Pointers - I

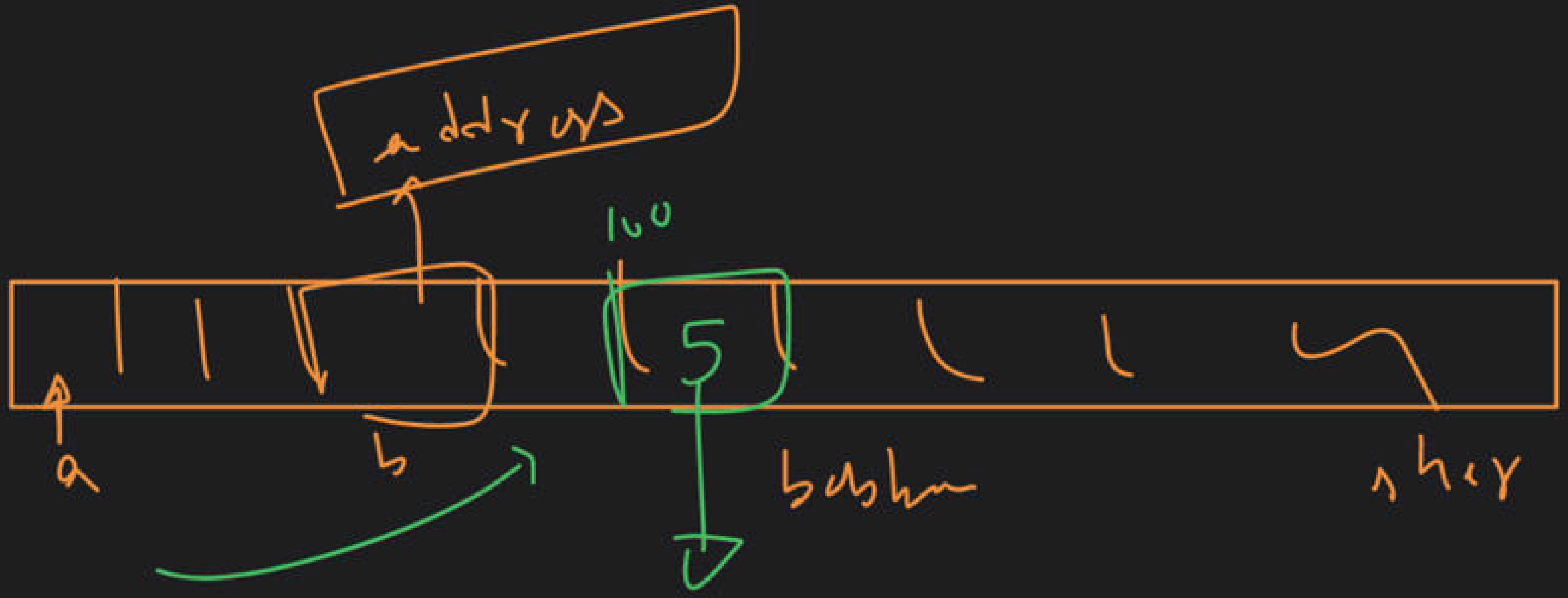
Foundation Course on Data Structures & Algorithm - Part I

Love Babbar • Lesson 16 • May 18, 2022

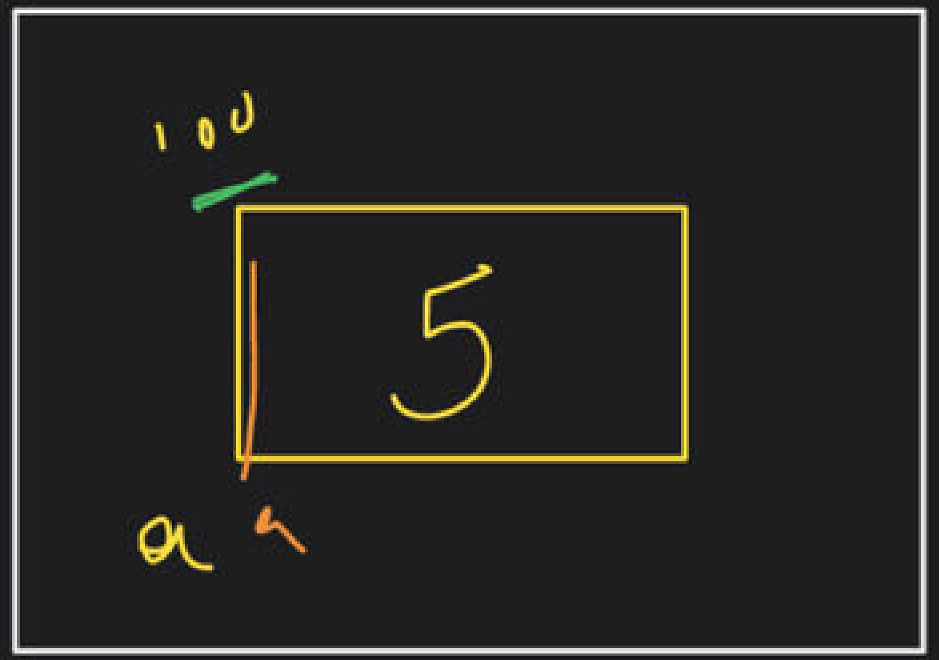
→ Pointers :-

cout << a ;

int a = 5; ?

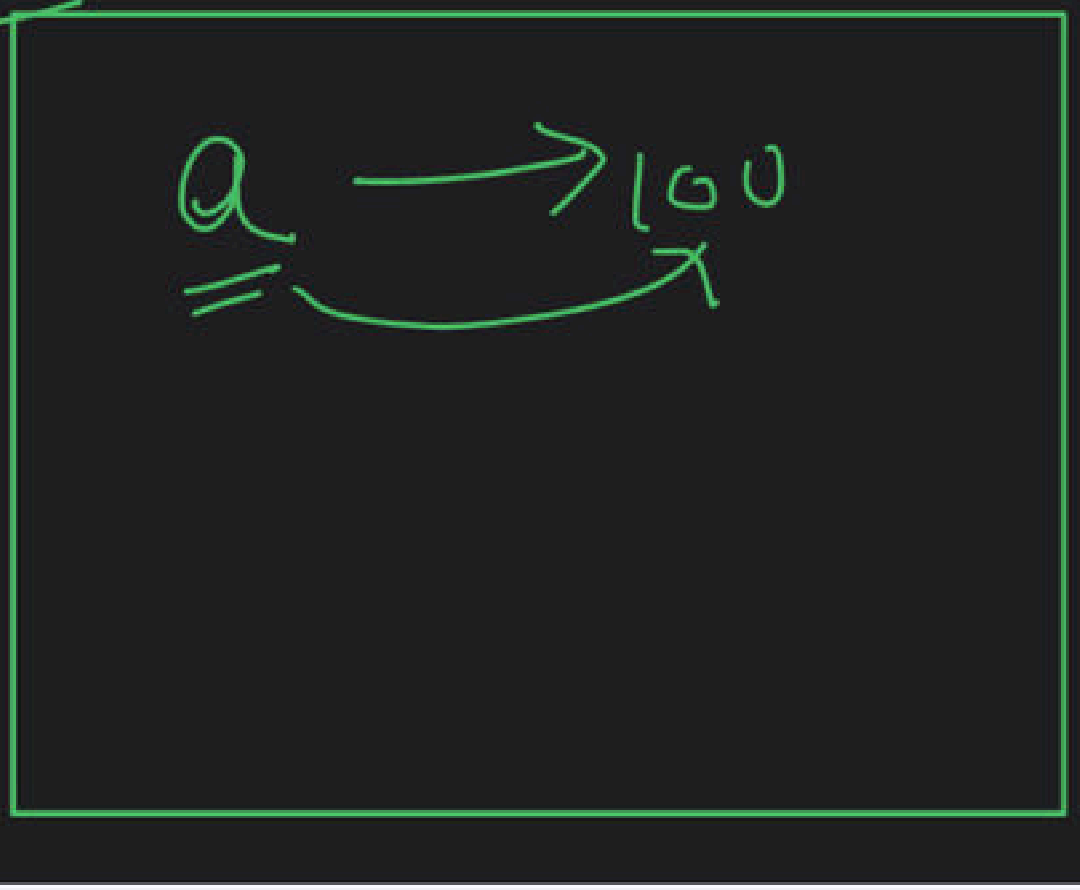


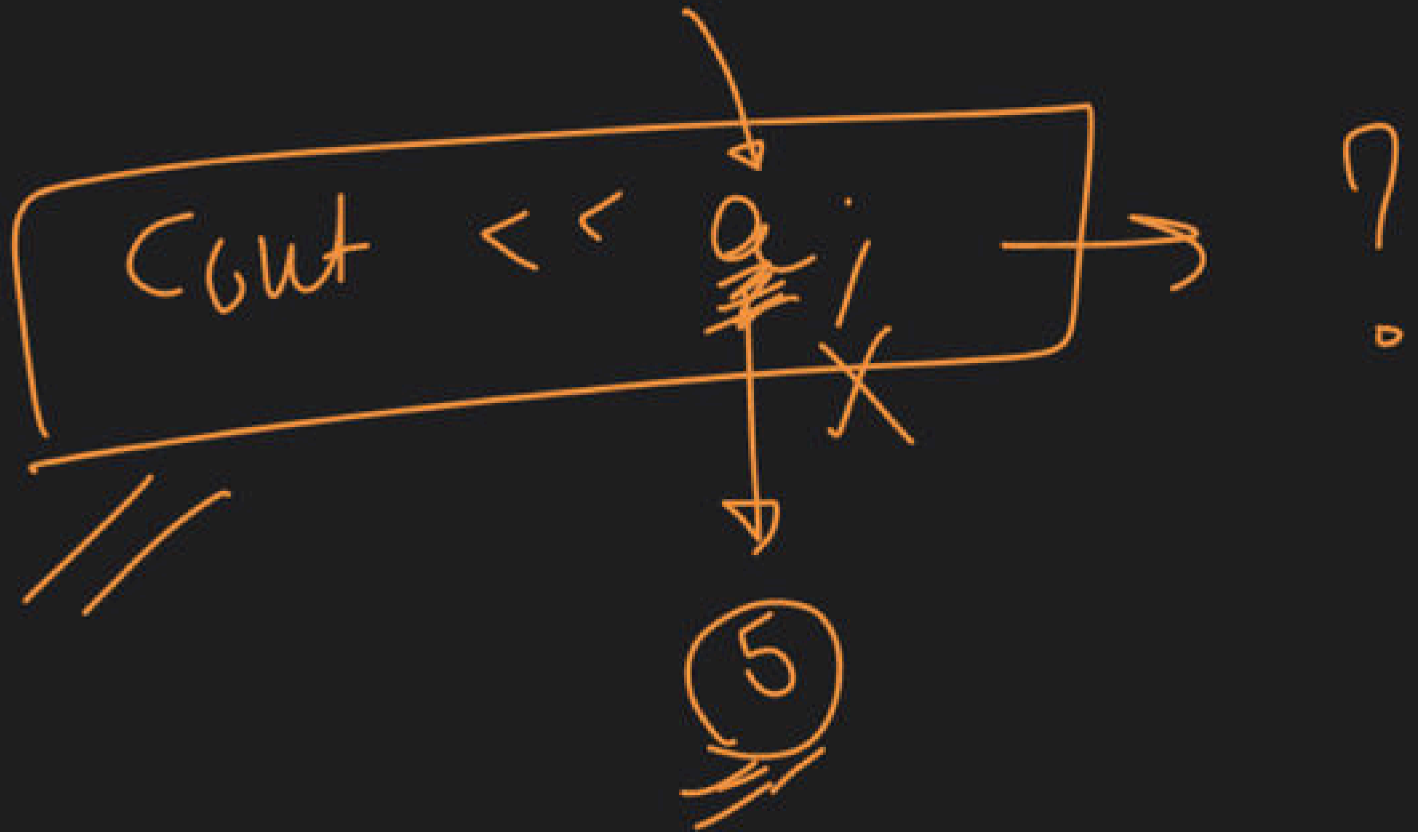
BTS → ?

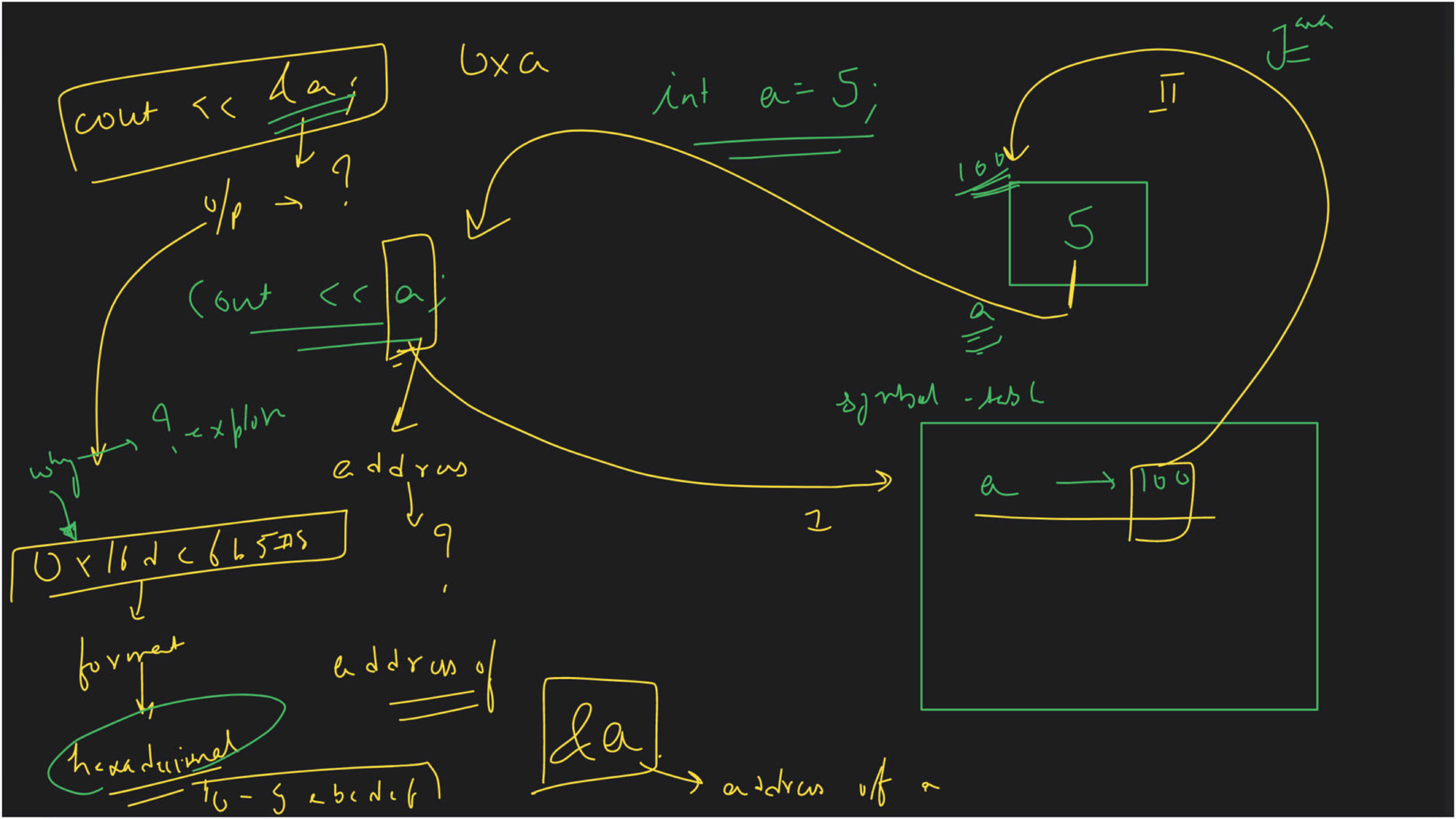


symbol table

D.1






cout << &a;
?
o/p → ?
0xa
int a = 5;
Jan
II
100
5
a
cout << a;
? explain
why
address
?
0x16dc6b5A8
format
hexadecimal
0 - 9 abcdef
address of
I
symbol - table
a → 100
&a
address of a

pointers:- stores address → Yes / No

```
int a = 5;
```

Create → `int *p = &a;`

address of

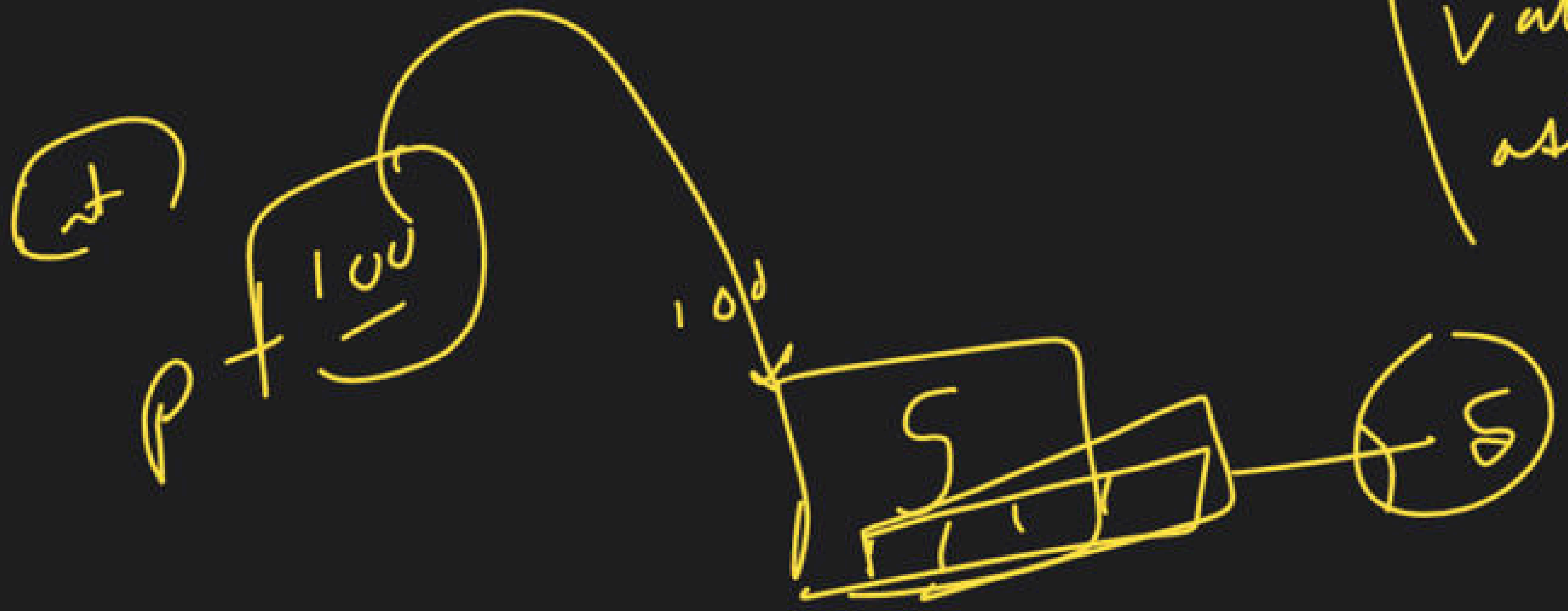
variable

int → datatype

* → format to create pointer

p is a pointer to integer data

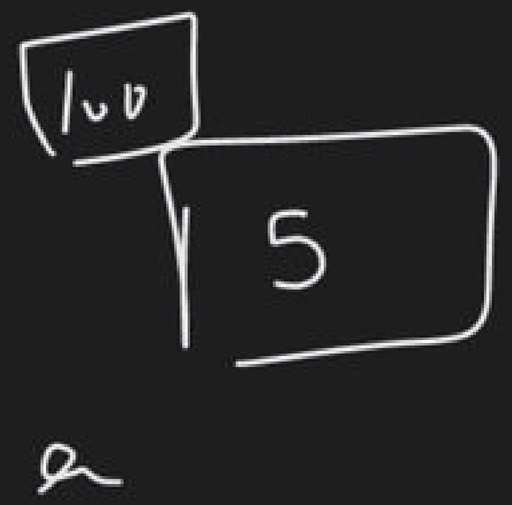
`cout << *p` → value at p



3

why → ?

int a = 5;

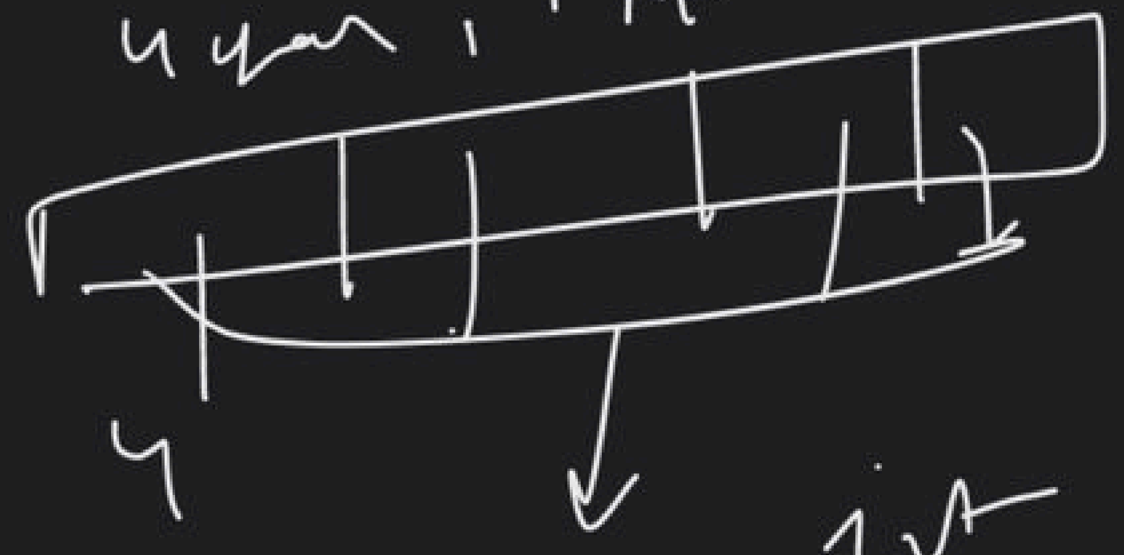


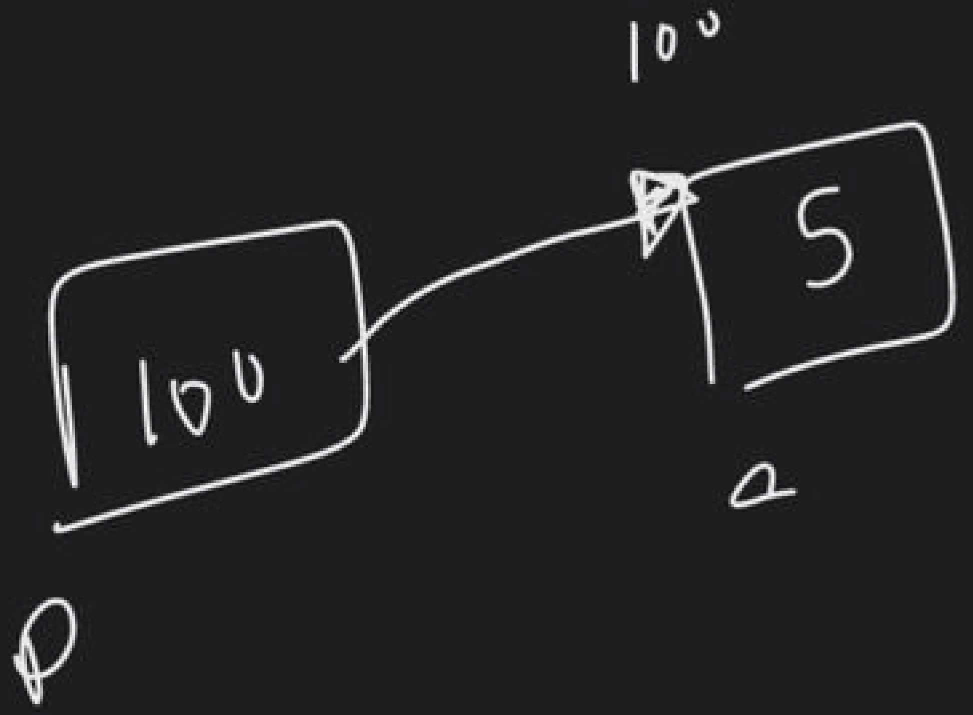
&a = 100

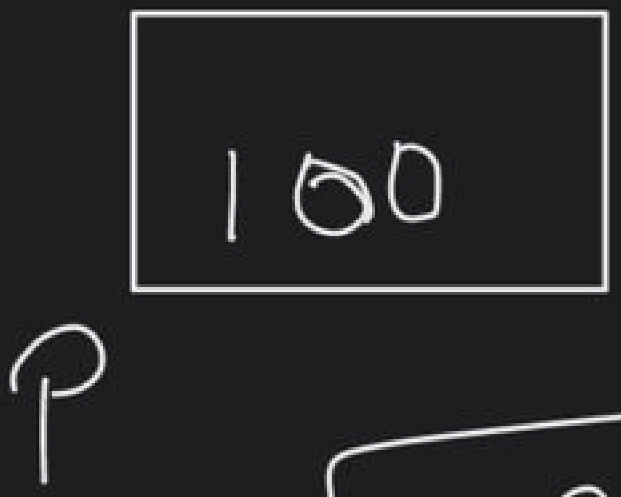
int *p = &a

char

4 char, 1 int → ?







sizeof(p) → ?

system → ?

why

8 bytes

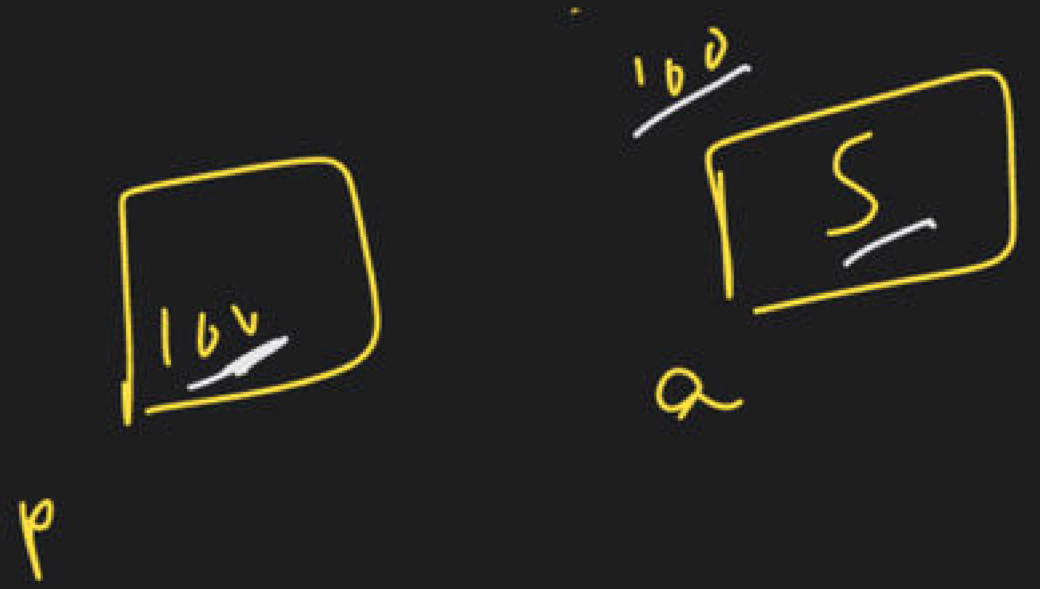
8 or 4 → article ?

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

→ ambiguity

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

*p → address



```
cout << *p << endl;
```

→ o/p → 5 → why

* → value at address pointed by p

* → dereference operator

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

```
int *p;
```

→ why not ?

↓

Danger

garbage value

0)

```
int *p = 0;
p = &a
```

pointer

```
int *p = 0;
cout << *p ?
```

→ o/p → Segmentation fault

→ play

KHELNA

& , * , create , Dangling

, NULL

int a = 5

int *p = &a ;

① cout << a → 5

cout << &a → 100 or address

cout << *a → ? → error → ?

cout << p → 100

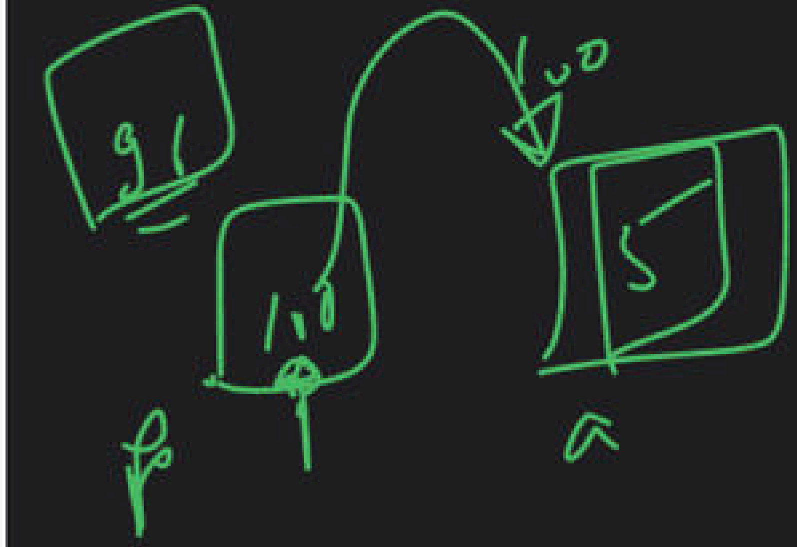
cout << &p → 91

cout << *p → 5

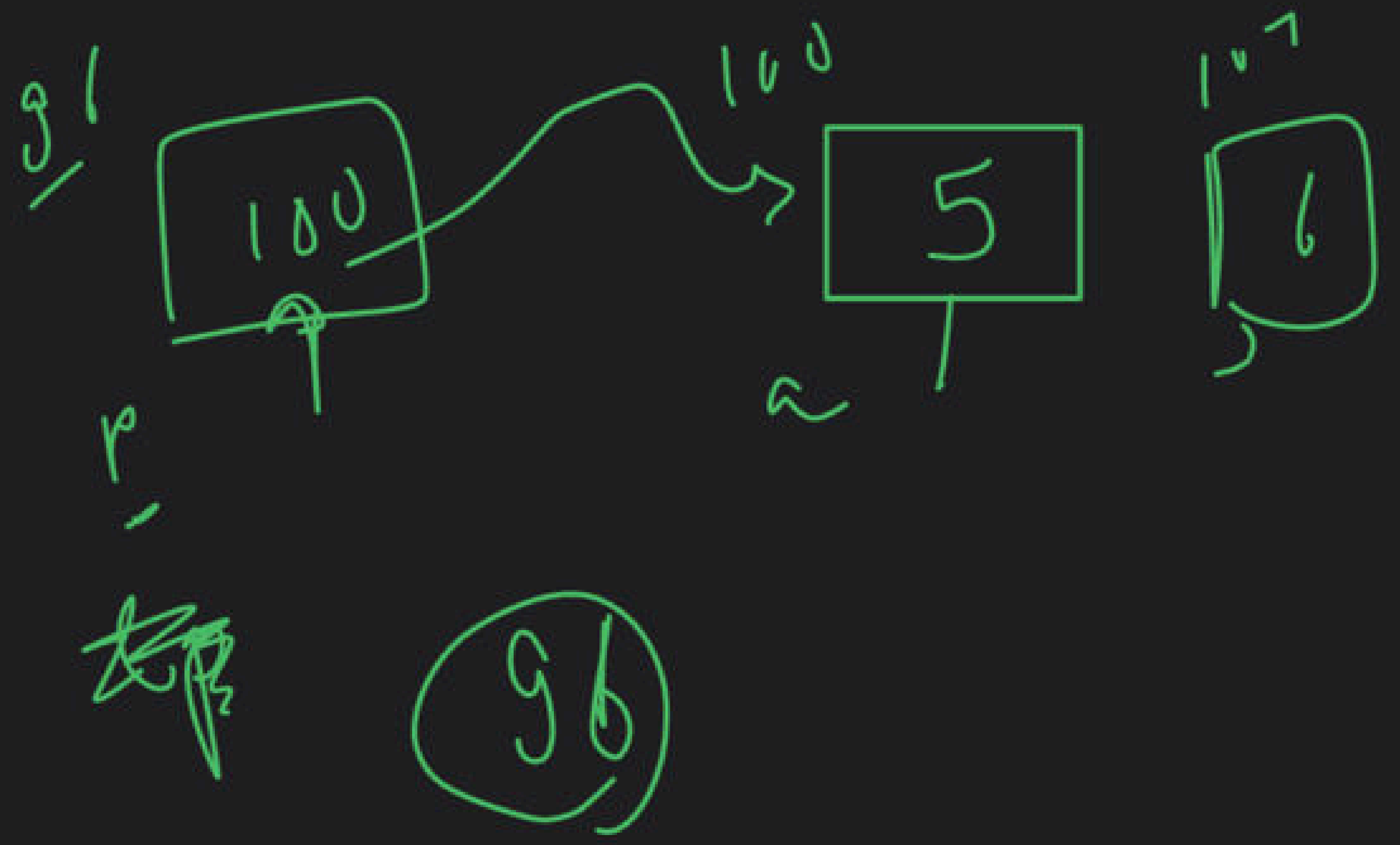
*a ✗

pointer

*p

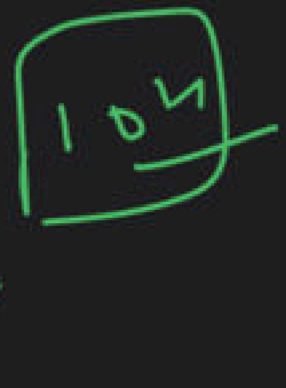


easy



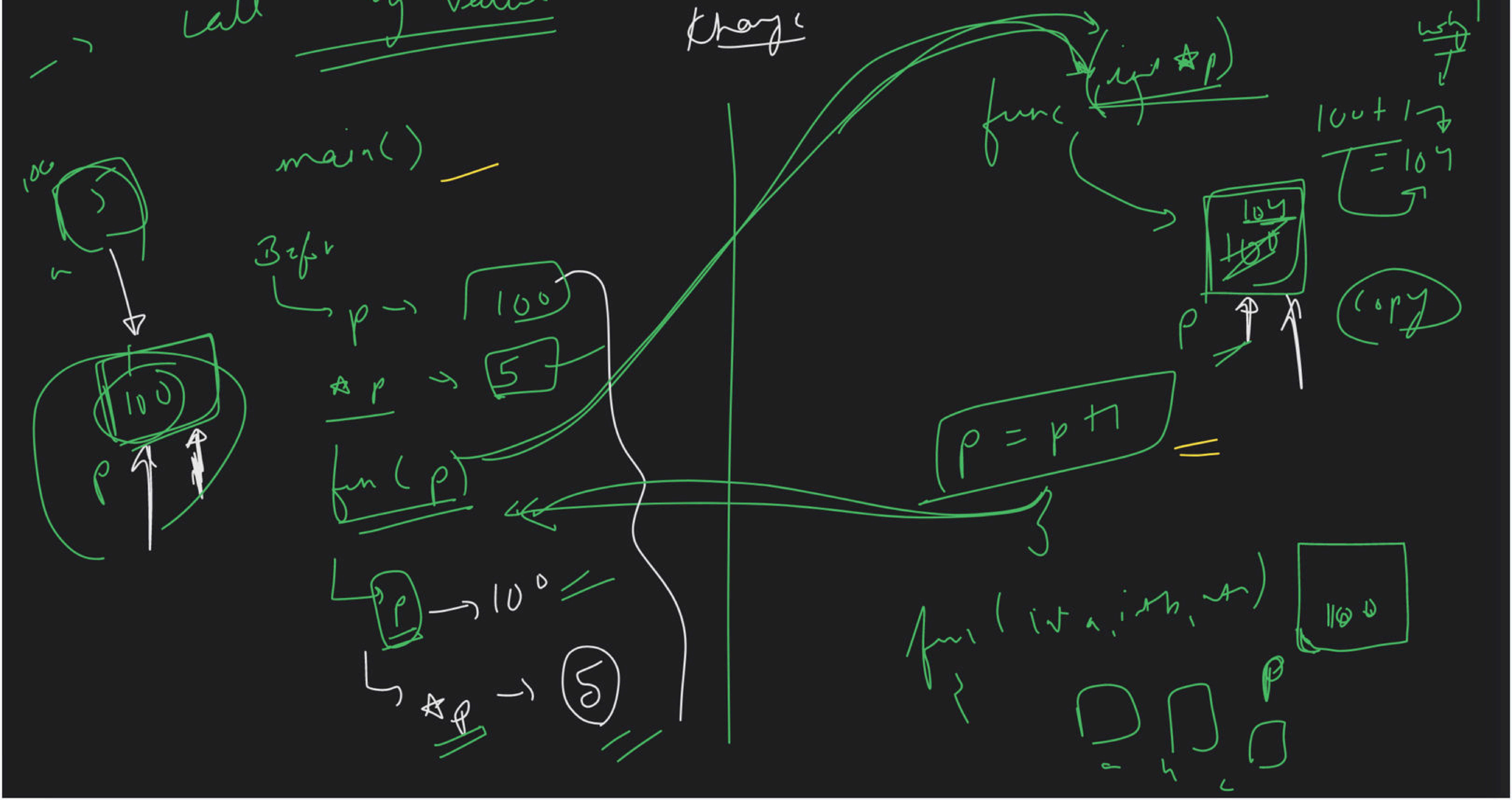
```
int main ()
{
    int a = 5

    int *p = &a

    cout << "Before" << p << endl   → (I) → o/p → (100)
                        *p << — ;   → o/p → (5)

    func (p);
    cout << "After" << p << endl;   → ? → 100 or 104 → (100) → ?
                       *p << endl;  → (5 or 6) → ?
```

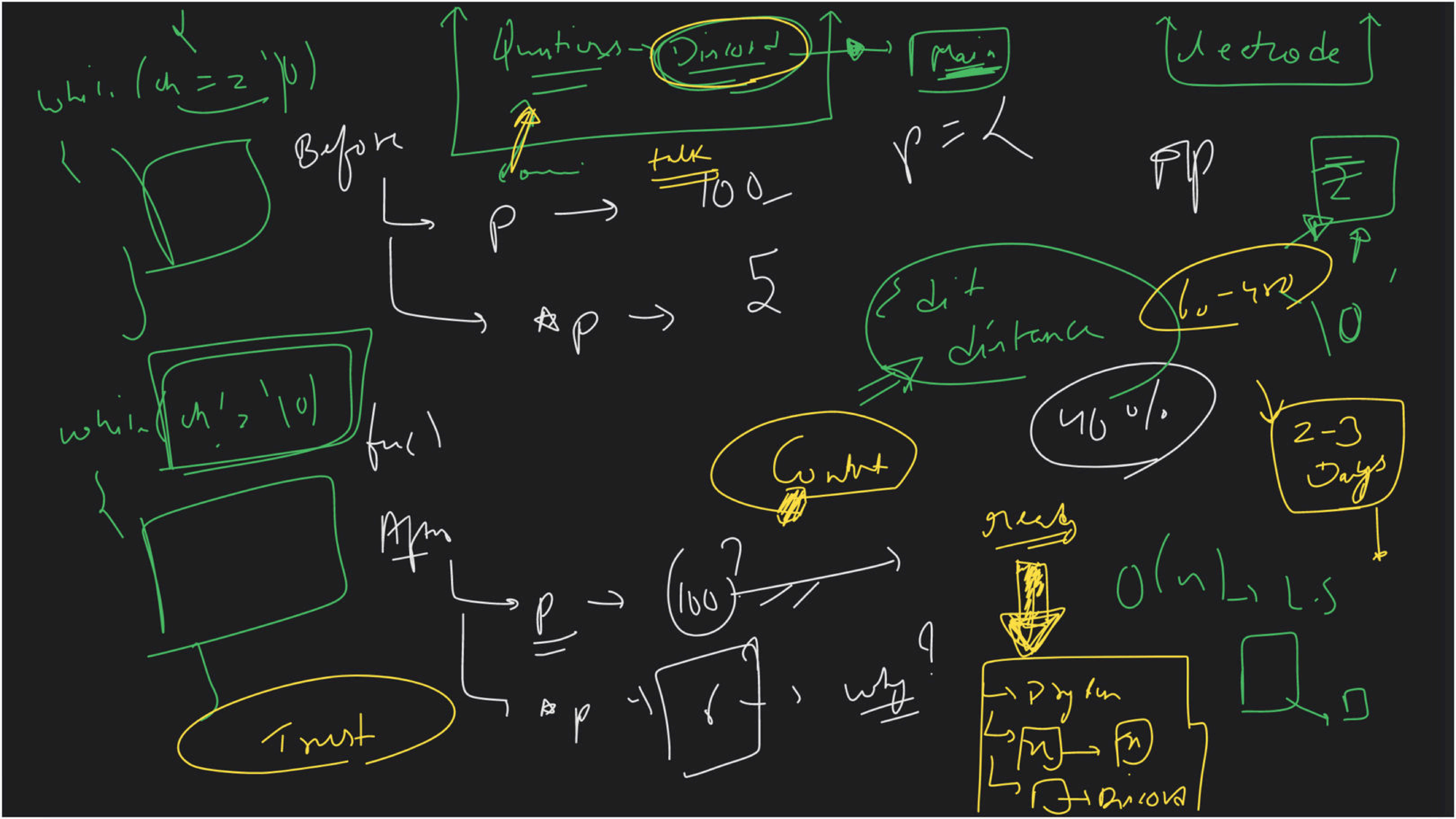
```
func (int *p)
{
    p = p + 1;
            100+1
}
```

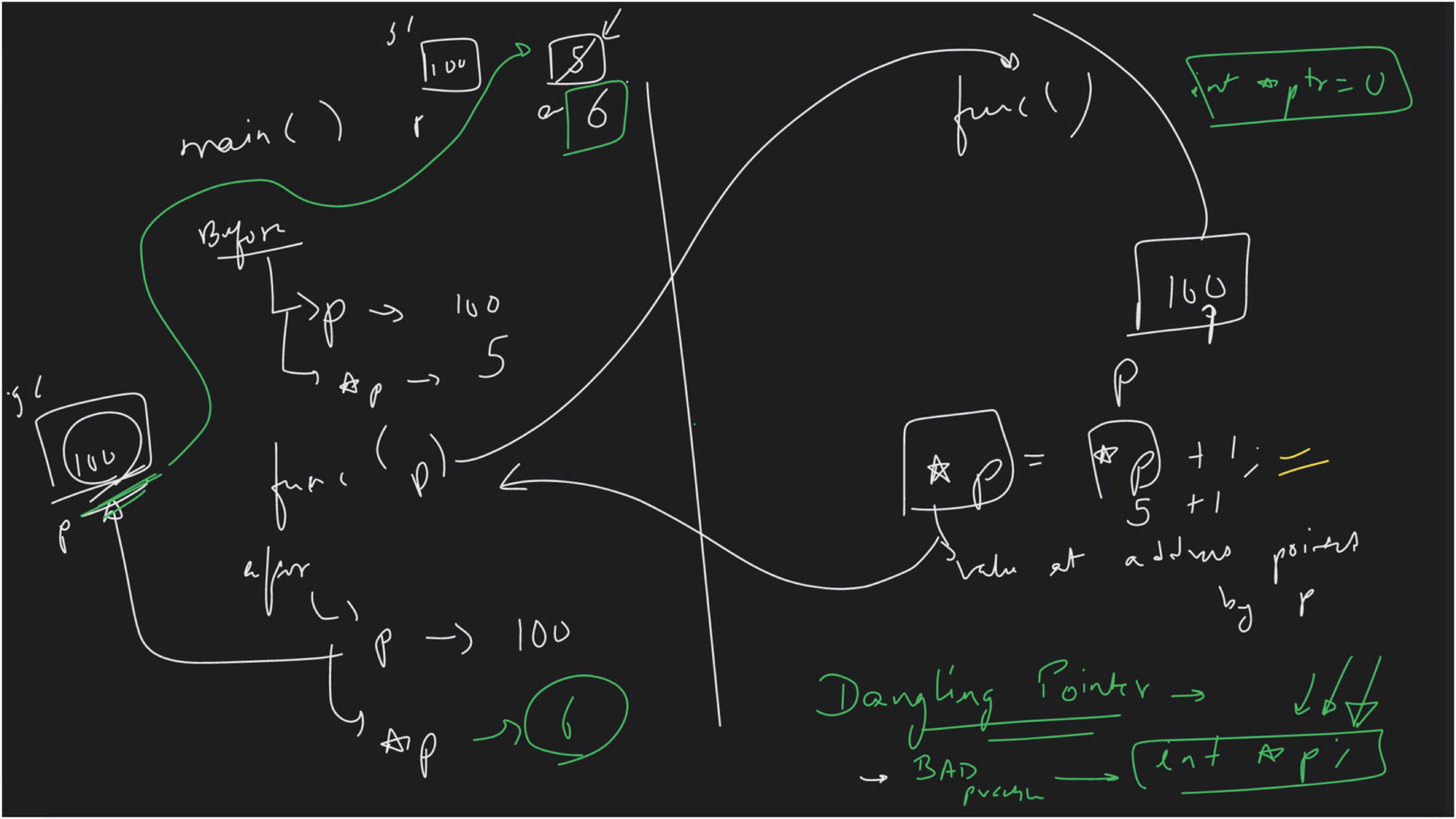


→ Call by Value

Change




while (ch == '\0')
while (ch != '\0')
fun()
Trust
Before
p → 100
*p → 5
After
p → 100?
*p → ? , why?
talk
Questions → Discord → Main
comm
p = L
PP
Electrode
2
Edit distance
60-40%
40%
10
2-3 Days
Content
greedy
O(n L L.S
Python
Discord


main( )
j1
100
p
~~5~~
a
6
Before
p → 100
*p → 5
func (p)
j1
100
p
after
p → 100
*p → 6
func( )
100
p
*p = *p + 1;
5 +1
value at address pointed by p
int *ptr = 0
Dangling Pointer →
→ BAD practice →
int *p;

Doubt :-

arr

pointer <-> Arrays

*arr → 1 ?

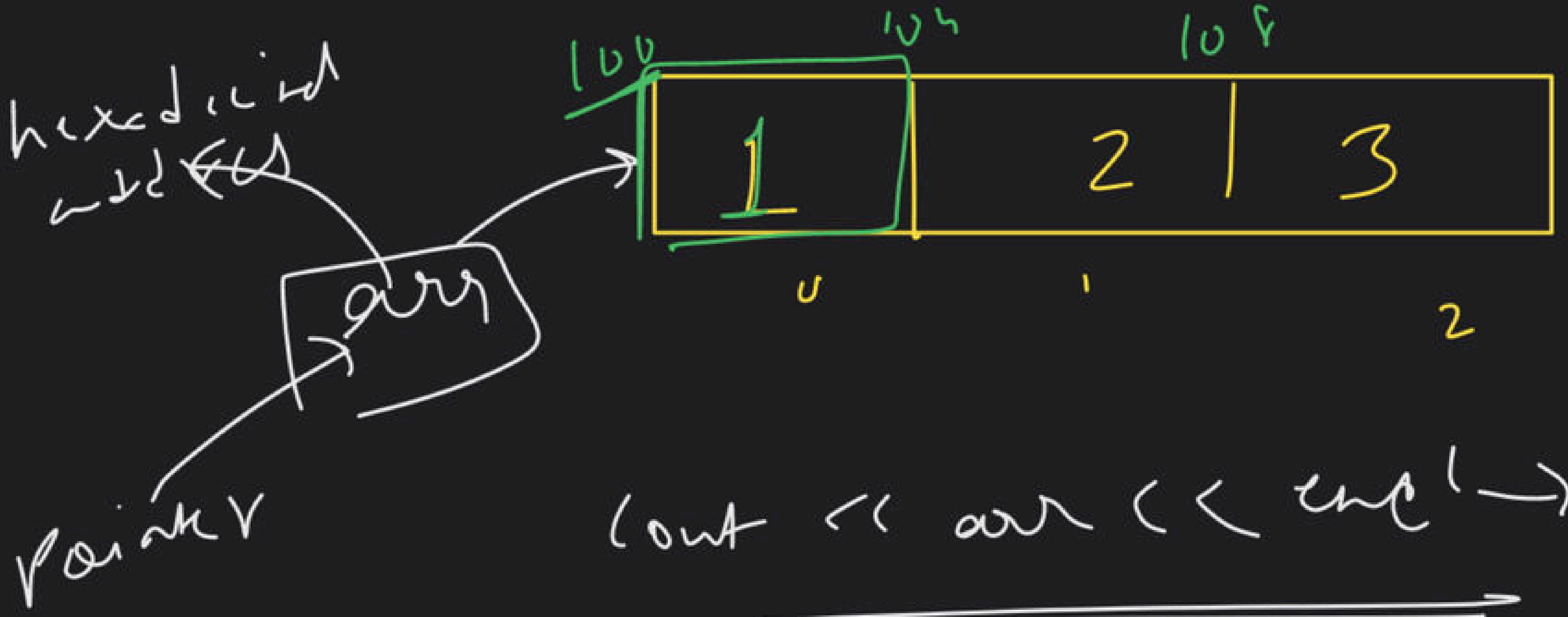
* arr

*(arr + 0) → ?

Repeat

*arr

int arr[] = {1, 2, 3}



cout << arr << endl → ?

arr → ?

↓

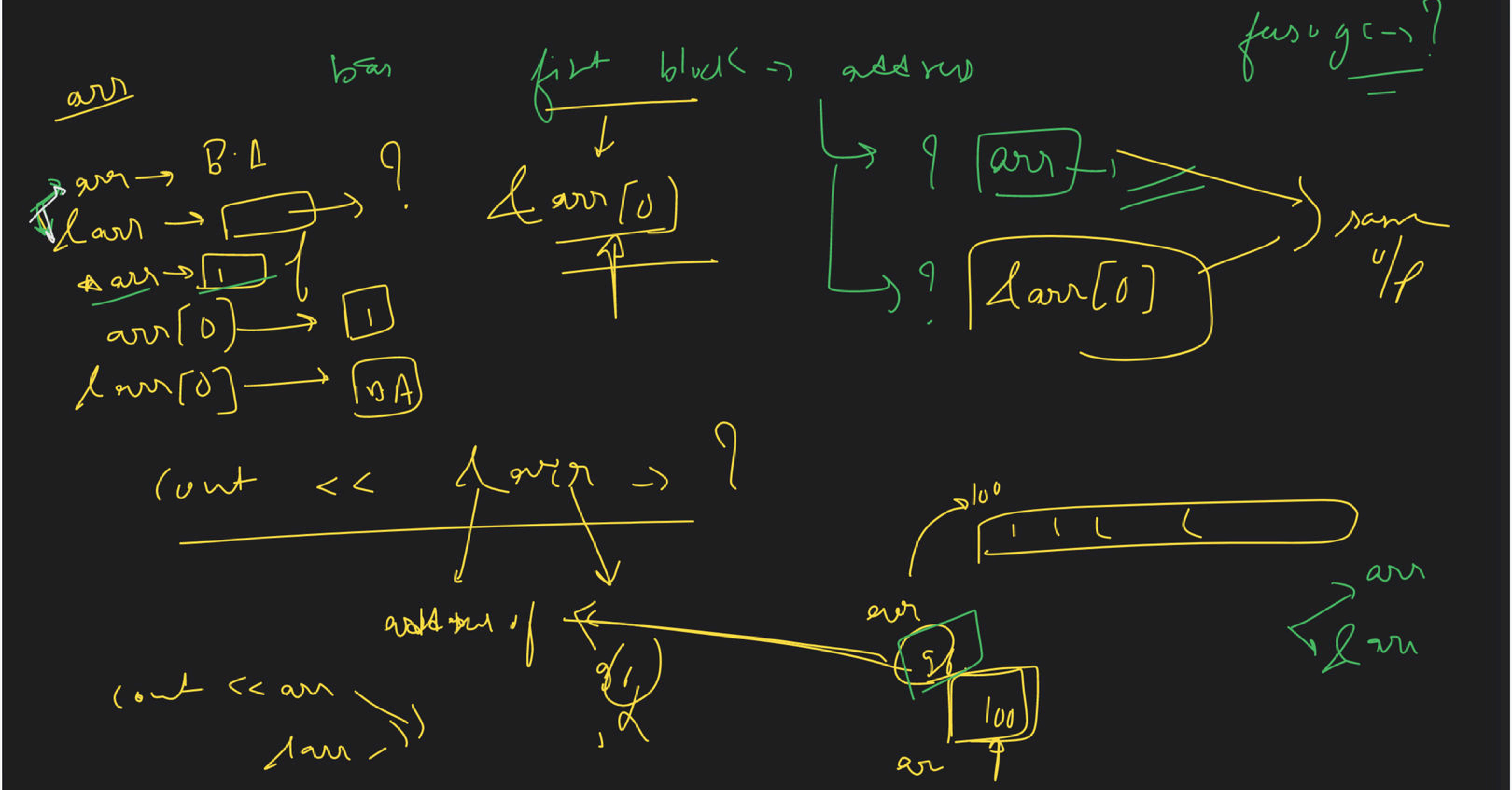
Base Address

↓

starting address

↓

starting point


arr
base
first block -> address
fasing -> ?
arr -> B.A
&arr ->
?
*arr -> 1
arr[0] -> 1
&arr[0] -> B.A
&arr[0]
? arr
? &arr[0]
same
o/p
cout << &arr -> ?
address of
cout << arr
&arr
100
arr
91
100
arr
arr
&arr

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

→ why — ?

└→ self referencing → ?

2 min

assignment

└→ *arr → → 1 → arr[0] → *(arr+0)

└→ *(arr + 1) → 2 → arr[1] → *(arr+1)

└→ *(arr+2) → 3 → arr[2] →

└→ *(arr+3) → 4 → arr(3)

└→ *(arr + 4) → 5 → arr[4)

arr[i] = *(arr + i) → Yes or No

any other way

Run

i[arr] = *(i + arr) → Yes or No

index

way

i[arr] → *(arr + i)

arr[i]

arr[i]

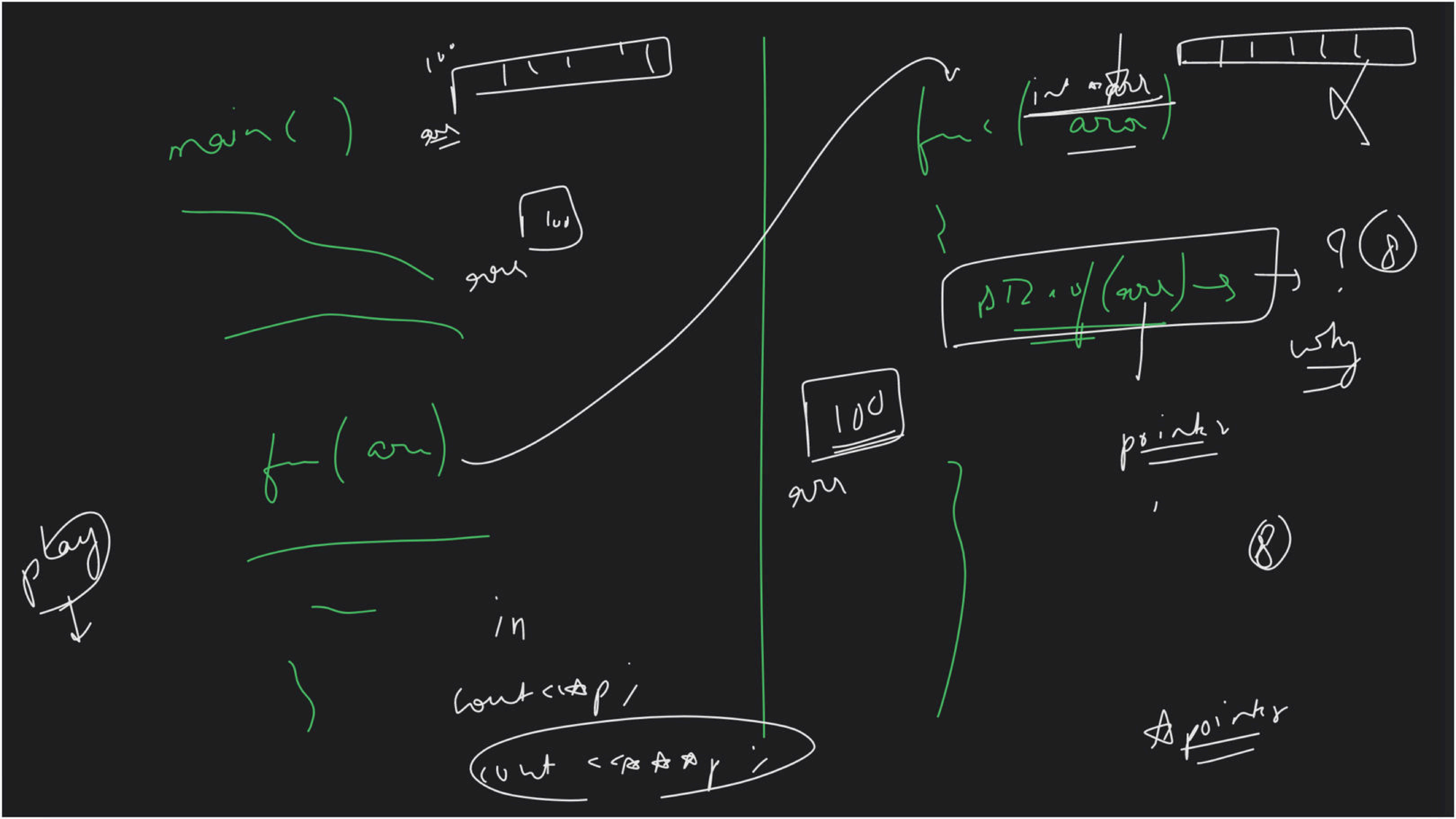
a th

b th

i[arr] — why = ?

→ cout << i[arr] << endl → Error

arr[i]


main( )
100
arr
100
arr
fun(arr)
play
in
cout<<*p;
cout<<**p;
fun(int *arr / arr)
100
arr
sizeof(arr)
? 8
why
pointer
8
*pointer

Pointer

→ adv :-

↳ memory efficiently utilise

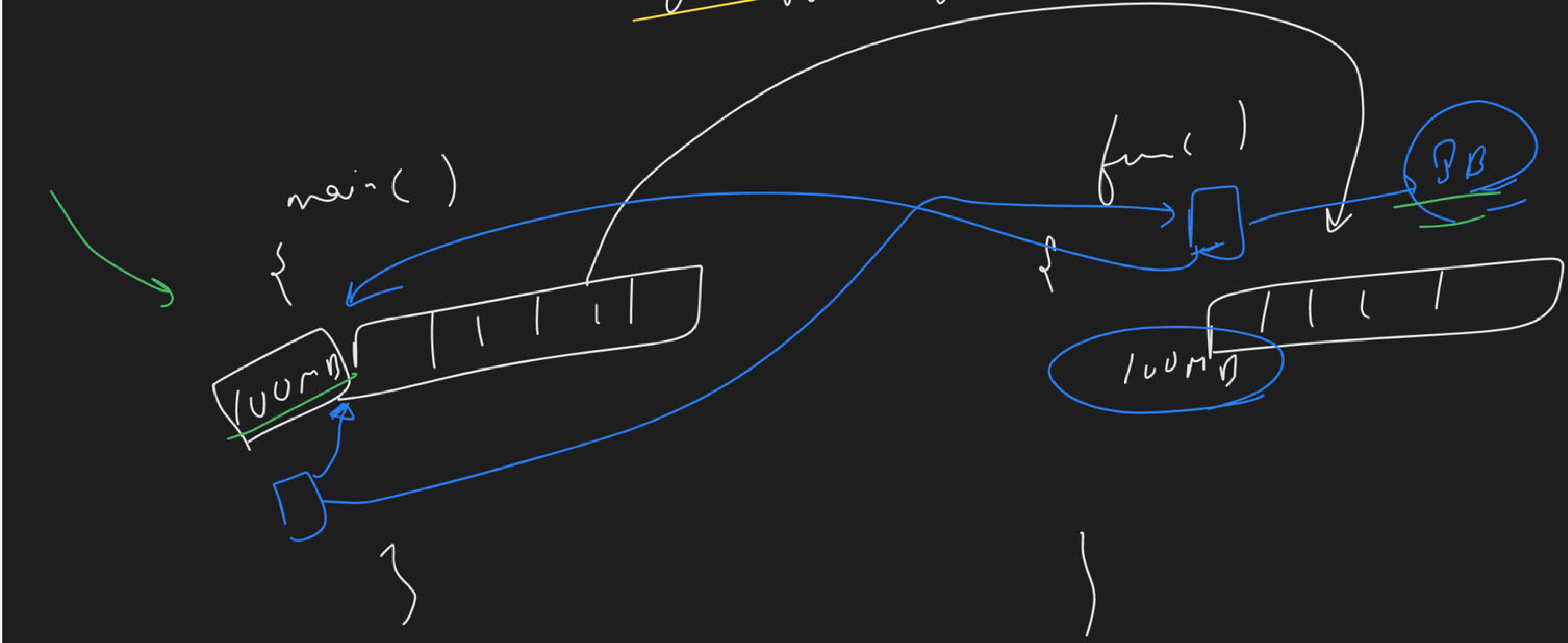
arr → Base address

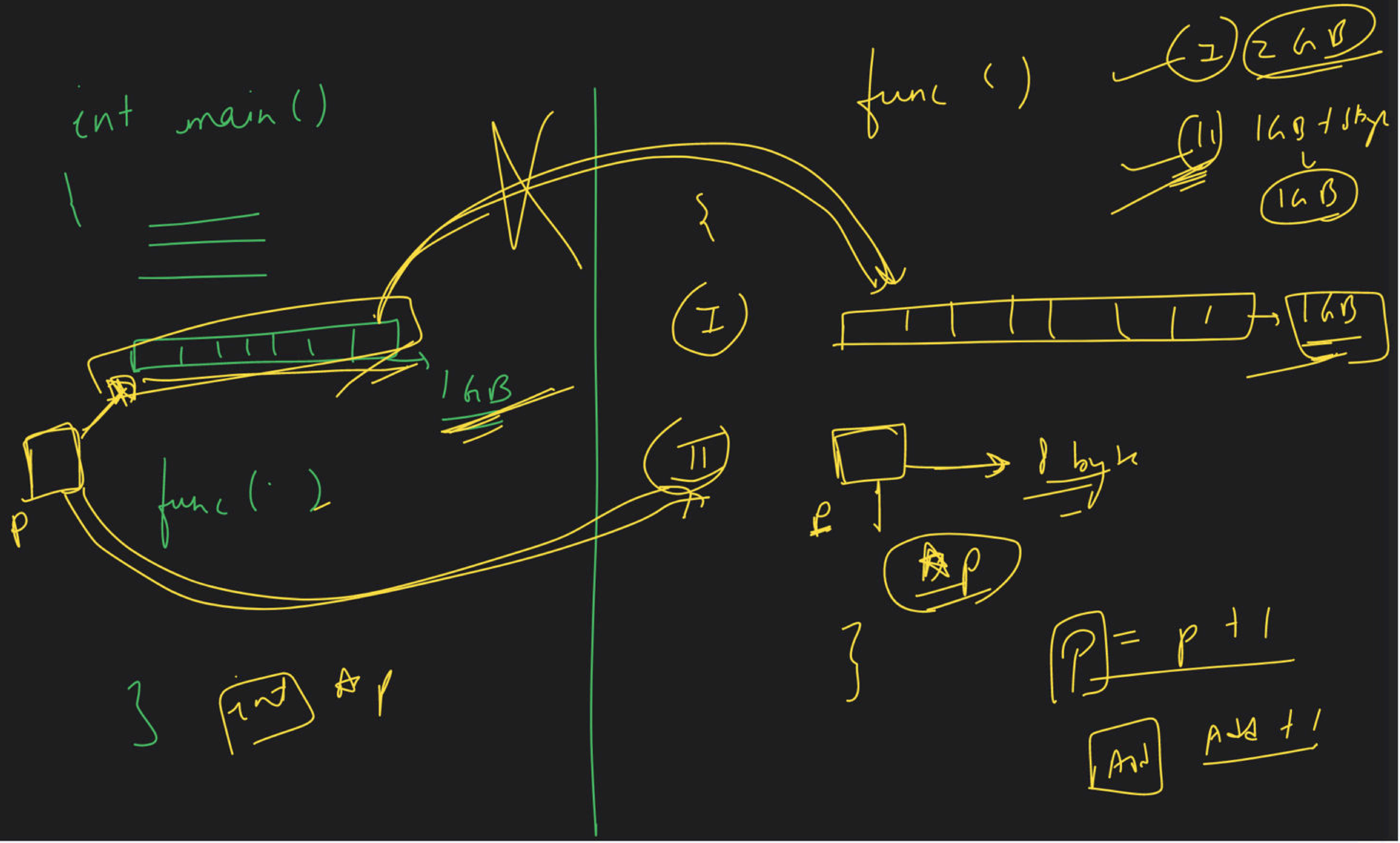
arr

100

sizeof(arr)

8




int main()
{
func( )
}
int *p
p
1GB
func ( )
{
I
II
1GB
8 byte
p
*p
}
I 2GB
II 1GB + 8 byte
1GB
P = p + 1
Add
Add +1

→ int *p = &a

sizeof(p)

char *ch = &a

sizeof(ch)

8

Diff

char a = 'z')

100 + 1 = 101

100 + 1 = 102

$\boxed{100 + 1 = 104}$

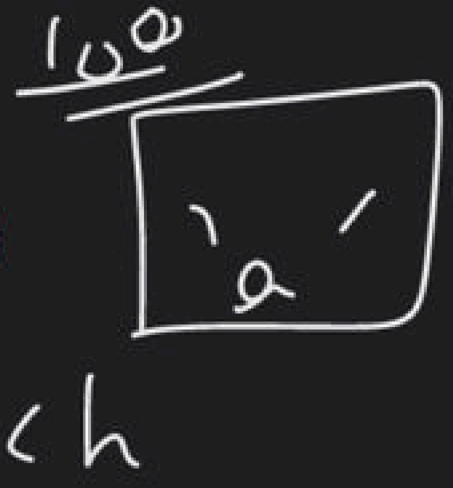
100 + 1 = 108

-> pointer -> array

-> char array

g X u ? 10

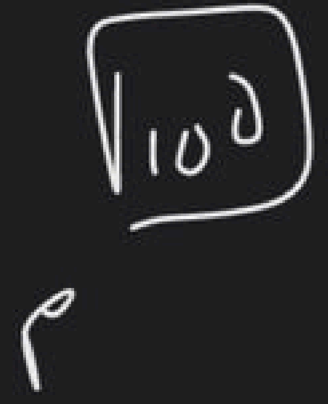
char ch = 'a'; -> 10



Repeat

cout

char *p = &ch



cout << p -> ?

-> ?

cout

char ch[5] = "abcd";

char *p = ch;

cout << p -> ? -> answer

entire string

why -> ?

yes or No

abcd

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

int *ptr = arr

cout << ptr -> address

ans -> 1

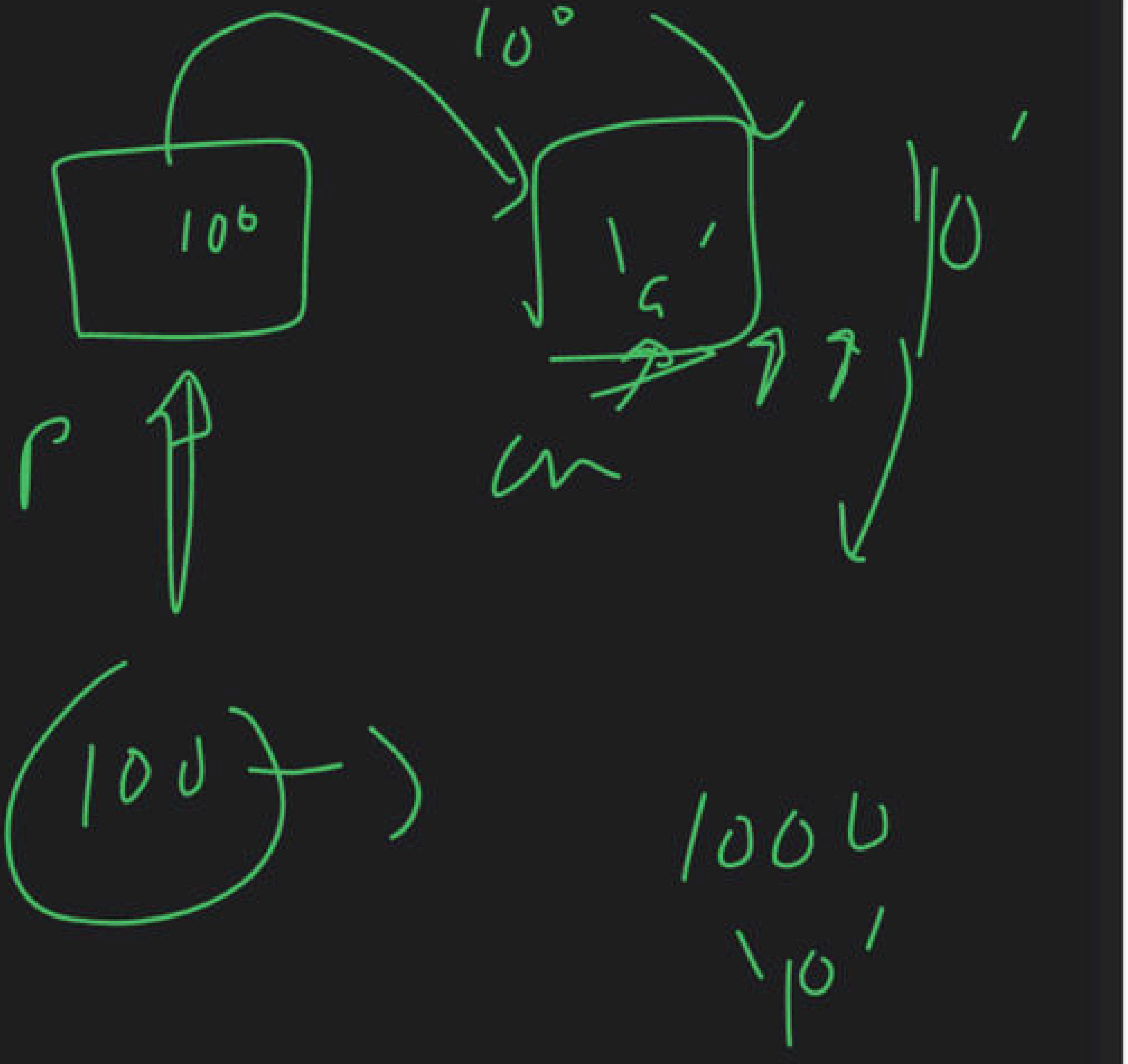
*arr

*(arr + 0)

arr[0]

O/P <

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

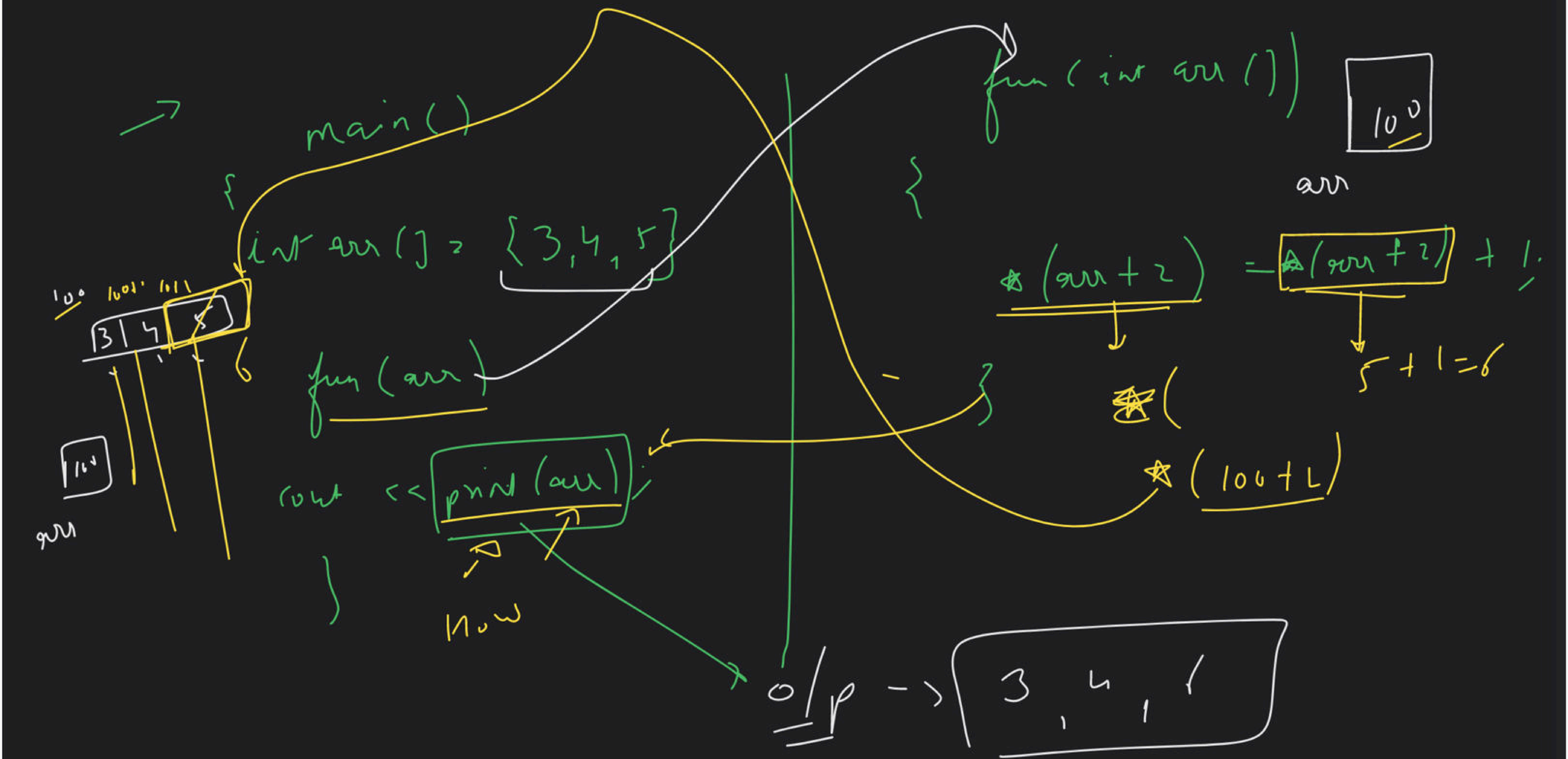


```
cout << *p →
```

$*p = *(p+0)$

$= *p[0] \rightarrow$ 'a'

1

main()
{
int arr[] = {3,4,5}
100 1004 1011
3 | 4 | 5
100
arr
6
fun(arr)
cout << print(arr)
How
}
fun(int arr[])
{
*(arr+2) = *(arr+2) + 1;
}
100
arr
*(
*(100+2)
5+1=6
o/p -> 3, 4, 6

```
int   main ()
{

   }
```

Syntax

cond^n → T/F

```
if (cout << "Barshan)
   {  0
      }
else
   {  0
      }
```

explain

Q

arr , &arr

name

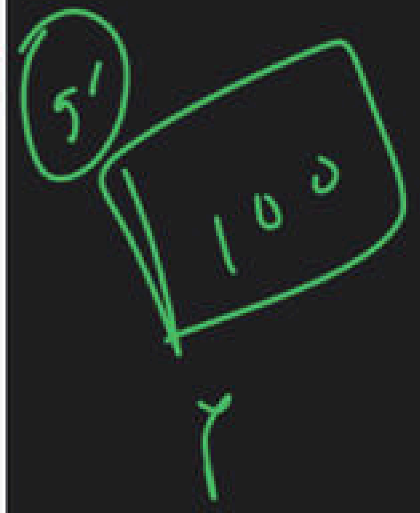
char sentence[] = "My name is Babbar";

char *p = sentence;

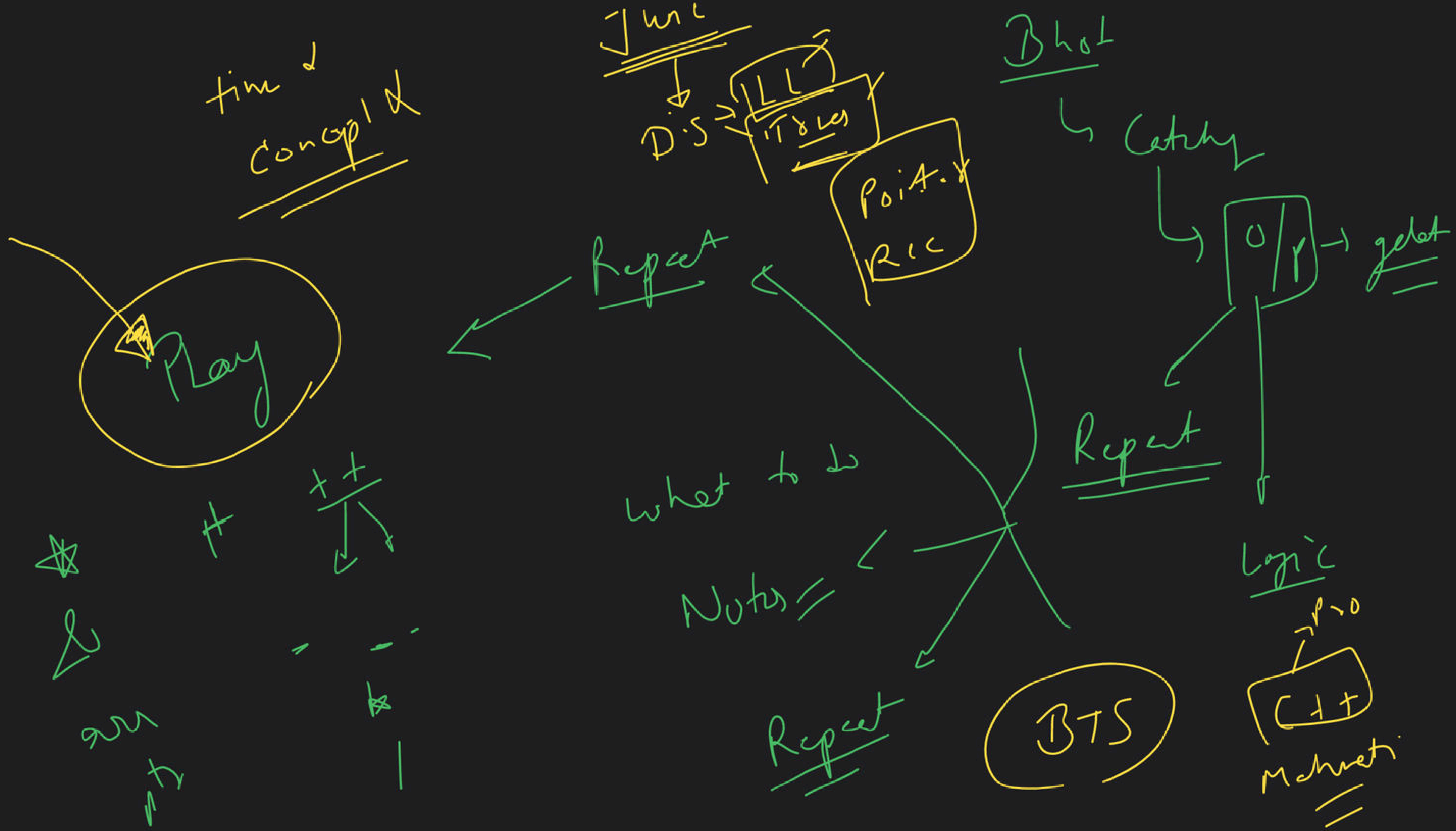
cout << p; -> ? ->

cout << *p -> ? -> My -> M

cout << &p -> ? -> address -> 96



*p
= *(p+0)
= p[0]



time &

Concept X

June

D.S →

LL

Trees

Poi A. X

R.C

Bhot

Catchy

O/P → galat

Repeat

Play

Repeat

what to do

Notes

Repeat

BTS

Logic

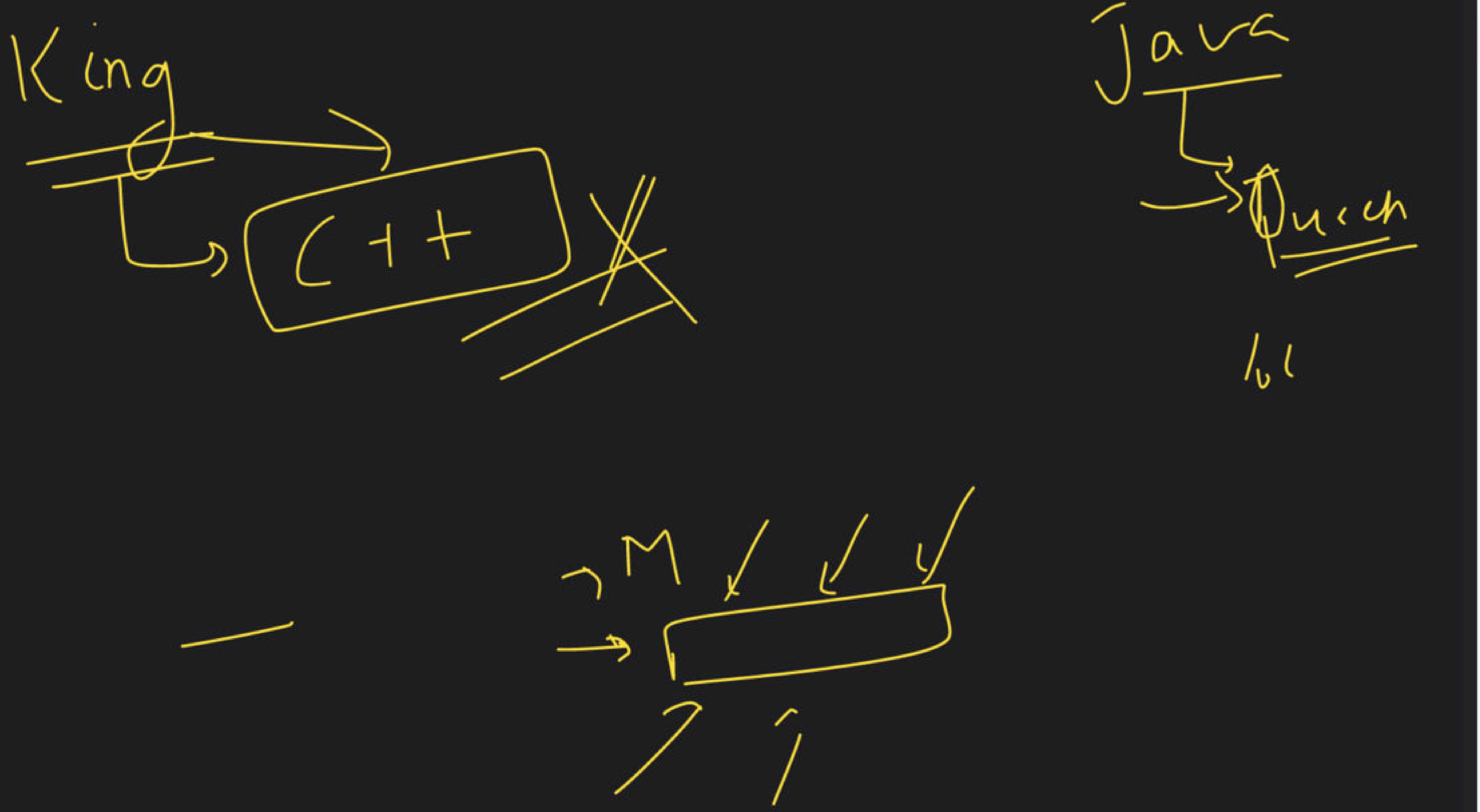
→Pro

C++

Mathemati

am

pts


King
C++
Java
Queen
M