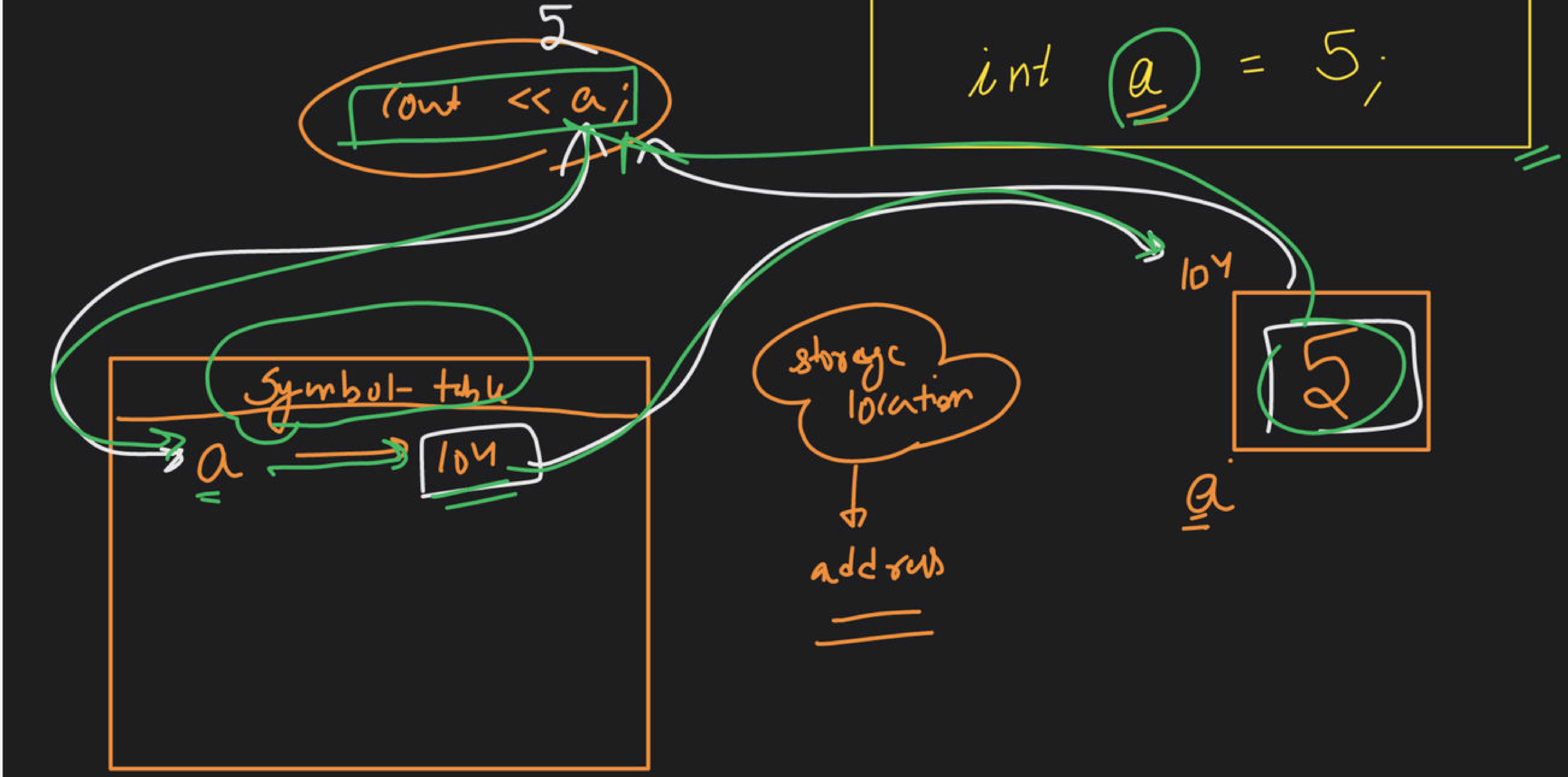
Pointers - Level 1

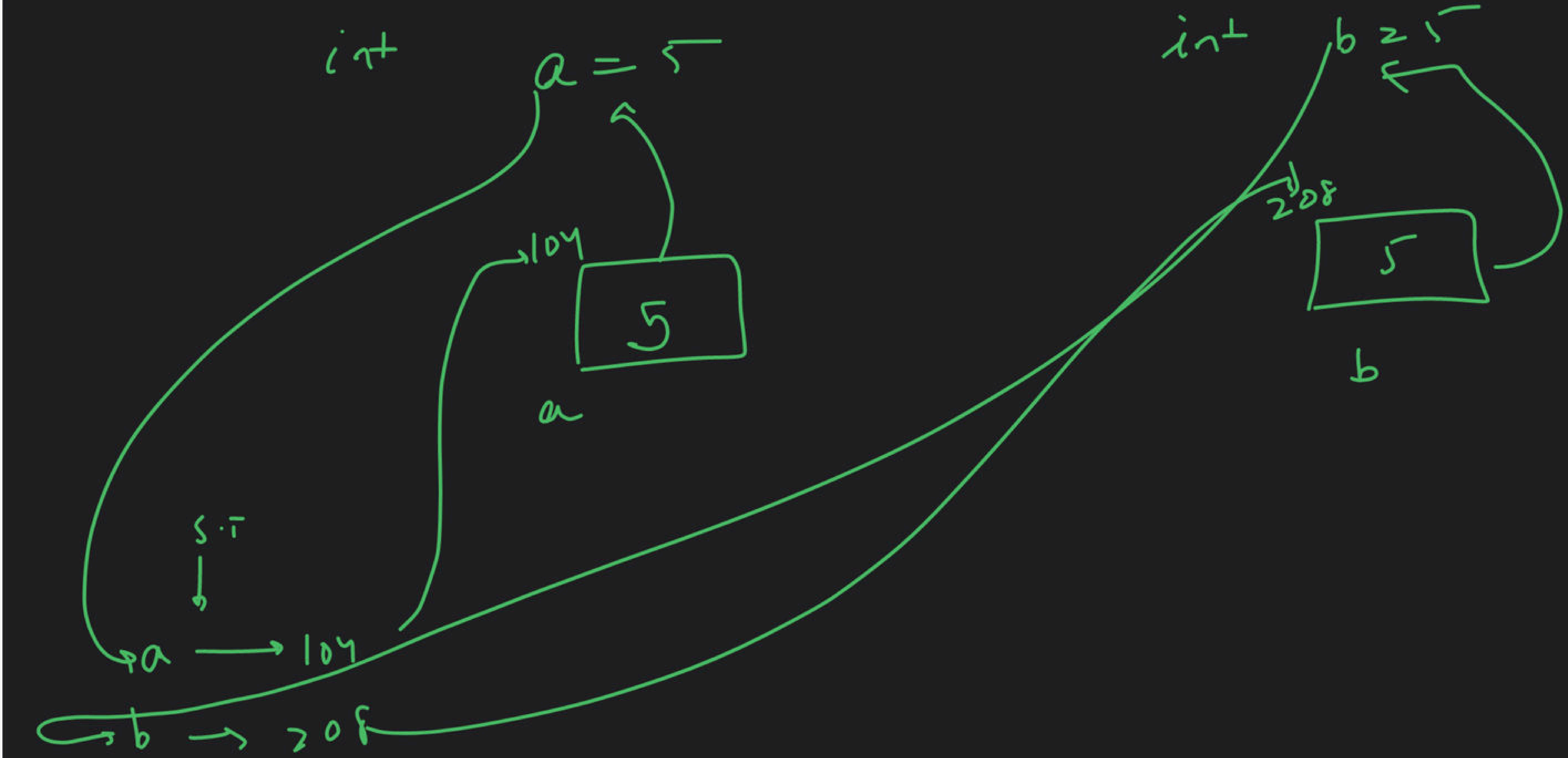
Special class

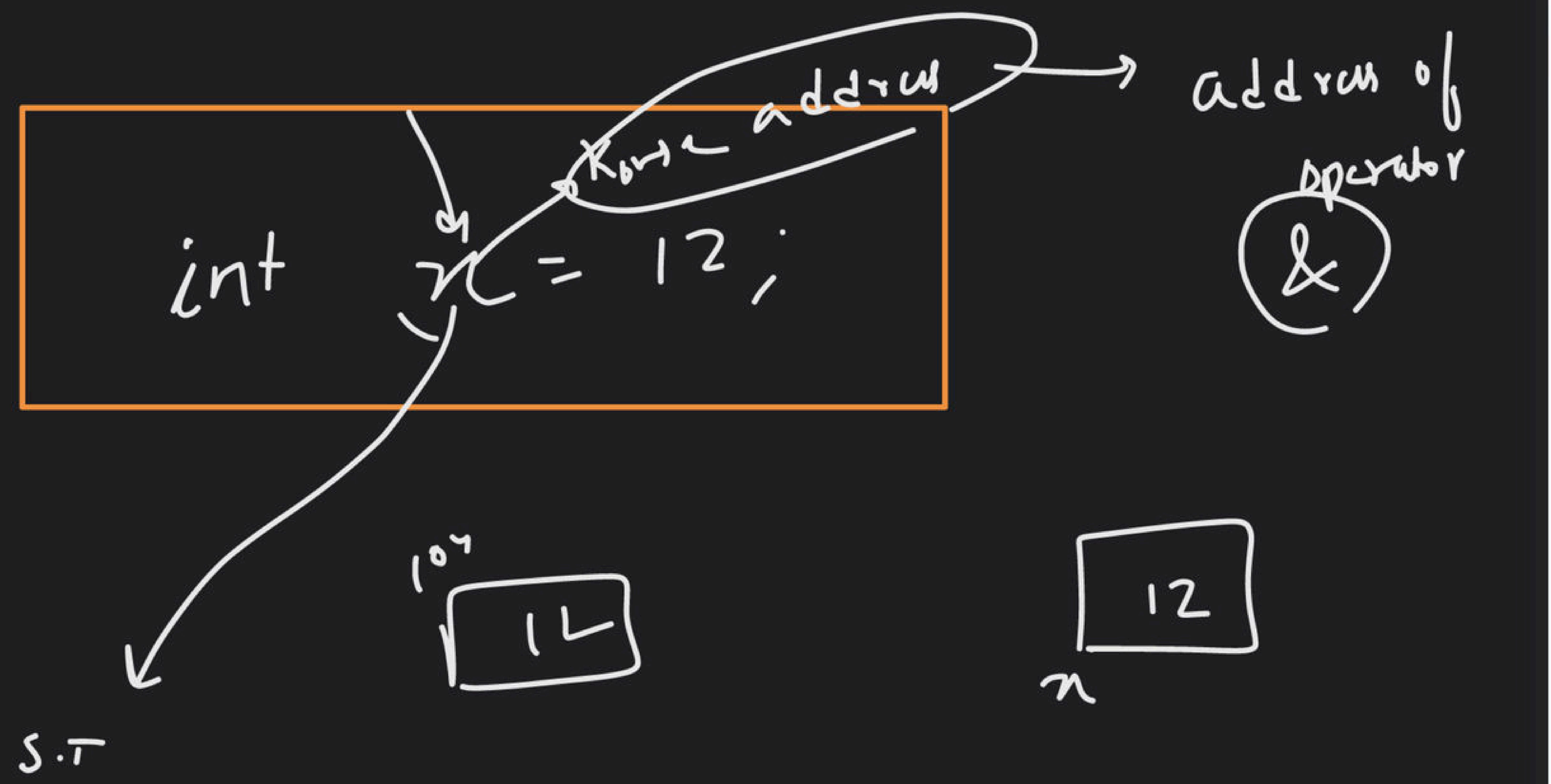
Love Babbar • Mar 3, 2023

① Pointers:-



5
cout << a;
int a = 5;
Symbol table
a
104
storage location
address
104
5
a


int a = 5
int b = 5
104
5
a
208
5
b
S.T
a → 104
b → 208



$\boxed{x} \rightarrow \boxed{104} \rightarrow$ find / verify → ji hae

① int a = 5;

② & → address of operator

Pointers:-

variable name

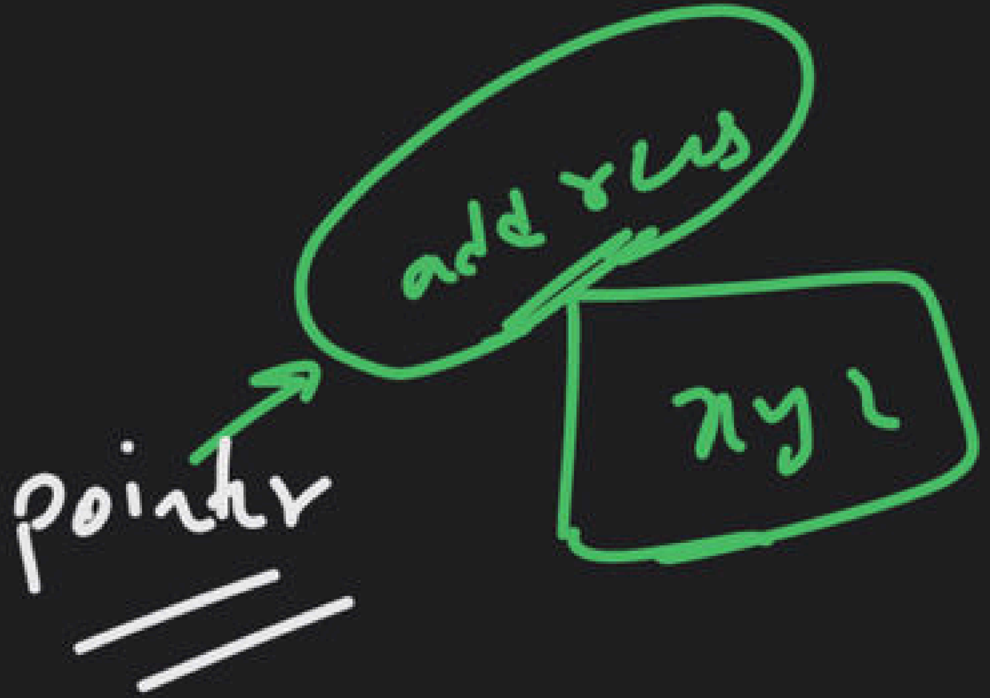
int* ptr = ?

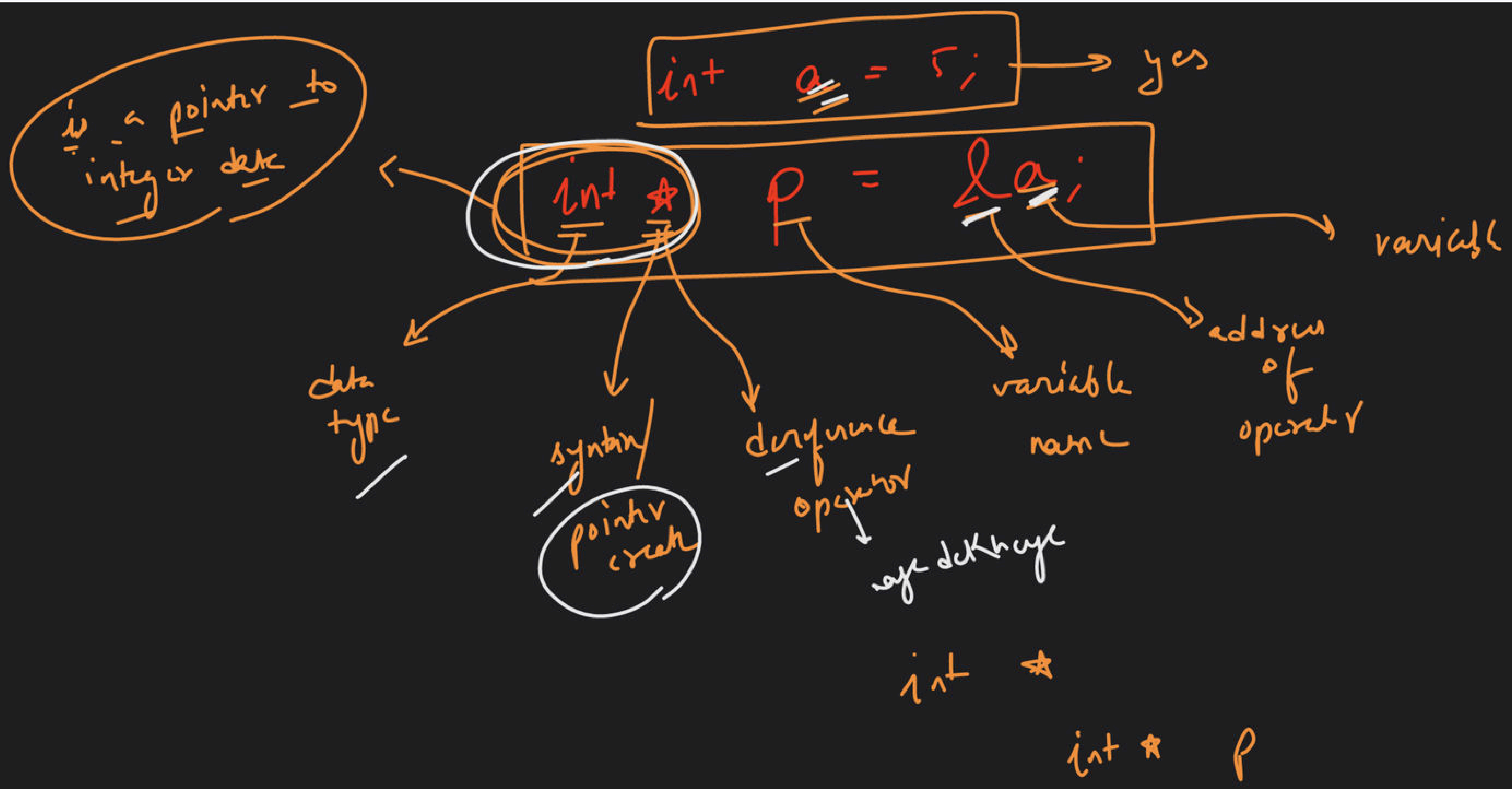
?

ptr is a pointer to integer data

int a = 5

a K अंदर integer type Ka data u can store




int a = 5;
yes
is a pointer to integer data
int * p = &a;
data type
syntax
pointer create
dereference operator
naye dekhenge
variable name
address of operator
variable
int *
int * p

```
char ch = 'a';
char * p = &ch;
```

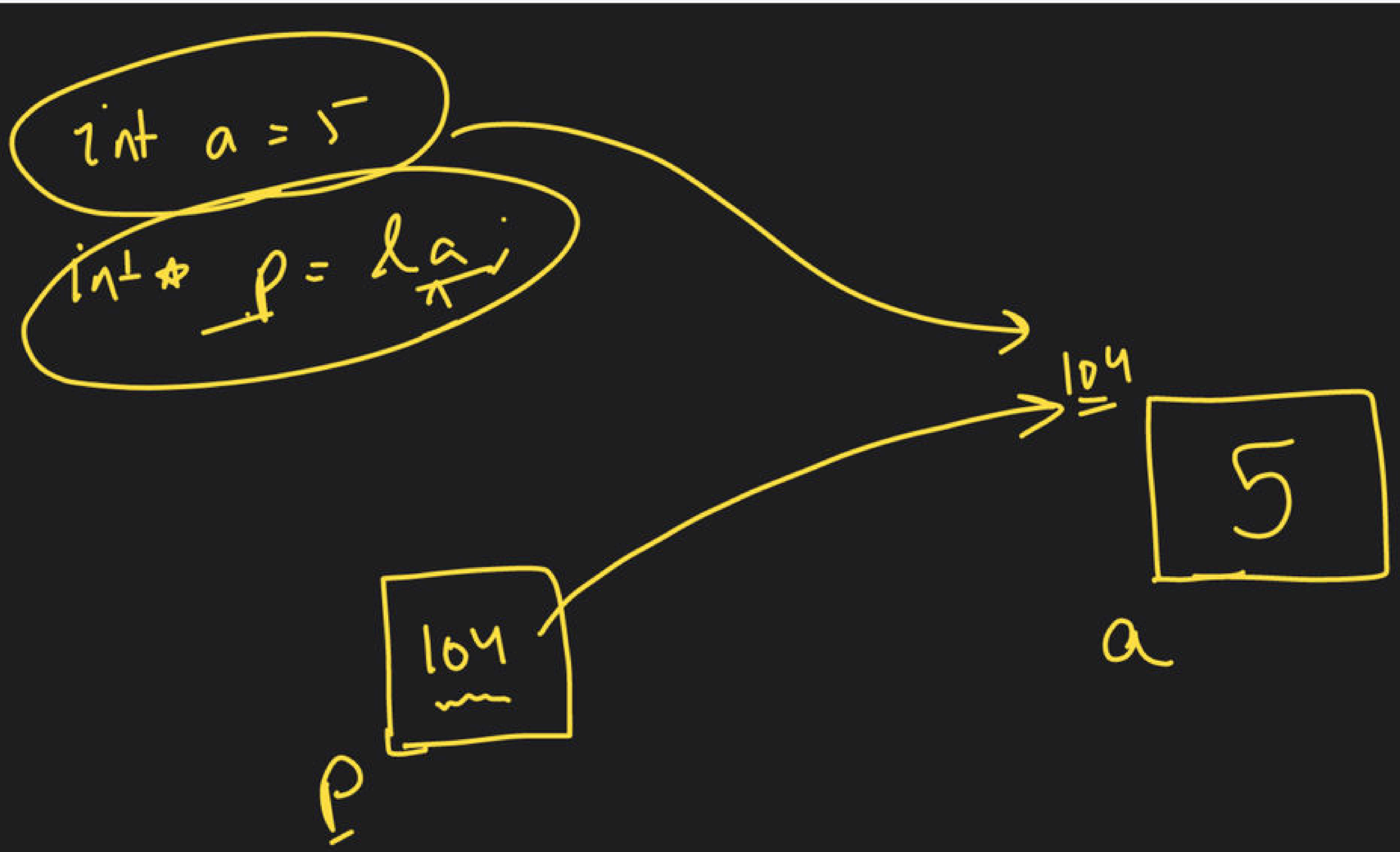
is a pointer to char data

```
bool * p = &a;
```

p is a pointer to bool ~~[illegible]~~ data long * p =

short * p

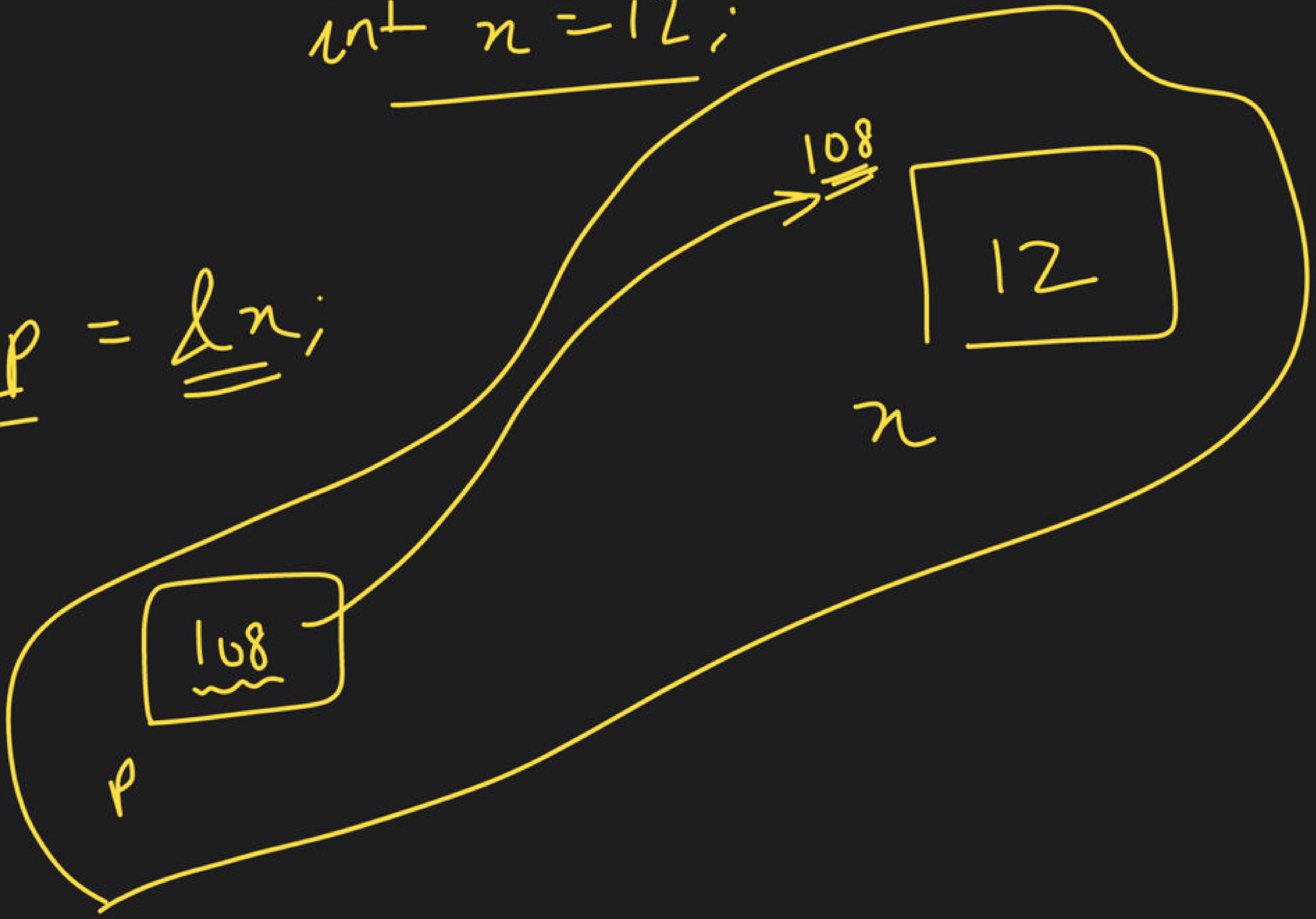
double * p

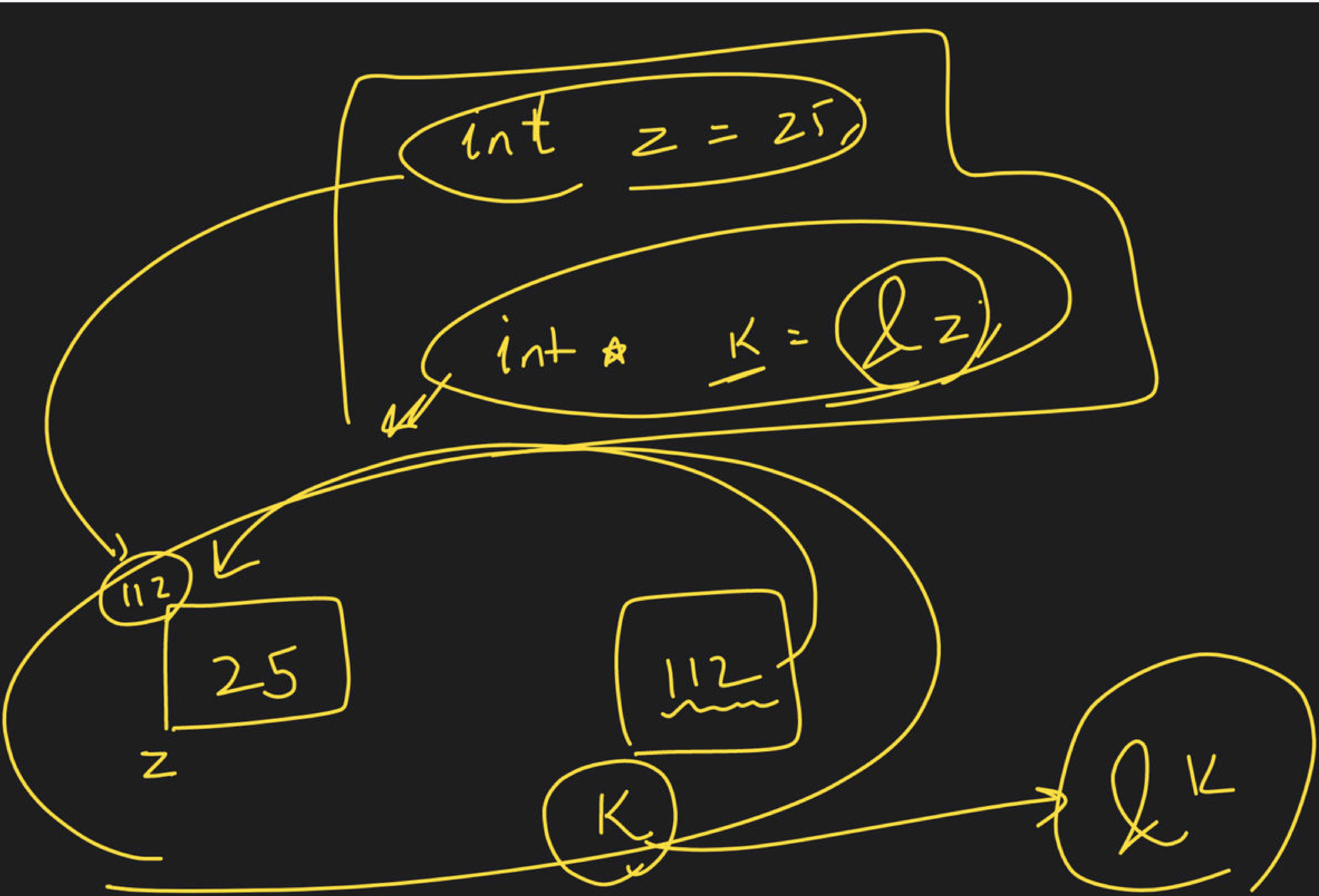


```
int a = 5
int* p = &a;
```

```
int x = 12;
```

```
int *p = &x;
```





```
int z = 25;
int * k = &z;
```

What?

pointer in C++ is a variable that storys address of another variable ptr → address

ptr → value

```
int a = 5;

int* ptr = &a;
```

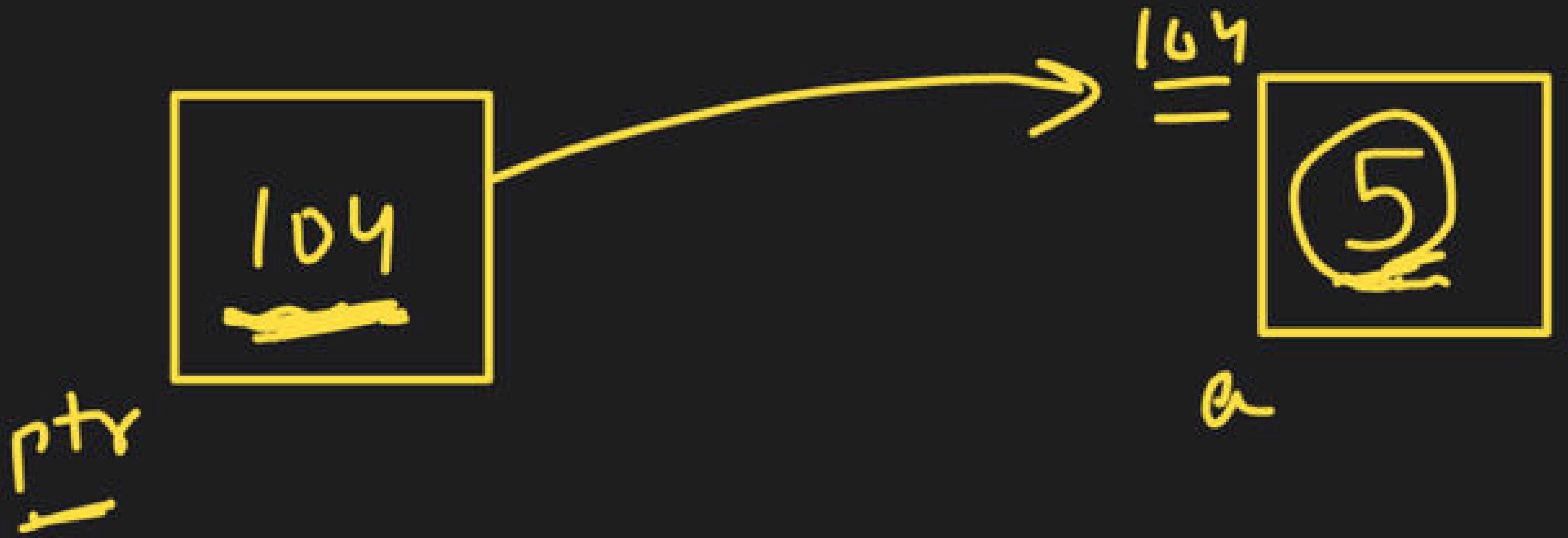
ptr → value acess

104

```
cout << ptr << endl;

cout << *ptr << endl;
```

5



*ptr → Value at location stored in ptr

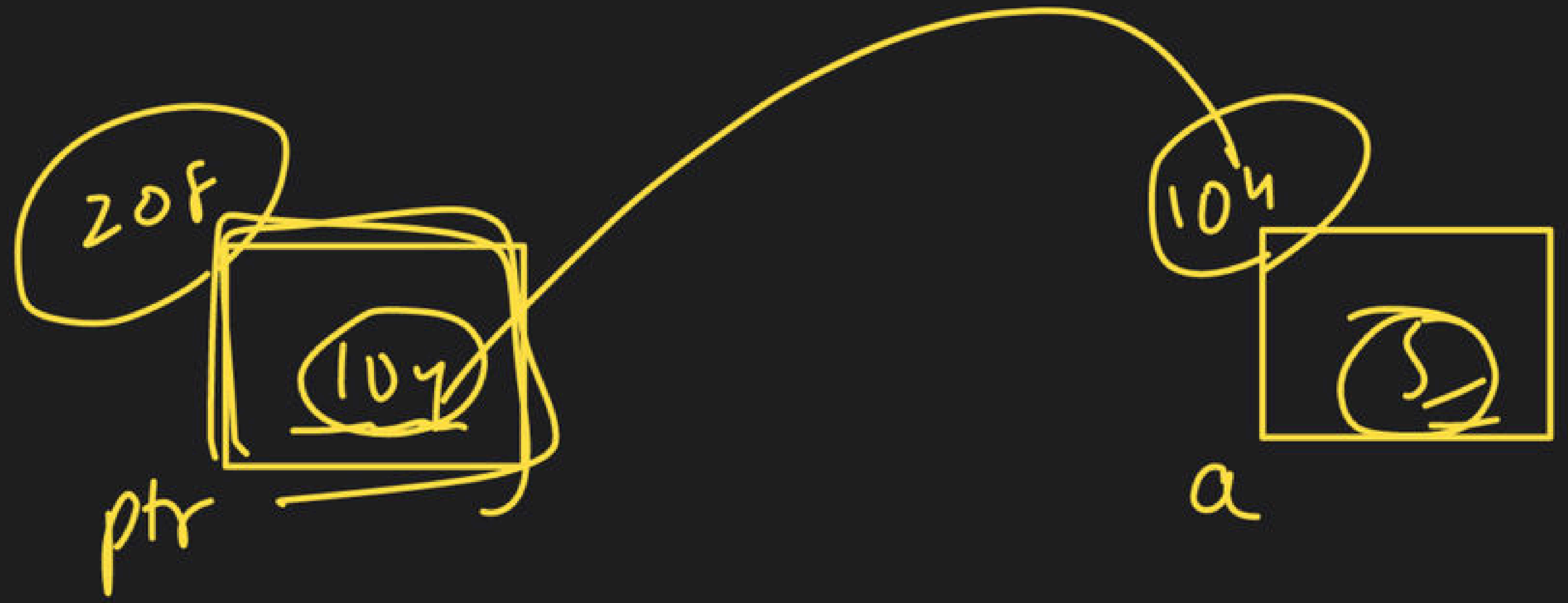
Creation

→ int* p = &a;

access

→ address → p → addr

value → *p → val



ptr → 104

&ptr = 208

*ptr = value at ptr

= value at 104

= 5

a → 5

&a → 104

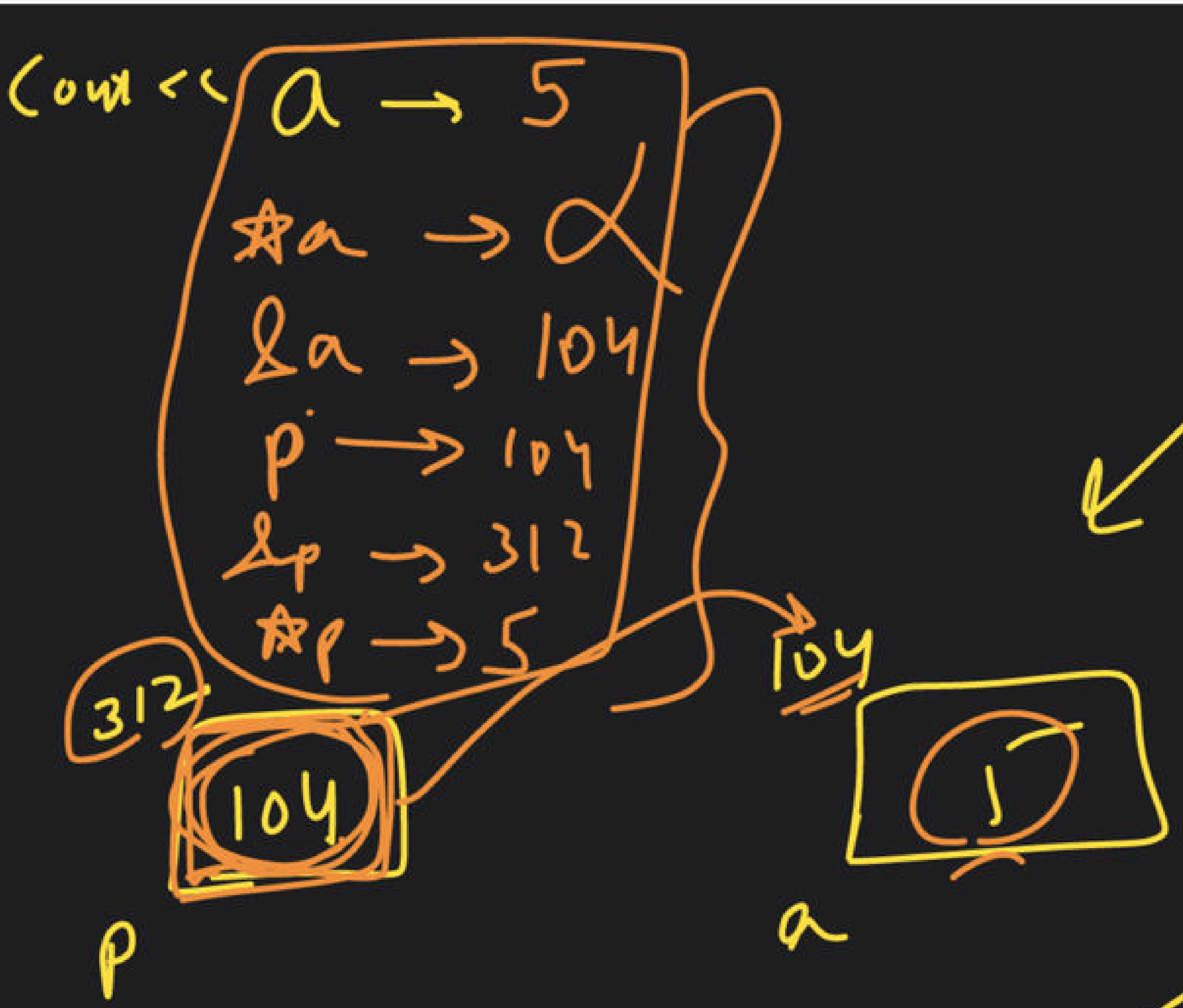
✓ *ptr → value stored at ~~ptr~~ location in ptr

✓ &ptr → address of ptr

✓ &a → address of a

✓ a → value of a

✓ ptr → value of ptr



cout << a → 5

*a → α

&a → 104

p → 104

&p → 312

*p → 5

int a = 5;

initialisation

a variable Ko 5 se initialise kie h

int* p = &a

p is a pointer to integer data & it is pointing to a wala variable int a = 5 ; int *p = &a; ——→ size

char ch = 'b'; char* c = &ch; ——→ size

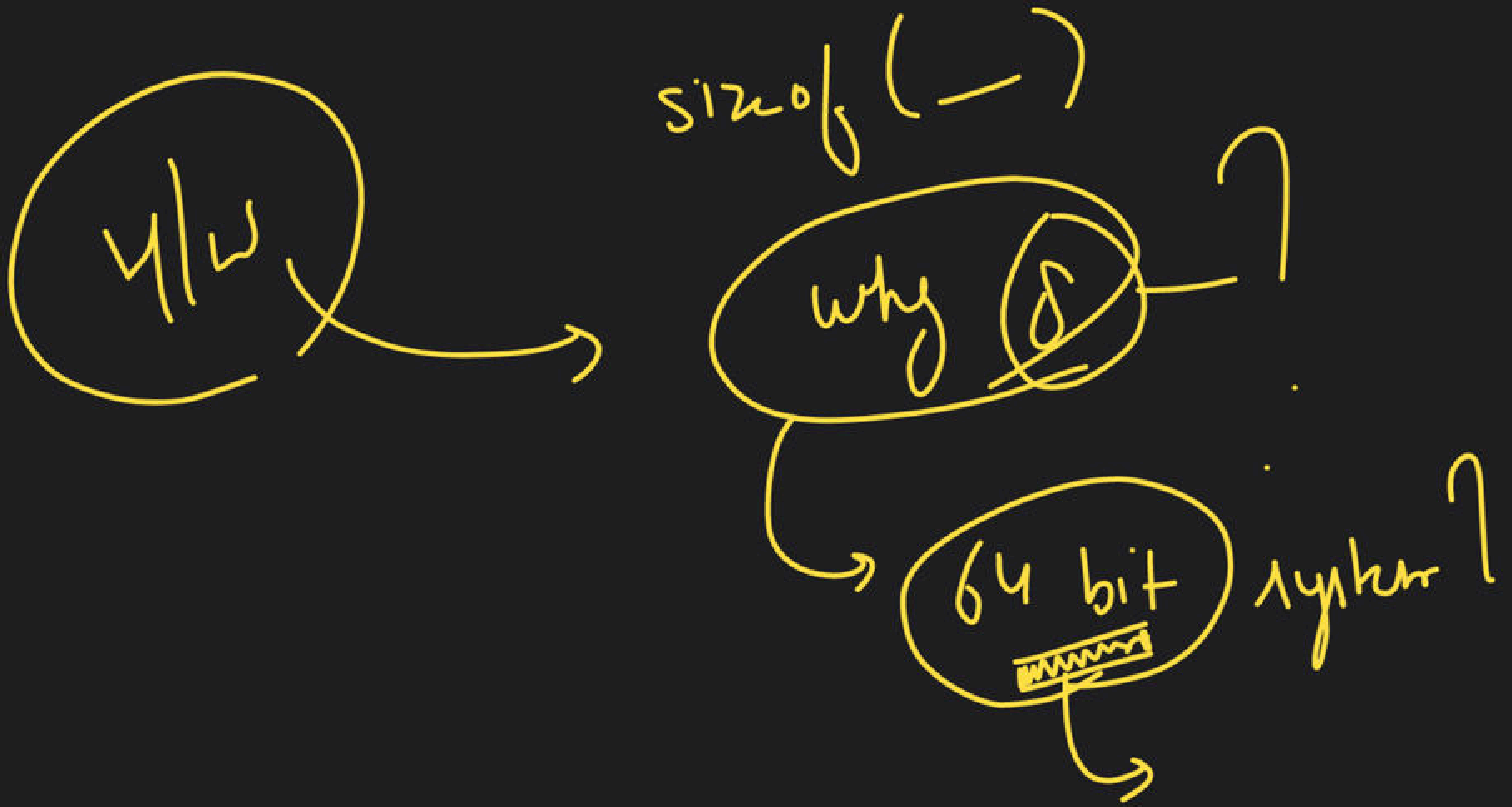
double d = 1.03; double * dtr = &d ——→ size

catch
↓
pointer
↓
size
↓
8

p is a pointer to integer data → 4

c is a pointer to char data → size → ? 1

dtr is a pointer to double data → 8


H/W
sizeof(_)
why 8 ?
64 bit
system?

pointer

- create
- access
- size

why ?

OS

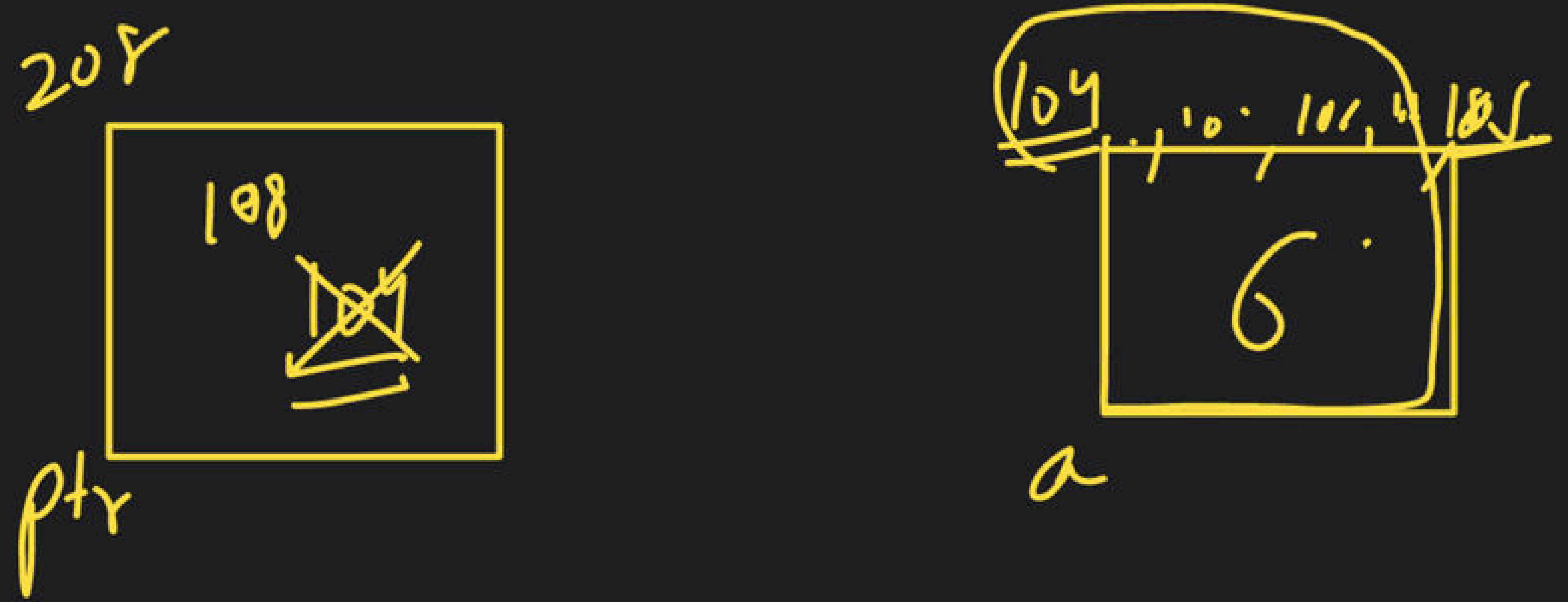
↳ Dynamic memory allocation

memory mgmt

H/W

Pointer to function

why ?



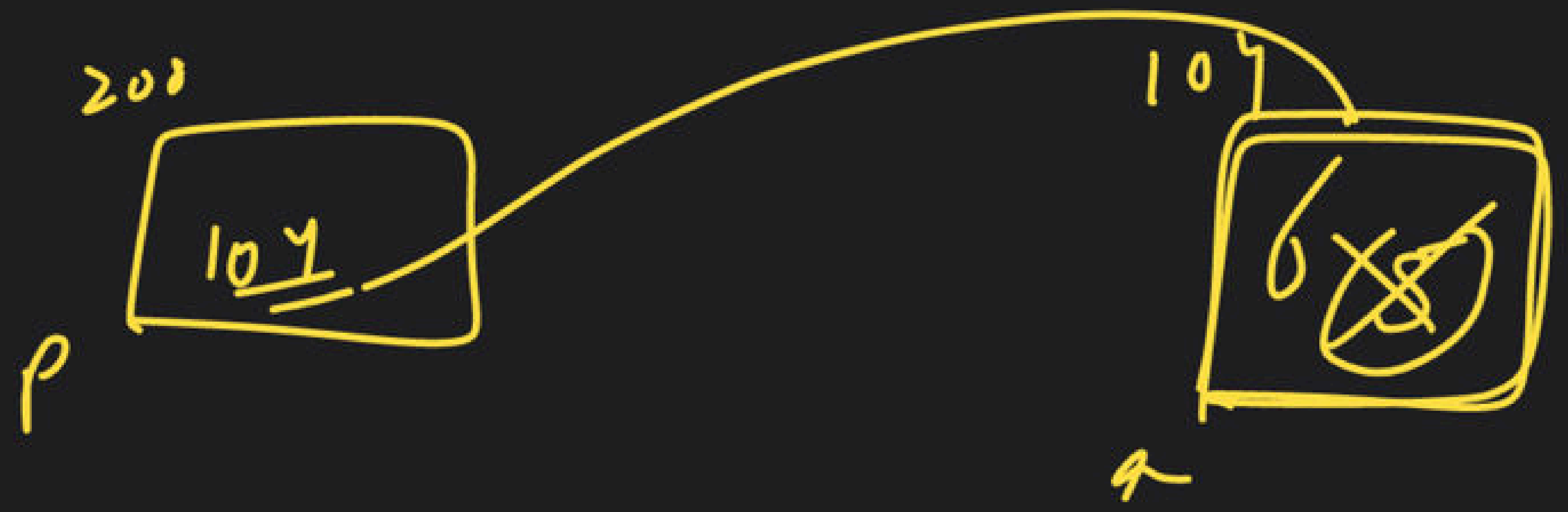
```
int a=5
int* ptr=&a;
```

a = a+1;

p = p+1

104 + 1

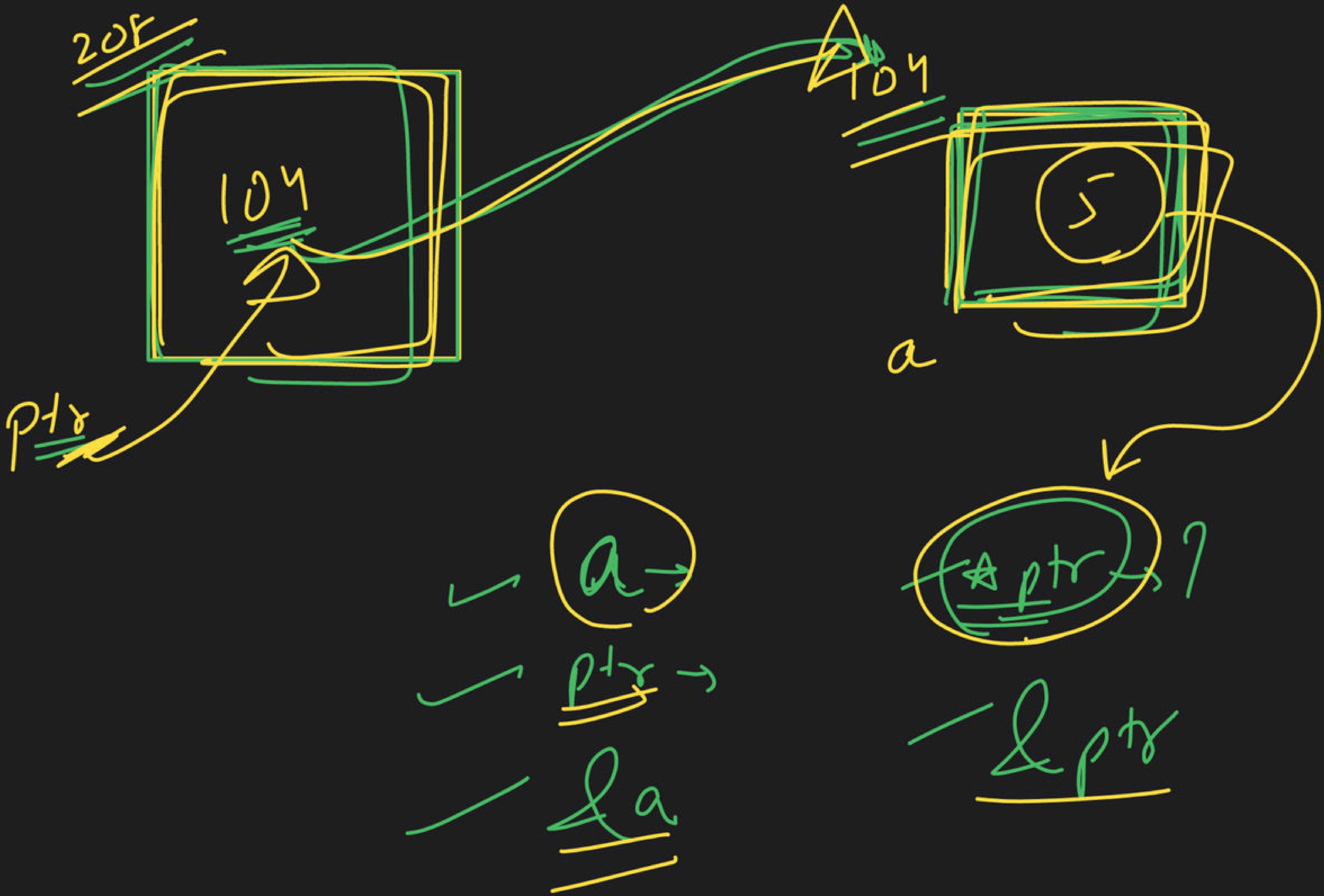
= 108

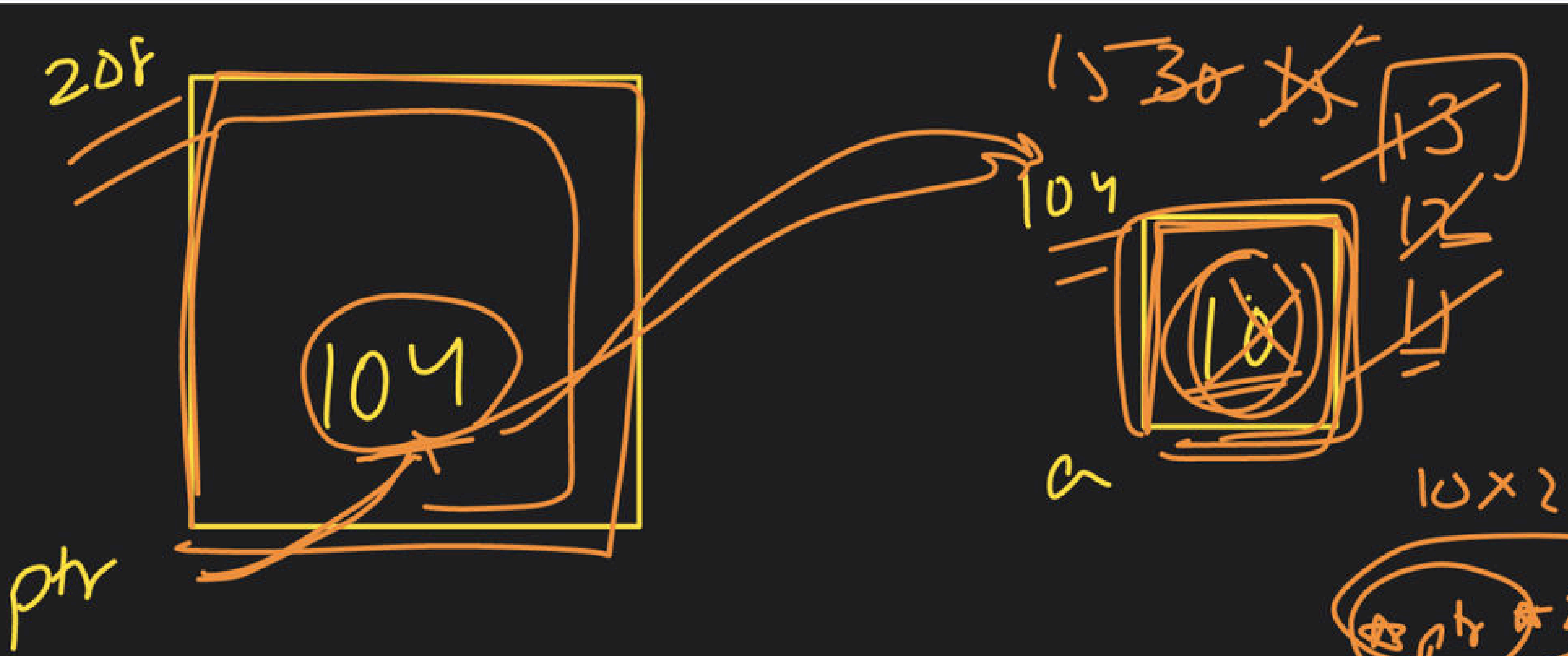


$$*p = *p + 1$$

Value at address stored in p = Value at address stored in p +1

$$\boxed{*p} = 5 + 1 = \boxed{6}$$


208
104
ptr
104
5
a
a →
ptr →
&a
*ptr → ?
&ptr



→ $*p = *p \times 2$

→ $*p = \frac{(*p)}{2}$

① → a → ? 10

→ &a → ? 104

→ ptr → ? 104

→ *ptr ? → 10

→ &ptr → 208

*ptr *2 = 20

→ *p *2

→ (*ptr)++

→ ++(*ptr)

→ a = a+1

→ *p = *p + 2

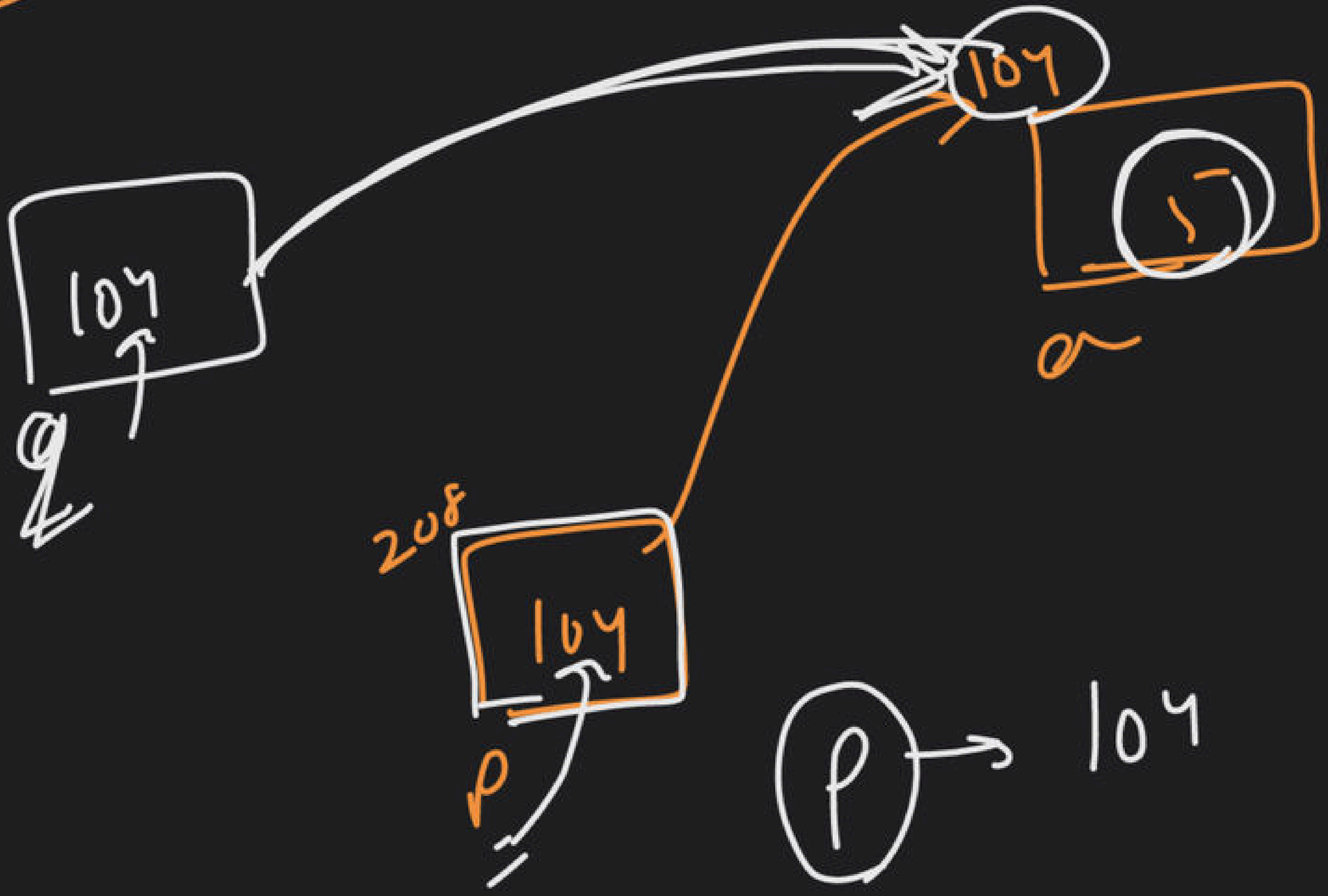
ek pointer → copy → other pointer

```
int a=5
int *p = &a
int *q = p;
int
```

(*p x 2) -> 10

*p -> 5

*q -> 5 (*q x 2) -> 10



```
int a = 5;
int* p = &a;
int* q = p;
```

→ a → 5

→ &a → 104

→ p → 104

→ &p → 201

→ *p → 5

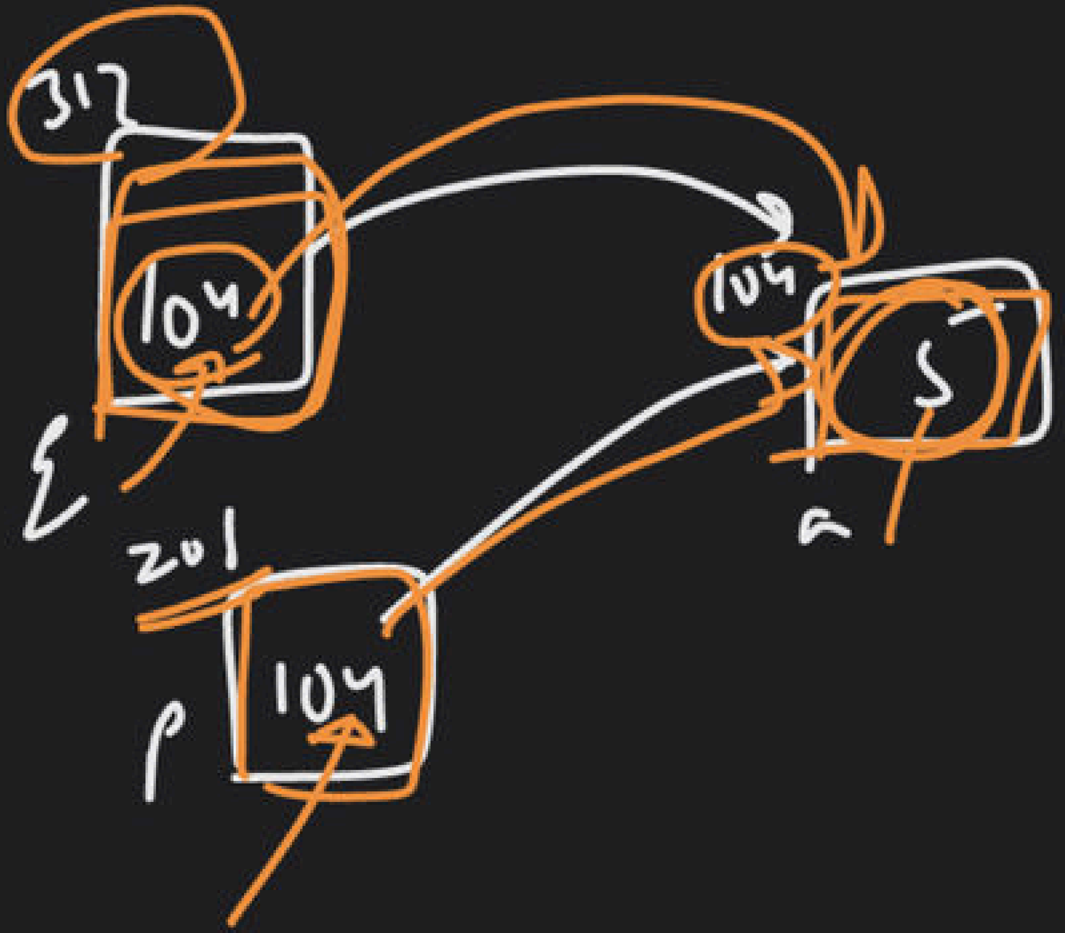
→ q → 104

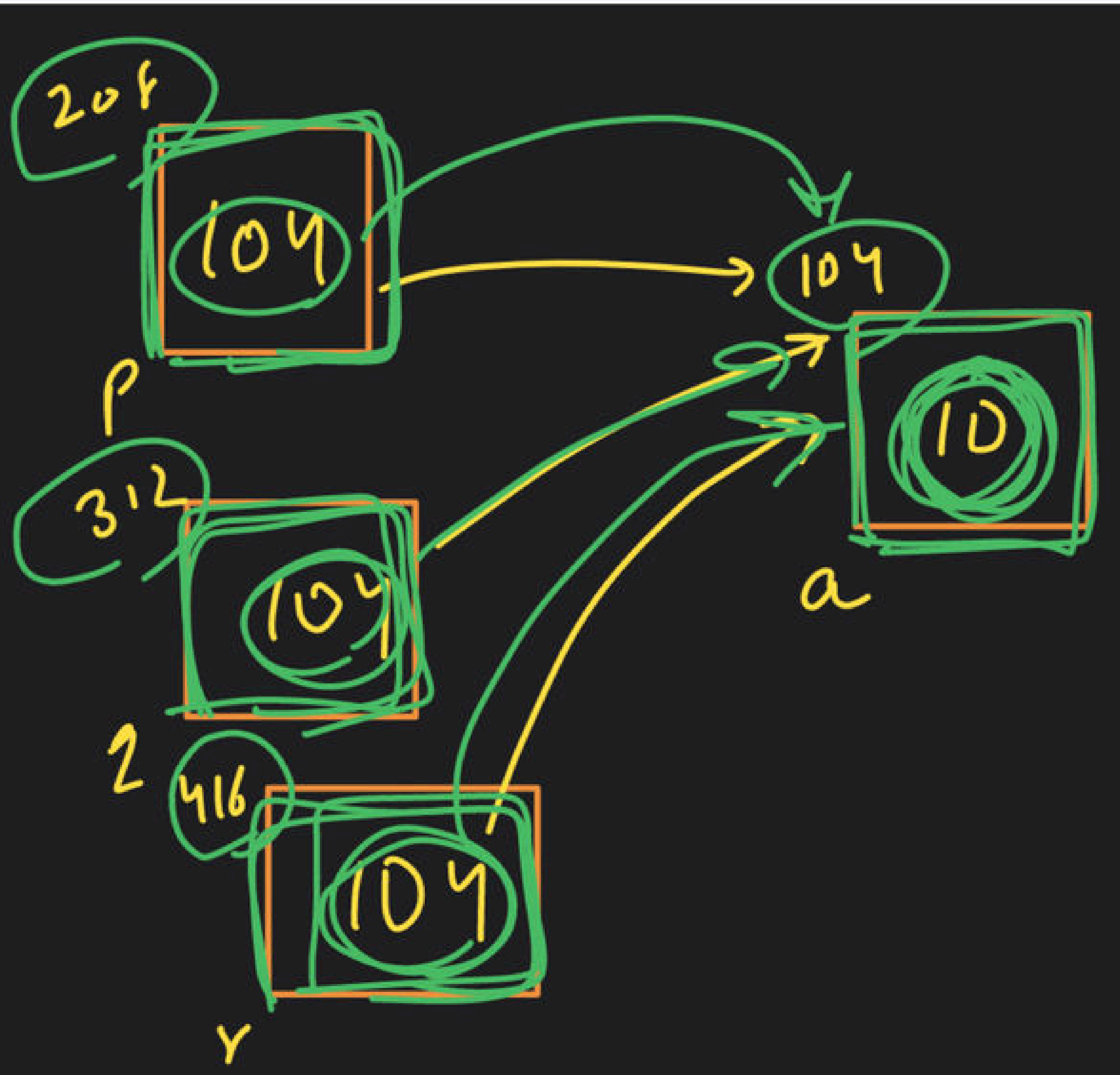
→ &q → 312

→ *q → 5

→ cout << (*p/2); 5/2

→ cout << (*q/2); 5/2





```
int a = 10;
int *p = &a;
int *q = p;
int *r = q:
```

a → 10

&a → 104

p → 104

&p → 208

*p → 10

q → 104

&q → 312

*q → 10

r → 104

&r → 416

*r → 10