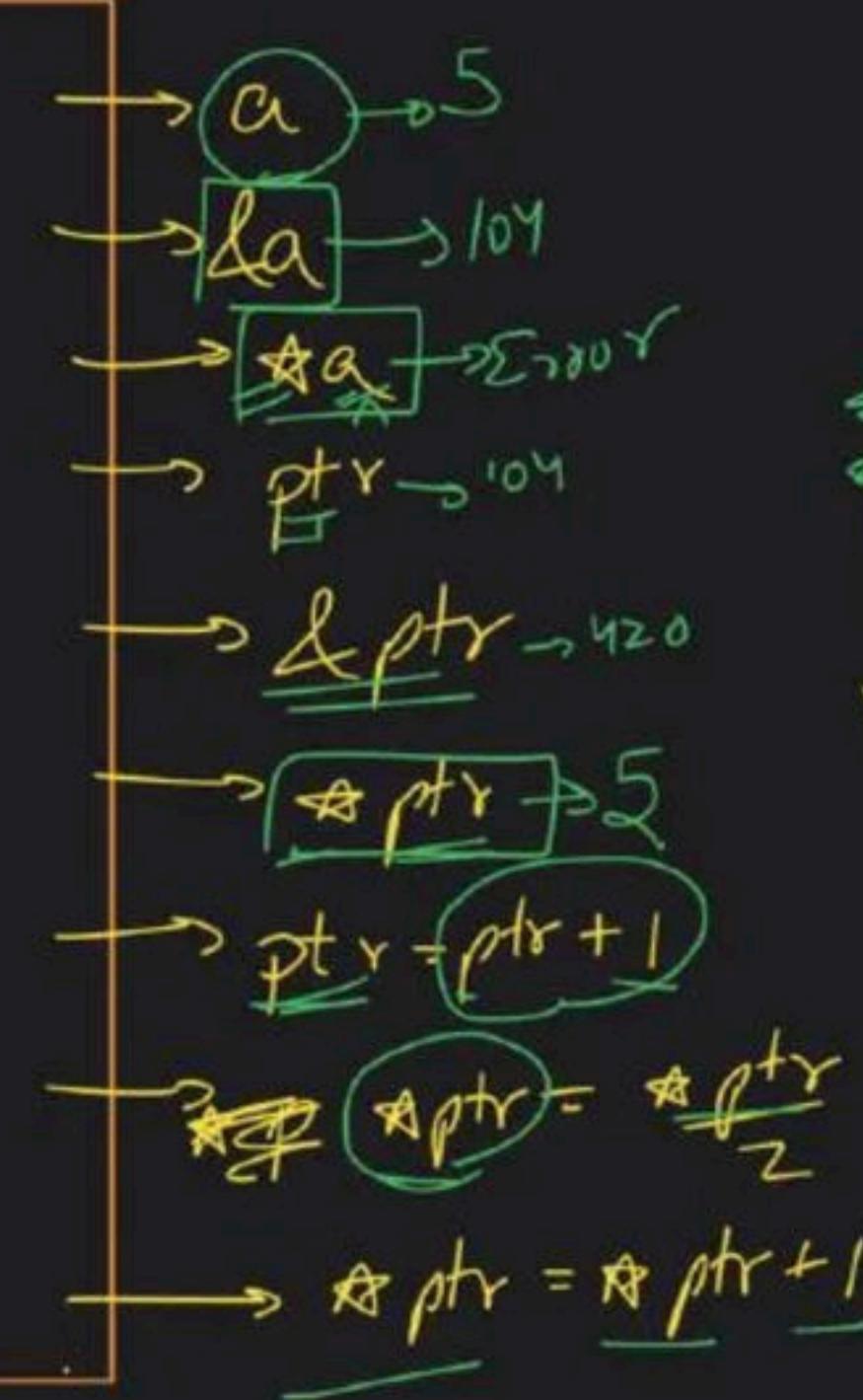
garbage
value

garbage
value

$\text{int } * \text{ptr}$

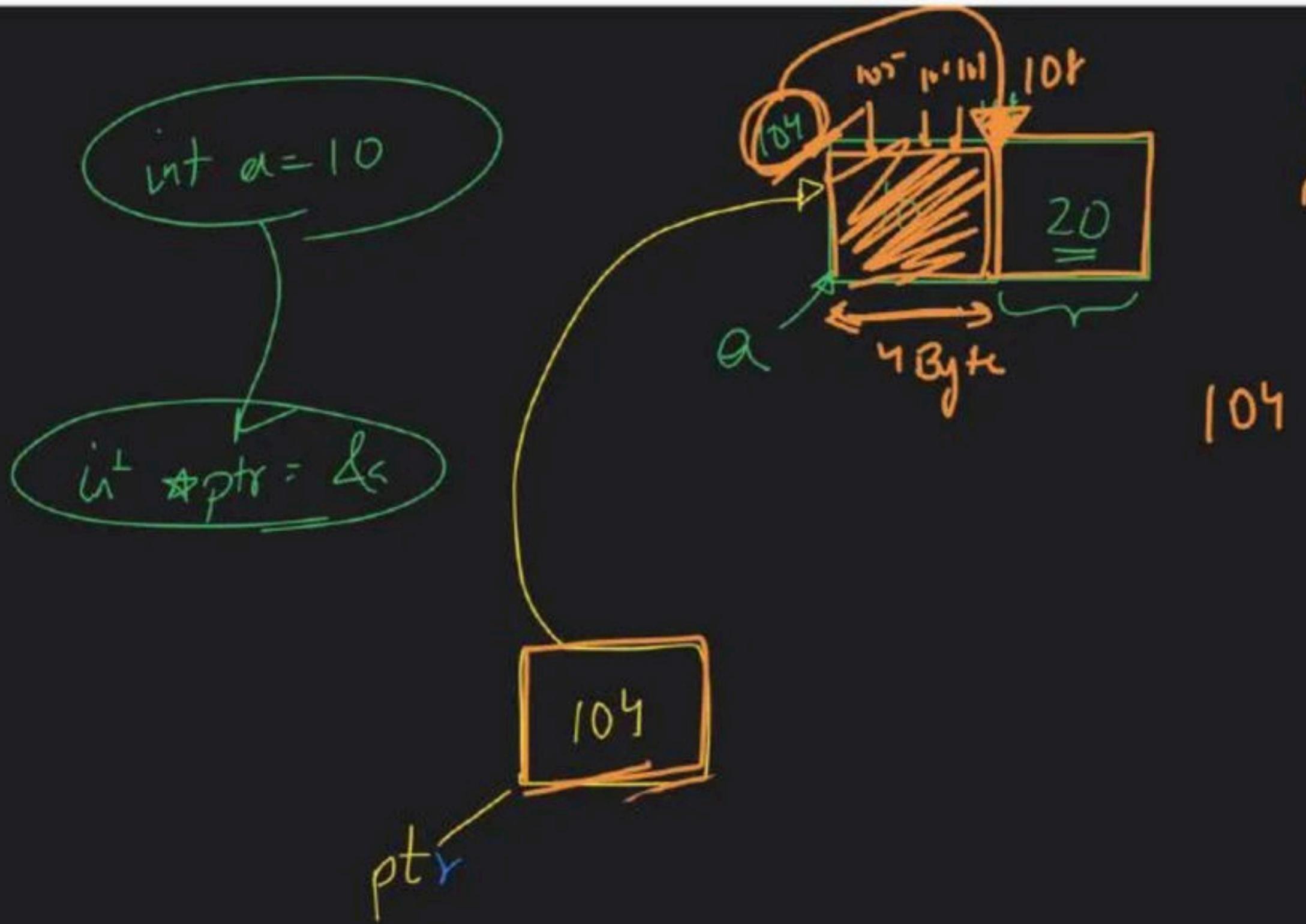


$\text{int } a = 5;$
 $\text{int } * \text{ptr} = \boxed{\&a}$

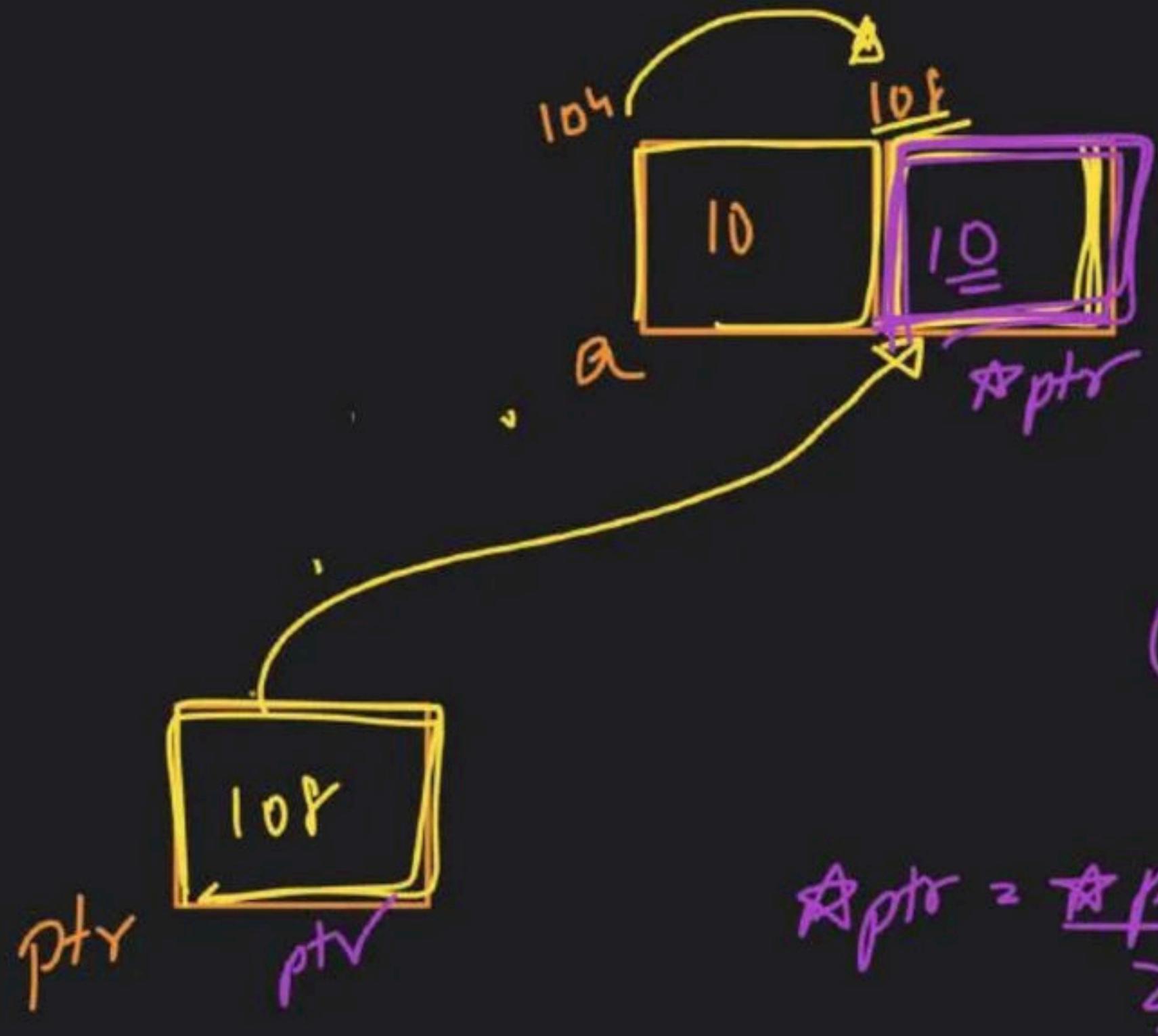


$$\frac{104 + 1}{105} = 7$$

$\& \& *ptr$



$$\begin{aligned}
 \text{ptr} &= \underline{\text{ptr}} + 1 \\
 \text{ptr} &\rightarrow \underline{104} + 1 \\
 &= 104 + 1 \\
 &= 105 \\
 &= \underline{105}
 \end{aligned}$$



$$ptr = ptr + 1$$

$$\star ptr \rightarrow 20$$

$$\star ptr = \star ptr + 1$$

$$\begin{aligned} &= 20 + 1 \\ &= 21 \end{aligned}$$

$$\star ptr = \frac{\star ptr}{2} = \frac{21}{2} = 10$$

int a = 10;

int *p = &a

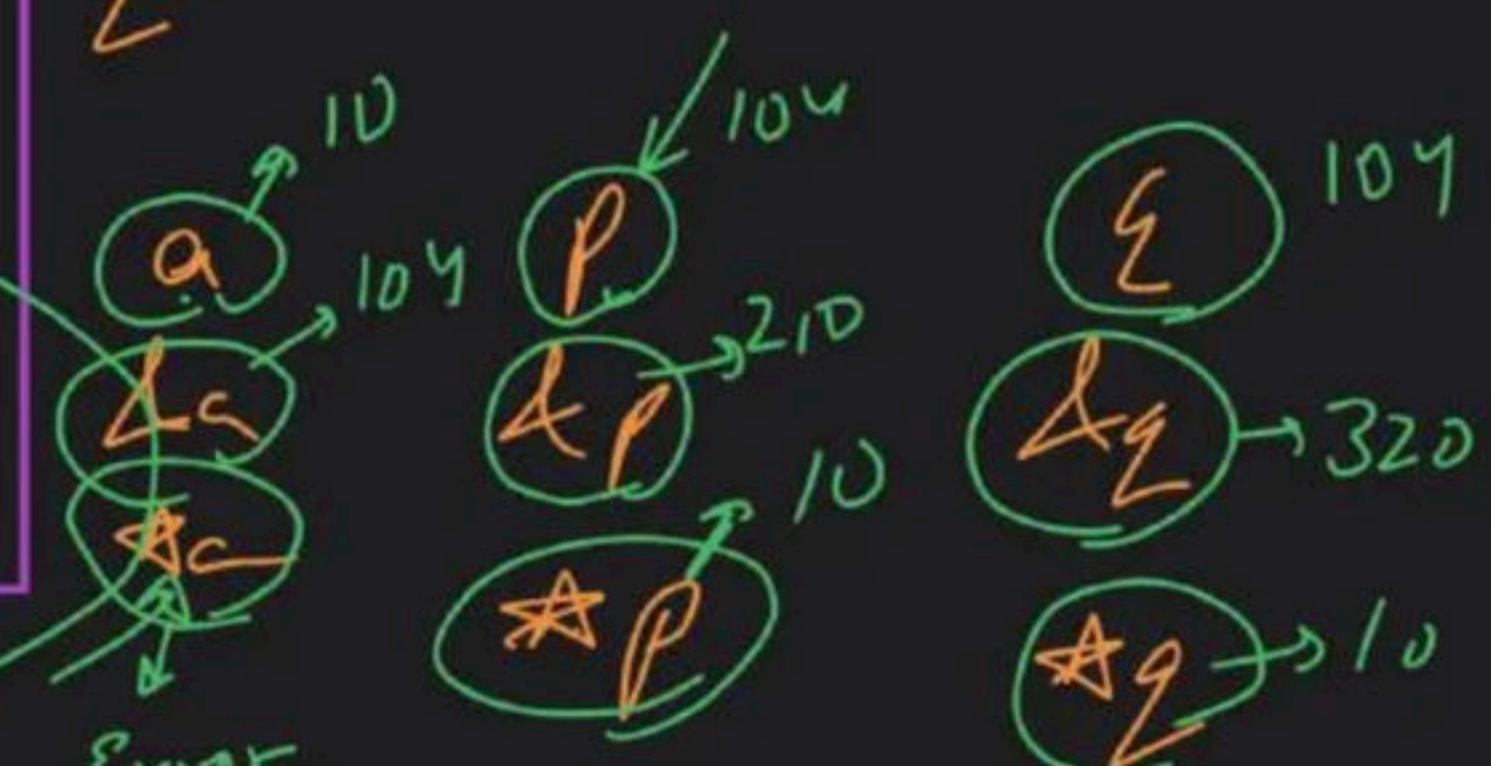
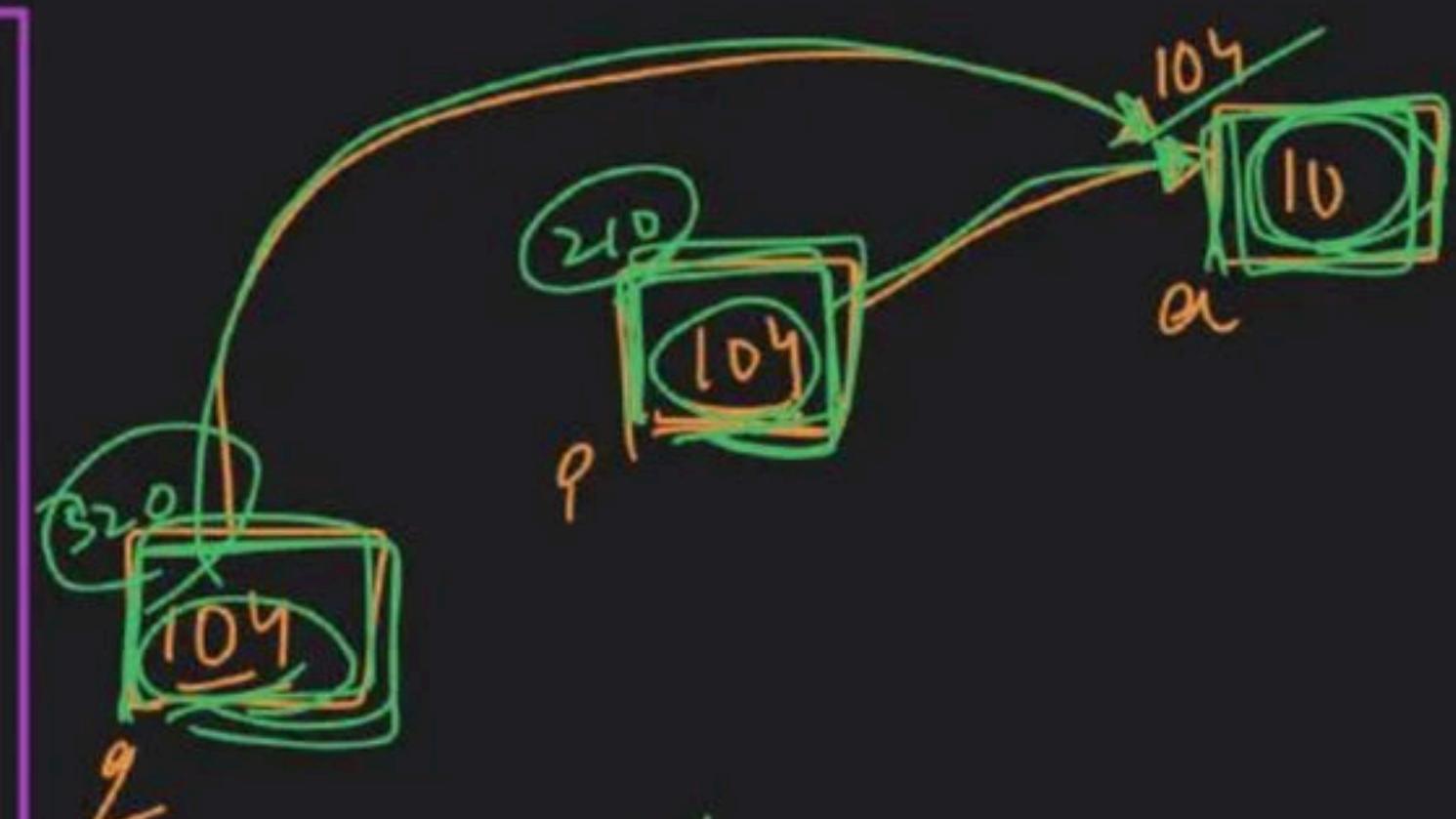
int *q = p → copy

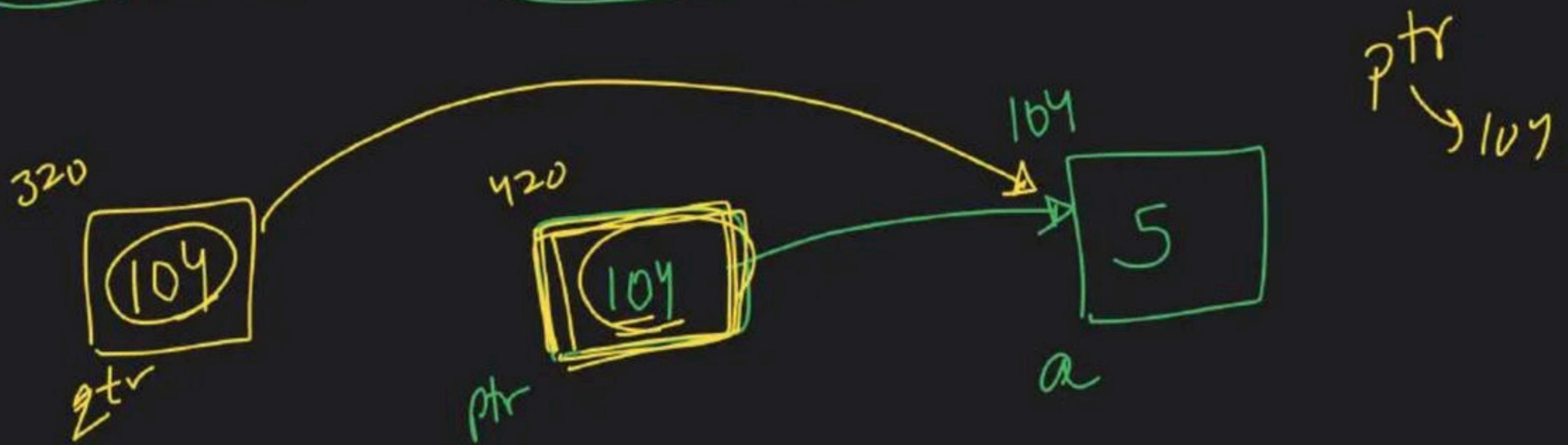
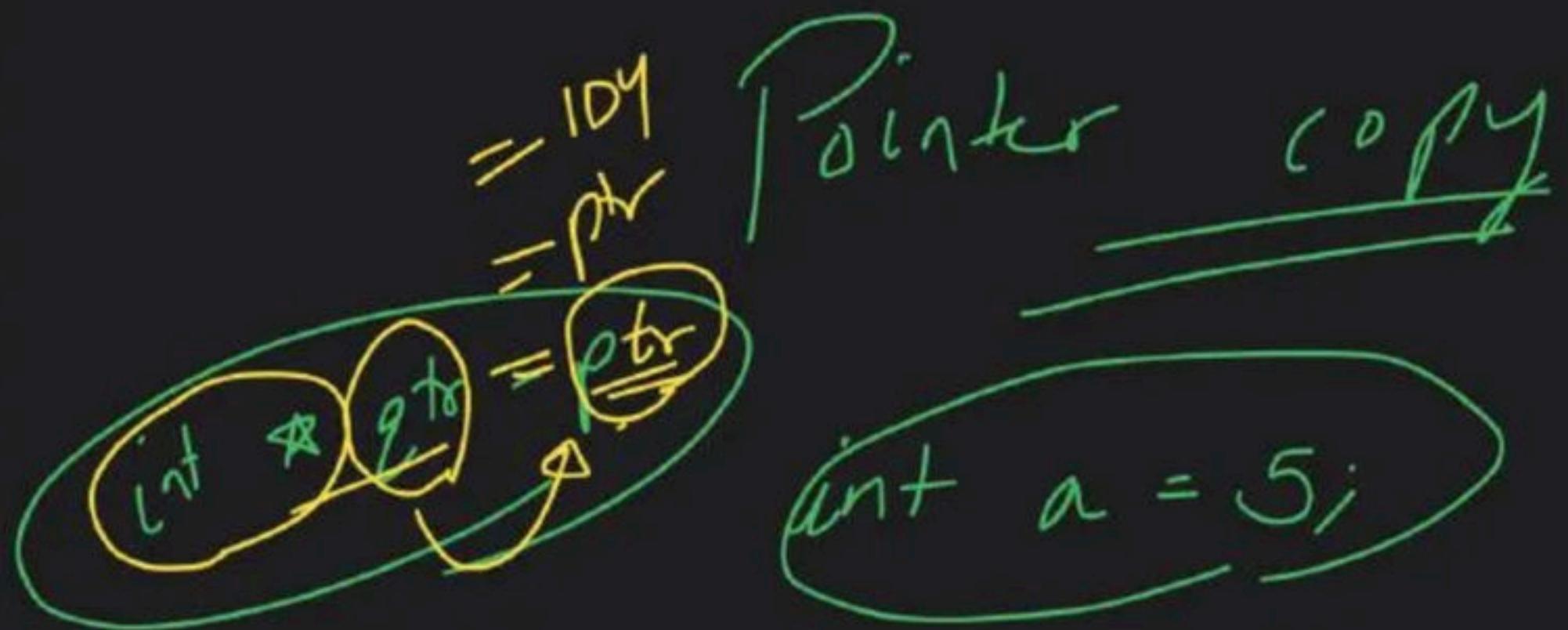
min

water

break

Error



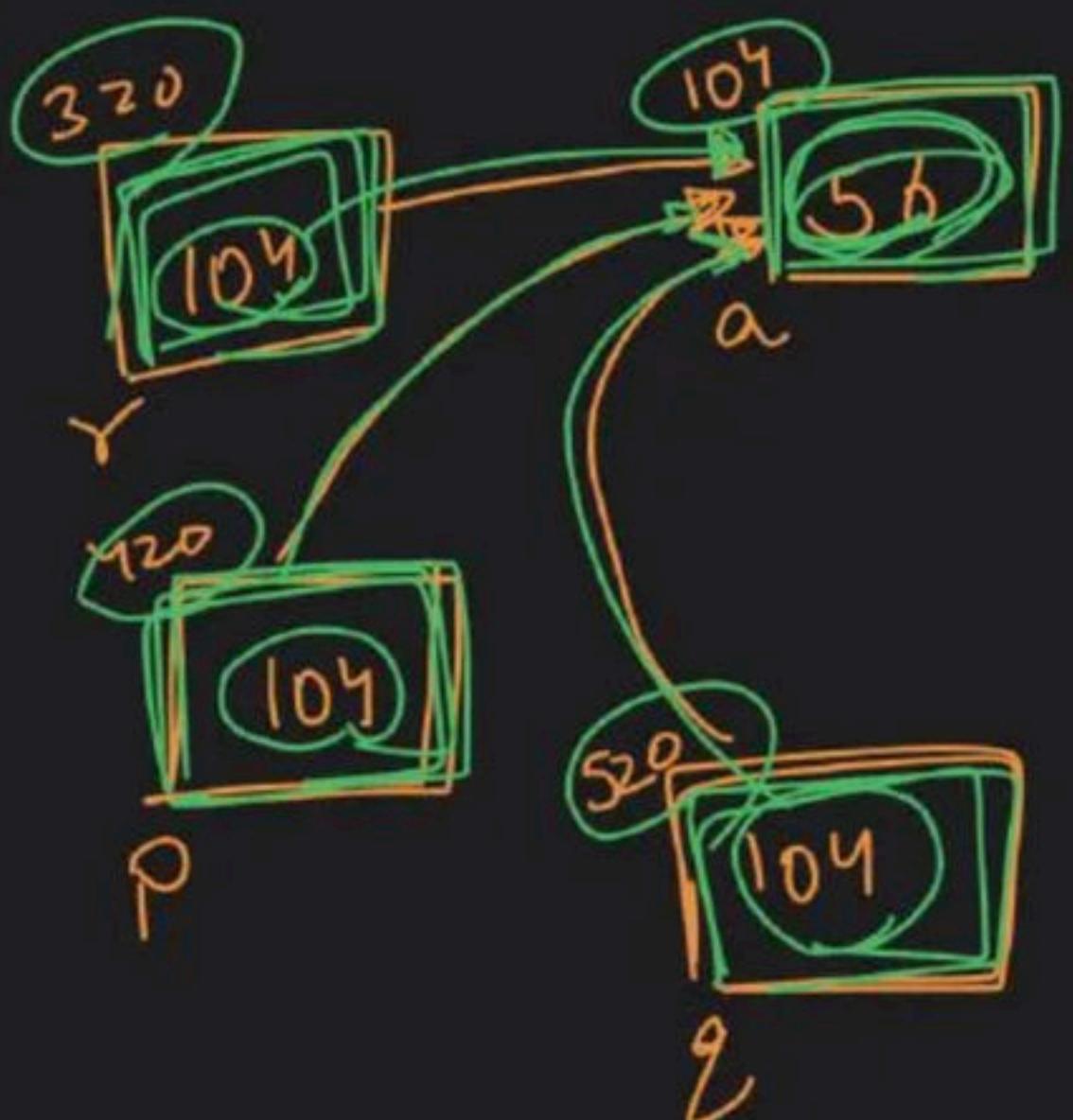


int a = 56;

int * p = &a;

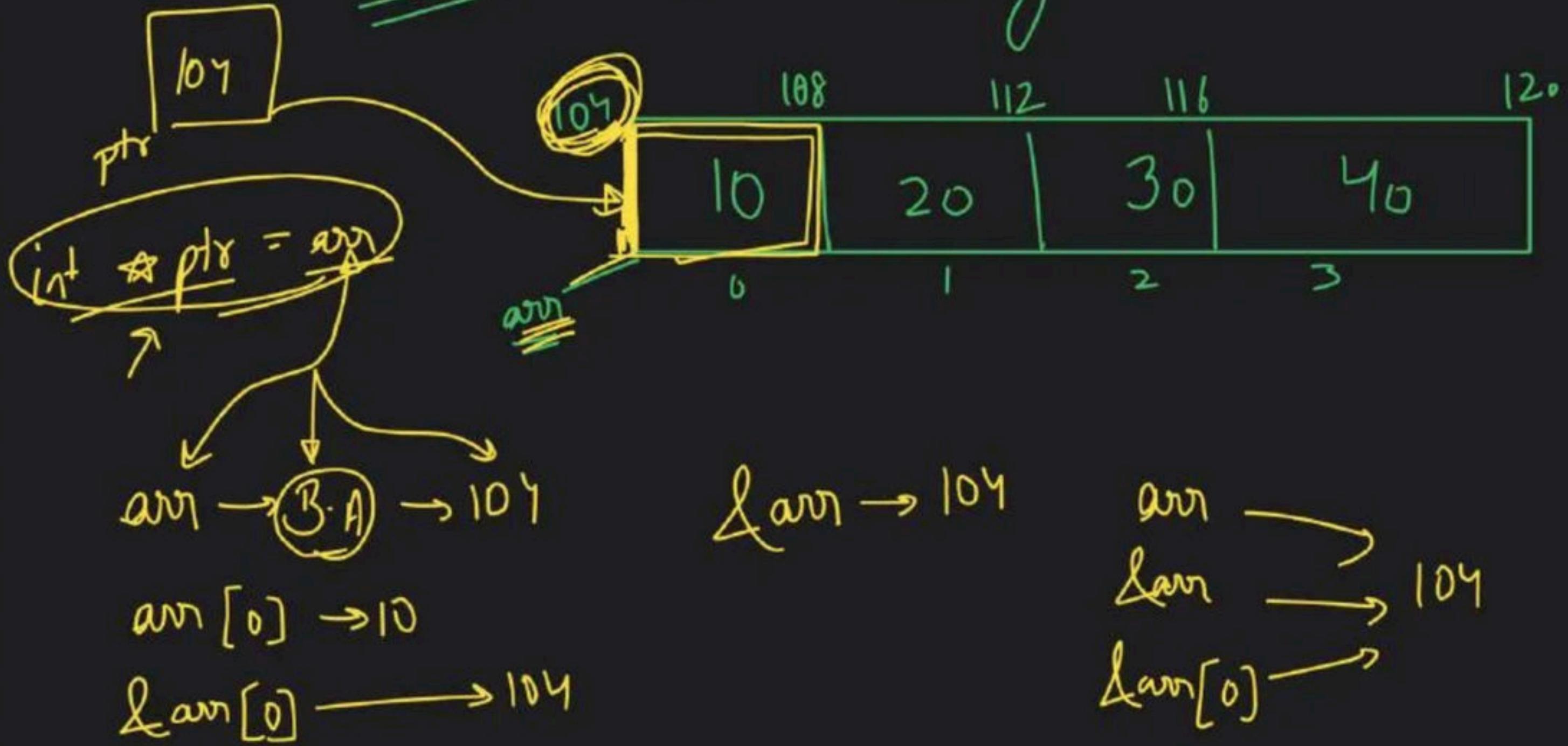
→ int R g = p;

→ int * y = g;



$a \rightarrow 56$
 $\underline{&a} \rightarrow 104$
 $\underline{\&p} \rightarrow 104$
 $\underline{p} \rightarrow 104$
 $\underline{\&g} \rightarrow 104$
 $\underline{g} \rightarrow 104$
 $\underline{\&y} \rightarrow 520$
 $\underline{y} \rightarrow 520$
 $\underline{\&1} \rightarrow 56$
 $\underline{1} \rightarrow 104$
 $\underline{\&Y} \rightarrow 320$
 $\underline{Y} \rightarrow 51$

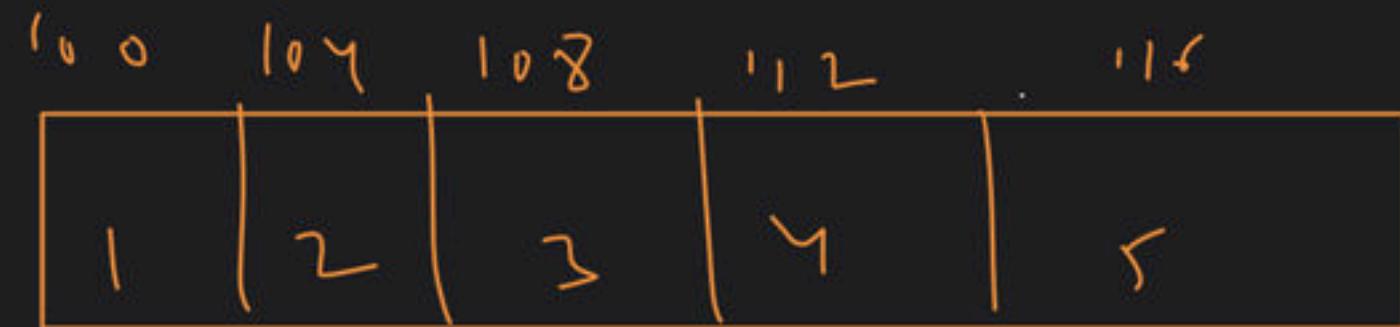
Pointers with Arrays



`int arr[5] = {1, 2, 3, 4, 5};`

`arr` \Rightarrow $\boxed{100}$

address of 1st element in an arr.



`arr`
`int *p = arr`

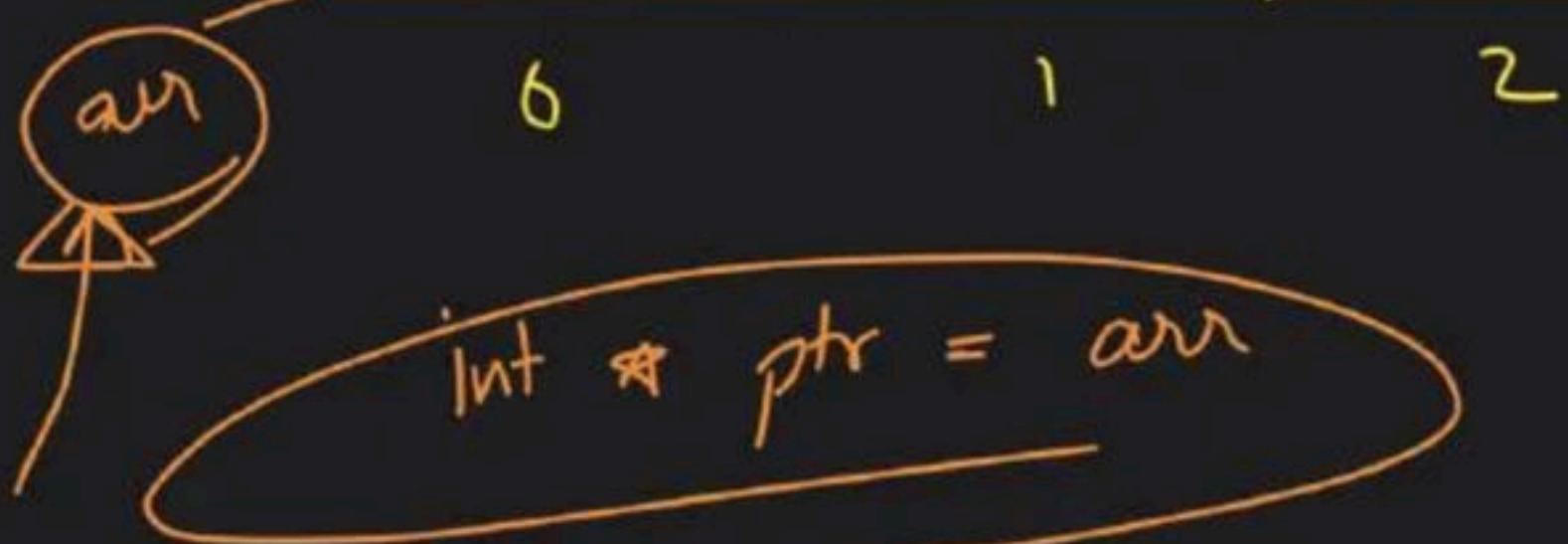
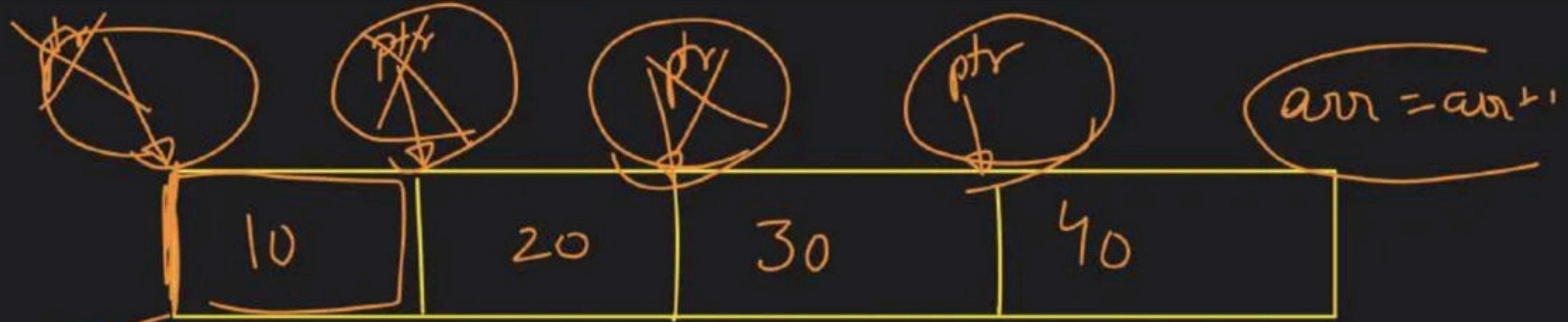
~~location of arr~~

`arr` \Rightarrow \downarrow

$\boxed{100}$

address of the whole array

`int (*P)[5] = &arr;`

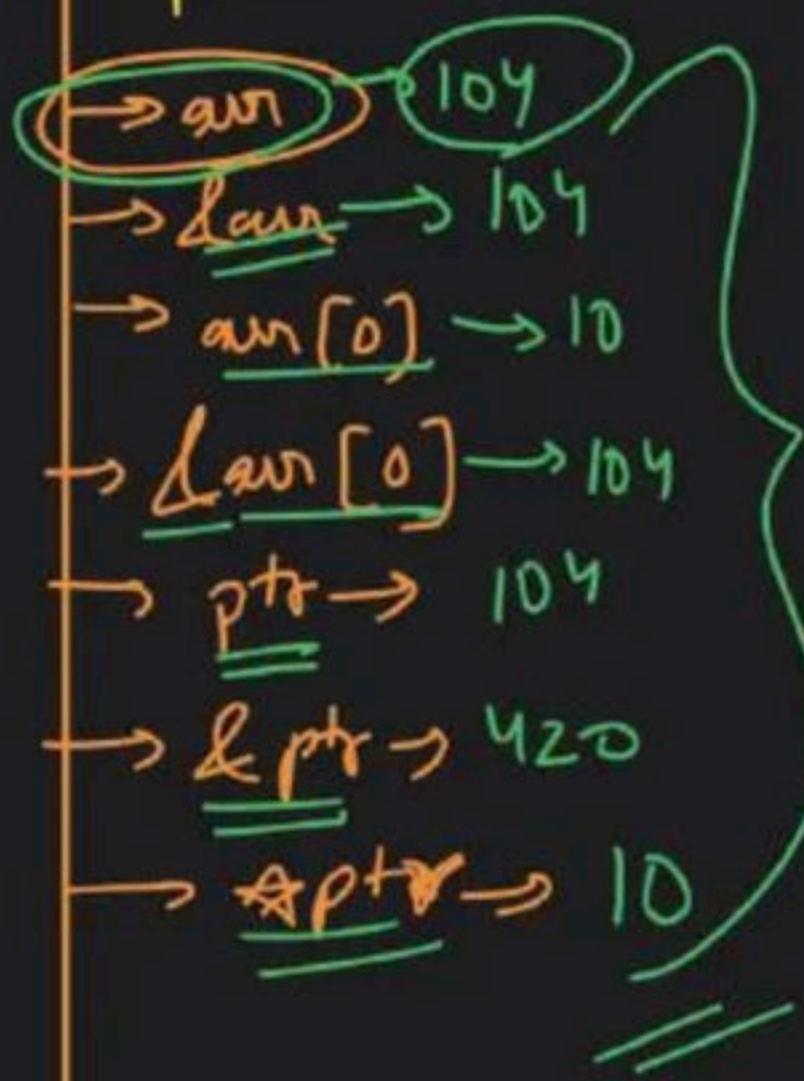
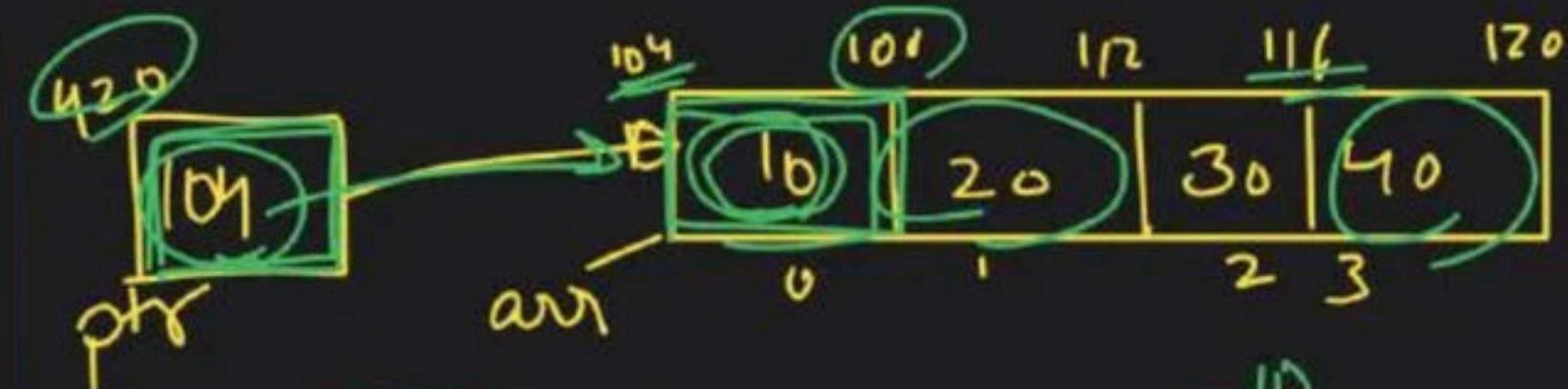


`ptr = ptr + 1`

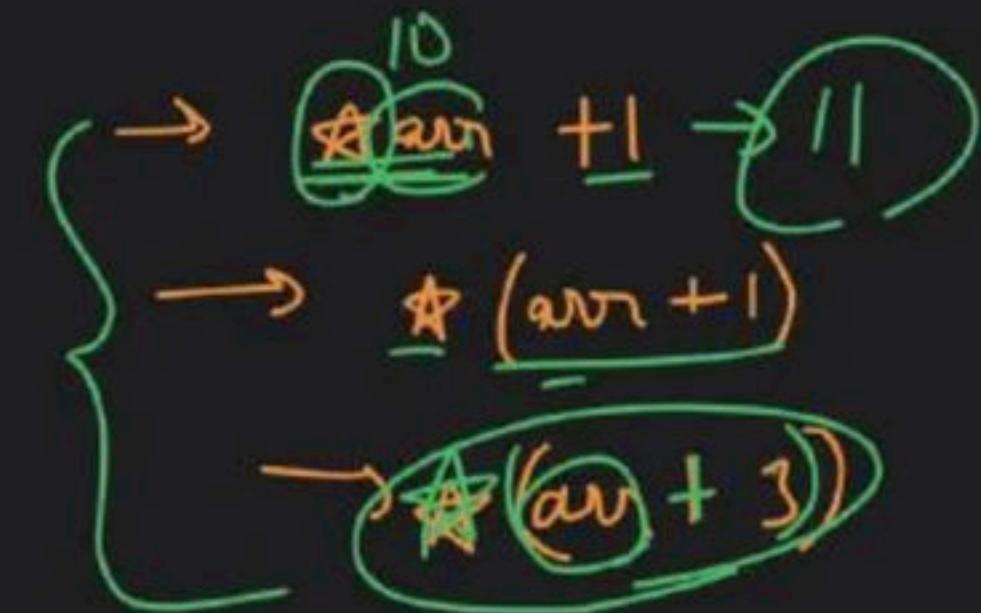
cont
`<<(arr+1)`
~~cout~~

int arr[] = {10, 20, 30, 40}

int *ptr = arr;



$\star(\text{arr} + 3) \rightarrow \star(104 + 3) \rightarrow \star(104 + 3 \times 4)$



$\text{arr} \rightarrow 104$
addr
 $\star(11)$
 $\star(104 + 1)$

$\star(\text{arr} + 1)$
 $\star(104 + 1)$
 $\star(105)$
 20

$$arr[i] = i[arr]$$

$$\star(arr+i)$$

$$\star(i+arr)$$

$$arr[2] \rightarrow \star(arr+2)$$

int arr[] → {100, 200, 300, 400, 500}

int *P = arr;

int *q = arr + 1;

$\star(q+4) \rightarrow \star(100+4)$

$\star(100+4 \times 4)$

$\star(p+2)$

$\star(100+11)$

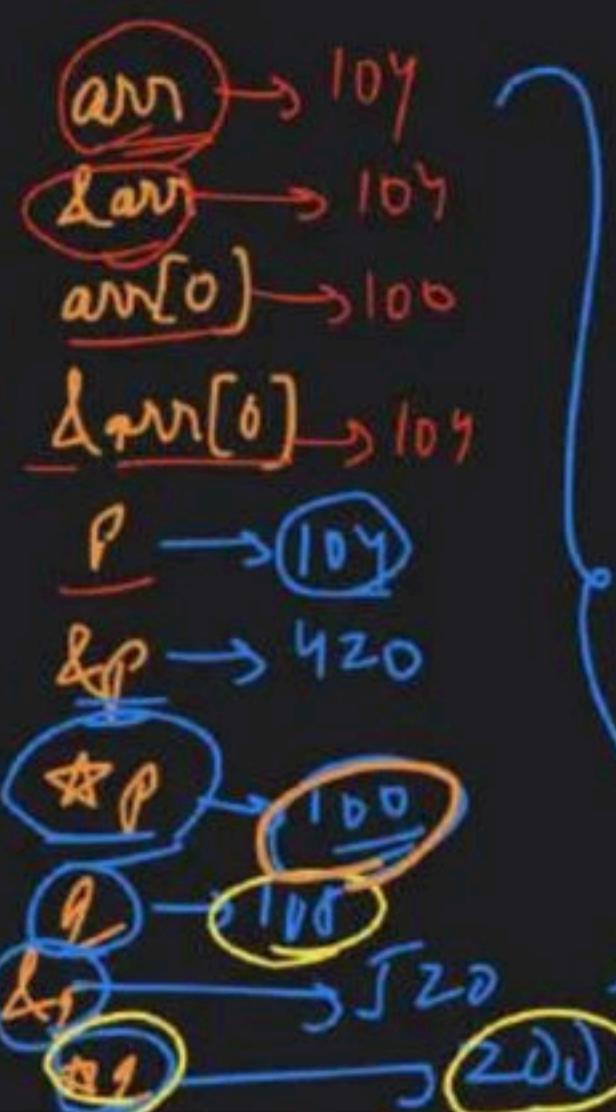
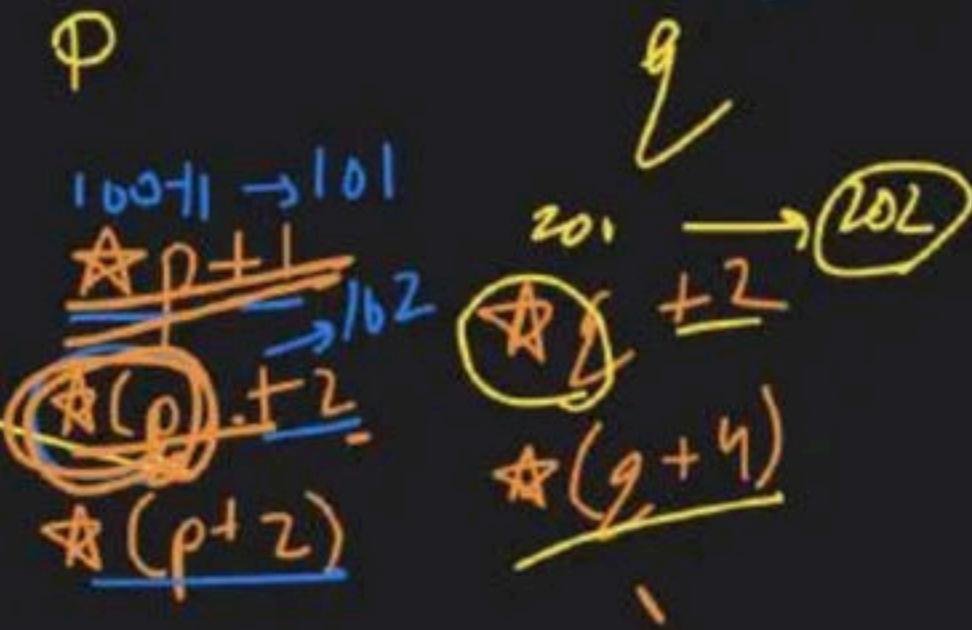
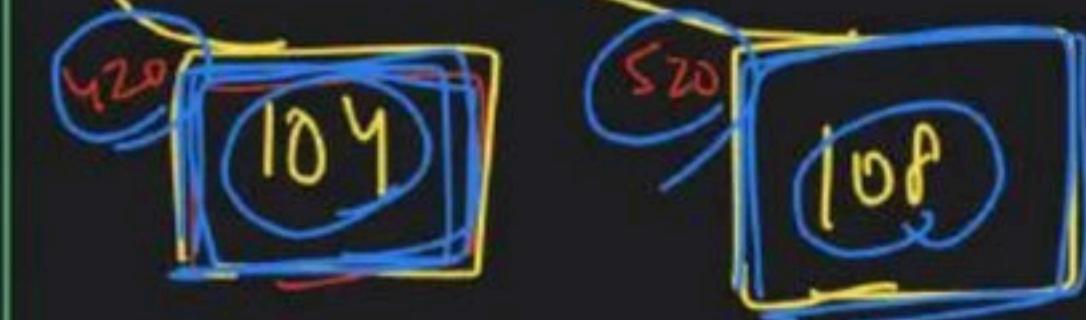
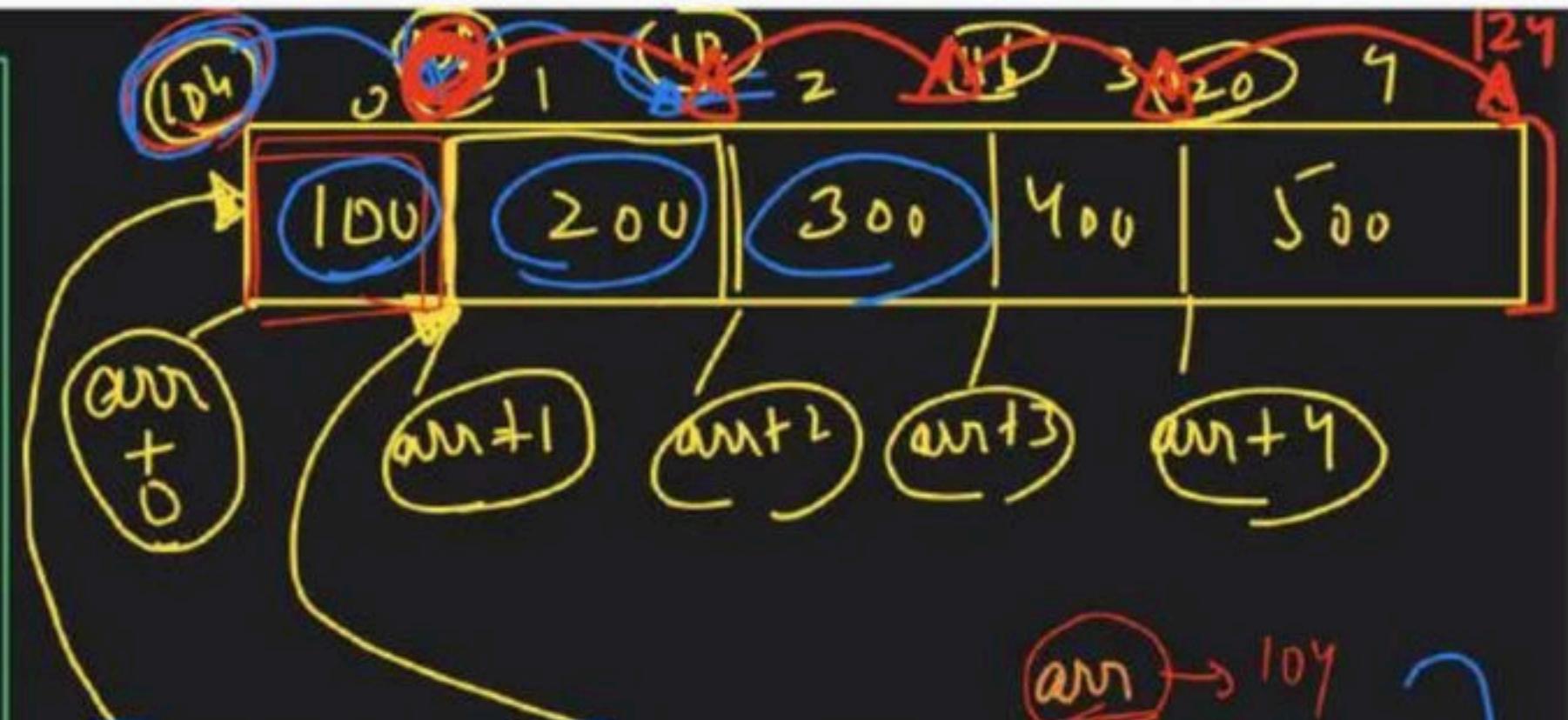
$\star(100+2)$

$\star(124)$

$\star(100+2 \times 4)$

$\star(100+6)$

$\star(112) \rightarrow 300$

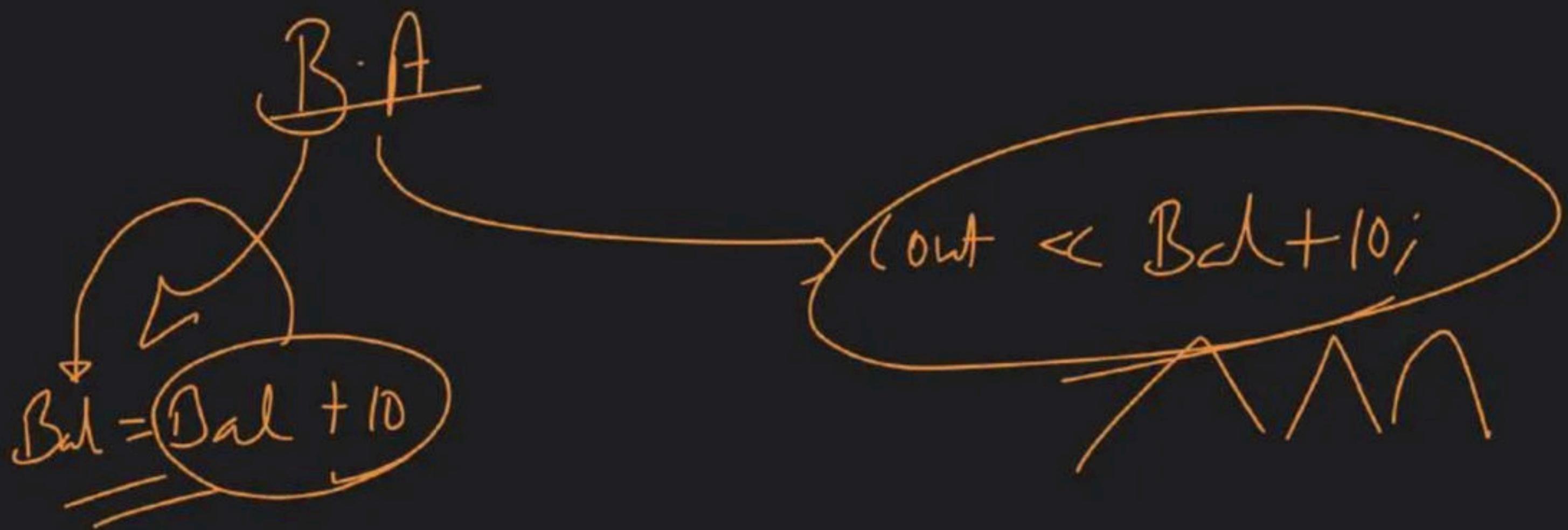


~~Char Arrays :-~~

2 min

Break

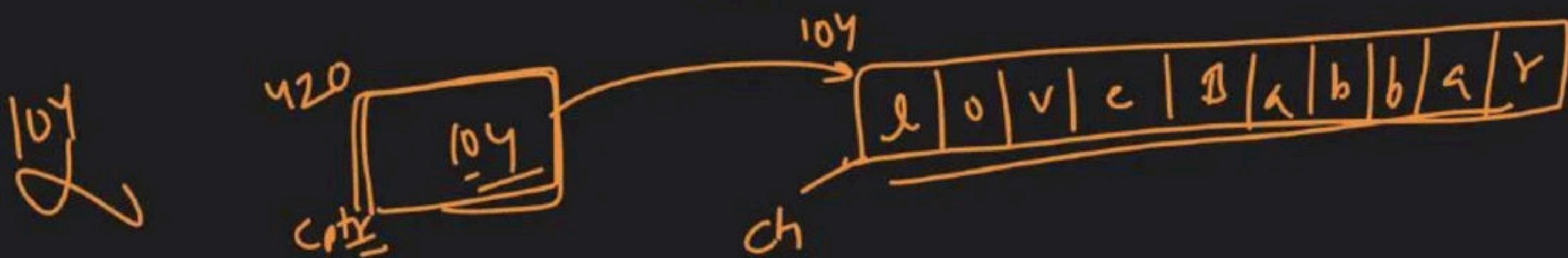
$\text{cout} \ll (\text{arr} + 1)$



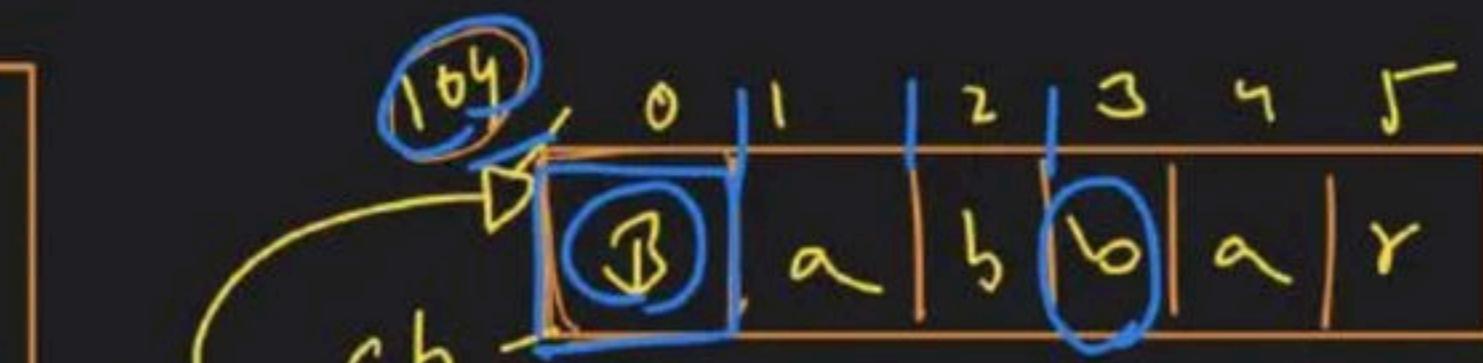
`char ch[100] = "lovebabbar";`

`char * cptr = ch;`

`cout << cptr;`

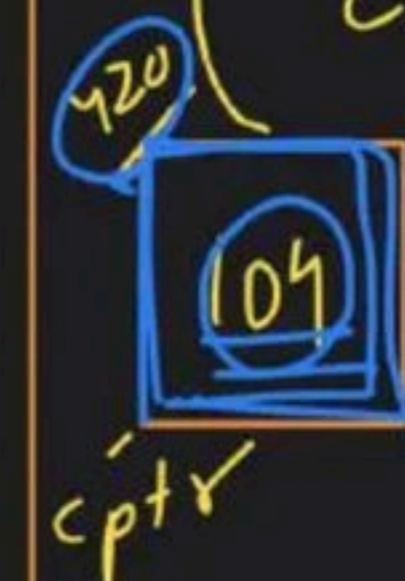


char ch[20] = "Babbar";



char * cpt = ch;

$\star(\text{cpt} + 3)$
 $\rightarrow (104 + 3 \times 1)$
 $\star(104 + 3)$
 $\rightarrow (107)$



$\boxed{\text{ch}} \rightarrow \text{Babba}$

$\&\text{ch}$ $\rightarrow 104$

$\text{ch}[0]$ $\rightarrow \text{B}$

$\&\text{cpt}$ $\rightarrow 104$

check
Karo
 $\star\text{cpt}$ $\rightarrow \text{B}$
check $\leftarrow \star(\text{cpt} + 3) \rightarrow \text{b}$
will be $\leftarrow \text{cpt} \rightarrow 104$

char

(ch) =

a'

(h)

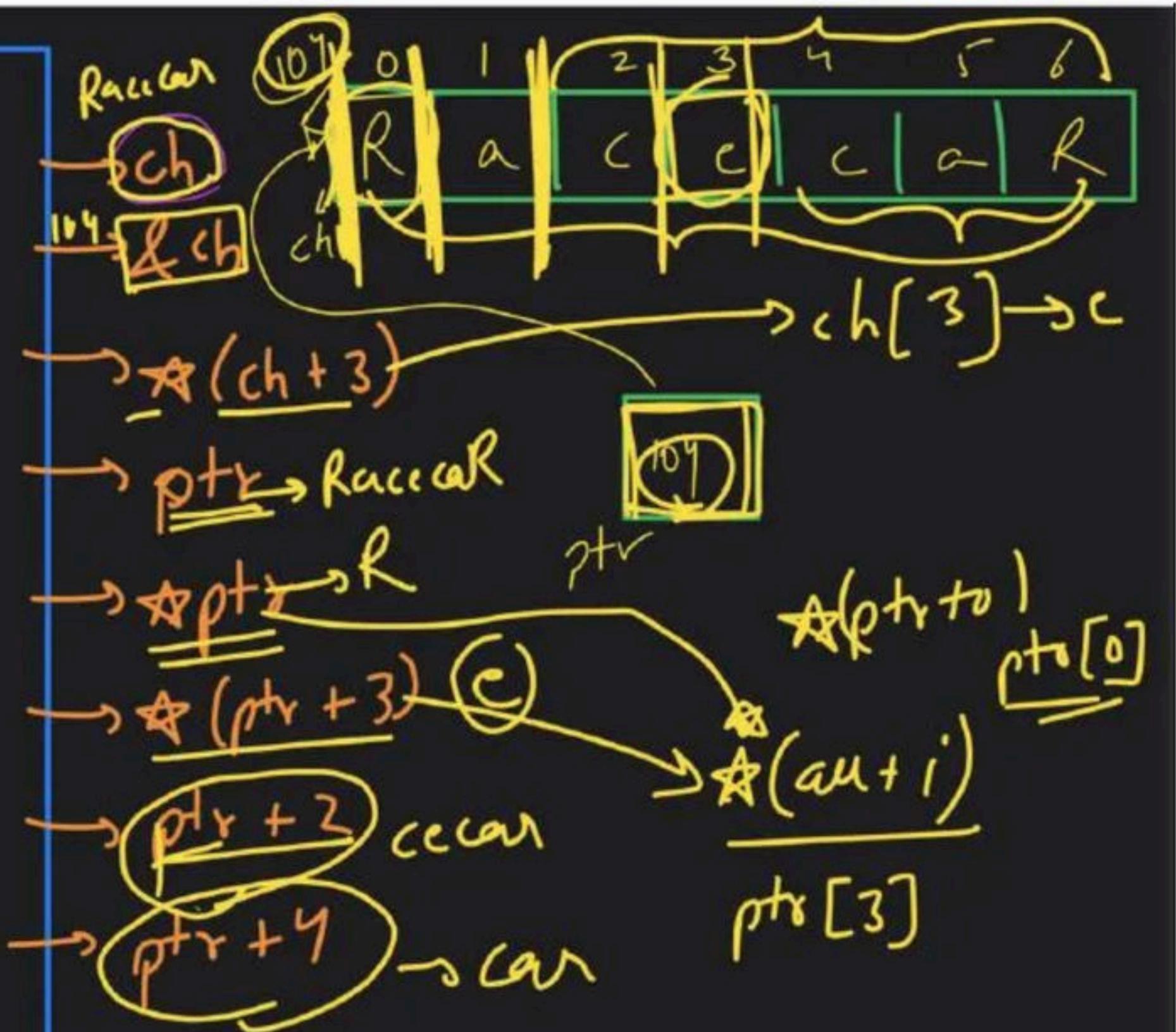
char * cpty = &ch

ch[0]

out << cpty

char ch[10] = "Raccccar"

char * ptr = &ch[0]



$\&ch[0]$

char ch[10] = "Balbar"

why

DP
ptr

char

*ptr =

"Balbar"

now

ptr































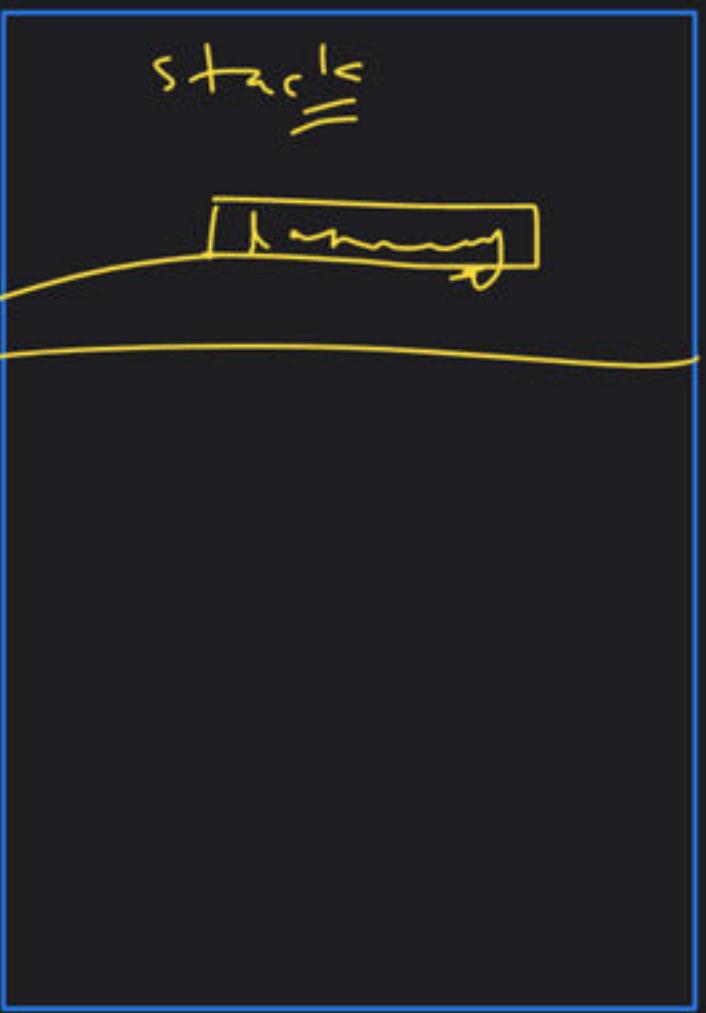


=>

char *ch = "Lakshay";

ch[2] = 'z'; → error/

undefined
behaviors



String literal =

String / char arr

read-only

memory

↓

Can't be
modified















Pointers - Class 2

Special class

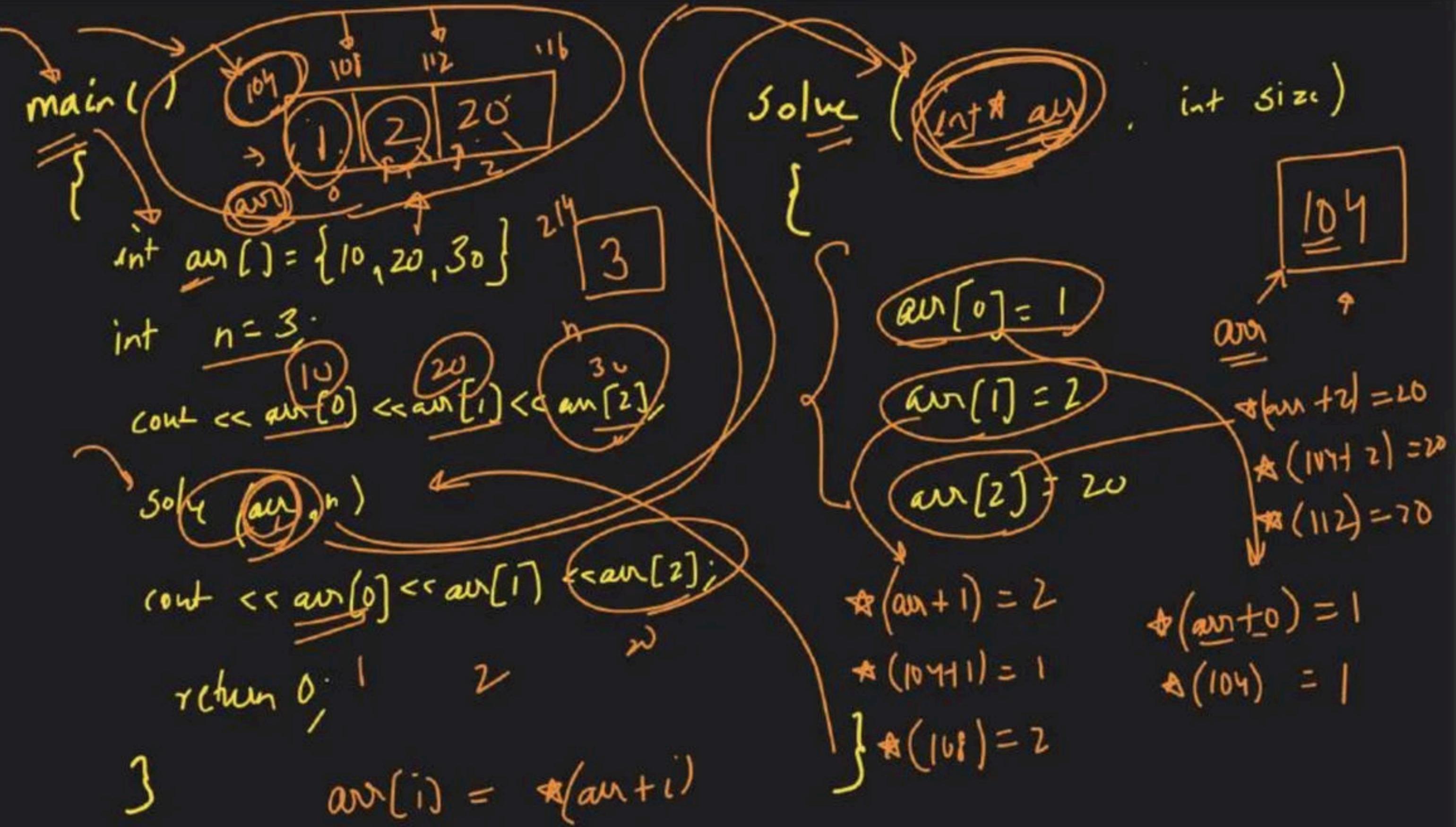
① Ref var

② Pointer

Arrays → function → by ref

Double-pointers

why?



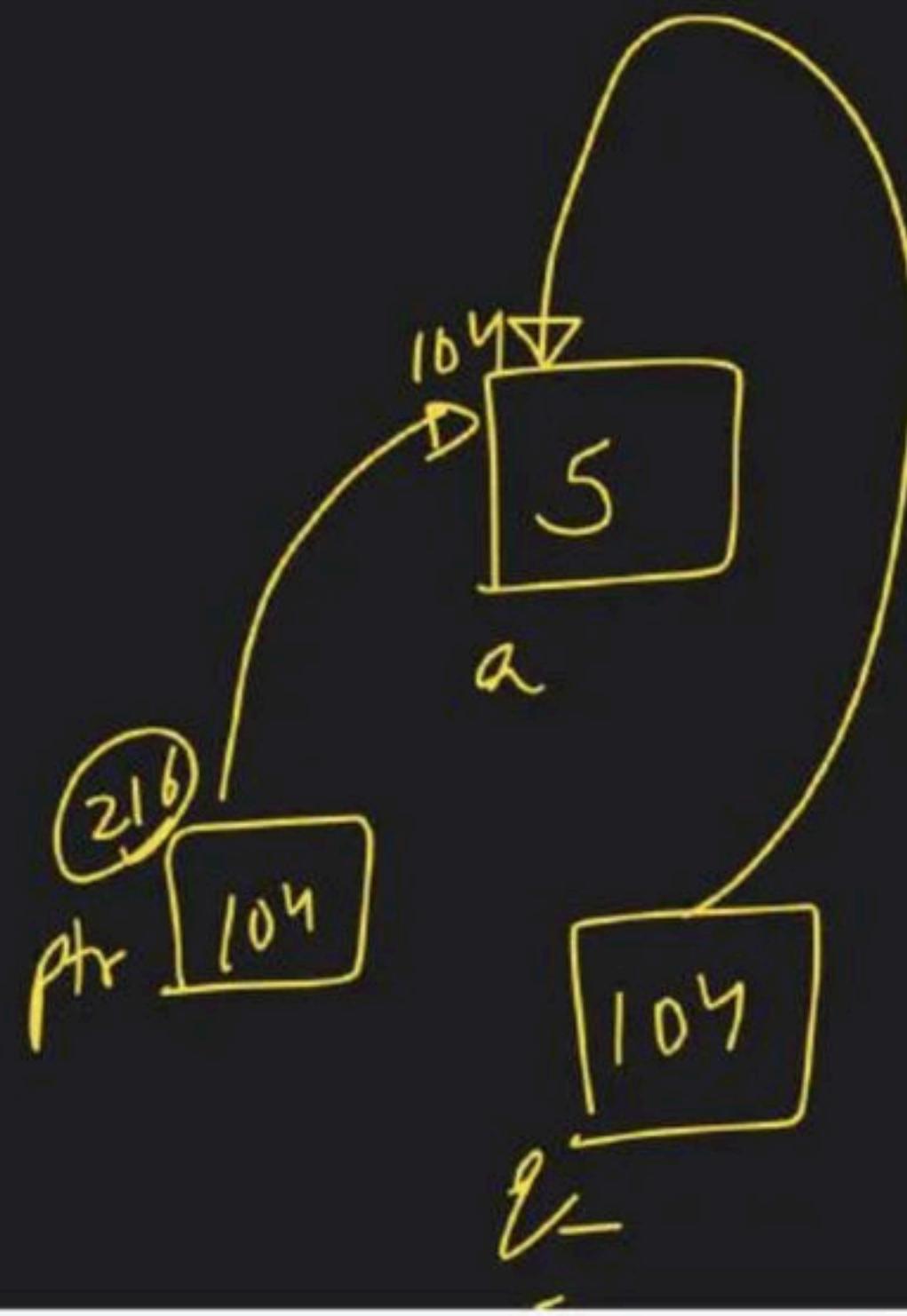
Double pointers :-

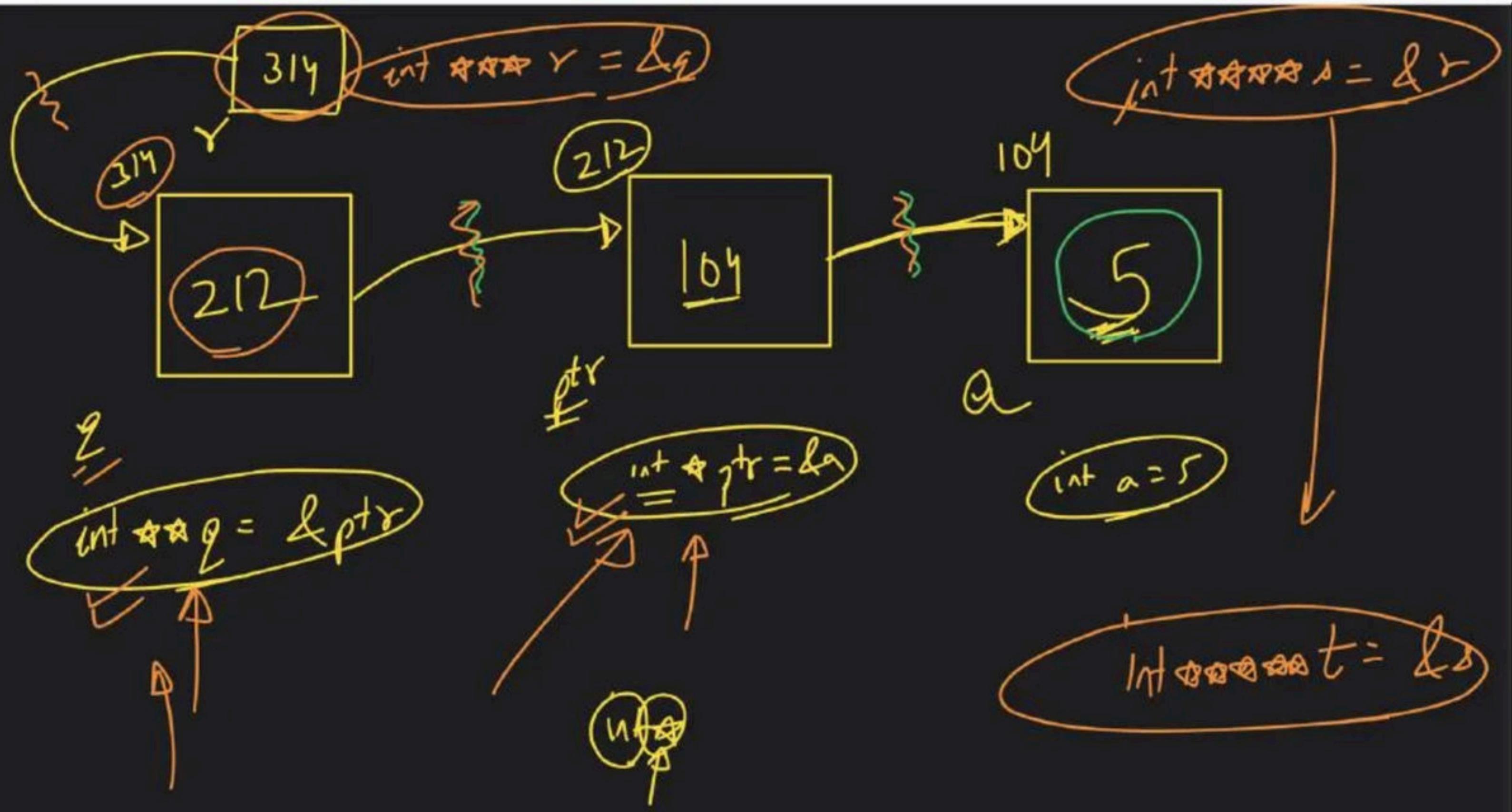
int a = 5;

int *ptr = &a

int **q = ptr

int *q = &ptr





int a = 10;

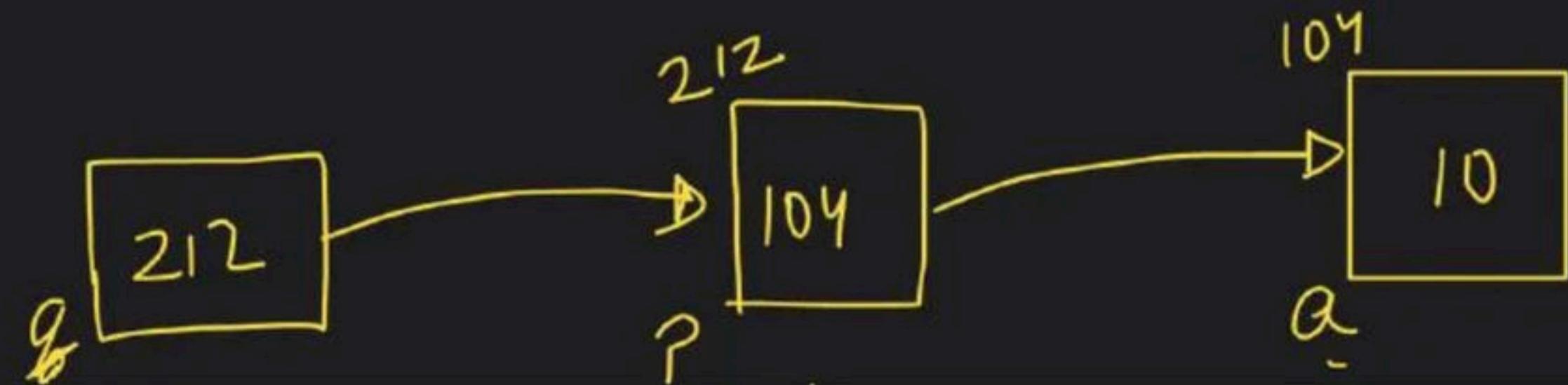
int * p = &a;

int ** q = &p;

int a → pointer to an integer

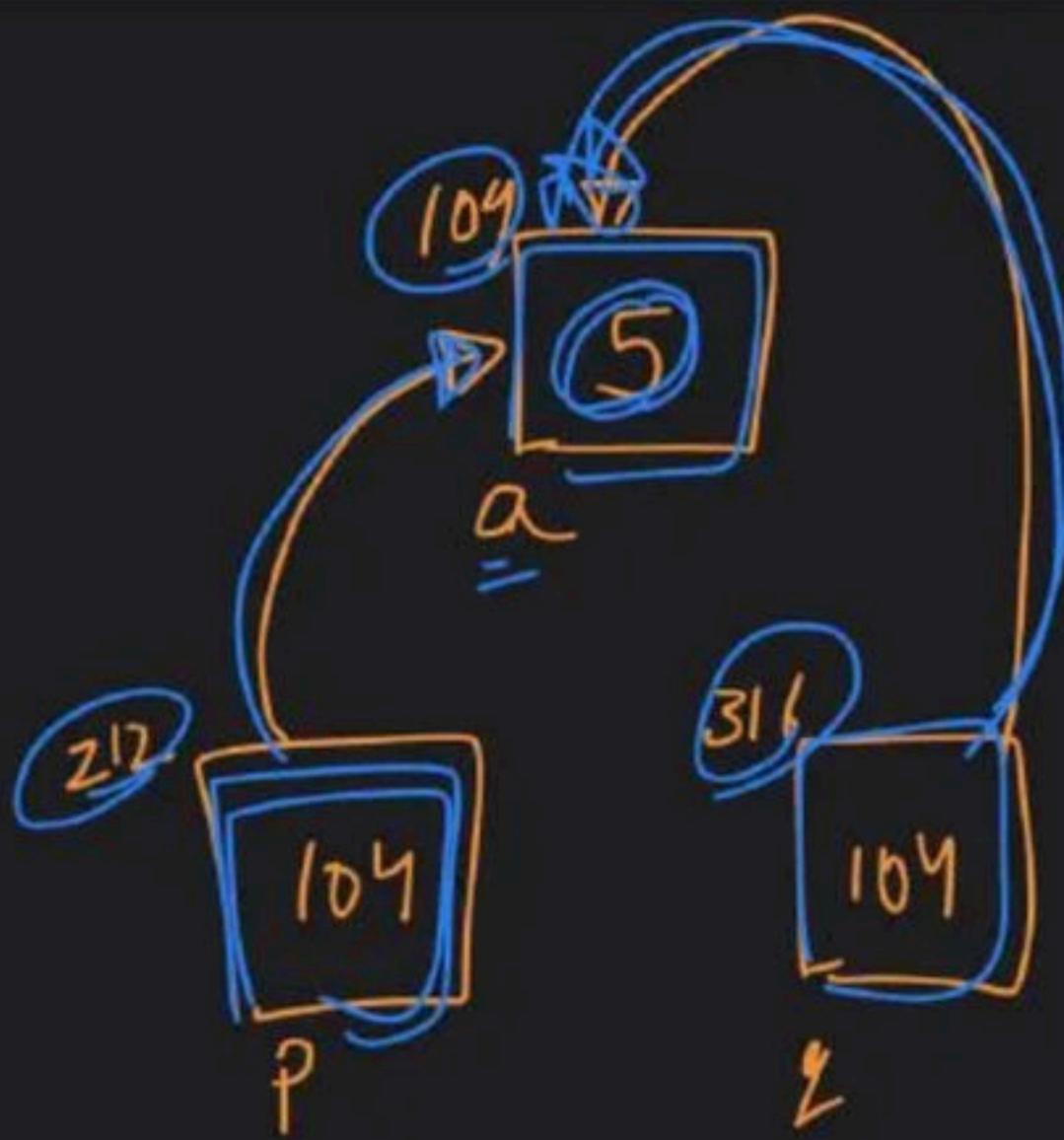
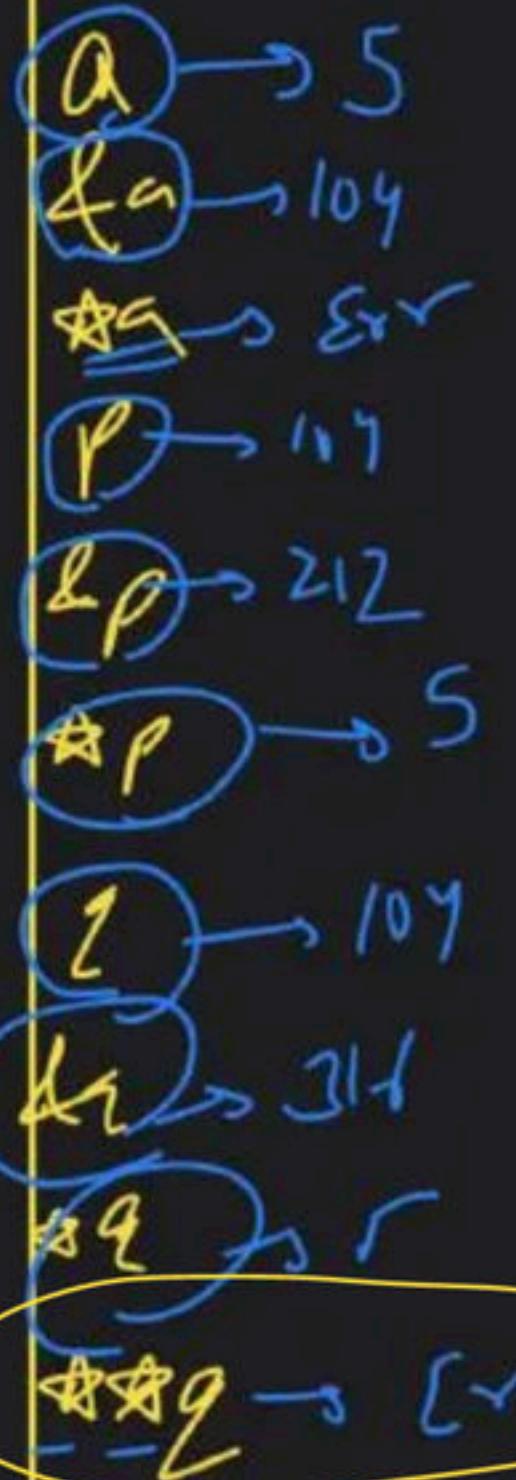
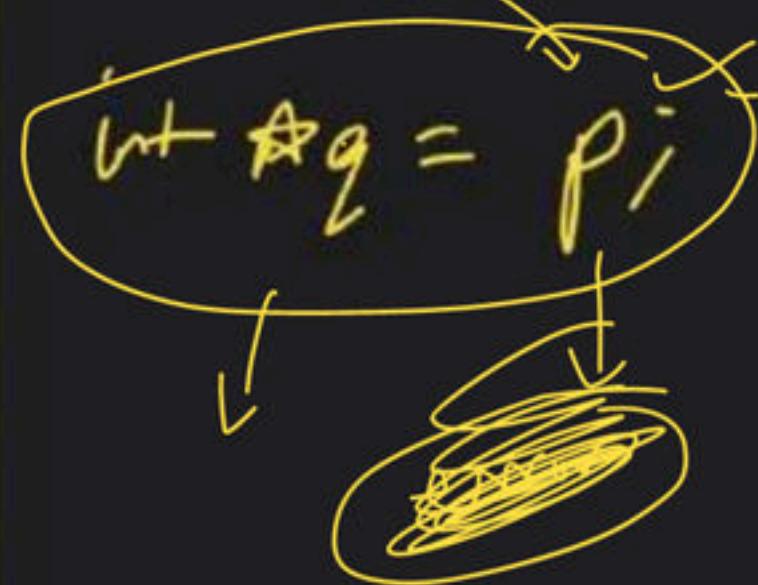
int * p → pointer to a int*

int ** q → pointer to pointer to an integer



int a = 5

int * p = &a

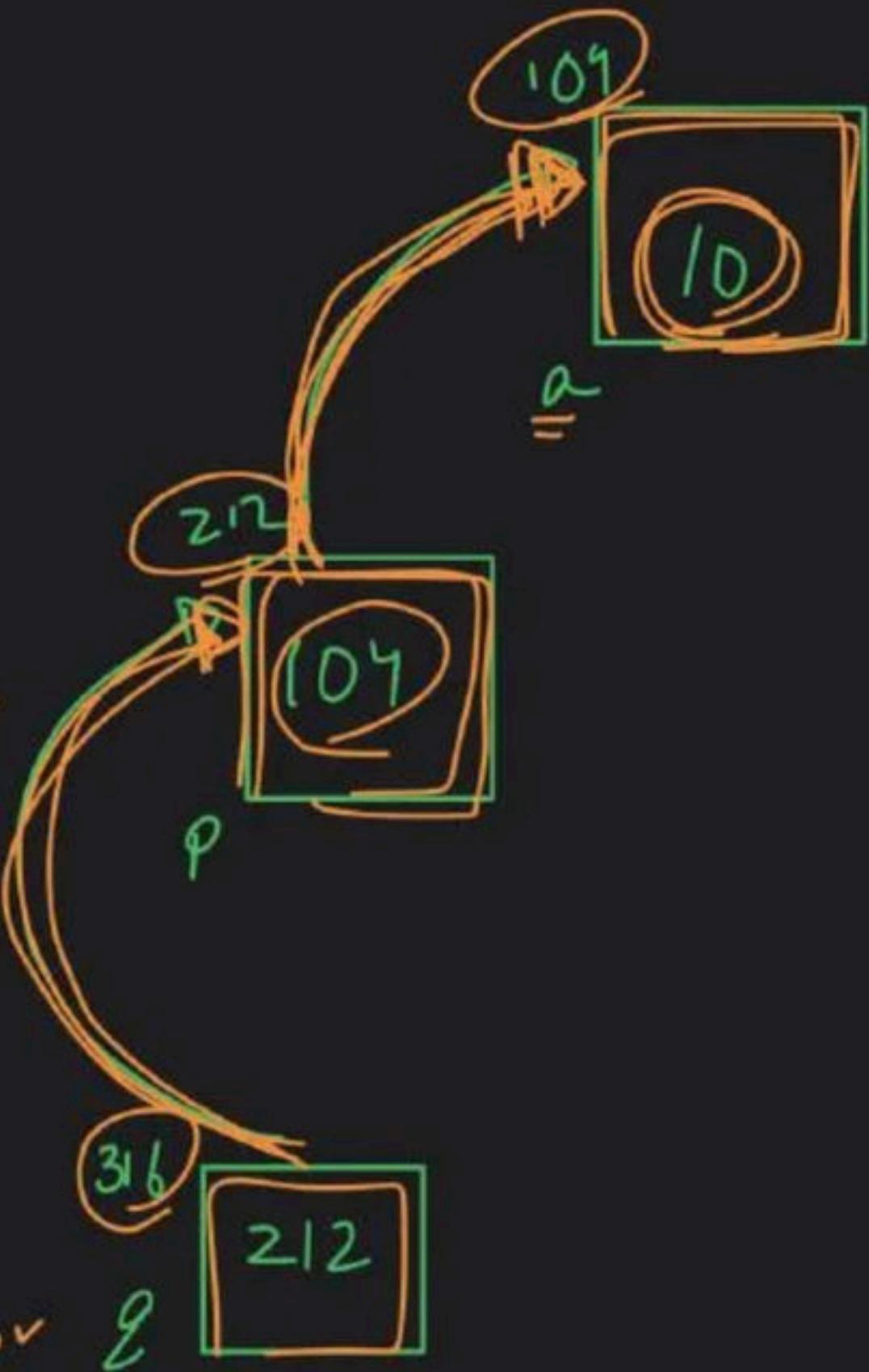
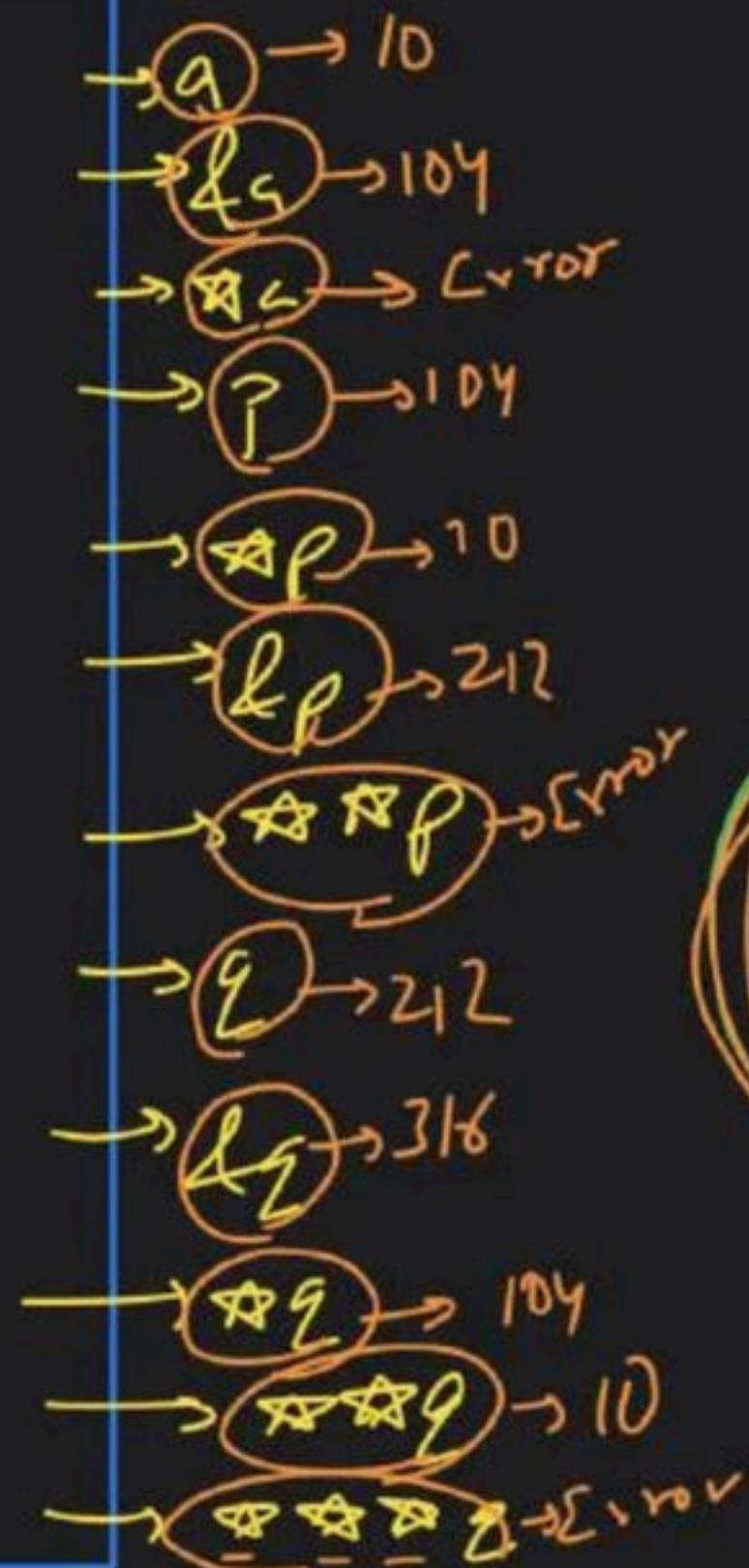


x ← q

int $\alpha = 10;$

int * $p = \underline{\&\alpha};$

int ** $q = \underline{\&p};$



int a = 50

int *p = &a;

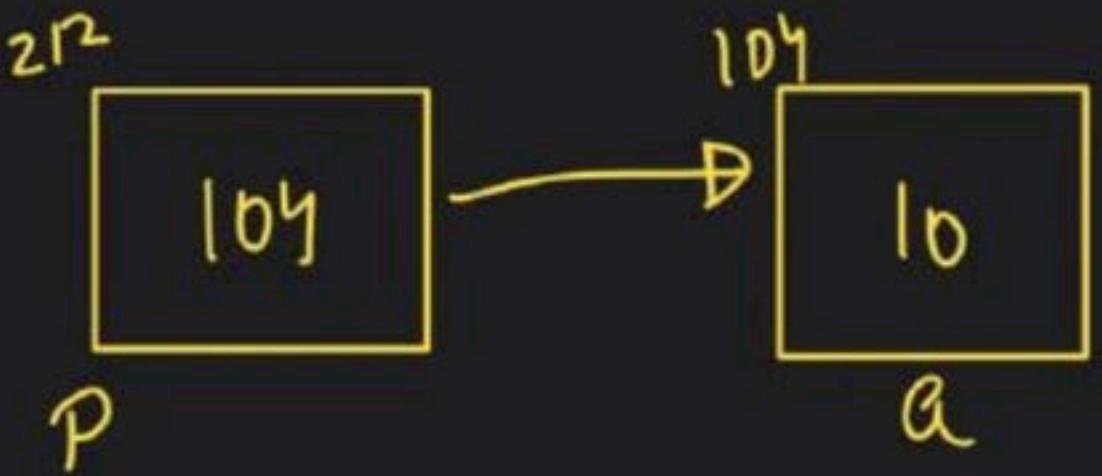
int **p2 = &p;

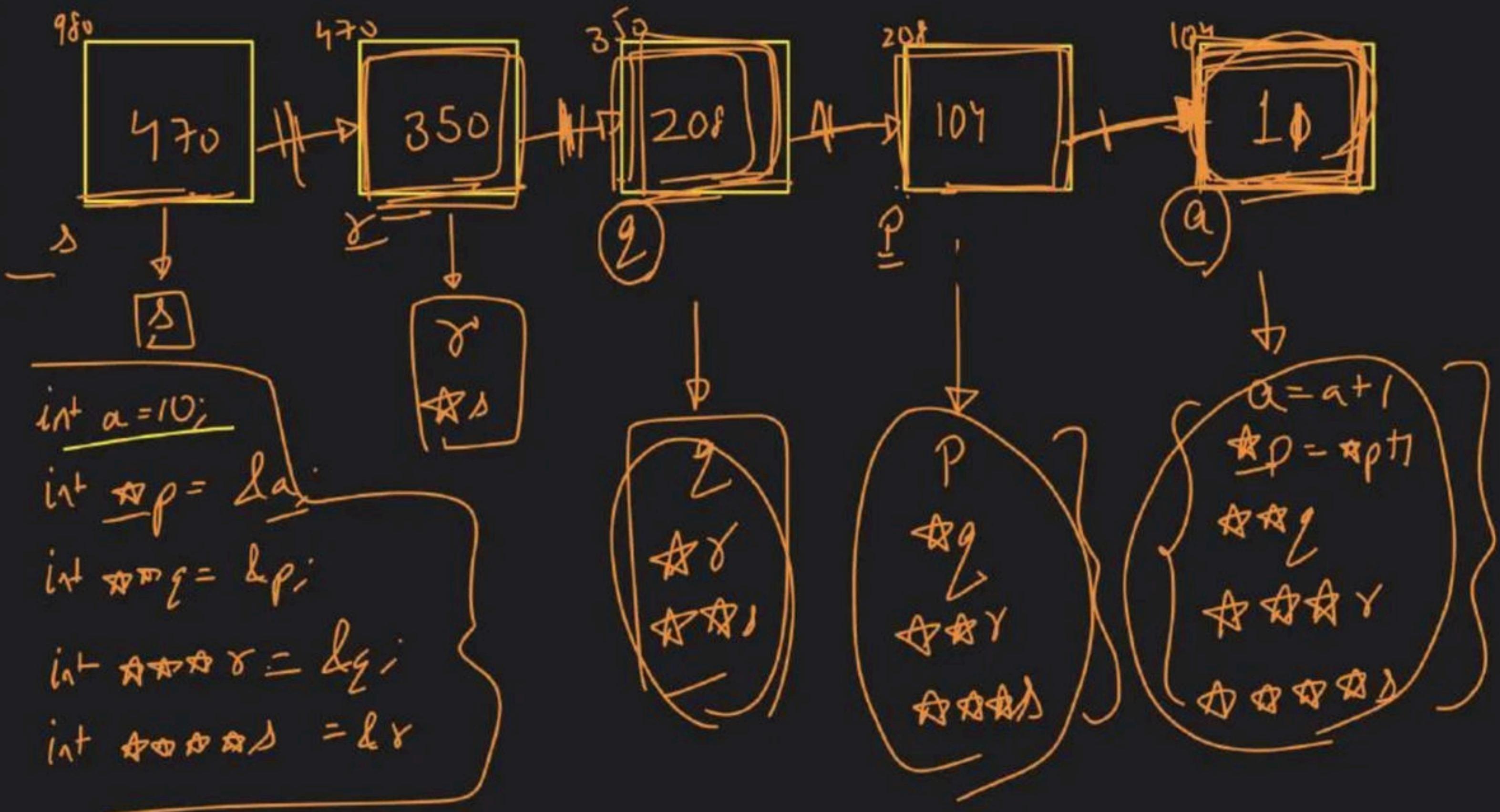
int ***p3 = &p2;

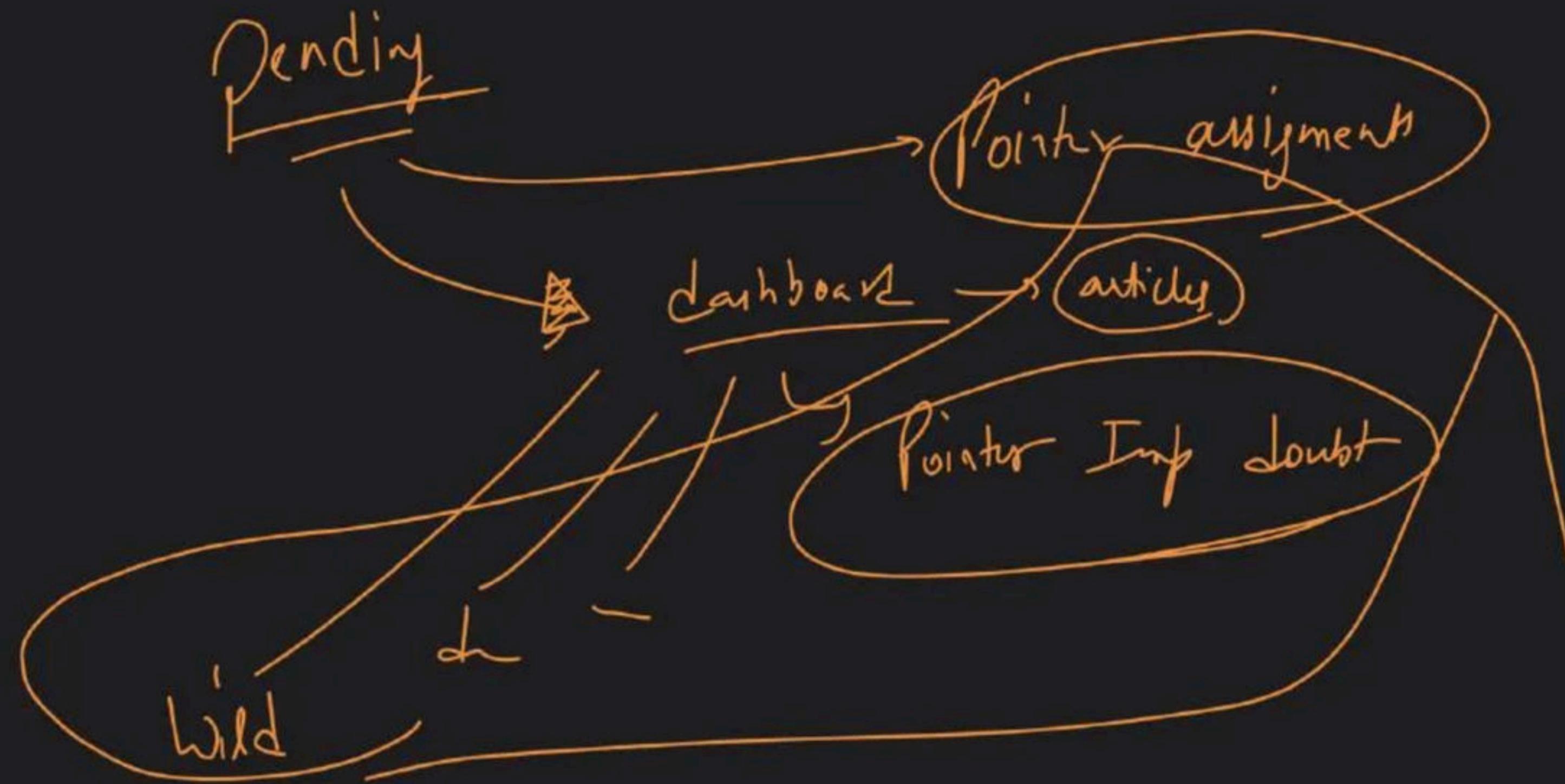
int ****p4 = &p3;

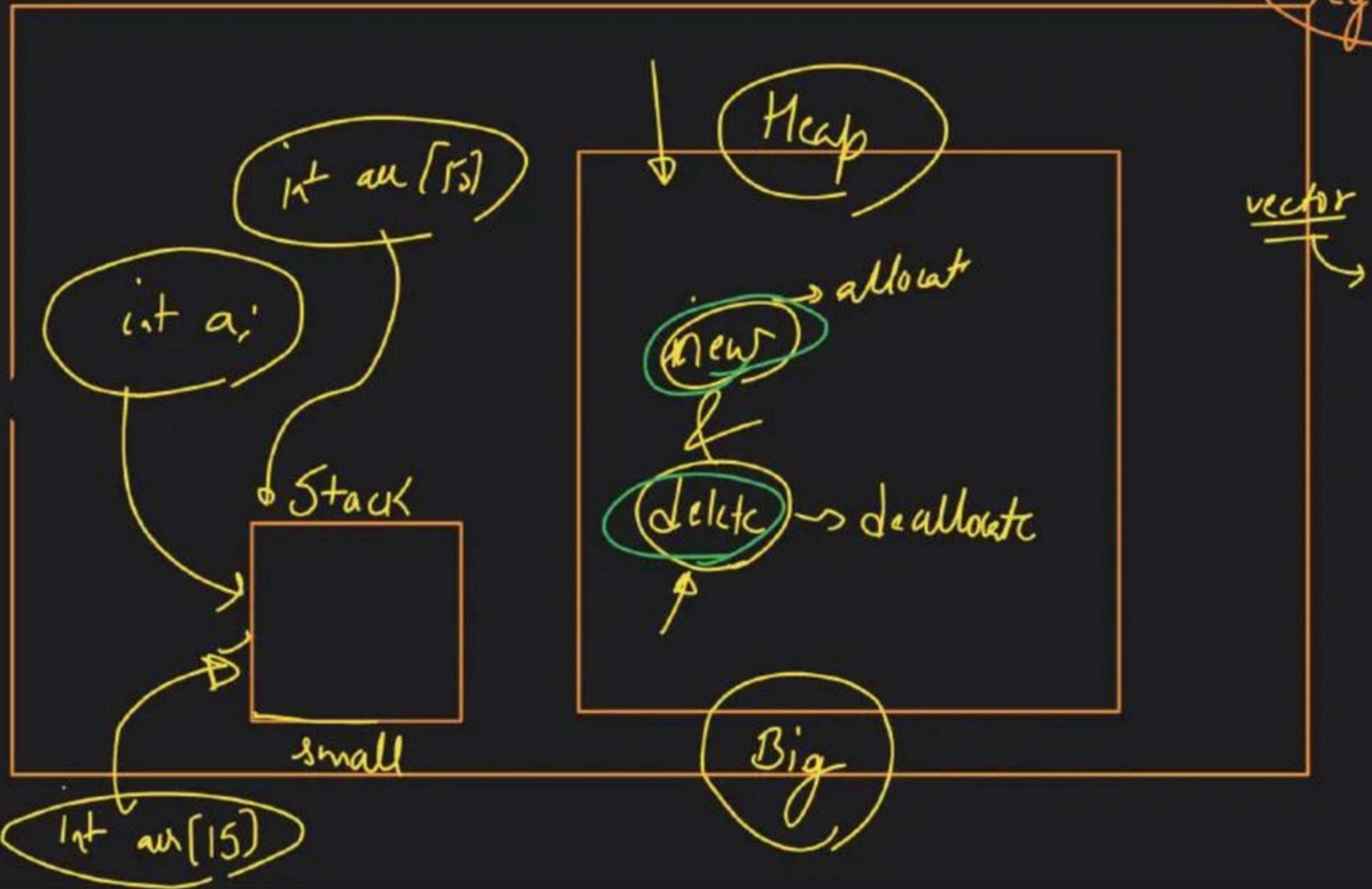


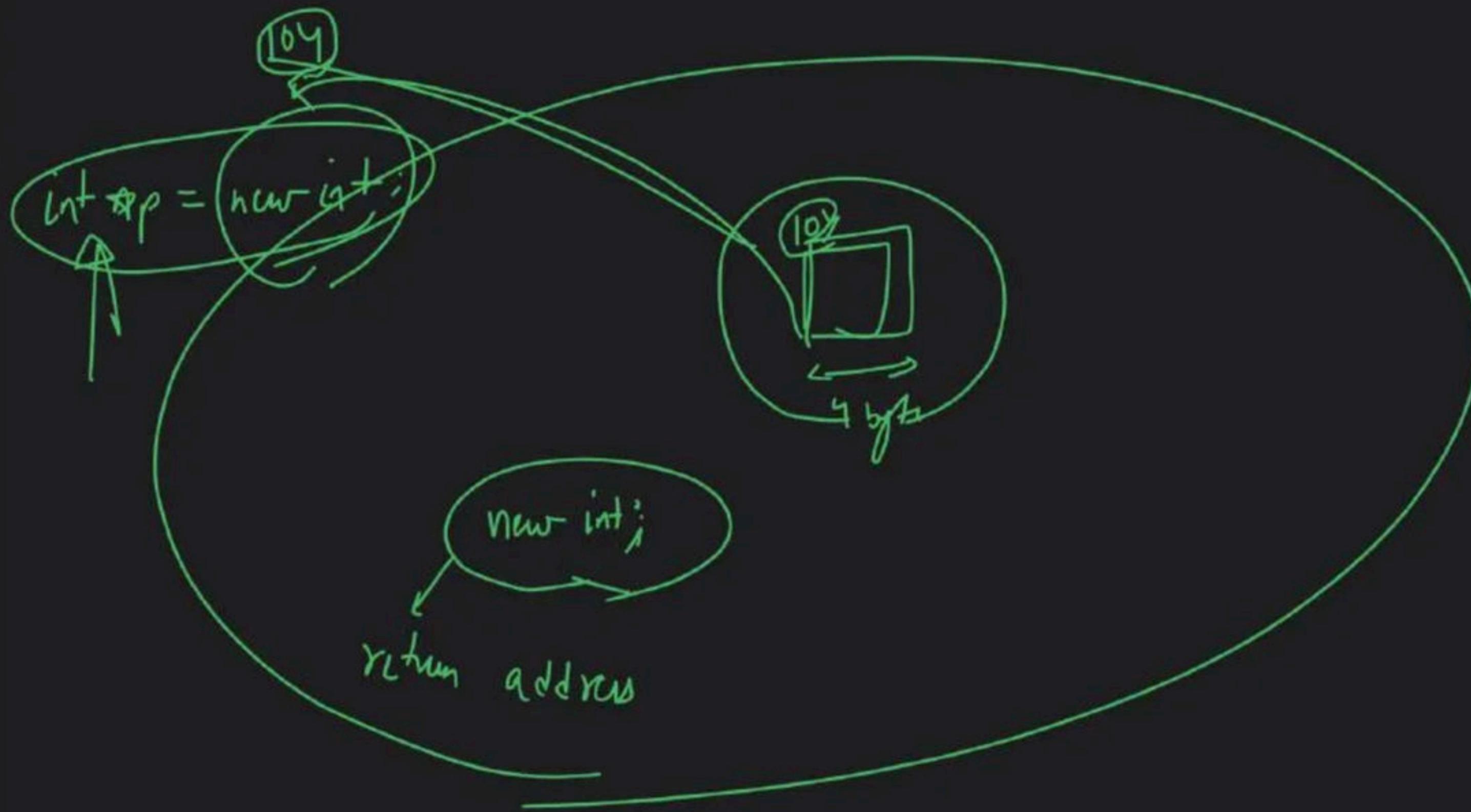
<u>a → 50</u>	50
<u>212 → 104</u>	104
<u>*a → Err</u>	Err
<u>p → 104</u>	104
<u>Δp → 212</u>	212
<u>*p → 50</u>	50
<u>q → 212</u>	212
<u>*1 → 104</u>	104
<u>lq → 314</u>	314
<u>*q → 50</u>	50
<u>y → 212</u>	212
<u>*y → 50</u>	50
<u>人 → 314</u>	314
<u>*人 → 104</u>	104
<u>*y → 50</u>	50

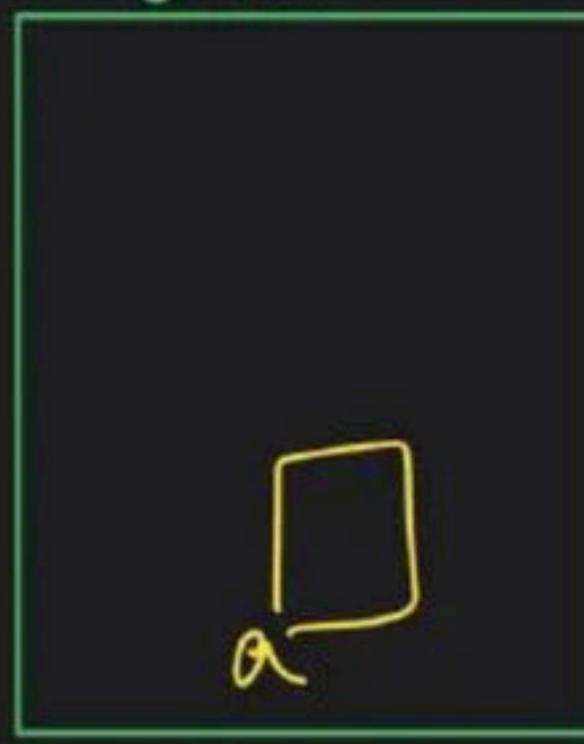




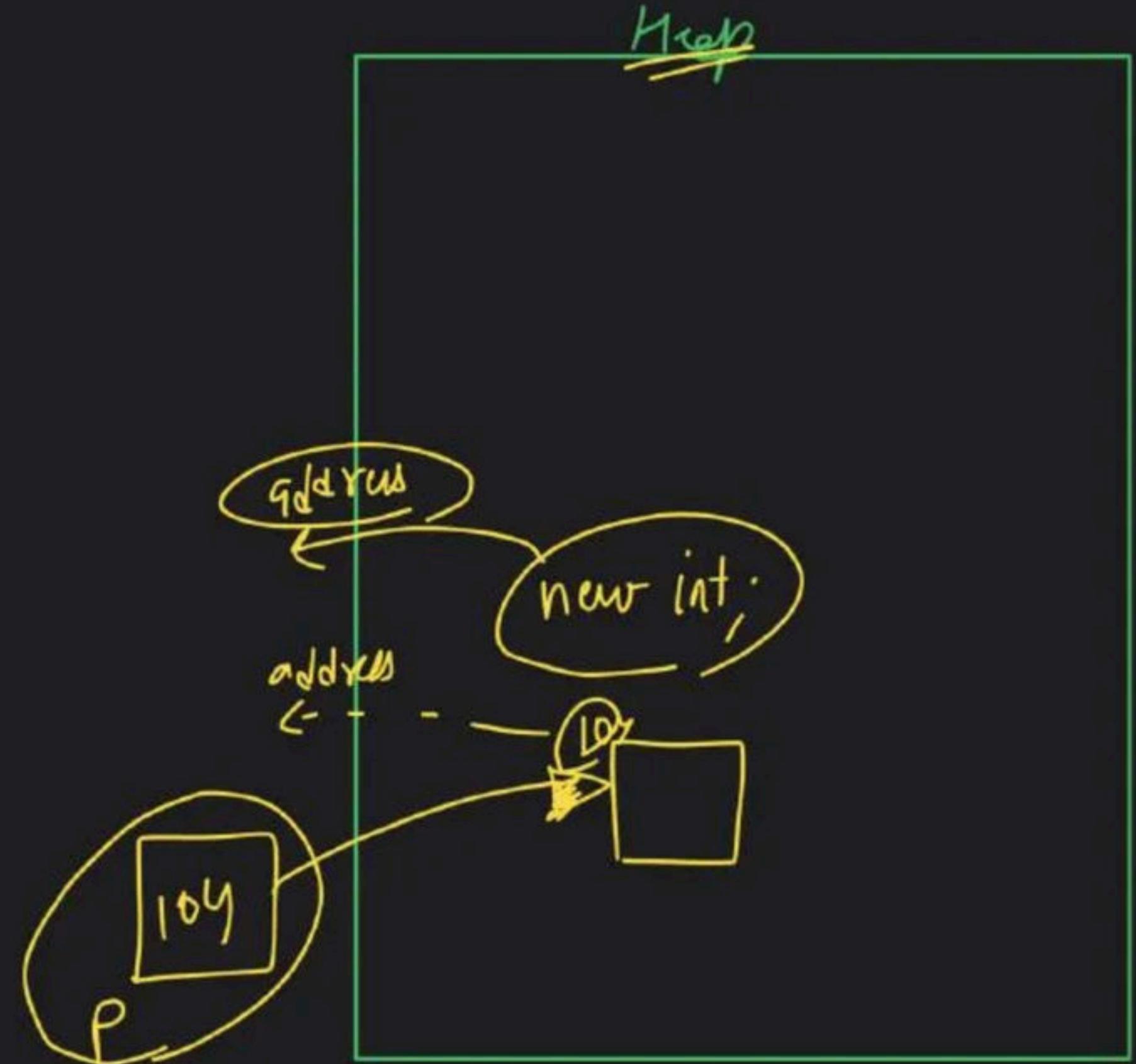


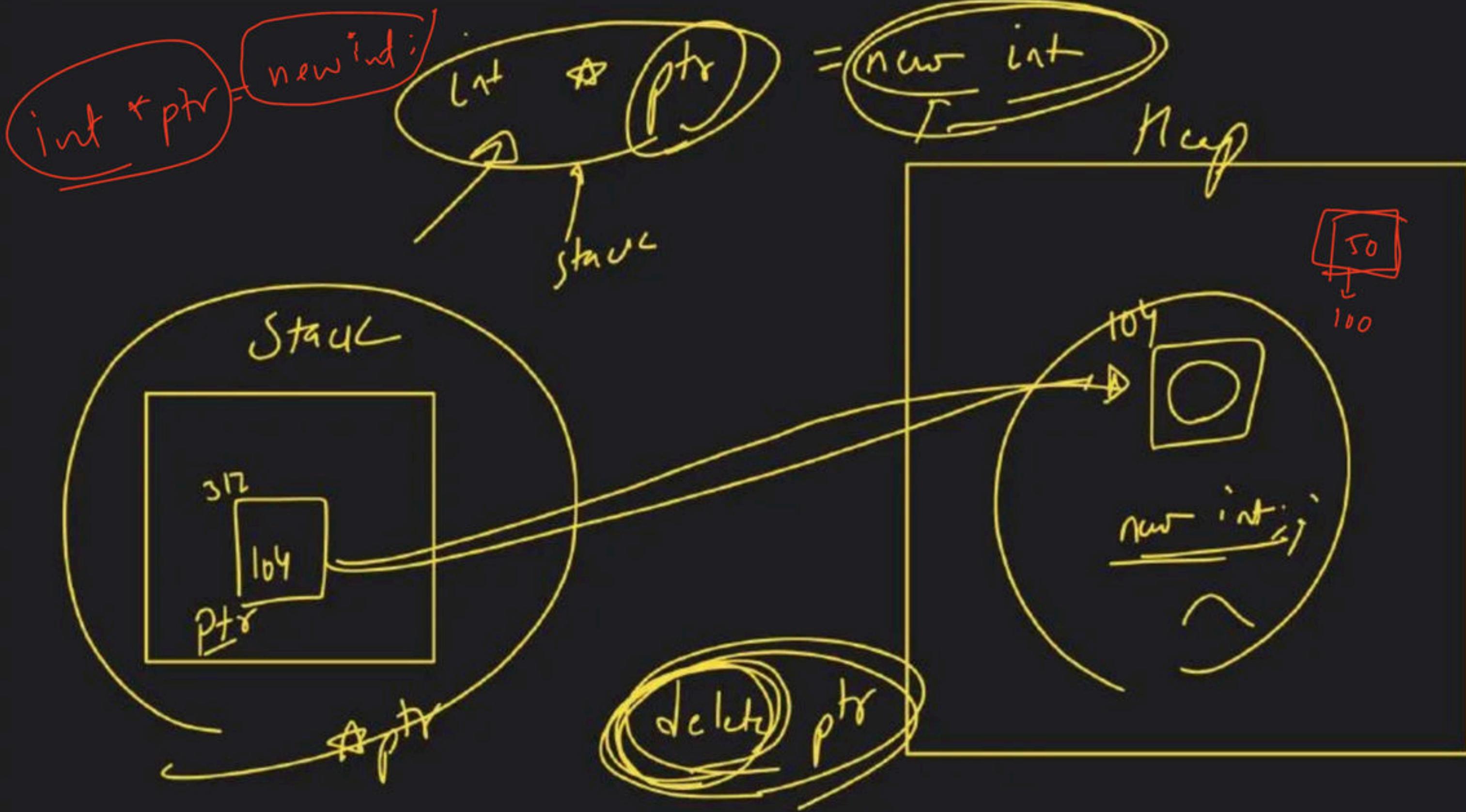


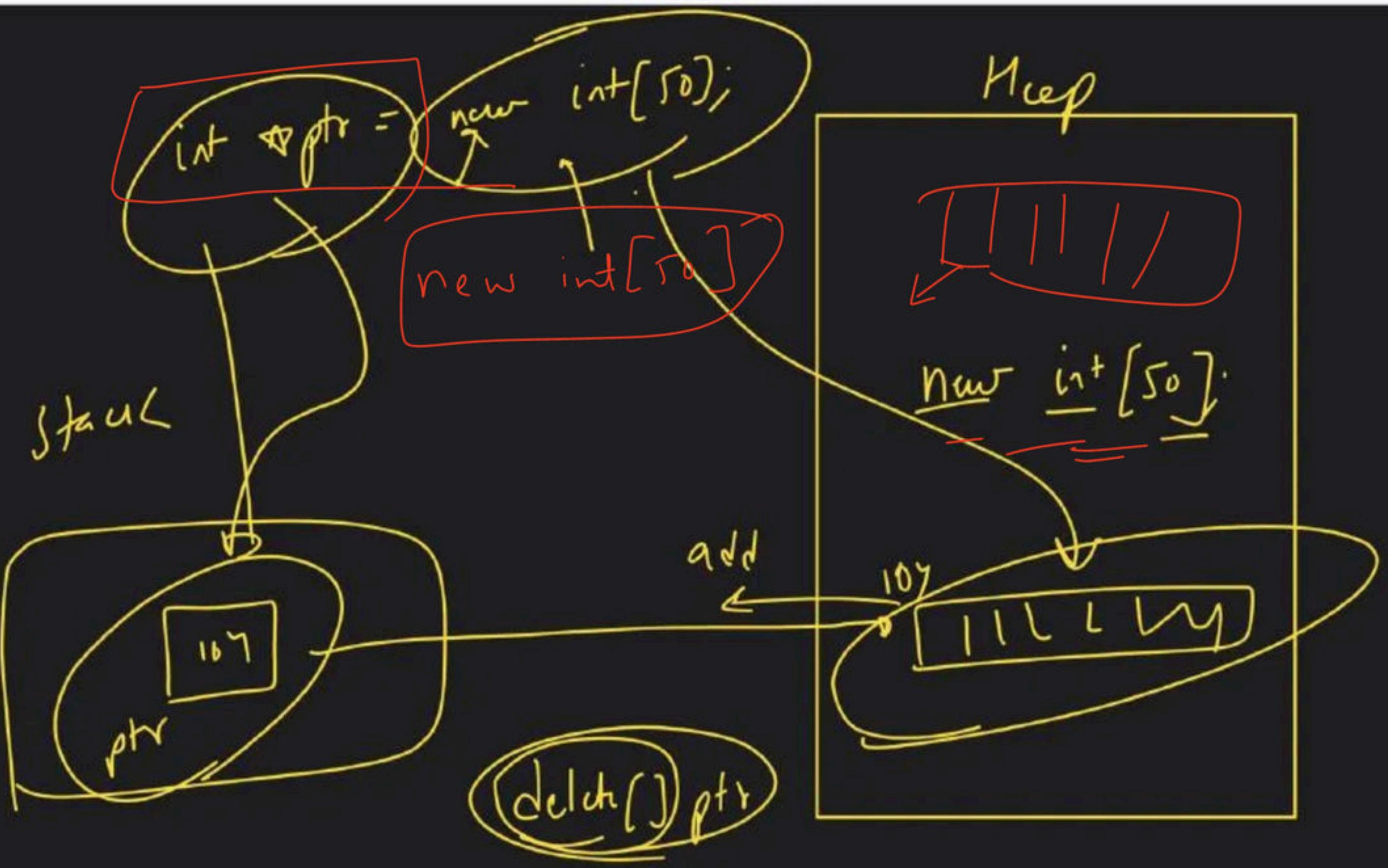




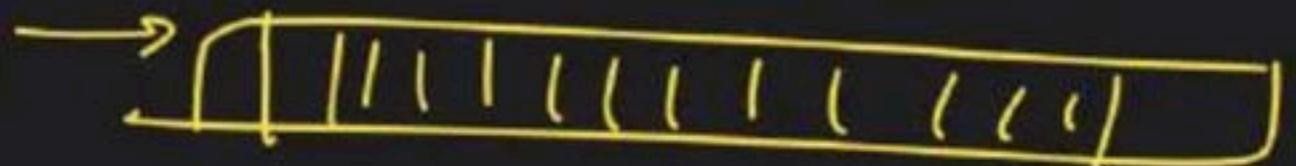
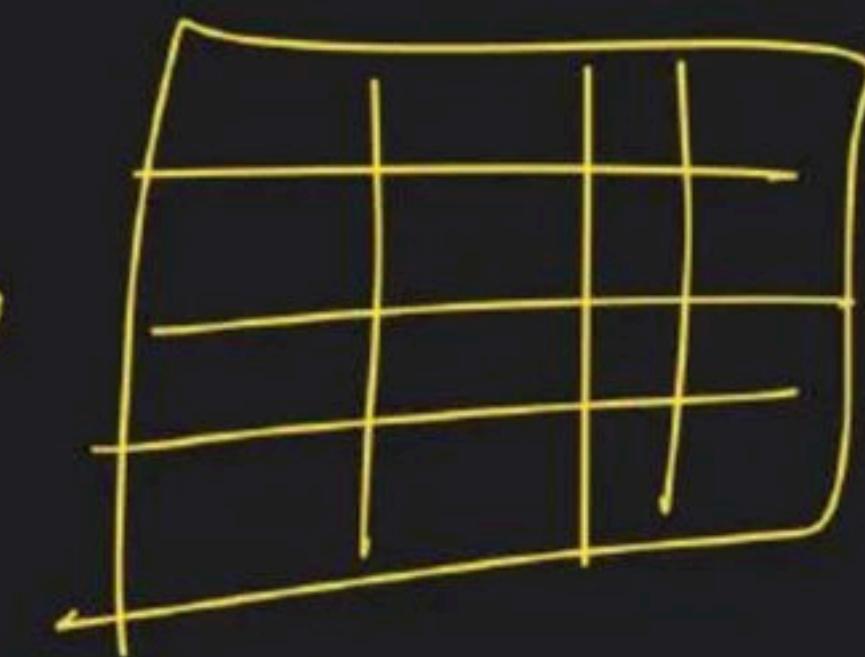
int a;

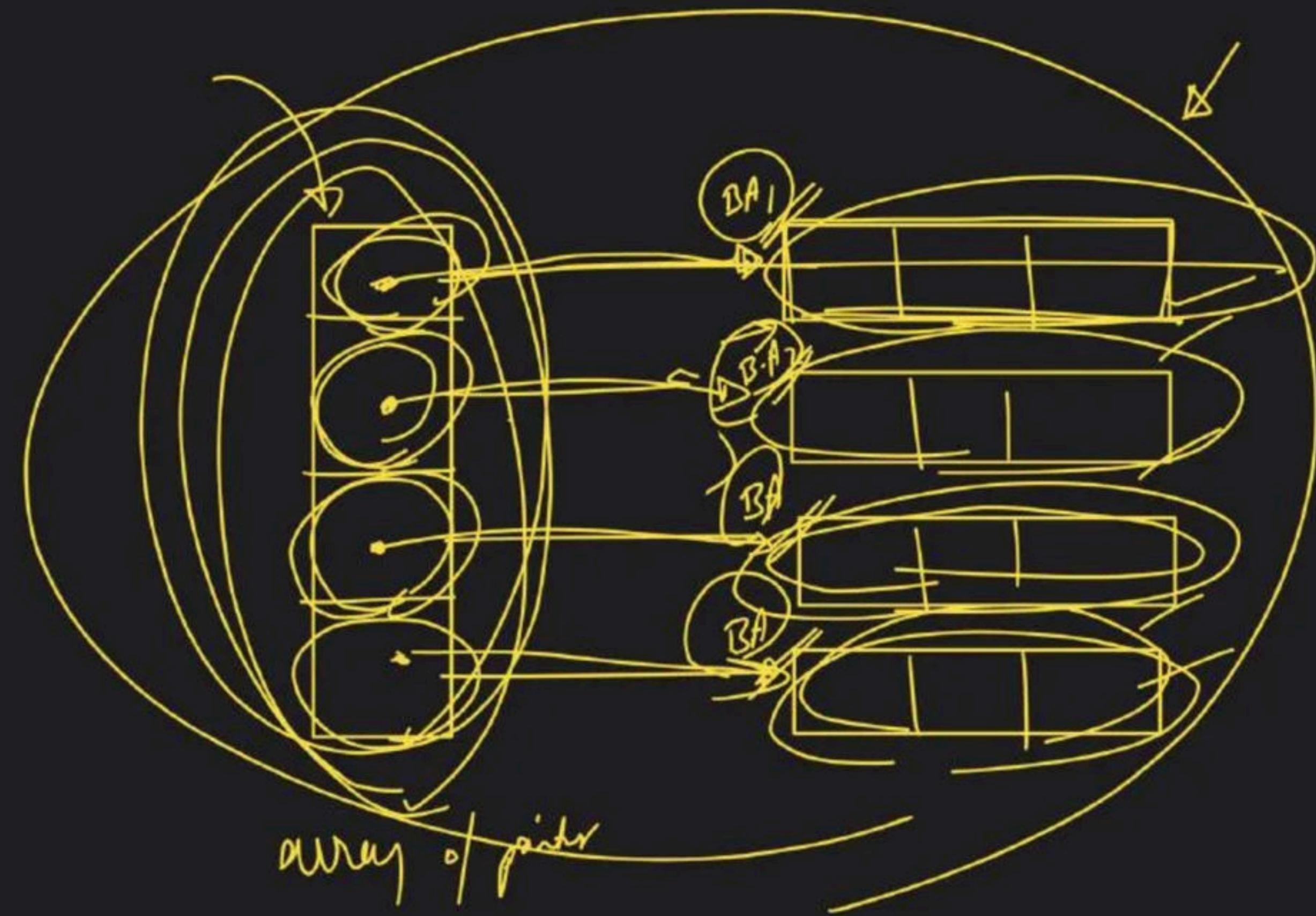


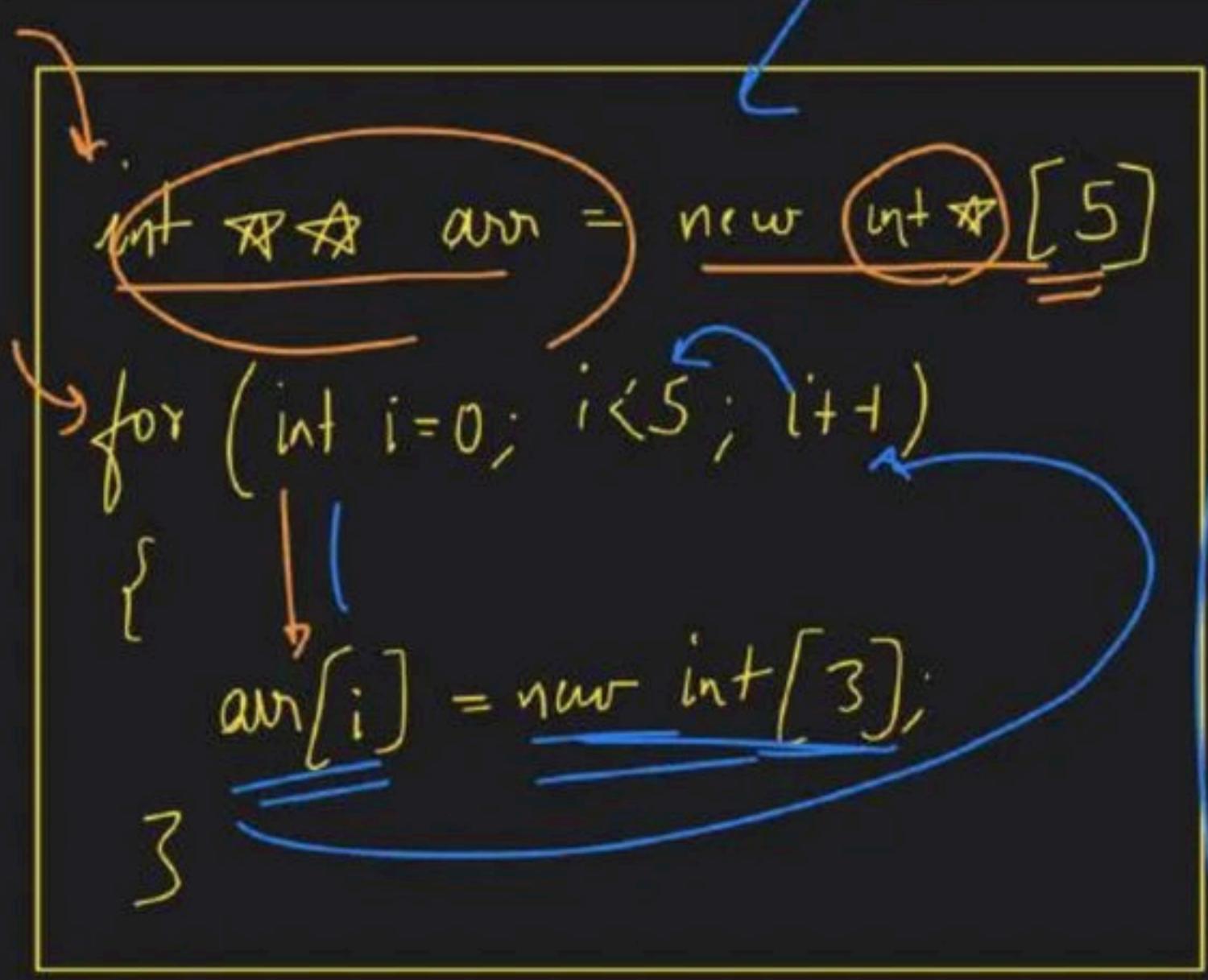




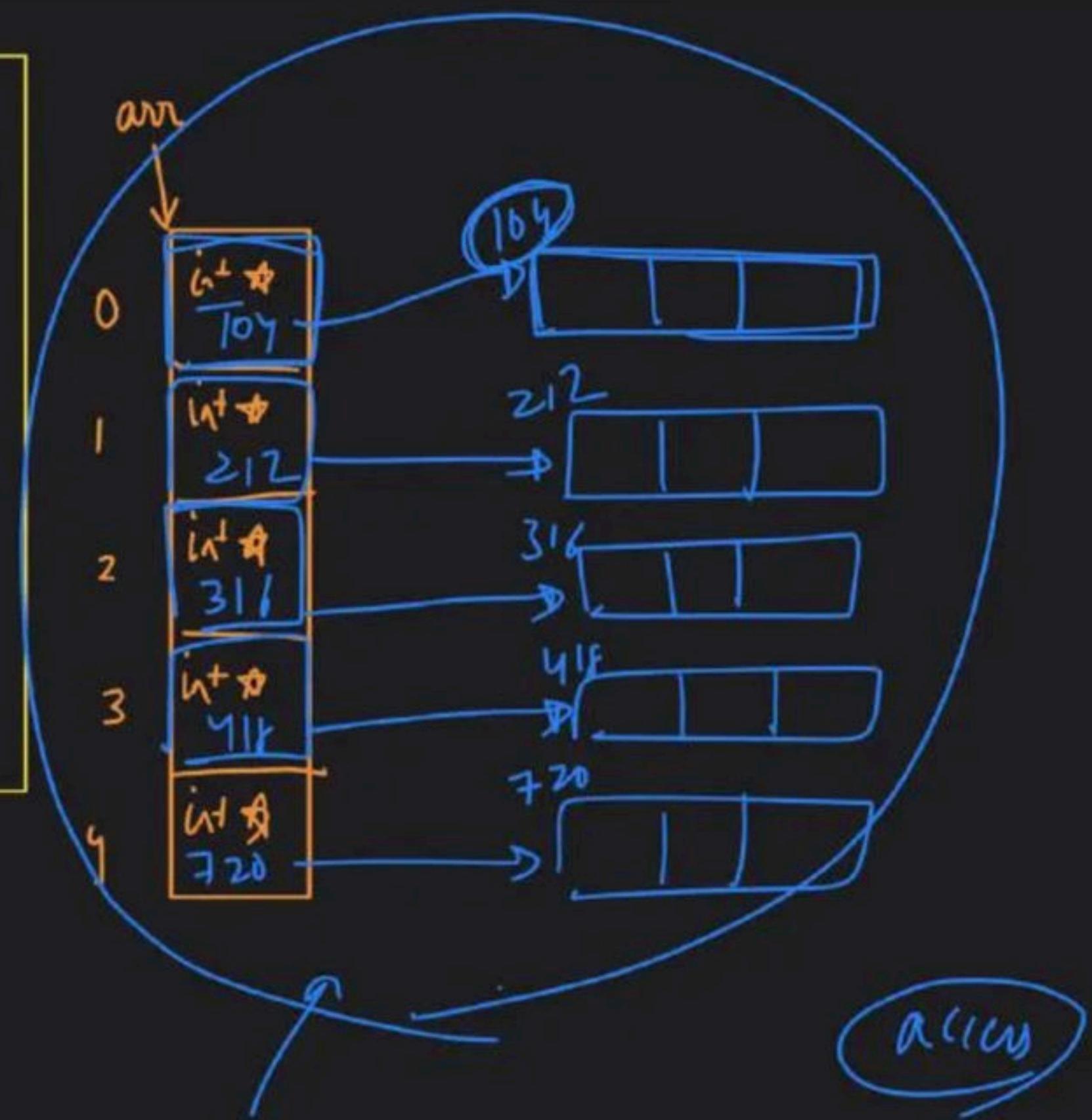
2D array → Heap memory

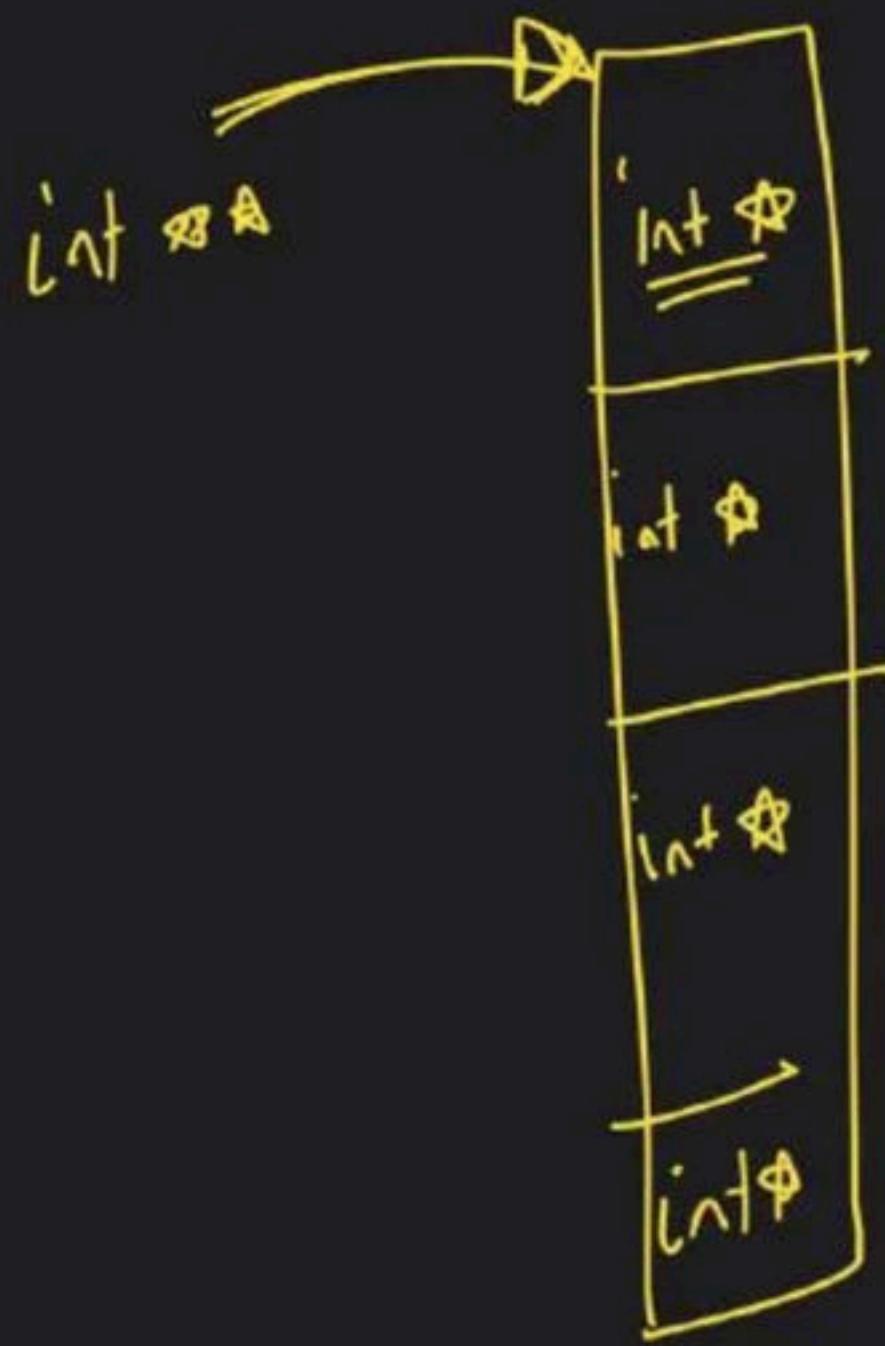






$i=0$
 $arr[0]$ $arr[1]$ $arr[2]$
 $arr[3]$ $arr[4]$





`new int;`

~~`int new int[10]`~~

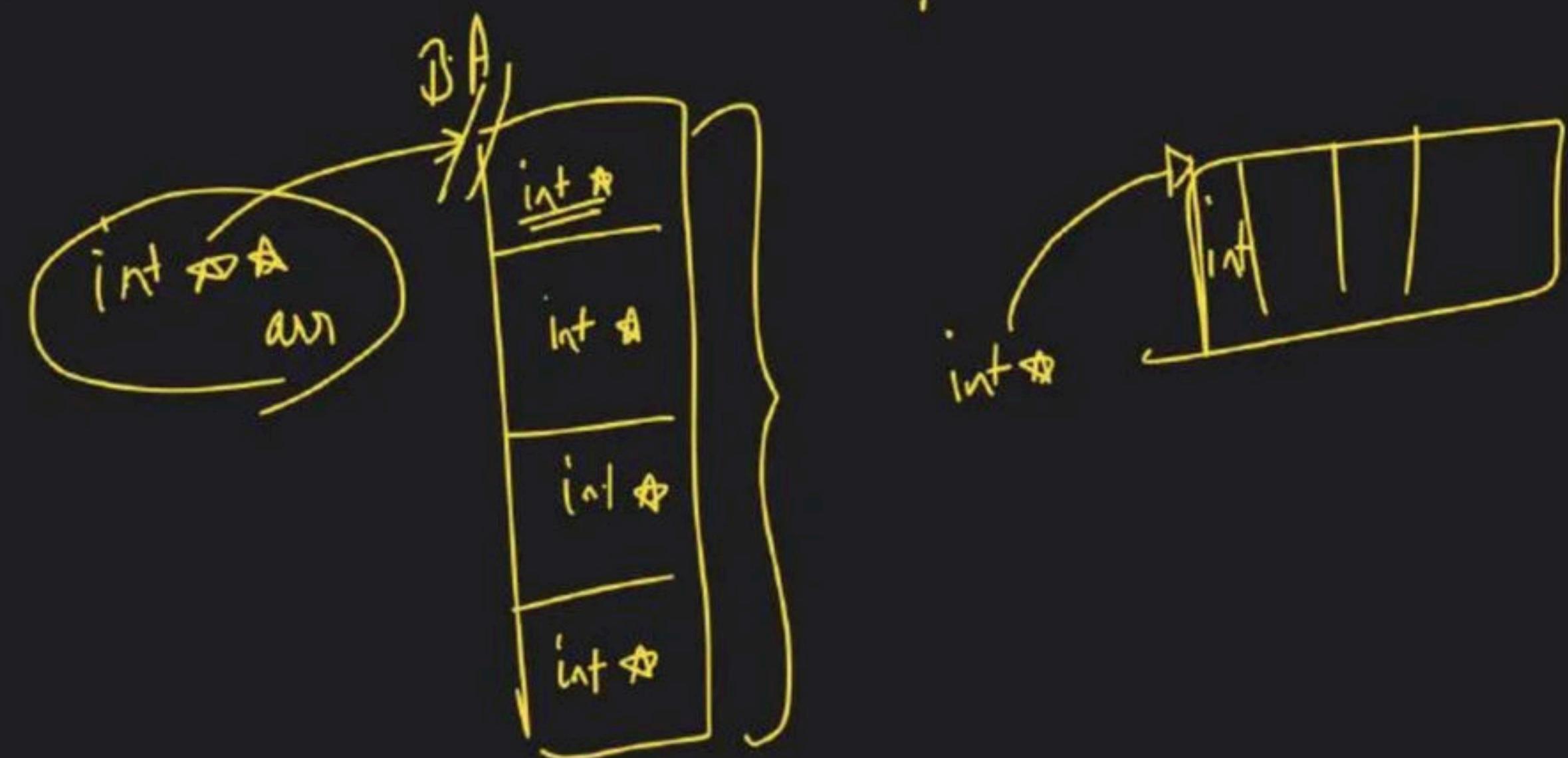
array of integer of size 10

`new int*[10]`

array of int * of size 10

~~array of [pointer to integer] of size 10~~

int * arr = new int [n]

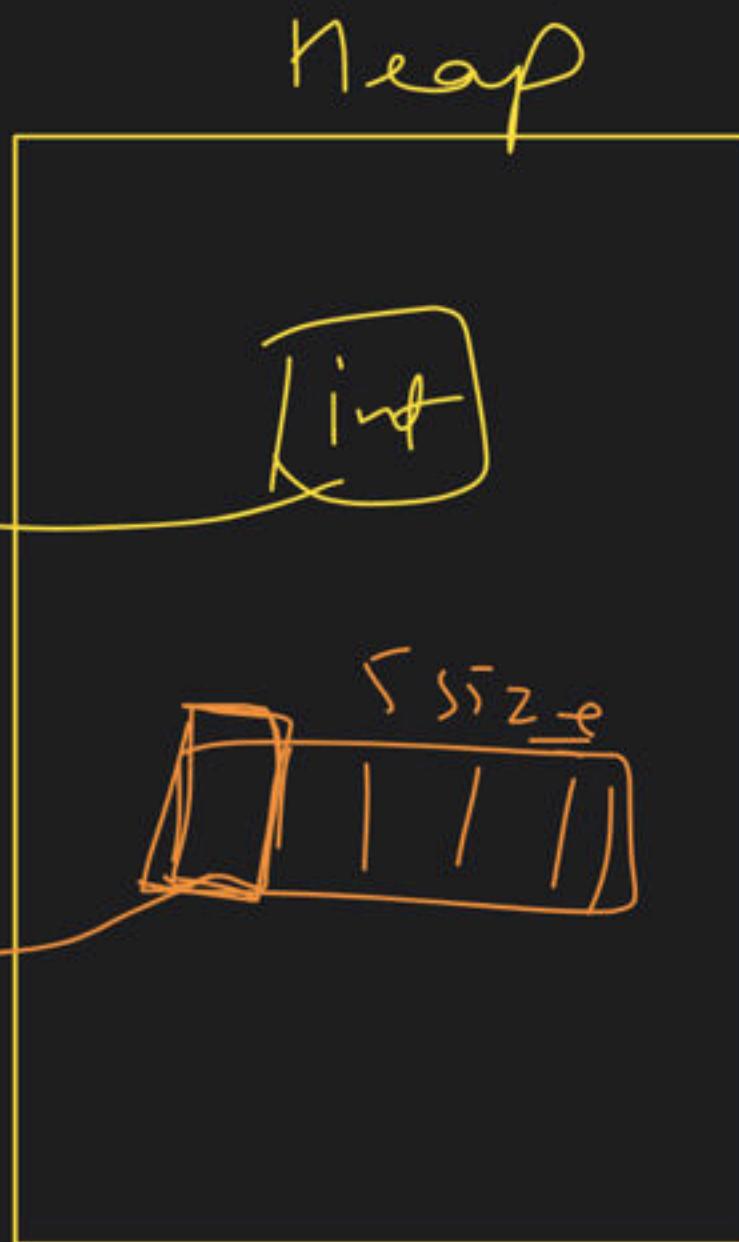


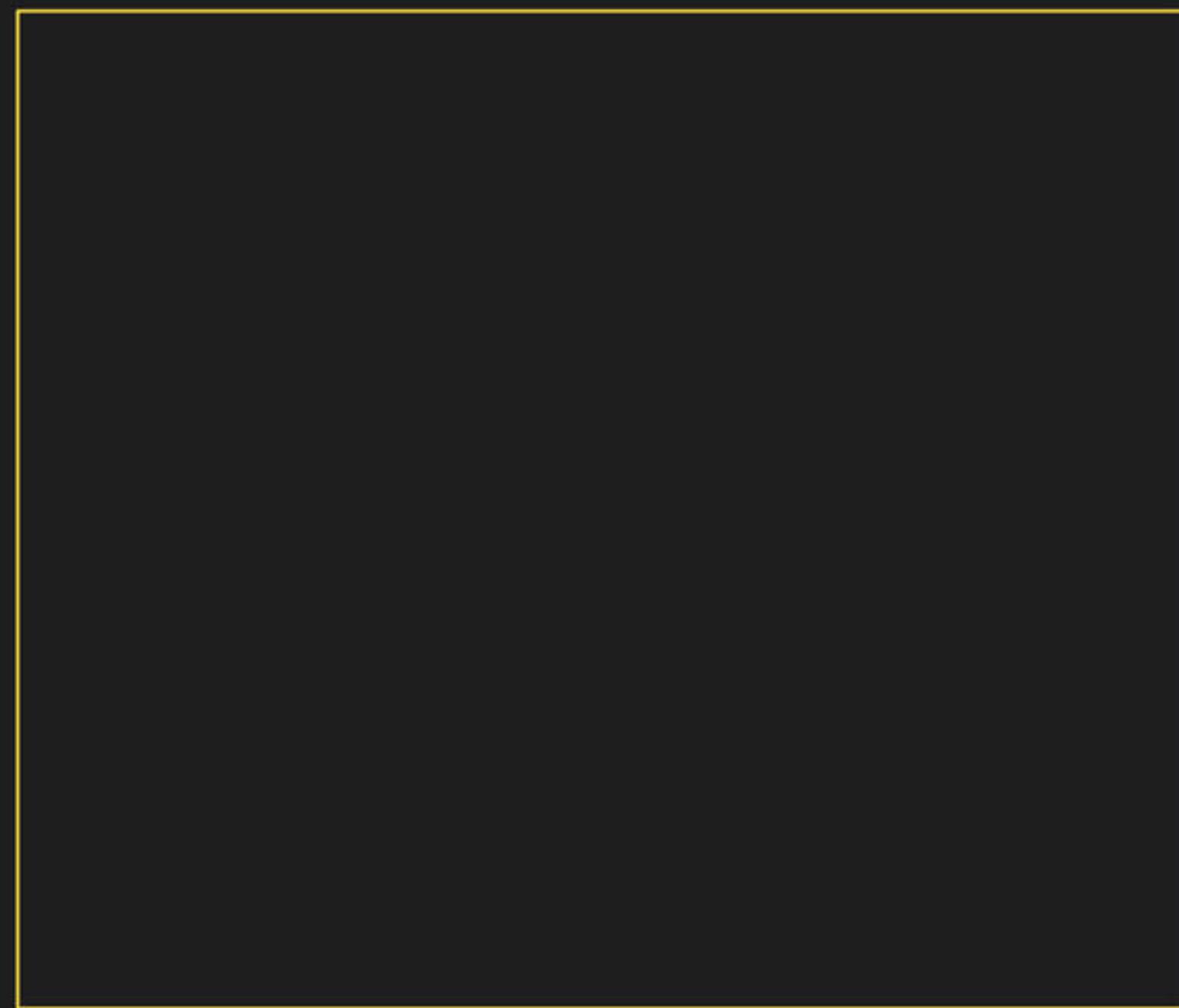
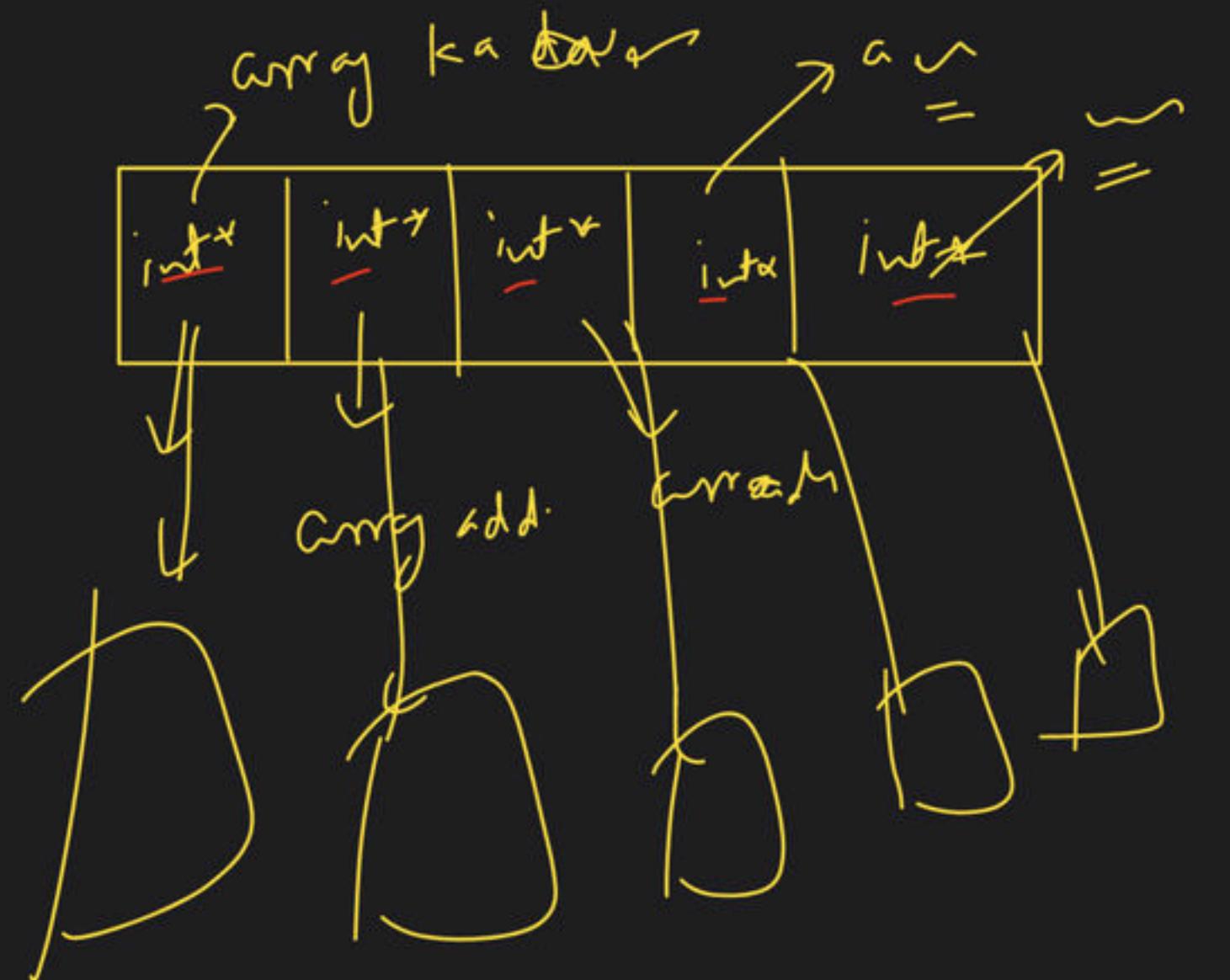
H/W

Memory Leak

H/W

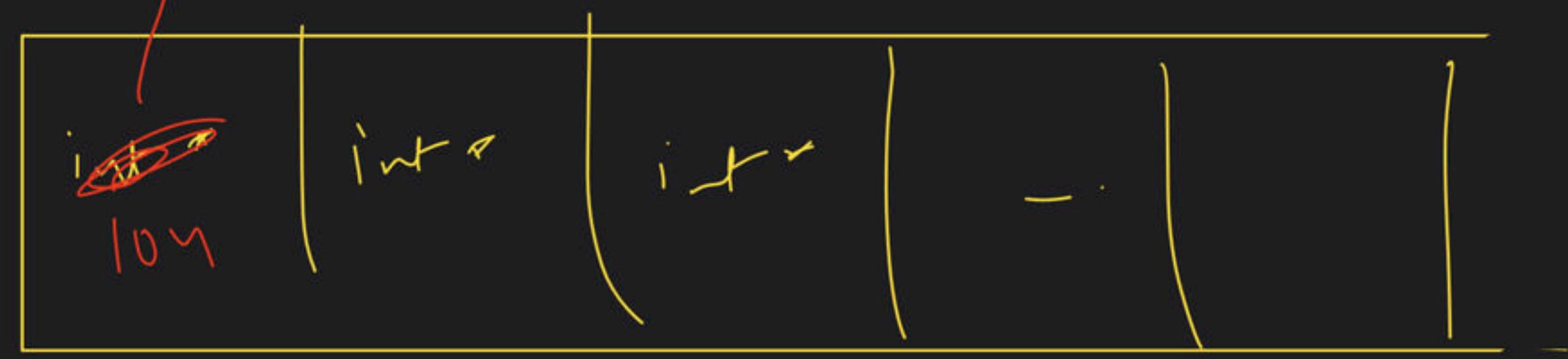
`int * ptry = new int` add. `int`





int * arr = new int [5]

$\text{int}^* \text{arr} = \text{new } \text{int}^*[5];$



fun.

`arr[i] = new int[6];`

New



delete

malloc



free

C

++
=

Kal → Extra

class

T-shirt
~~'Early 500'~~

~~400 address~~

~~100~~

Recursion

8:30

2030 hours















































