



Pointers - Class 1

Special class

Variable

int a = 5;

l,wood
0

vegg
0 0

Milk

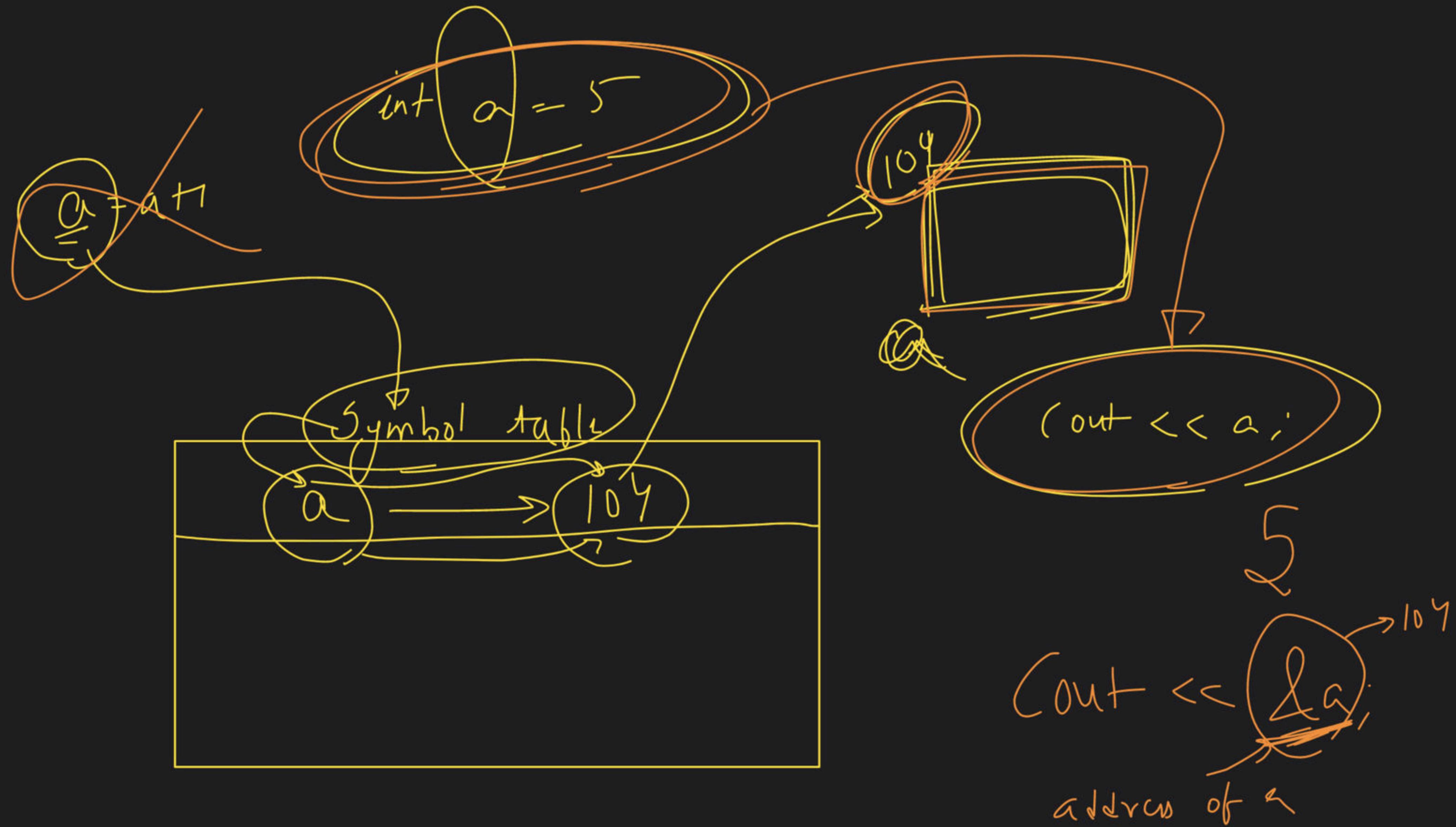
COLD STORED

Pointers



address

store address



`int a = 5;`

`cout << a;`

`int b = a;`

~~Int~~

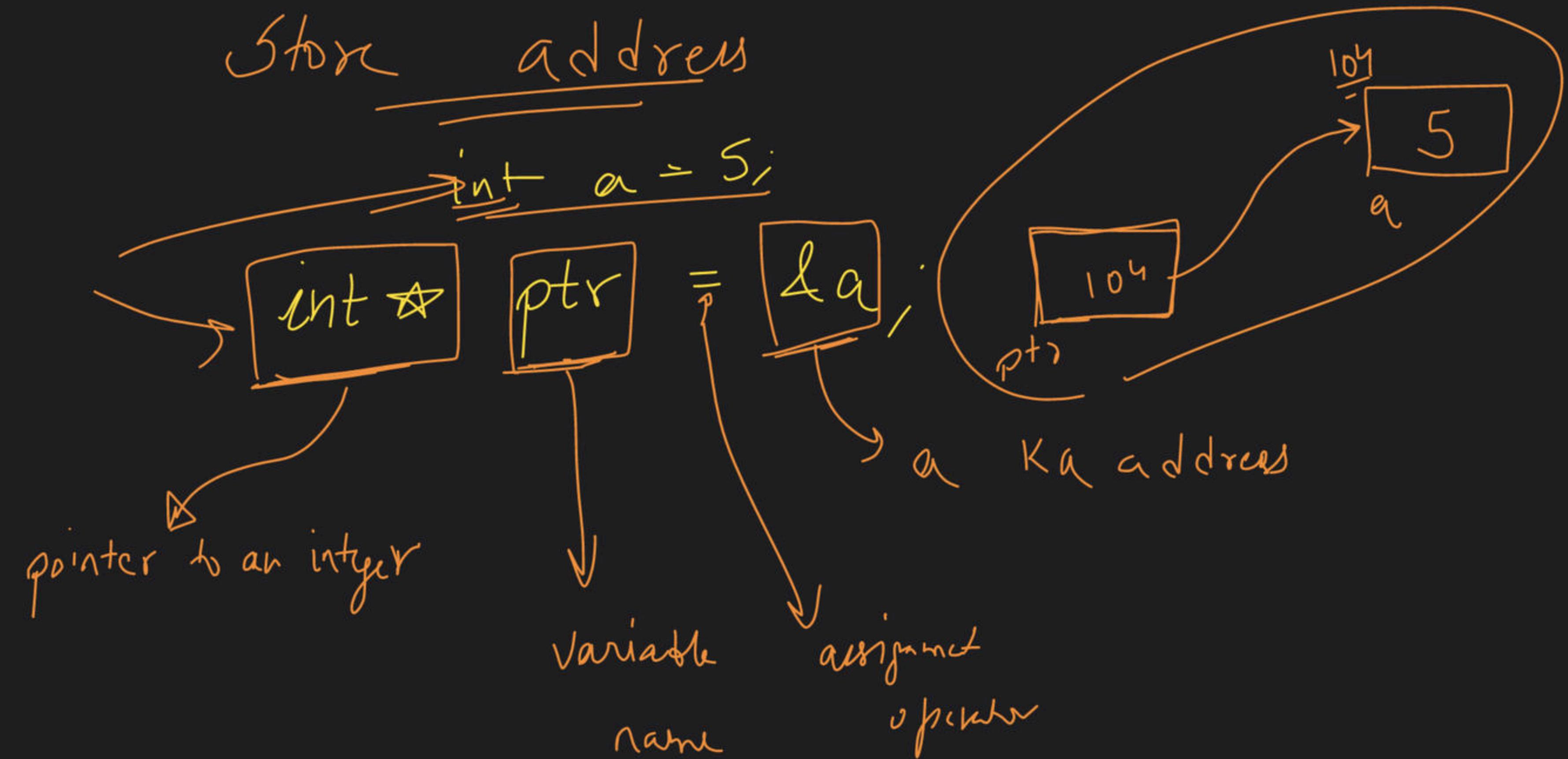
~~c~~

~~a~~

~~var~~

~~pointer~~

~~addr~~





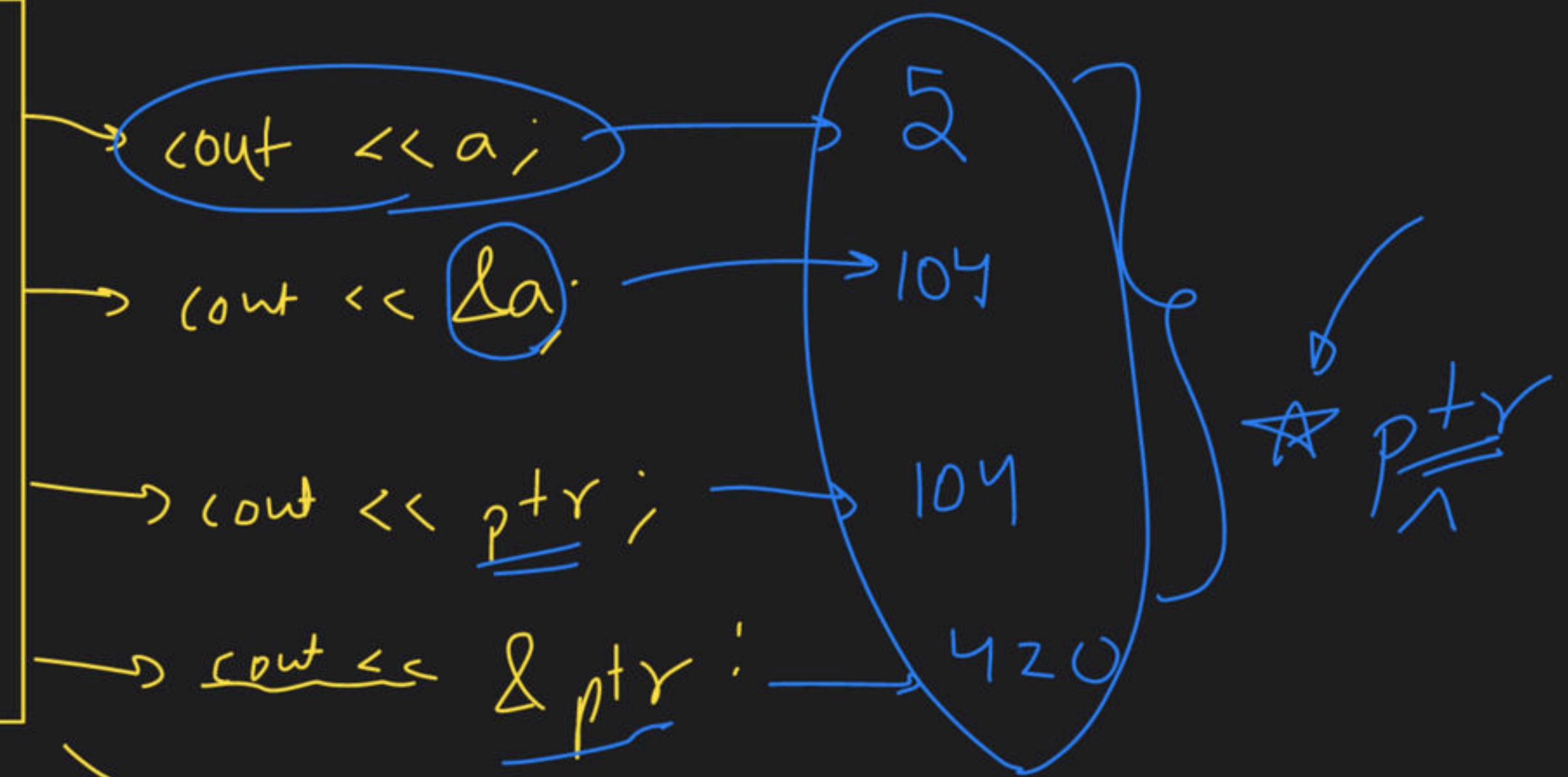
Custom
Datatype → Babbar

Babbar * ptr

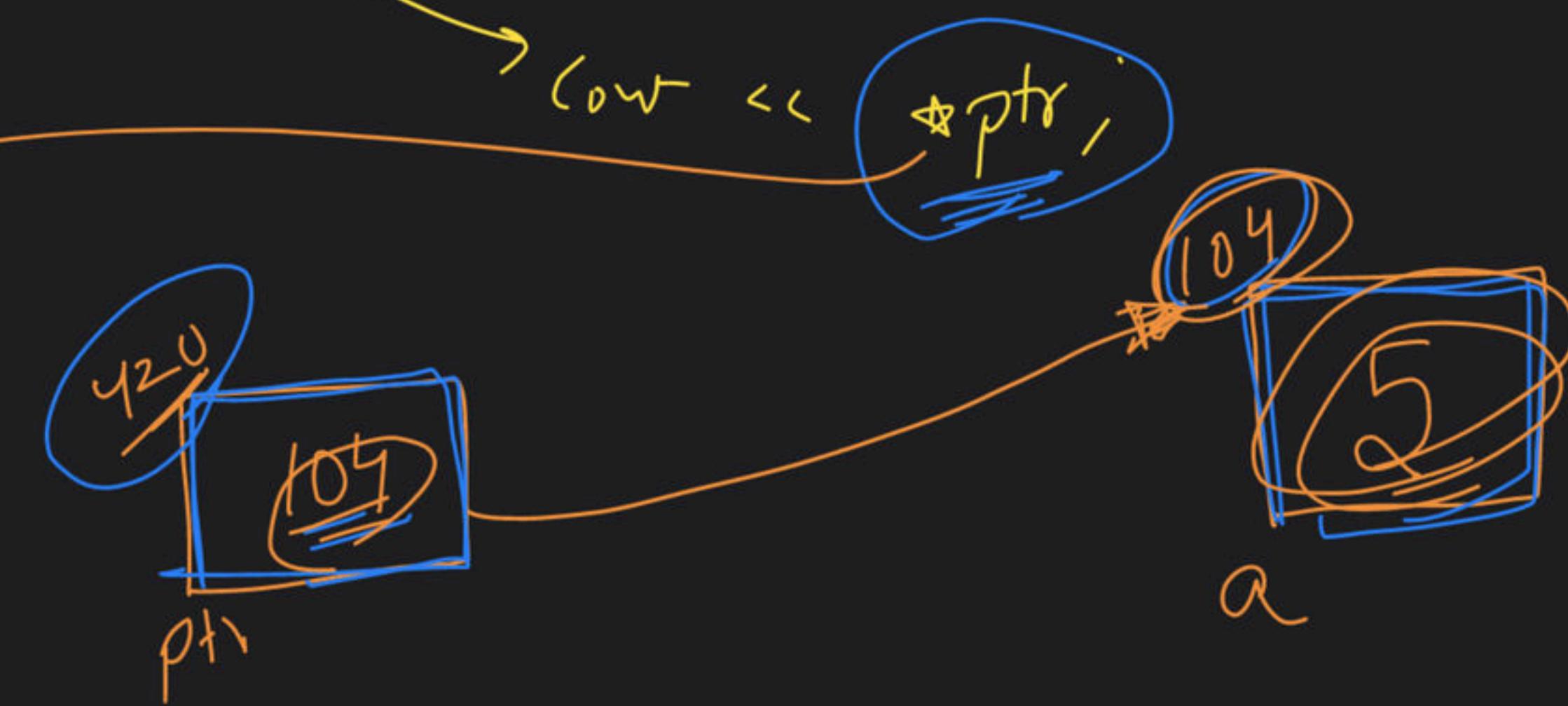
ptr is a pointer
& pointer to Babbar type
data

int a = 5

int *ptr = &a;



dereference
operator



value at

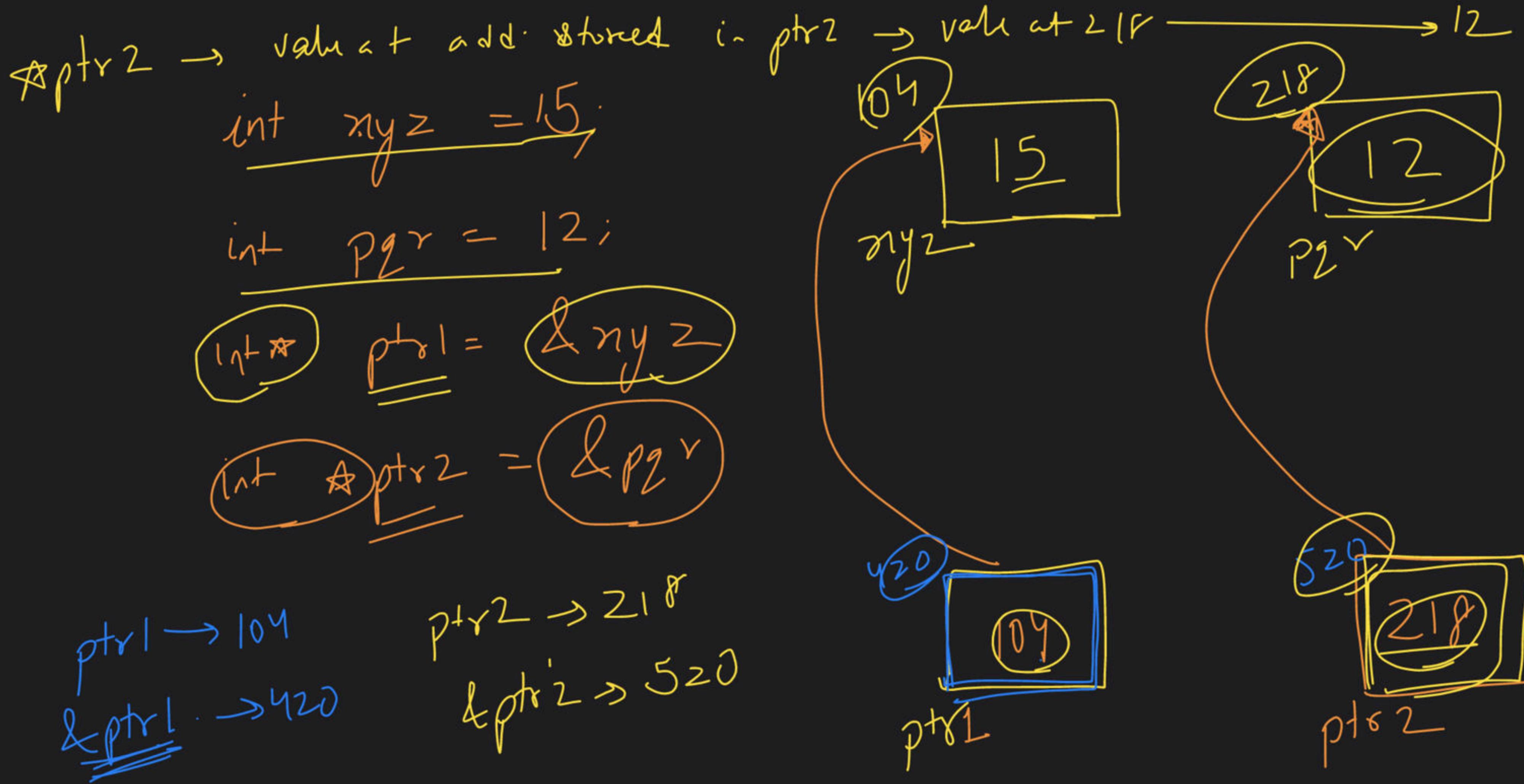


value at \underline{ptr}
or

value at address stored in ptr

value at $\underline{104}$





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

Obj (array)

Ref (reference variable)

diffuse

H/W

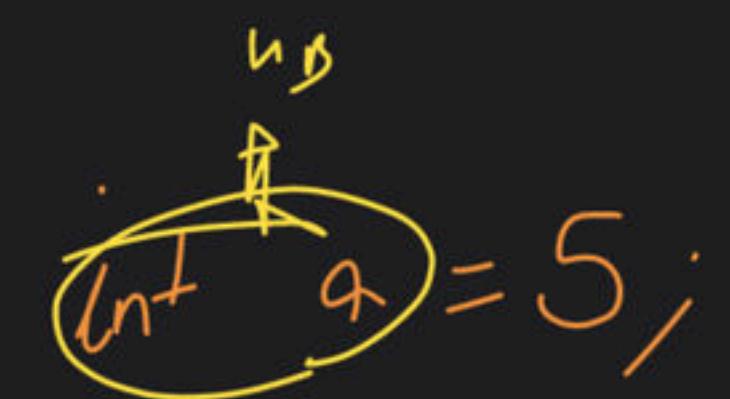
pointers

int a=5

int * b=&a

int a=5

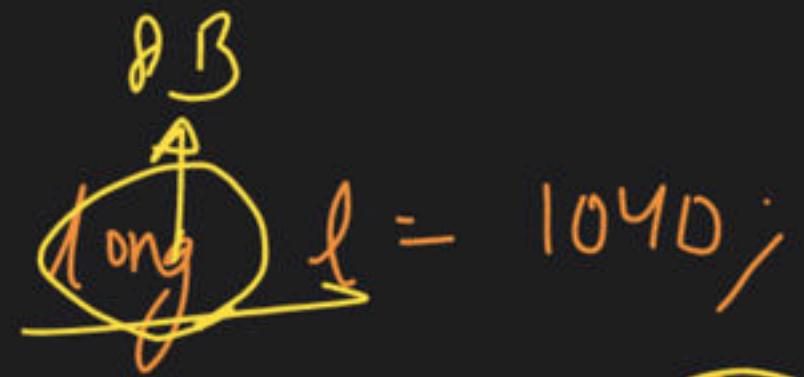
int &b=a



`int * pptr = &a`



`char * cptr = &ch`



`long * lptr = &l`

`sizeof(ptr)`

`sizeof(ptr)`

`sizeof(lptr)`

why size is coming 8?

`H/w`



declaration

int *ptr;

cout << ptr

104



garbage
value

int a;

a

int *ptr;

420



garbage
value



Syn

BAD
Practices

ptr

cout << *ptr

`int *ptr`

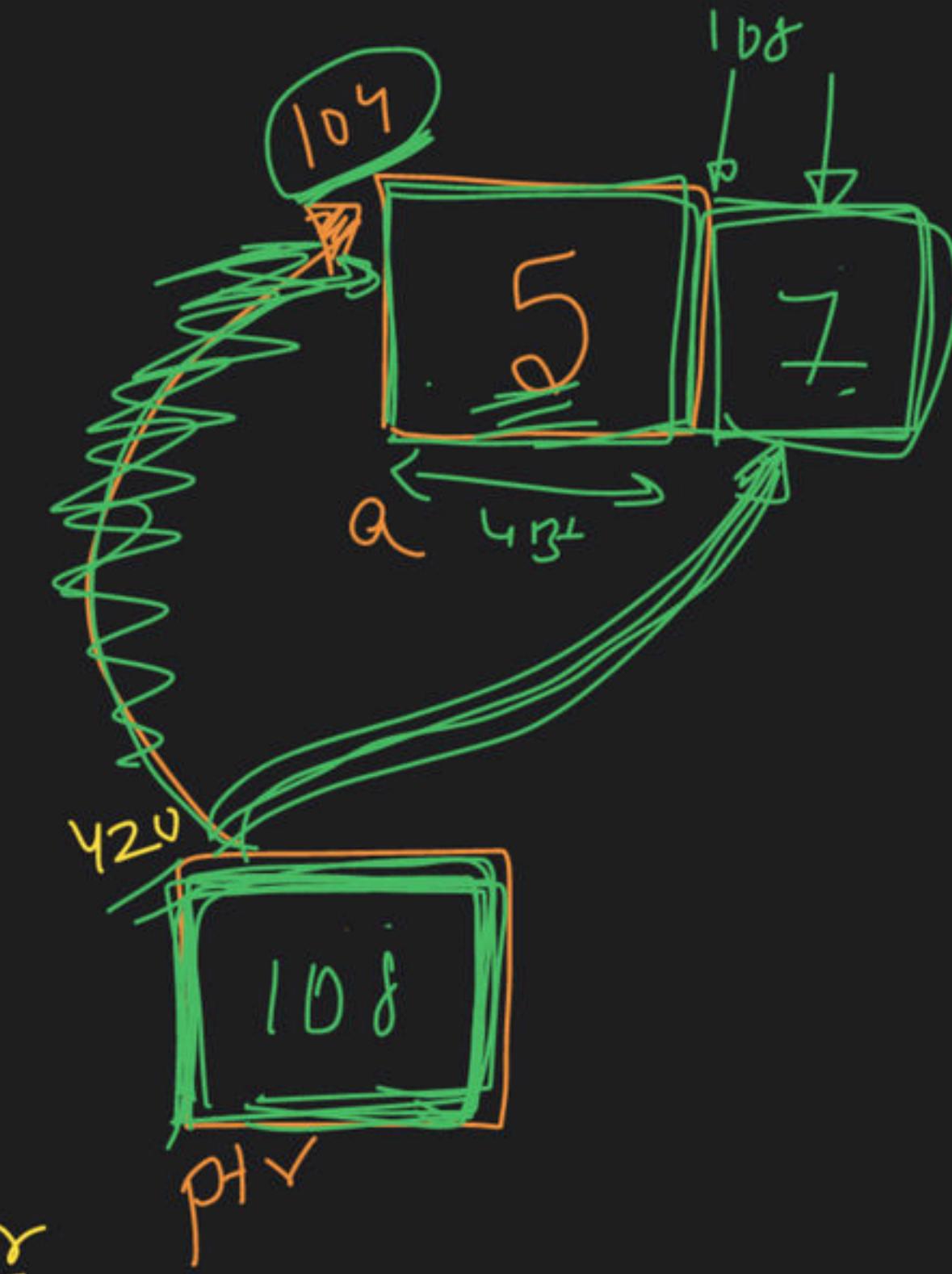
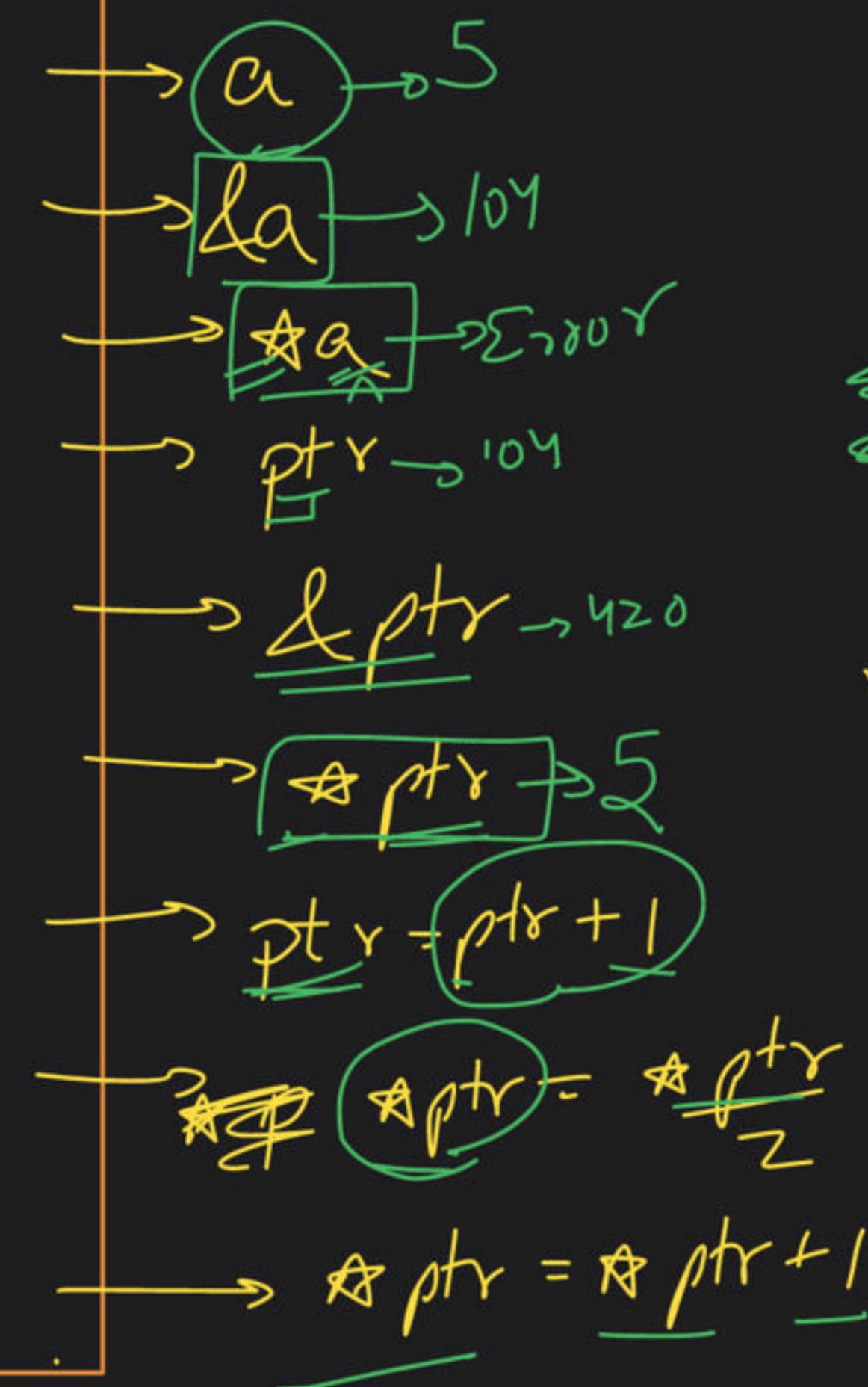
null pointer

`int *ptr = 0`

```

int a = 5;
int *ptr = &a;

```



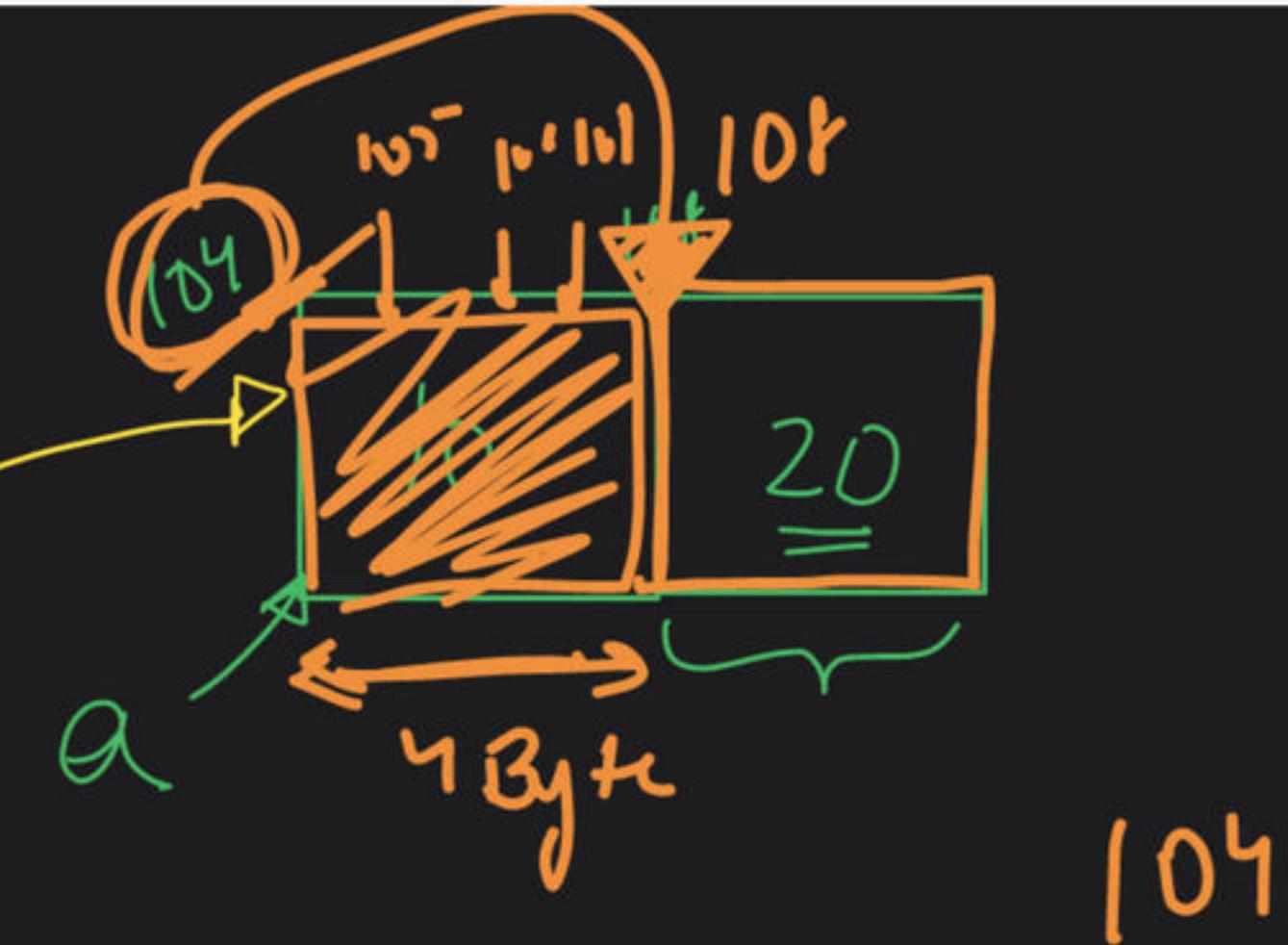
~~\star~~ ~~\star~~ ~~\star~~ \star ptr

$$\frac{104 + 1}{105} \neq 104$$

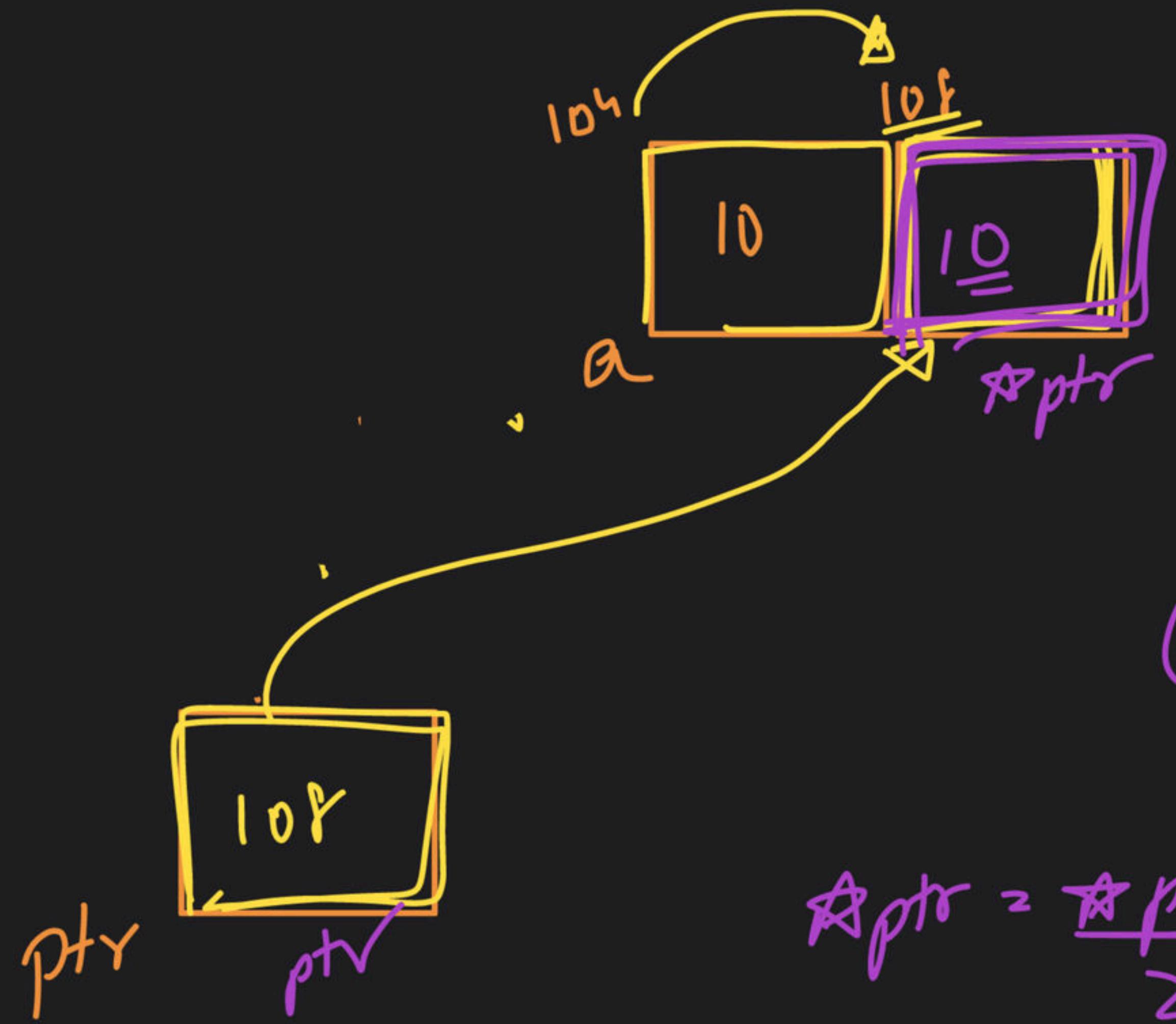
int a = 10

i⁺ *ptr = &a

ptr



$$\begin{aligned} \text{ptr} &= \underline{\text{ptr}} + 1 \\ \text{ptr} &= \underline{104} + \underline{1} \\ &= 104 + 1 \star 4 \\ &= 104 + 4 \\ &= 108 \end{aligned}$$



$$p^{tr} = p^{tr} + 1$$

$$\begin{aligned}
 \star p^{tr} &\rightarrow 20 \\
 \star p^{tr} &= \star p^{tr} + 1 \\
 &= 20 + 1 \\
 &= 21
 \end{aligned}$$

$$\begin{aligned}
 \star p^{tr} &= \frac{\star p^{tr}}{2} = \frac{21}{2} = 10
 \end{aligned}$$

`int a = 10;`

`int *p = &a`

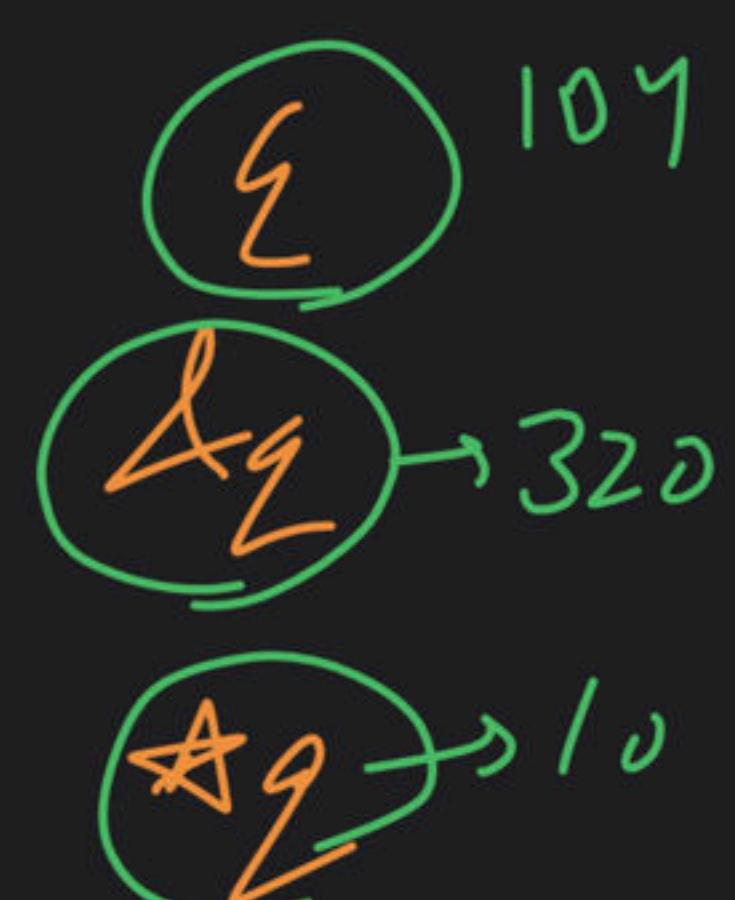
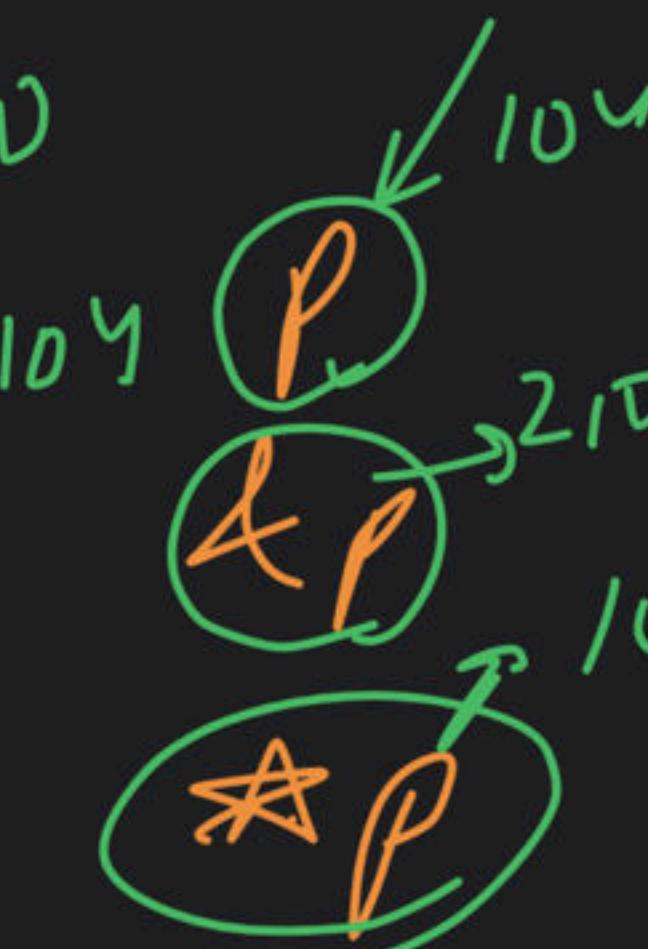
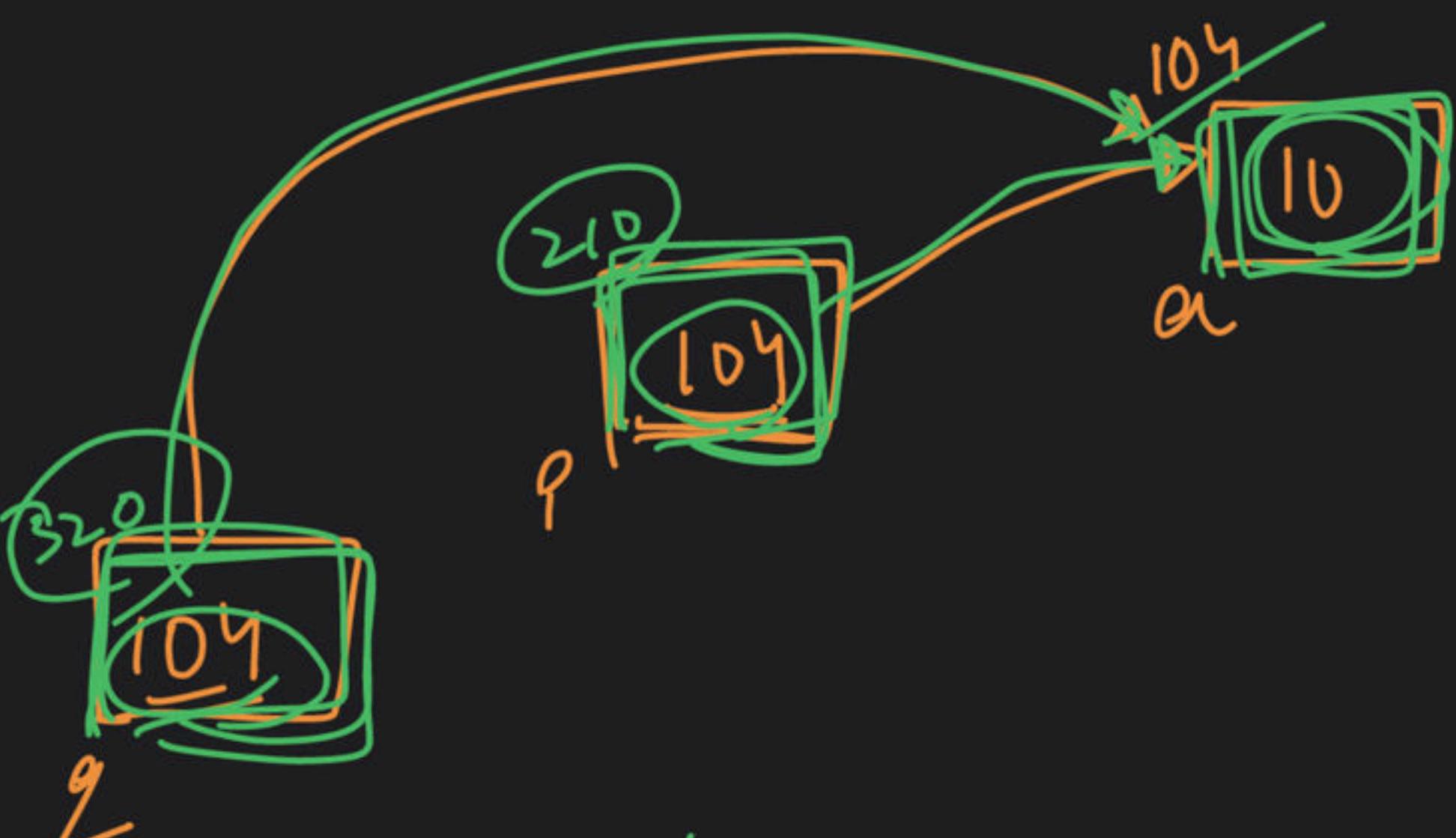
`int *q = p` → copy

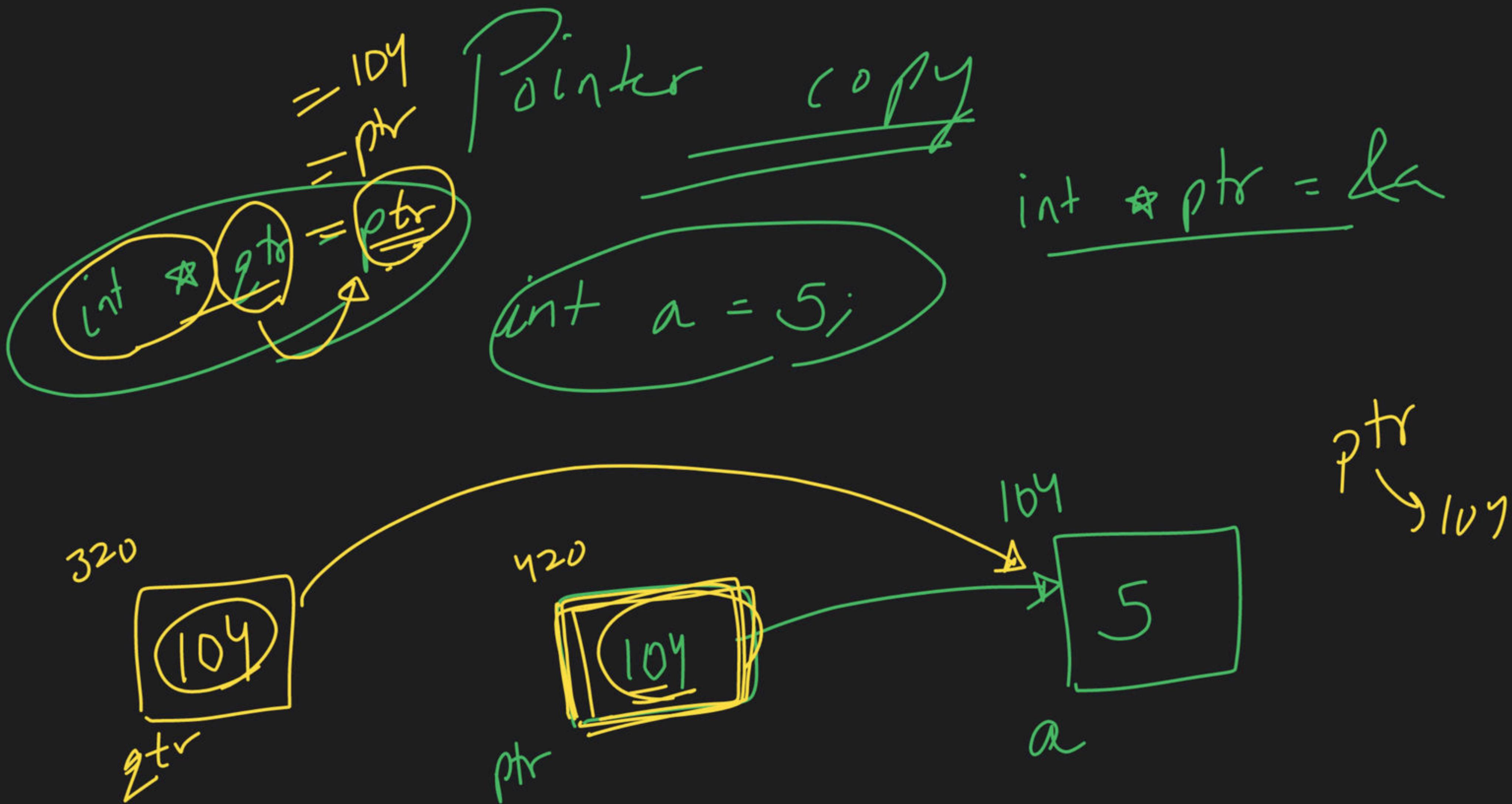
min

water

break

Error



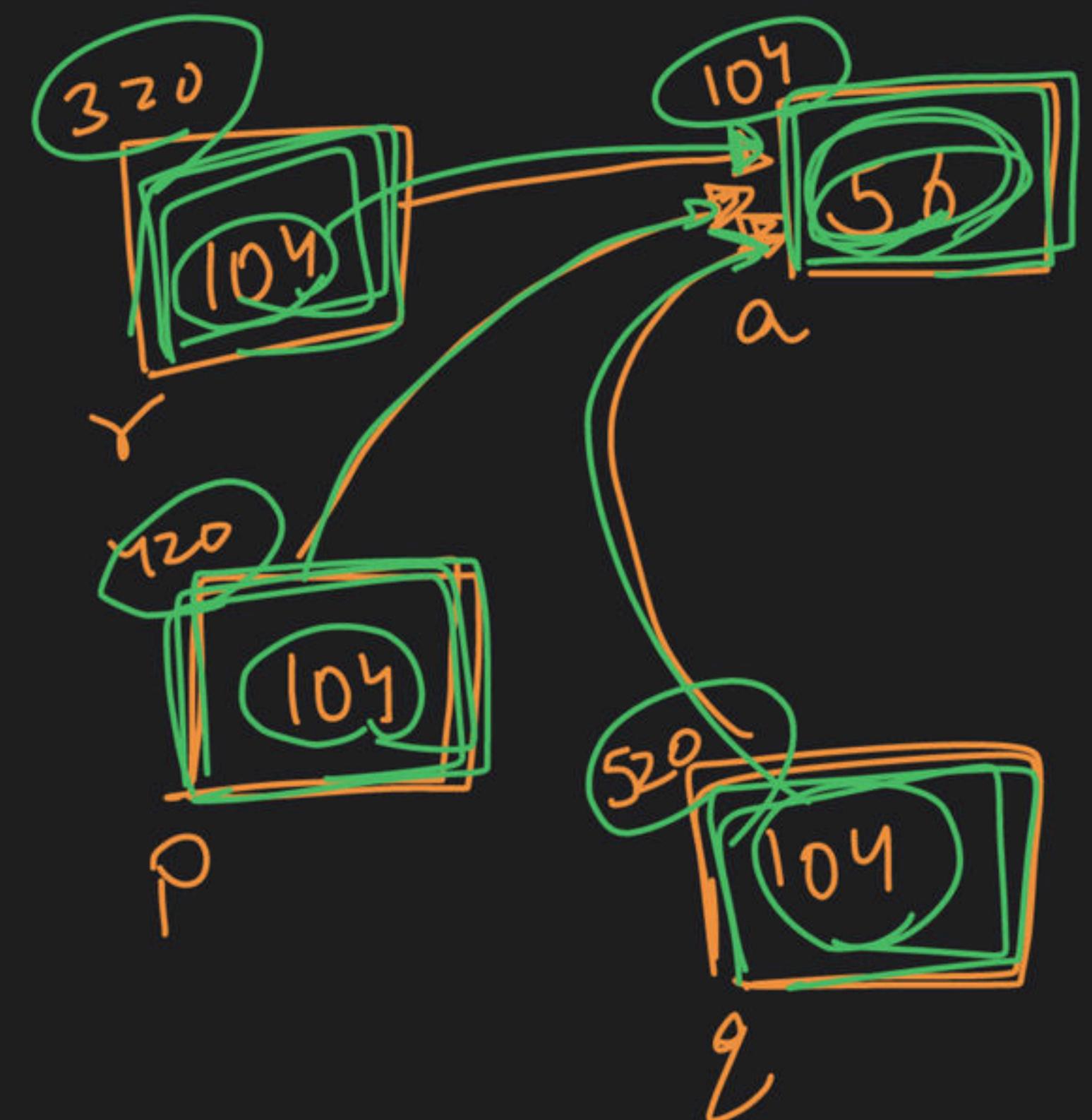


int a = 56;

int * p = ba;

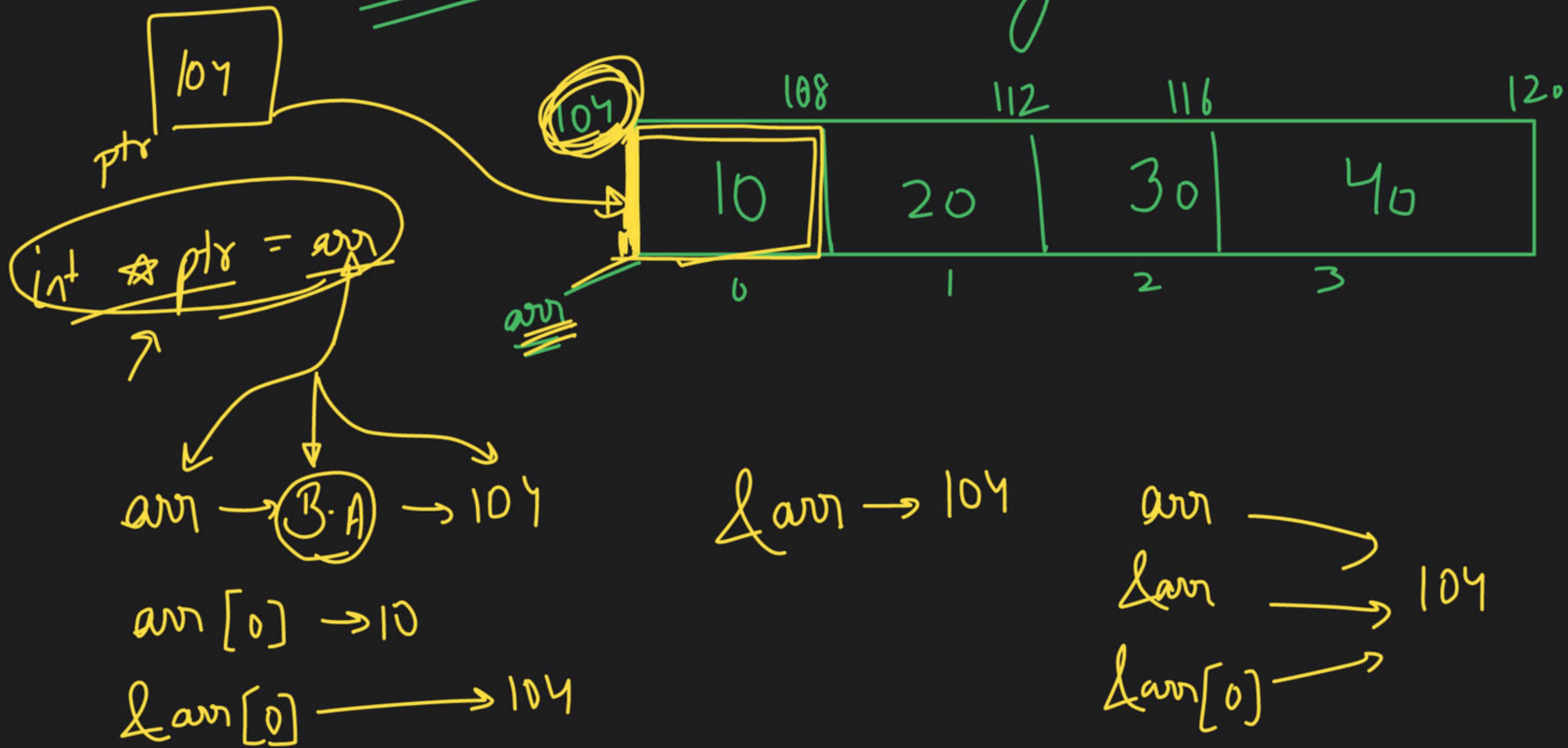
→ int * q = pi;

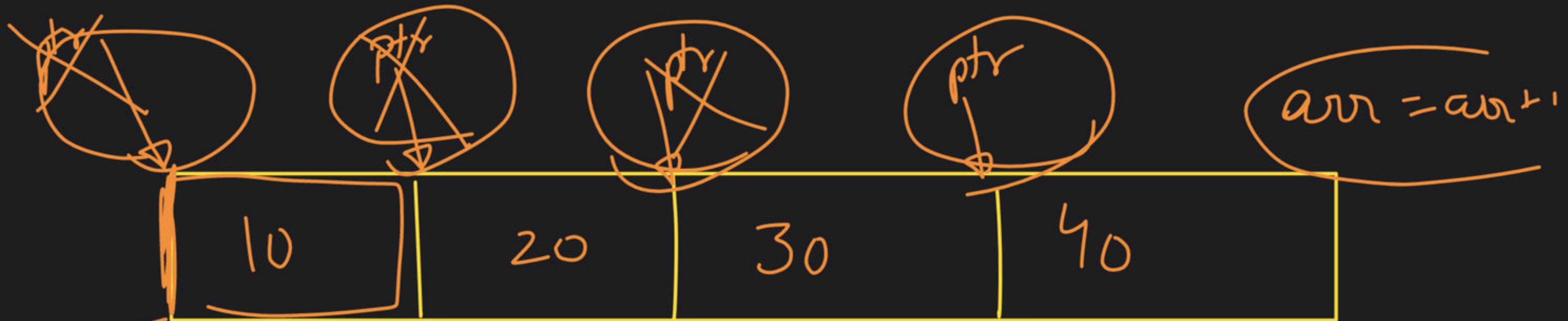
→ int * y = g;



$\underline{a} \rightarrow 56$
 $\underline{\&a} \rightarrow 104$
 $\underline{\&(\&a)} \rightarrow 61404$
 $\underline{p} \rightarrow 104$
 $\underline{\&p} \rightarrow 420$
 $\underline{\&(\&p)} \rightarrow 56$
 $\underline{q} \rightarrow 104$
 $\underline{\&q} \rightarrow 520$
 $\underline{\&(\&q)} \rightarrow 56$
 $\underline{y} \rightarrow 104$
 $\underline{\&y} \rightarrow 320$
 $\underline{\&(\&y)} \rightarrow 56$

Pointers with Arrays





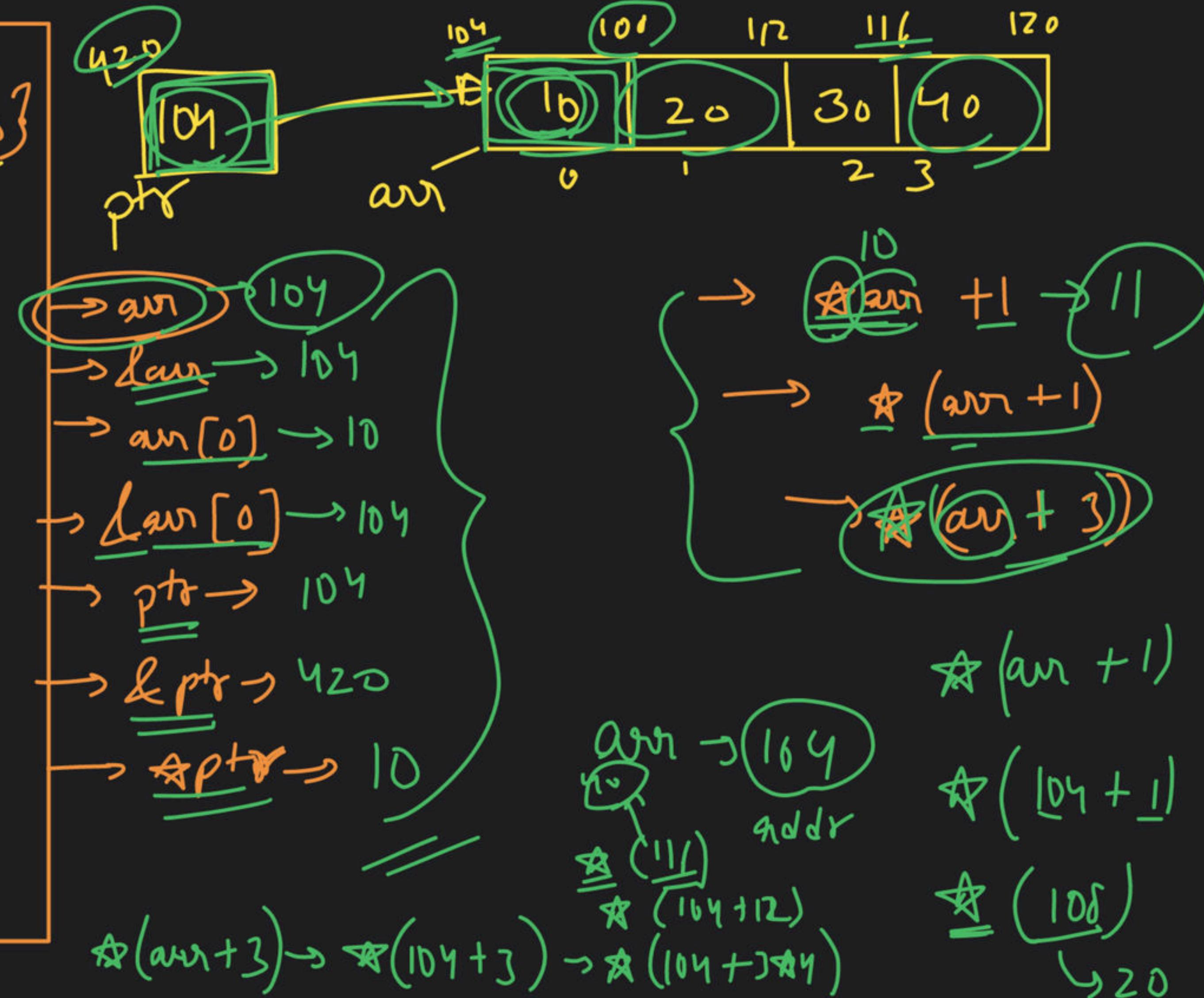
$int * ptr = arr$

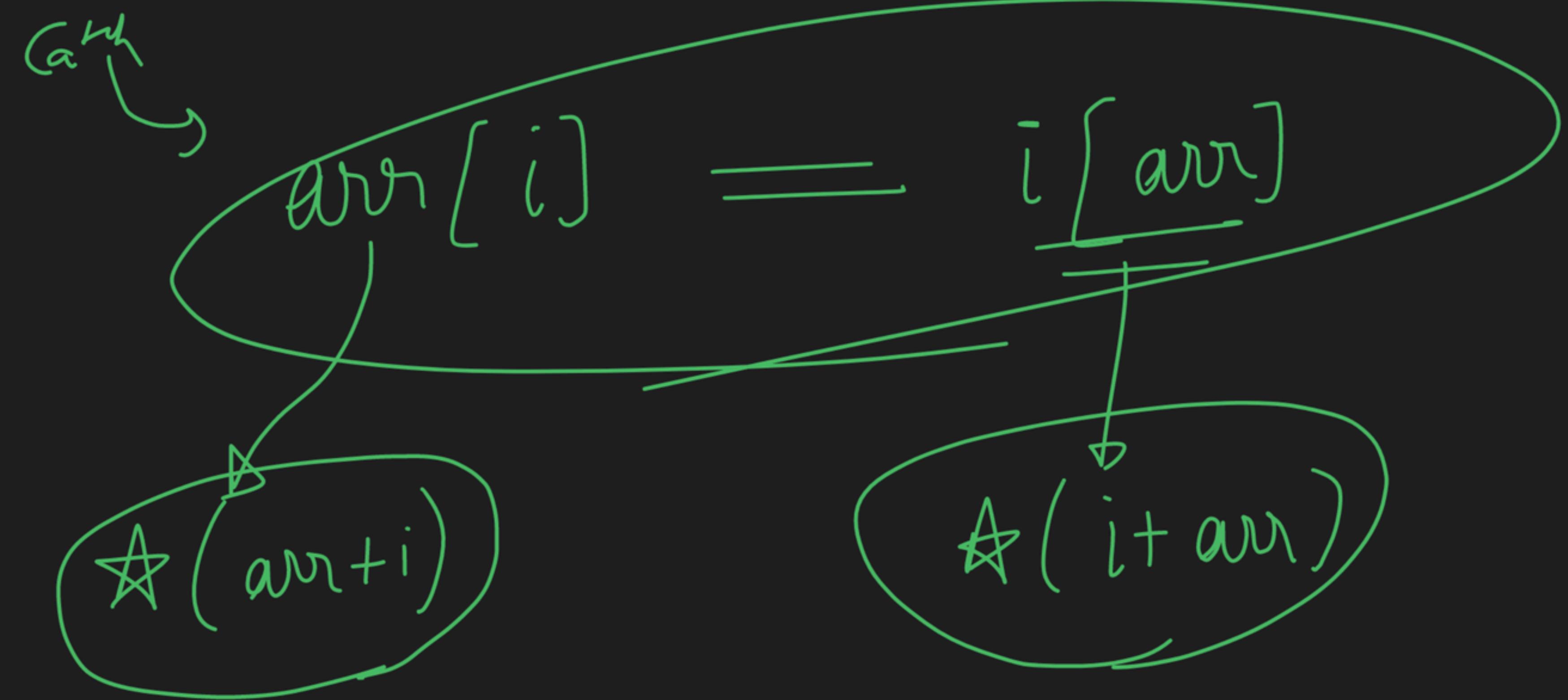
$ptr = ptr + 1$

$cout << (arr + 1)$

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

int *ptr = arr;





int arr[] → {100, 200, 300, 400, 500}

int *P = arr;

int *q = arr + 1;

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

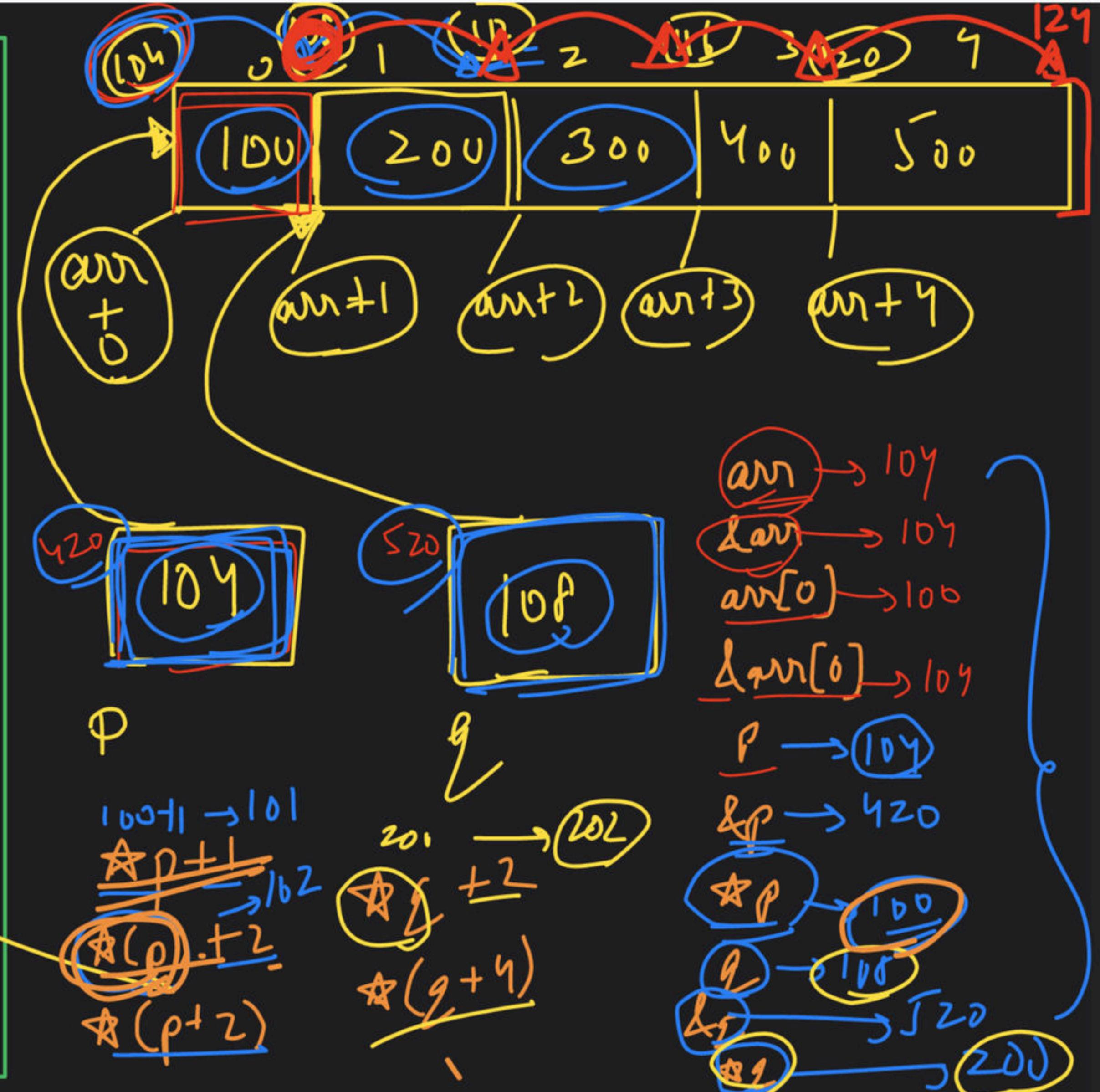
$\star(p+2) \rightarrow \star(100+16)$

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

$\star(124)$

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

$\star(112) \rightarrow 300$

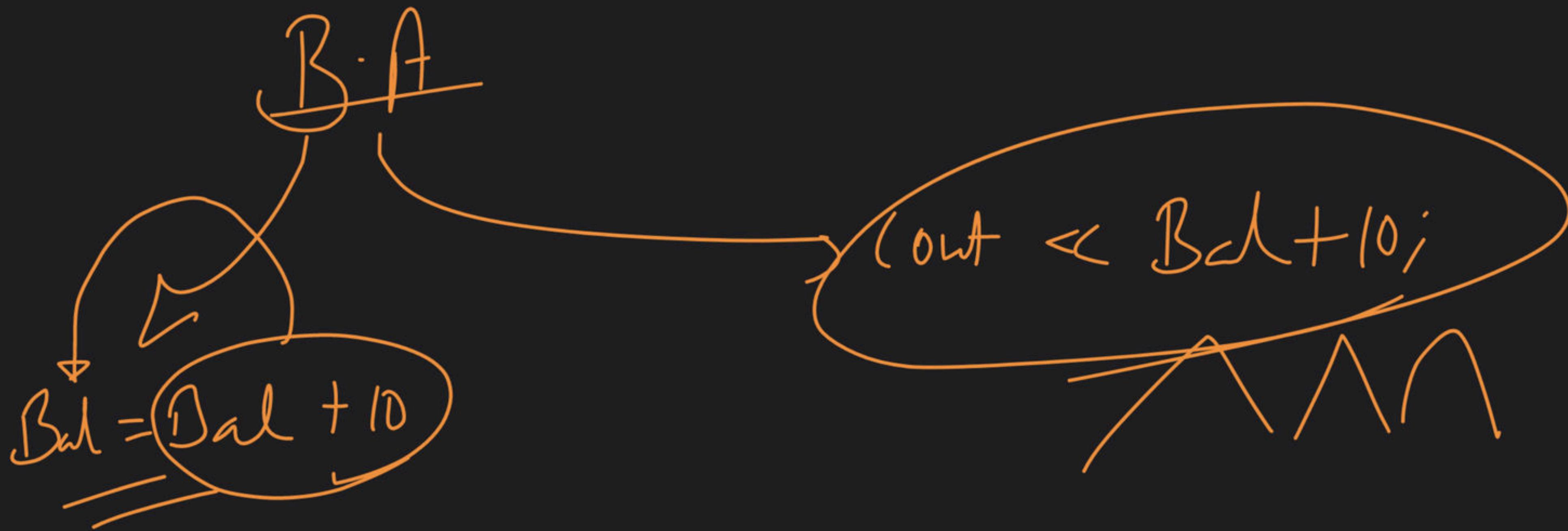


~~Char Arrays :-~~

2 min

Break

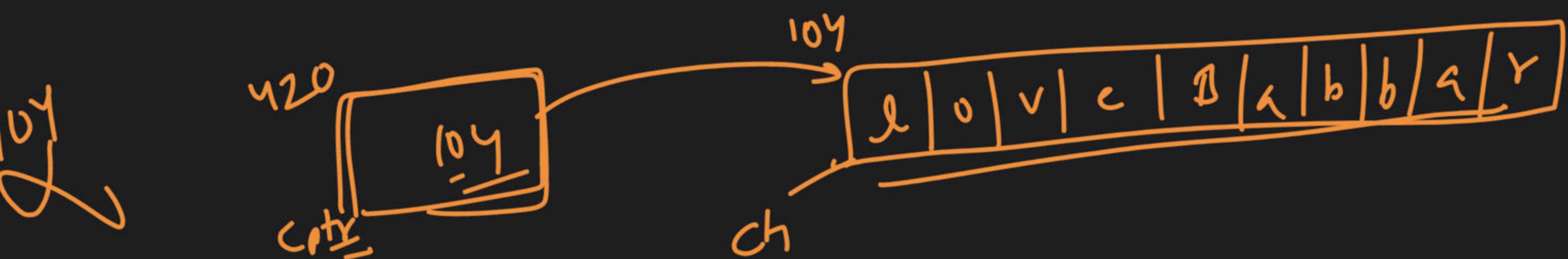
$\text{cout} \ll (\text{arr} + 1)$



char ch[100] = "loveBabbar";

char * cptr = ch;

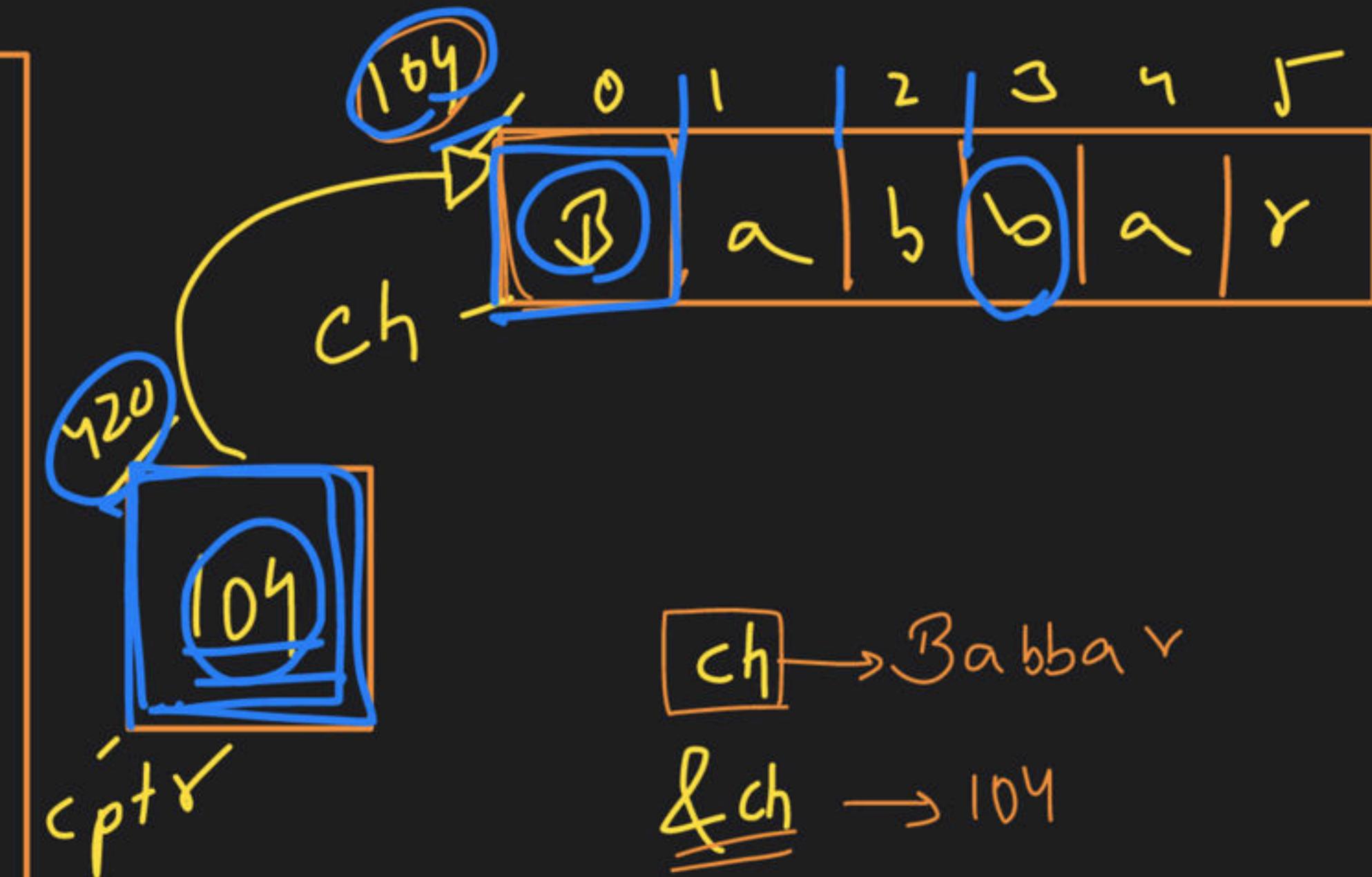
cout << cptr;



char ch[20] = "Babbar";

char * cptr = ch;

$\star(cptr + 3)$
 $\triangleright (104 + 3 * 1)$
 $\star (104 + 3)$
 $\triangleright (107)$



ch → Babbar

&ch → 104

ch[0] → B

&cptr → 104

check
Kaw

check
willal

$\star cptr$ → B
 $\star (cptr + 3)$ → b
 $\star cptr$ → 104

char

(ch) =

x

a

'

a

v

v

v

v

v

v

char *

(p1)

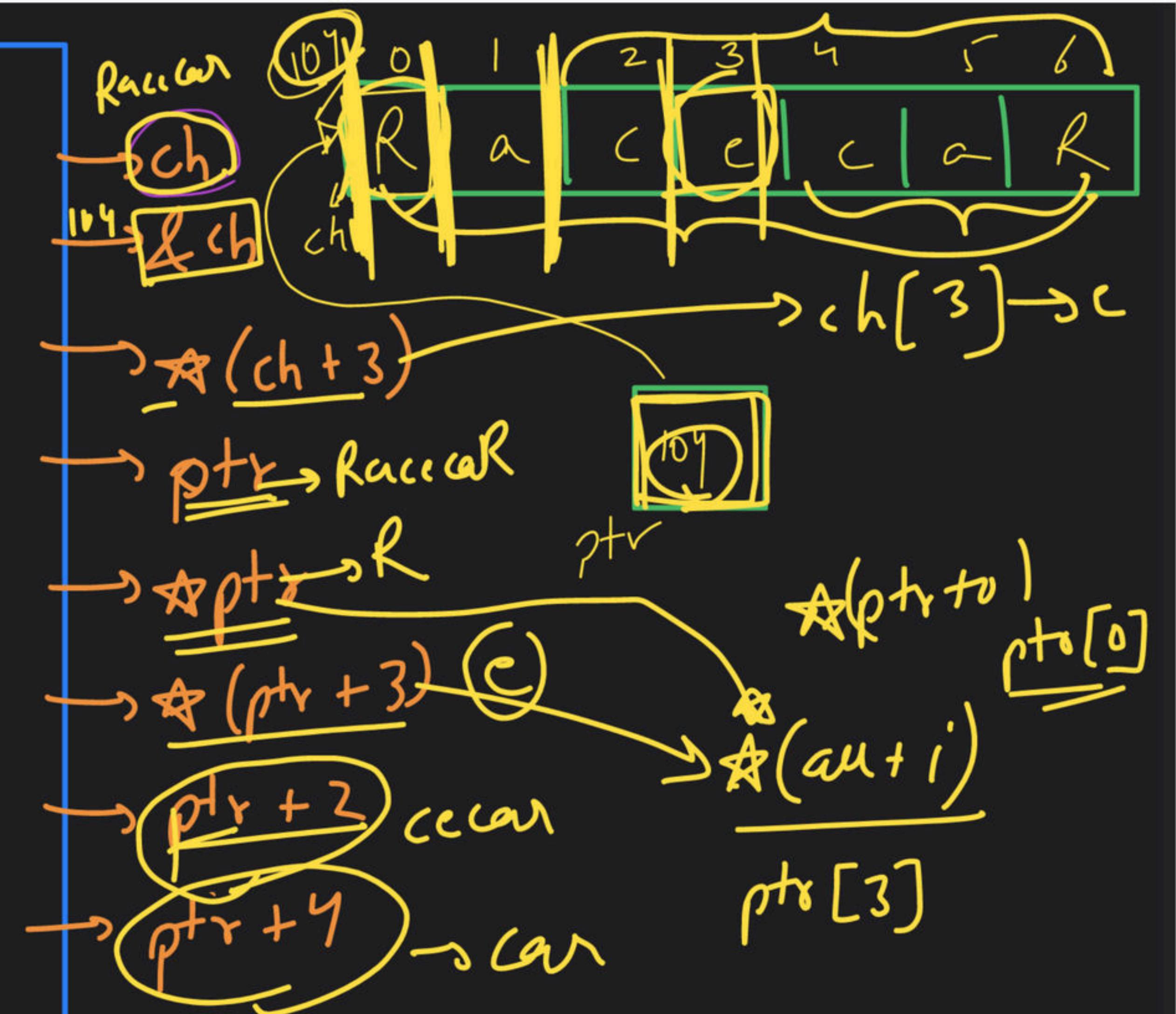
= &ch

cout << (p1)

&ch[0]

`char ch[10] = "Racecar"`

`char *ptr = &ch[0]`



$\&ch[0]$

Char ch[10] = "Balbar"

why

radio

char *ptr = "Balfur"

H/w

ptr













































