Static/Dynamic Allocation

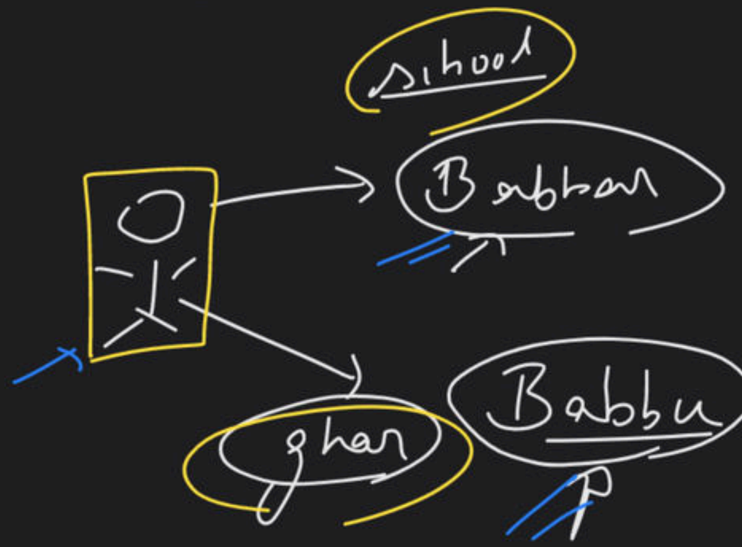
Foundation Course on Data Structures & Algorithm - Part I

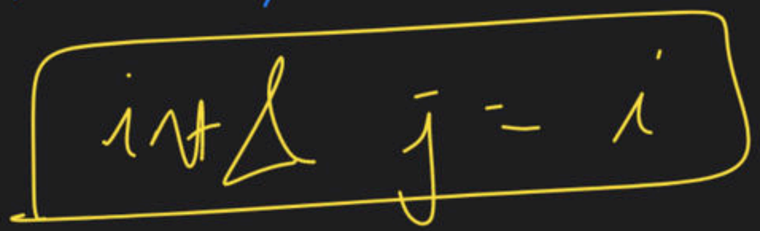
Love Babbar • Lesson 18 • May 21, 2022

→ Reference Variable :-

[same memory / different names]

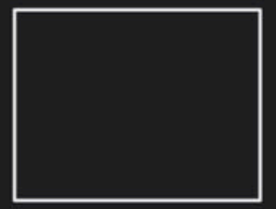
int = &i

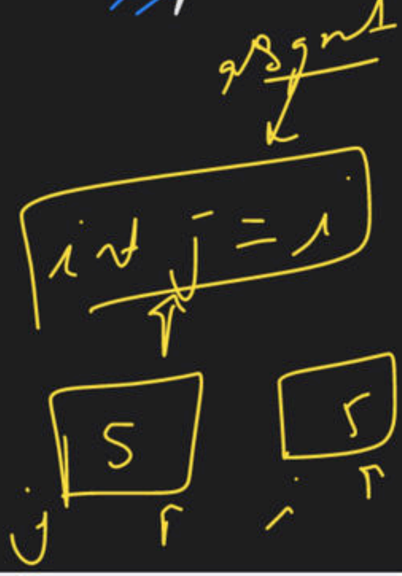


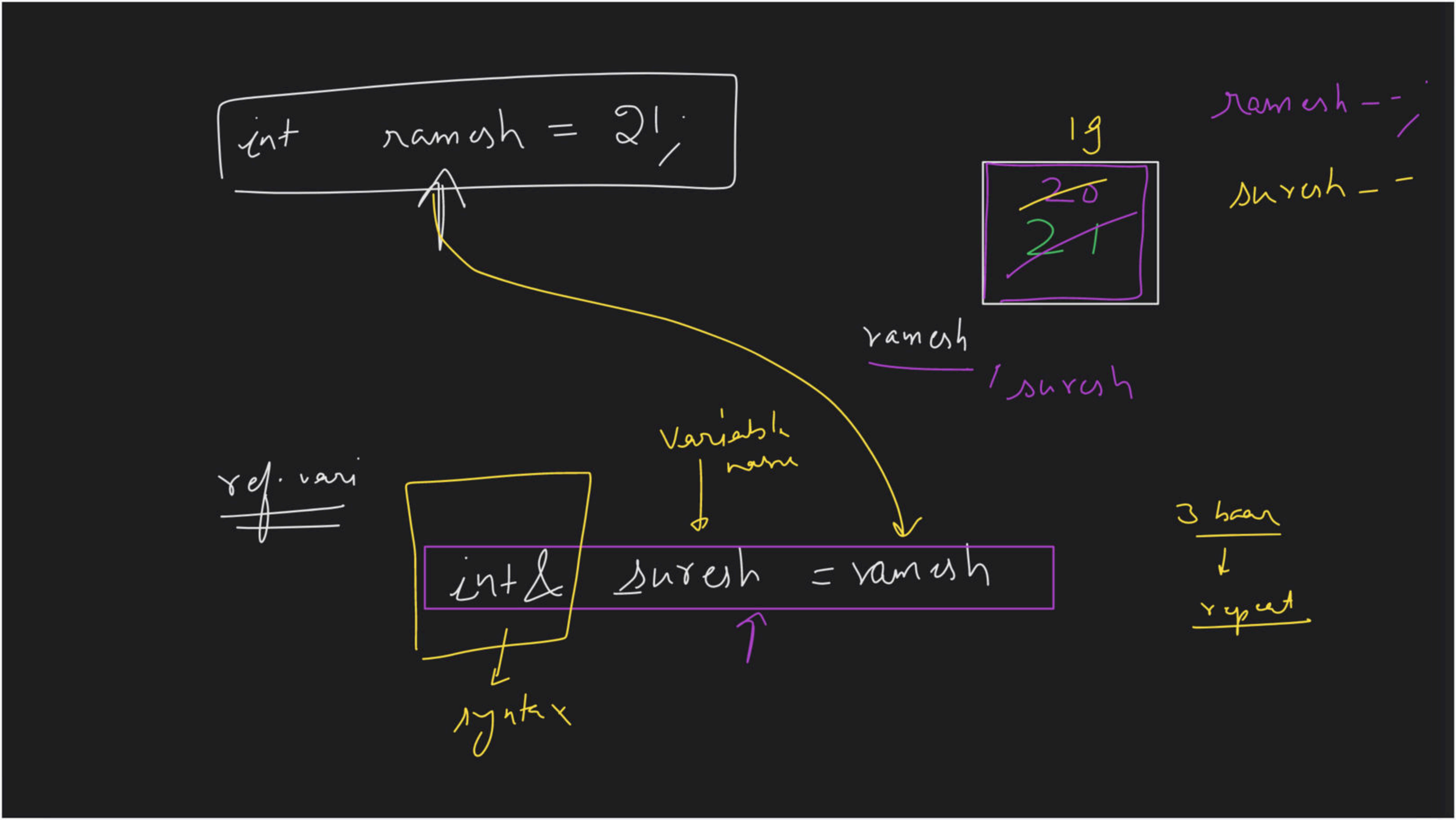


i++

j++

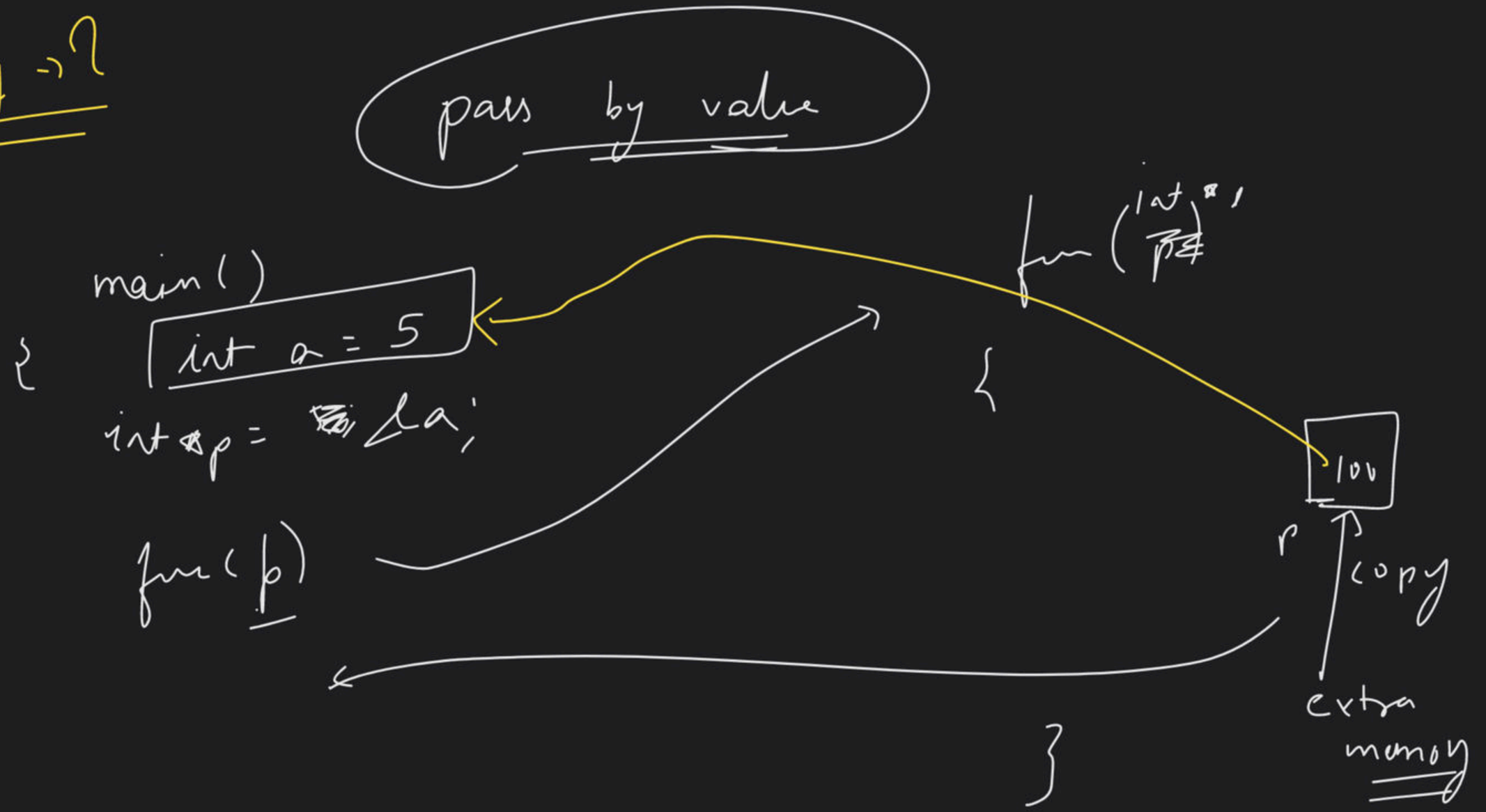


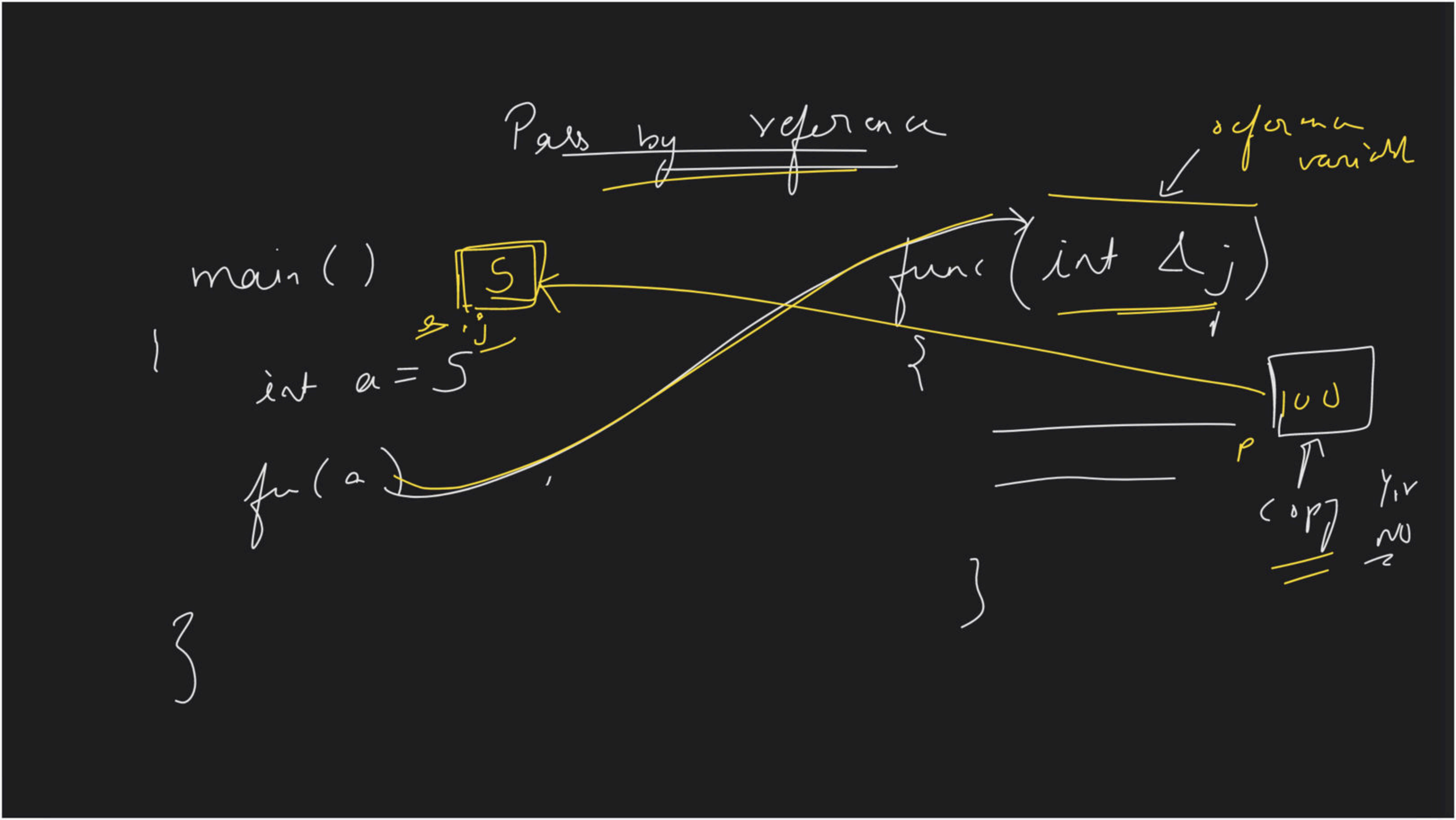


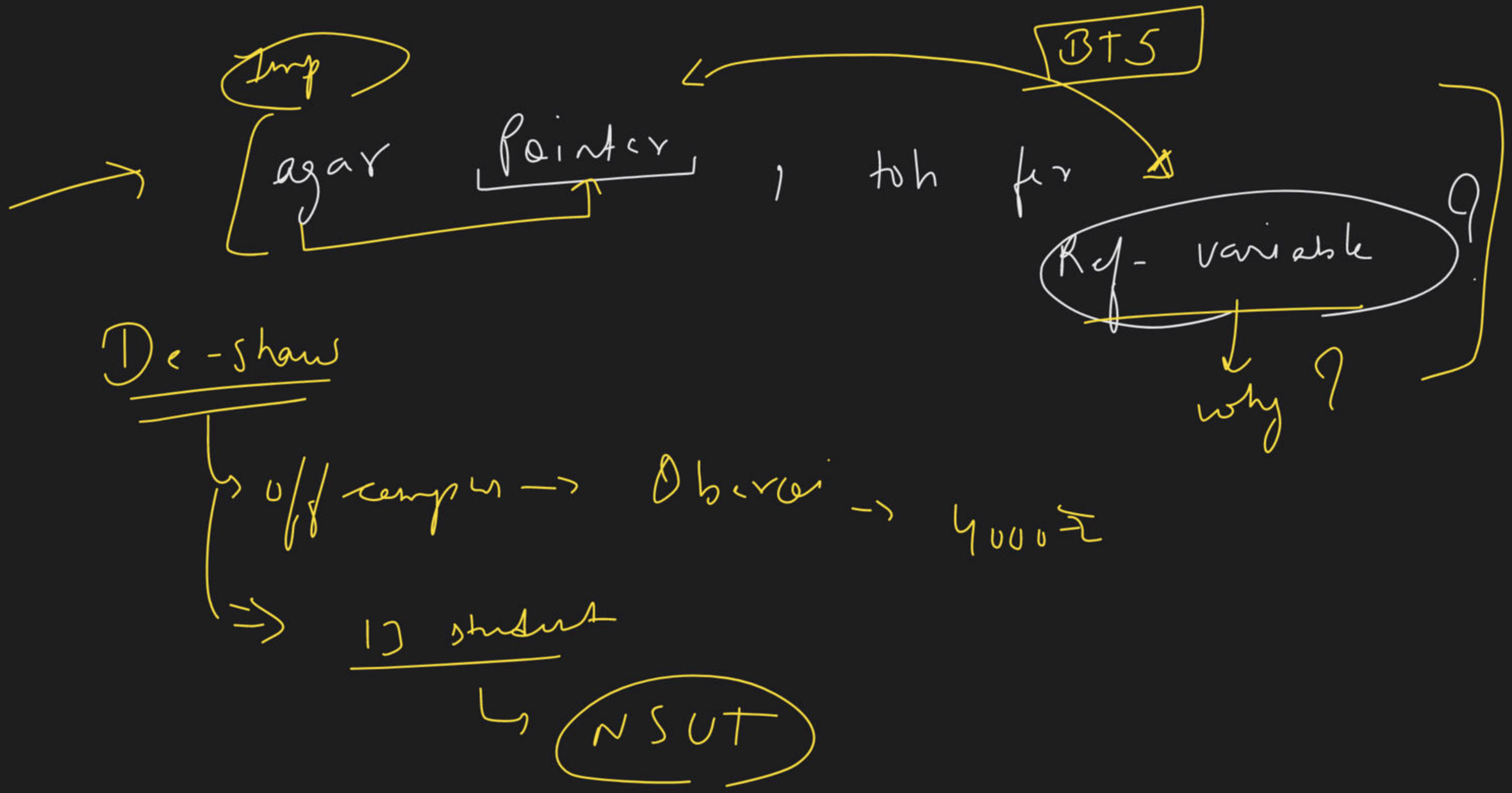

int ramesh = 21;
ramesh--;
suresh--
1g
20
21
ramesh
suresh
ref. vari
Variable name
int& suresh = ramesh
syntax
3 baar
repeat

why ->?

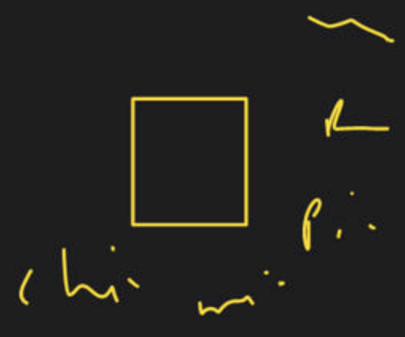
pass by value

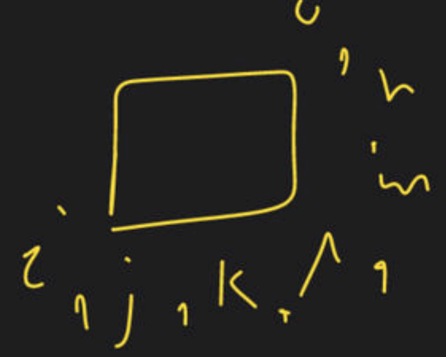


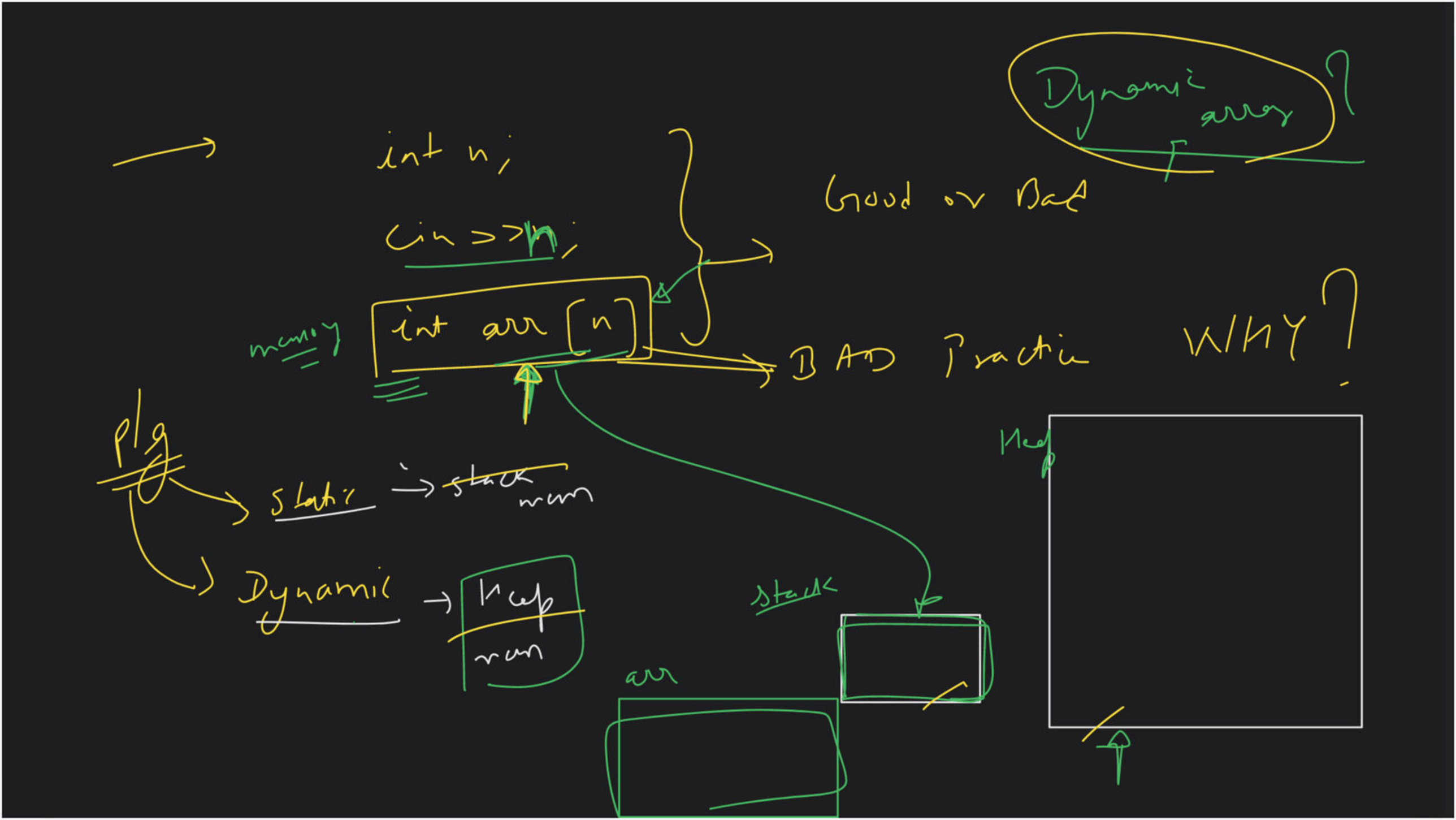

Pass by reference
reference variable
main()
{
int a = 5;
fun(a);
}
5
a
j
func(int &j)
{
j = 100;
}
p
copy
Yes
No


Imp
BTS
agar
Pointer
, toh for
Ref- variable
?
why ?
① De-shaw
↳ o/f campus → Observer → Yuuu ₹
⇒ IIJ student
↳ NSUT

→ what is ref. variable ?



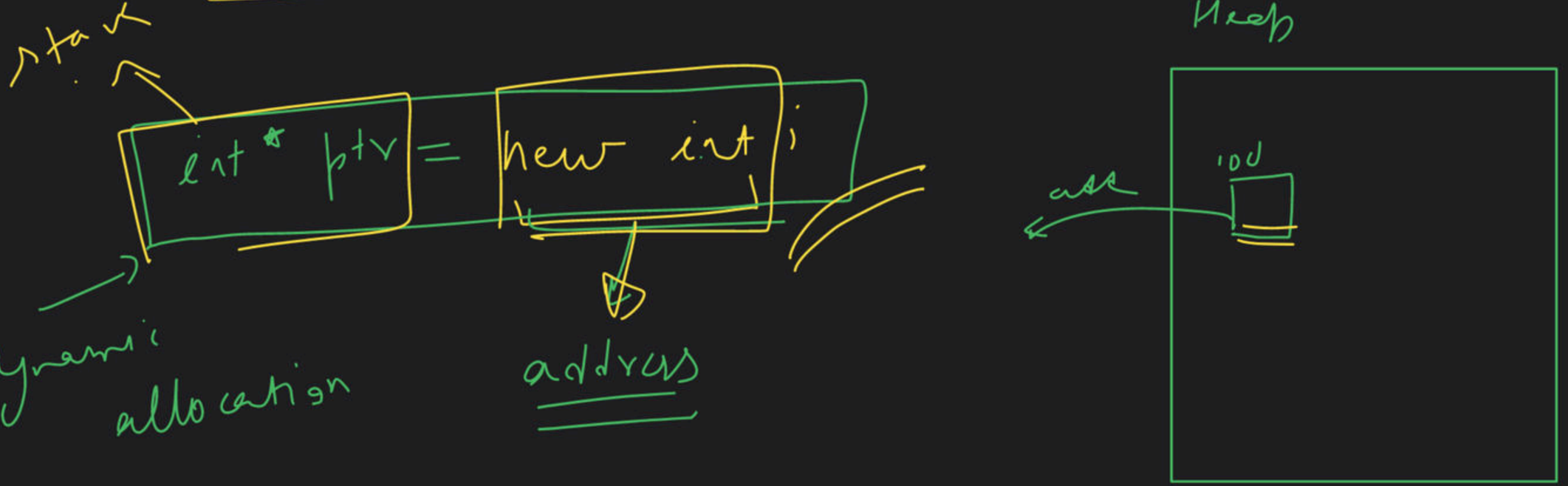


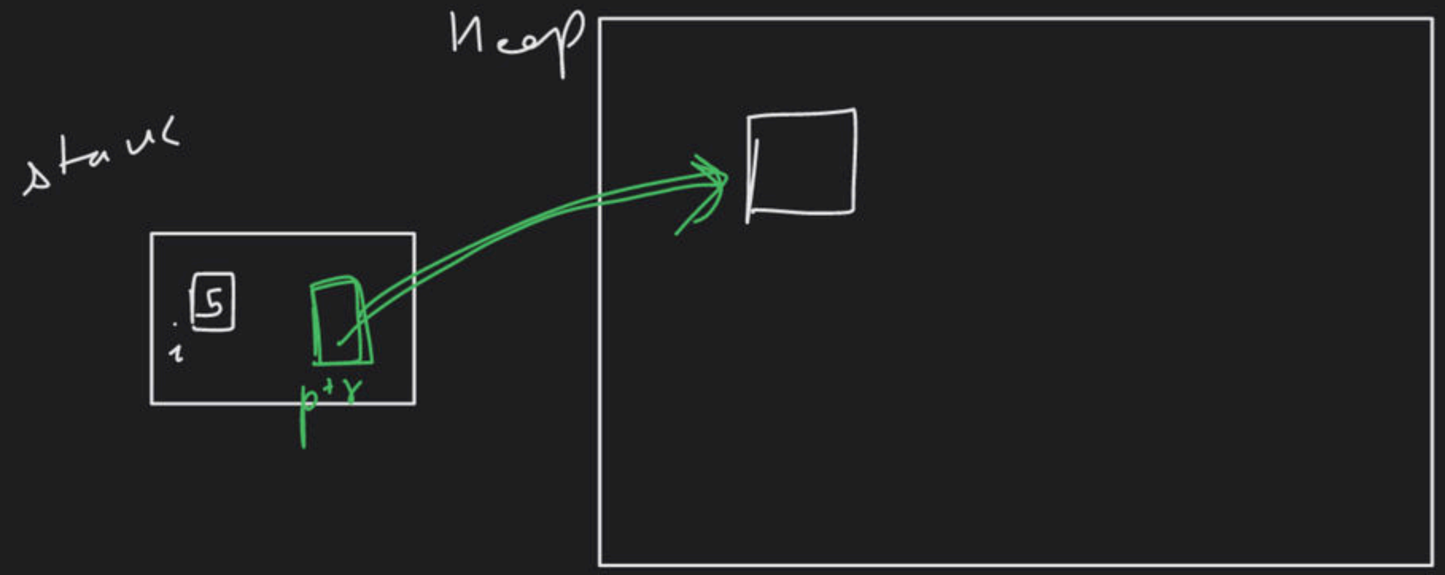

Dynamic array ?
int n;
cin >> n;
Good or Bad
memory
int arr [n]
BAD Practice
WHY ?
Heap
p/g
Static → stack
num
Dynamic → Heap
num
stack
arr

→ Dynamic Memory Allocation

"new"

`int i = 5` → static allocation



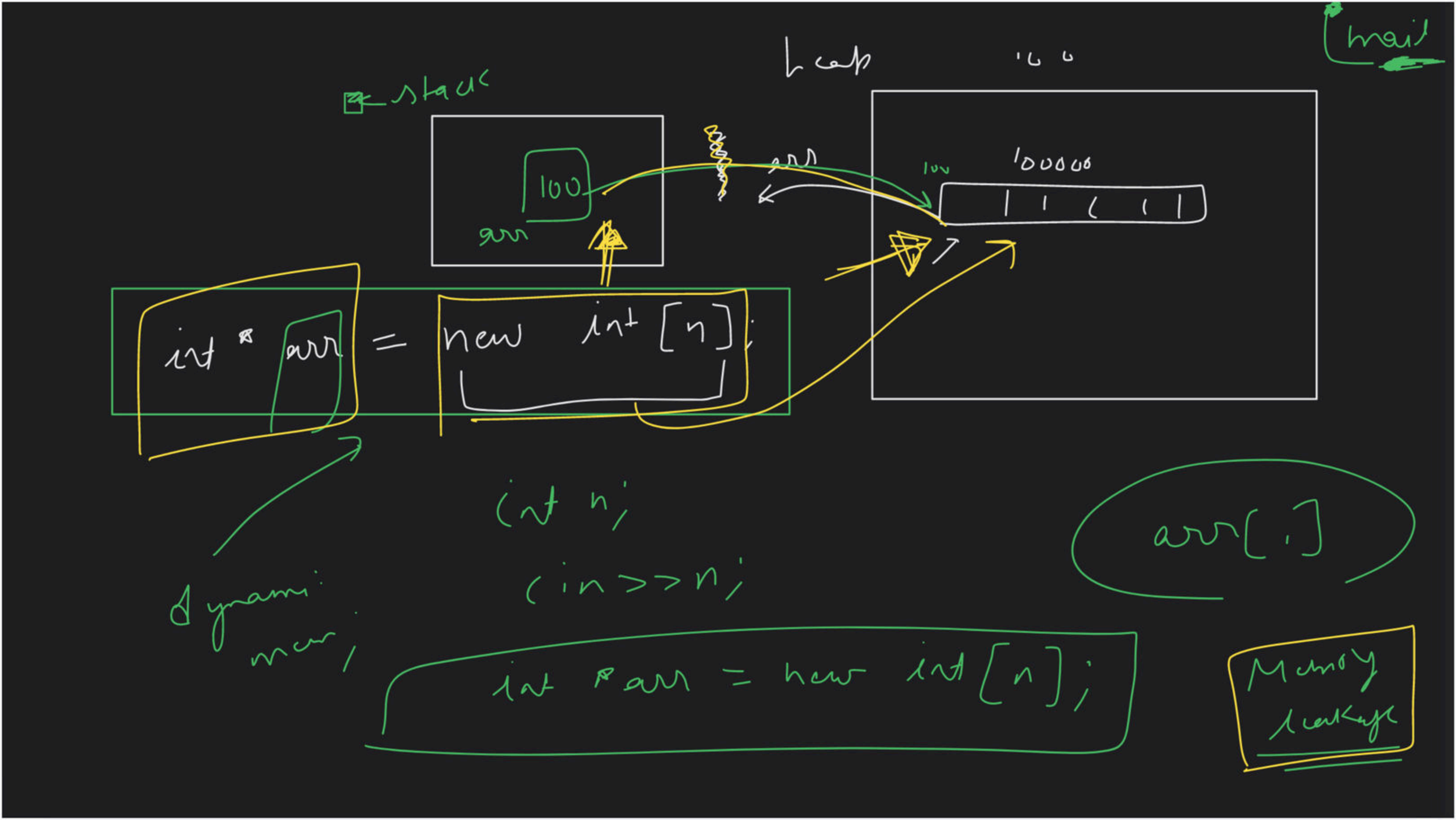



```
int i = 5
```
→ static memory allocation

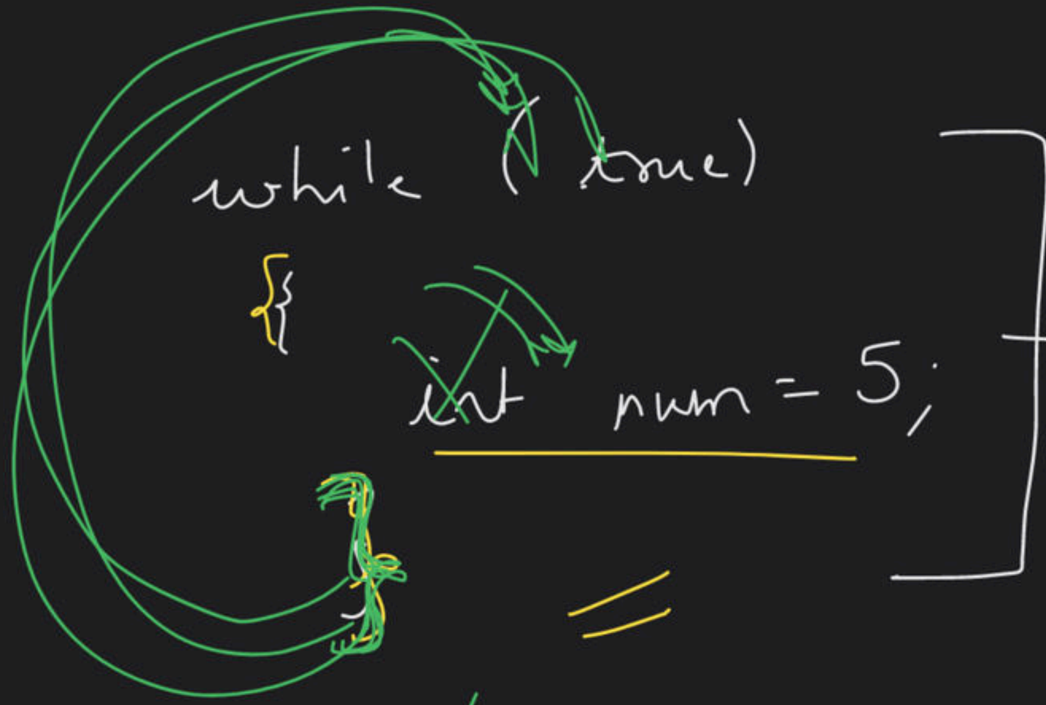
```
int *ptr = new int;
```
→ Dynamic mem allocat

stack

heap


main
stack
heap
100
arr
100
100000
int * arr = new int [n];
int n;
cin >> n;
dynamic memory
arr[i]
int *arr = new int [n];
Memory leakage

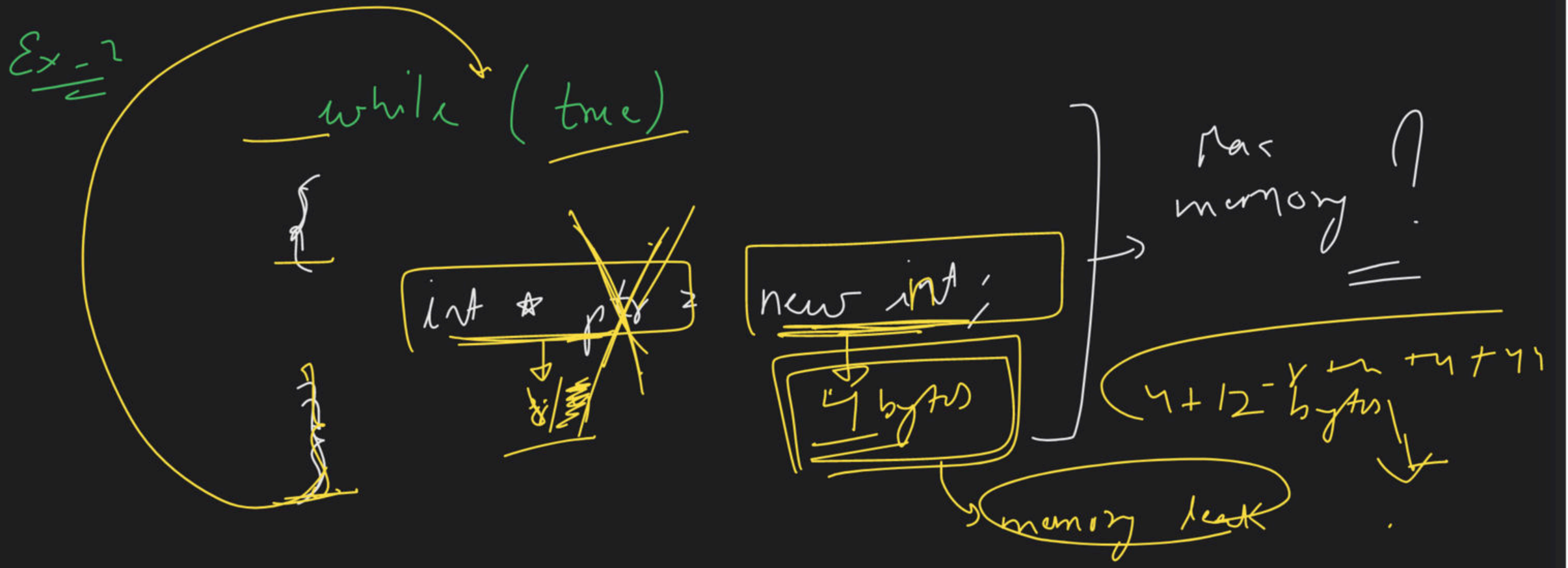
Ex - 1



→ max memory used ?

= 4 byte

infinite - ?
↳ 50 %



static memory allocation → Yes or No

-> stack
   -> automatically

-> heap
   -> release
      -> manually
         -> How -> ?

(1) -> while (true)
(2) {
(3)   int i = 5;
(4) }

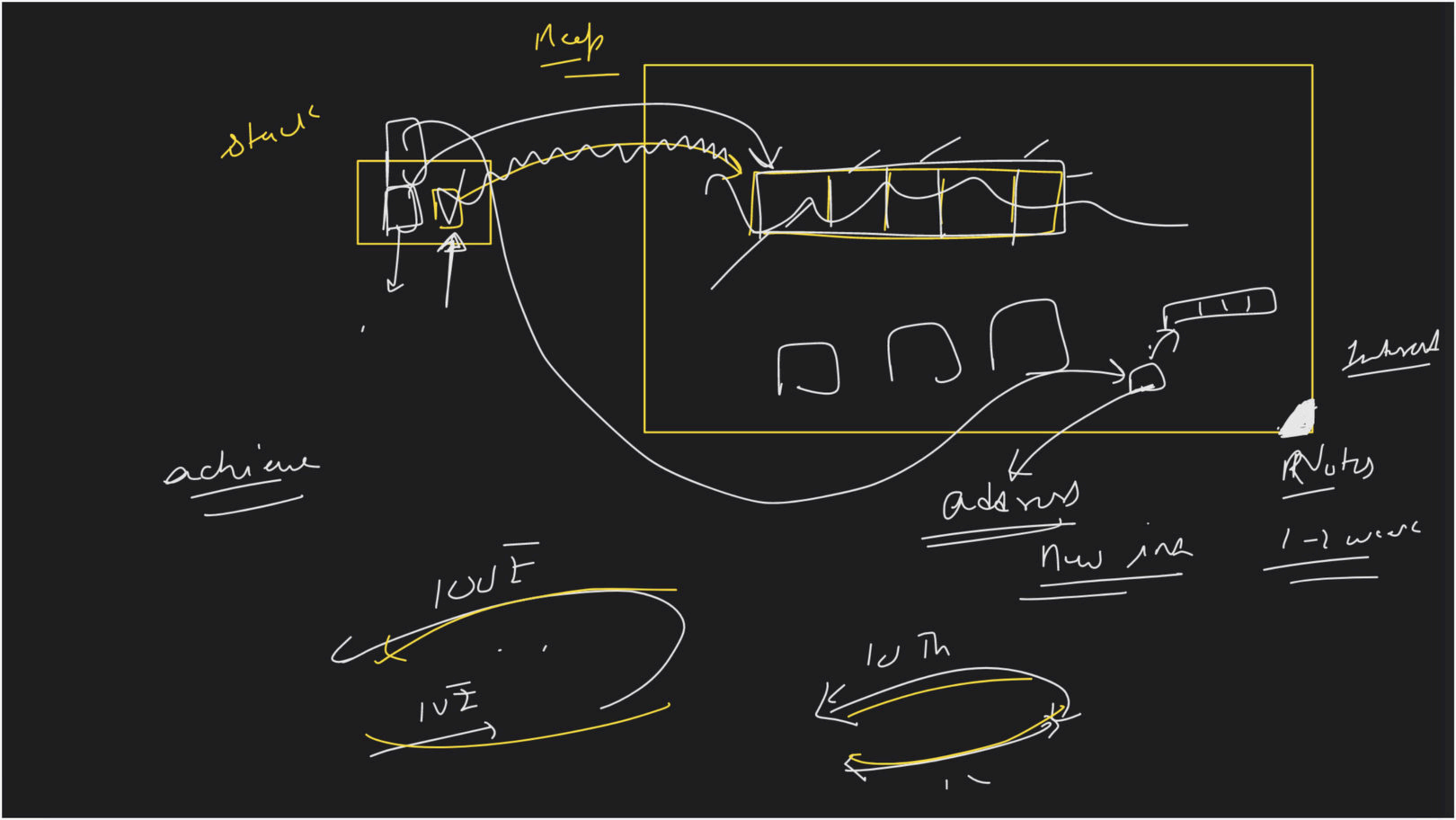
static allocation X

"delete" Keyword


```
int * ptr = new int;
```


```
int * arr = new int[5];
```

arr[]


Heap
Stack
achieve
address
New ins
Routes
1 - 2 more

→ Void pointer →

void *

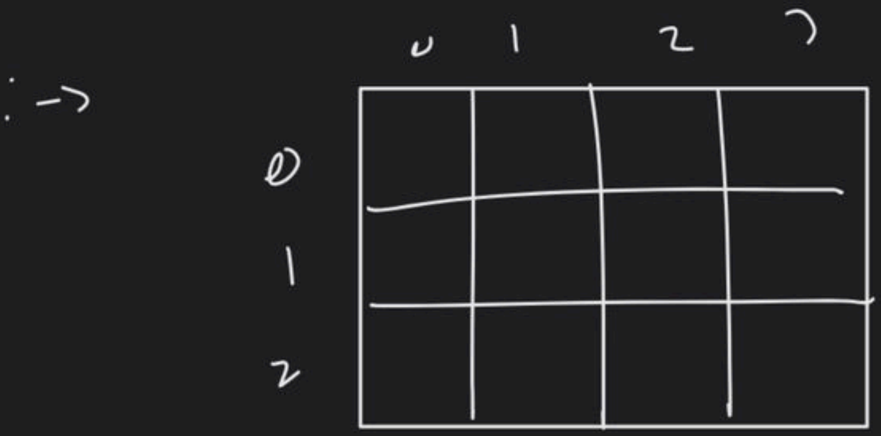
→ explore → 2 min

↳ any data type

why - ? → generic

what - ?

→ 2 D array :->

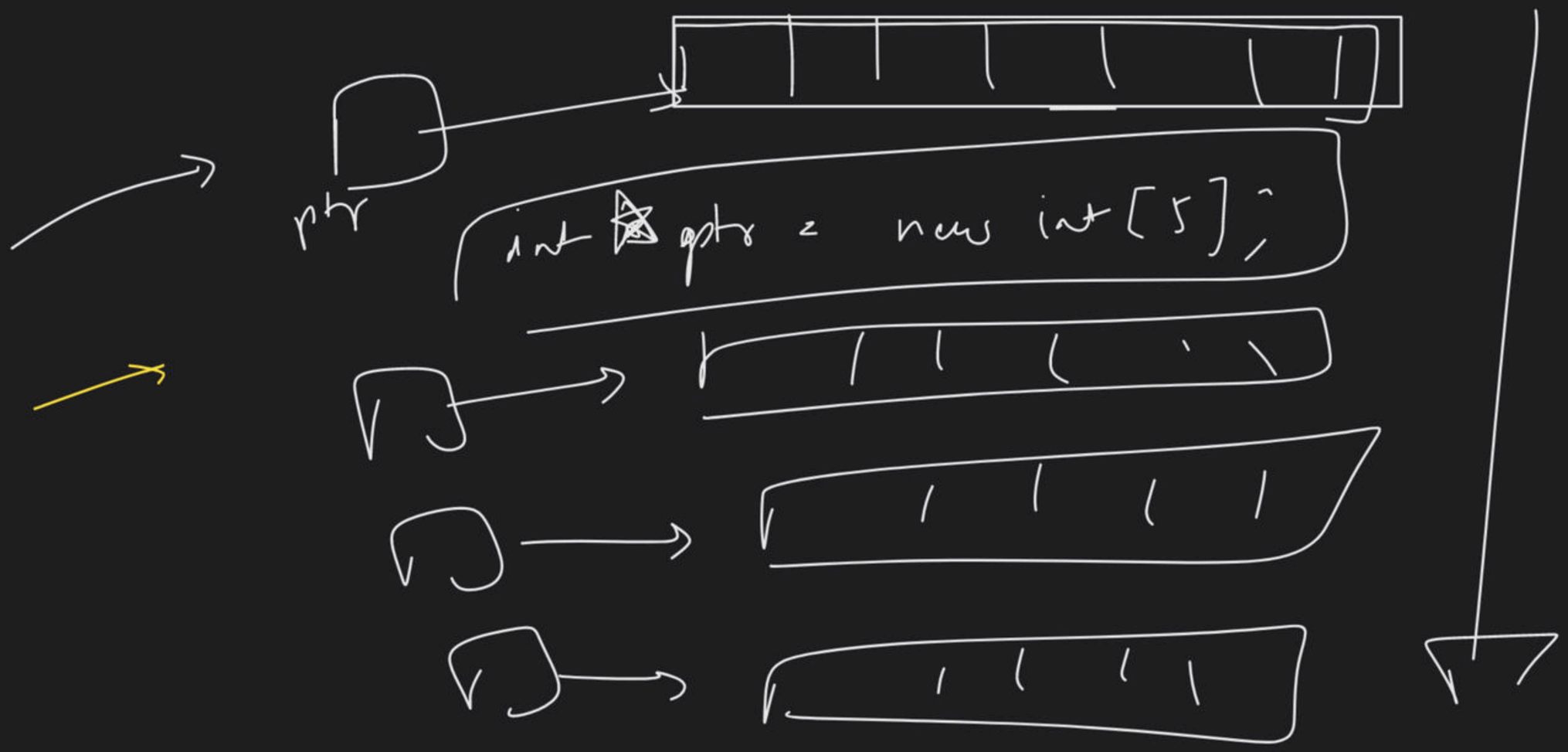


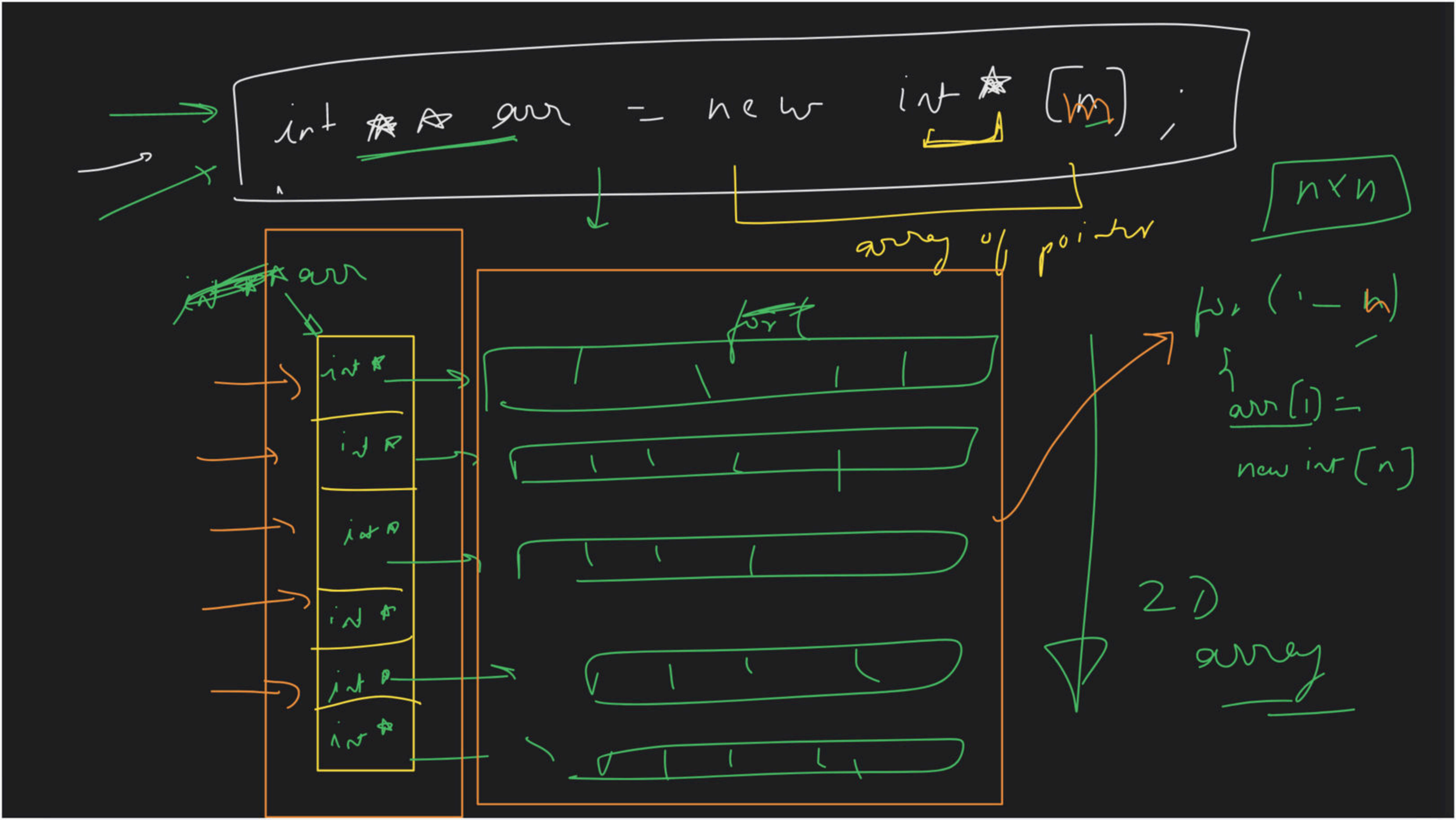
arr [3] [4];

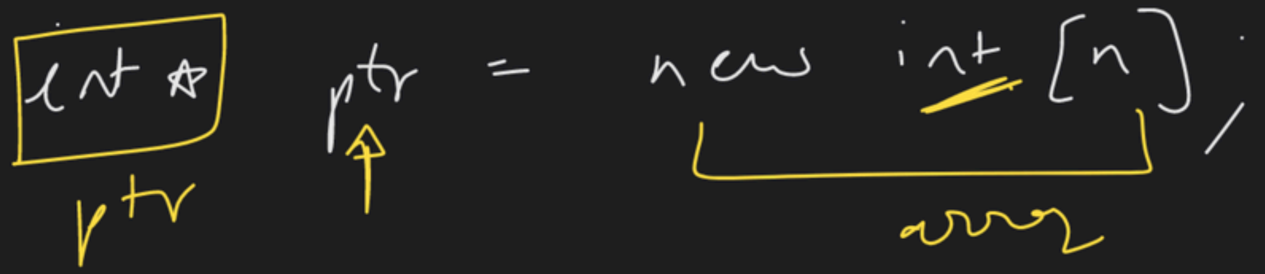
arr[m][n] , arr[n][n] -> Good or Bad ?

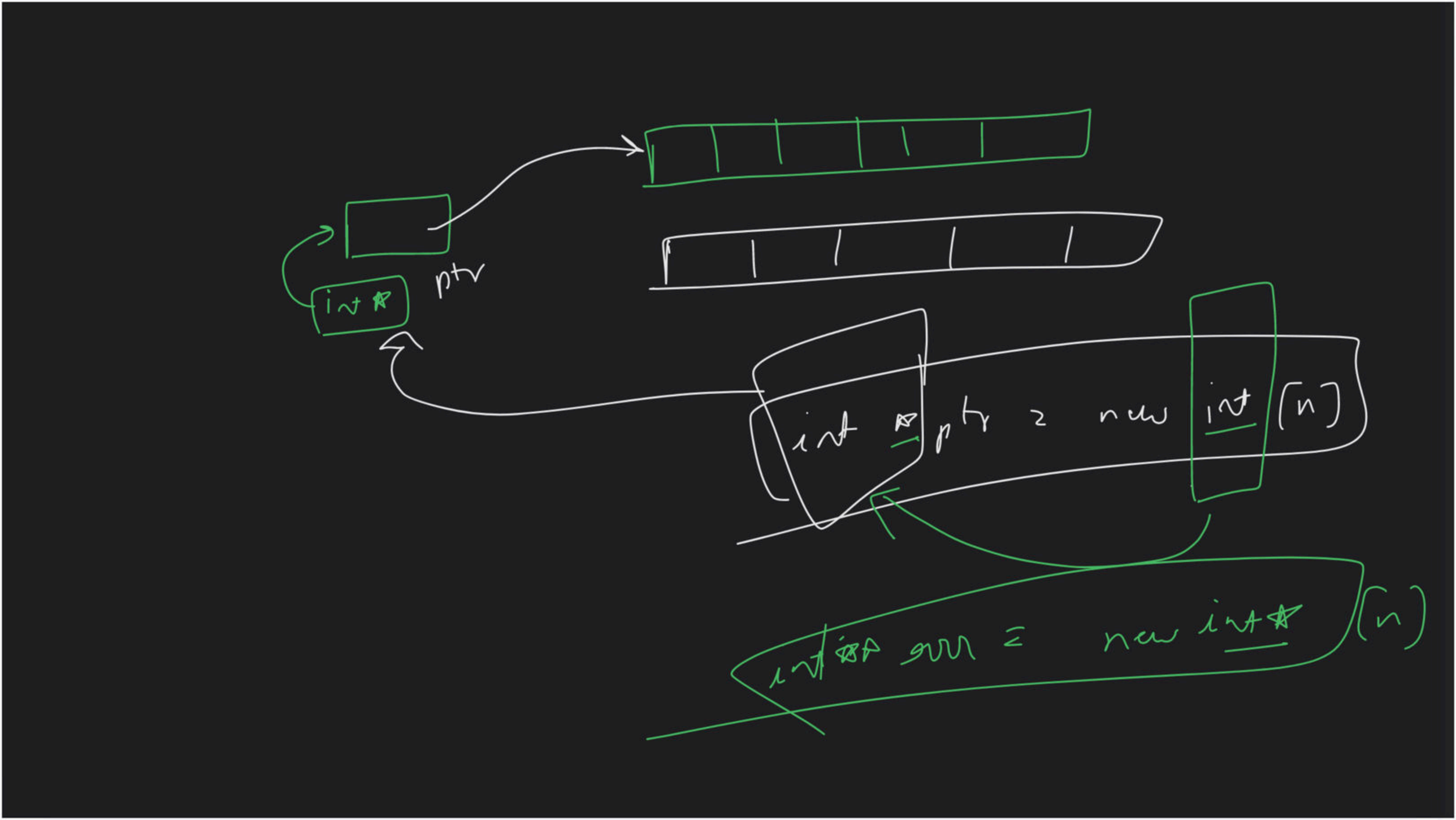
→ 2D Array → Dynamic Allocation

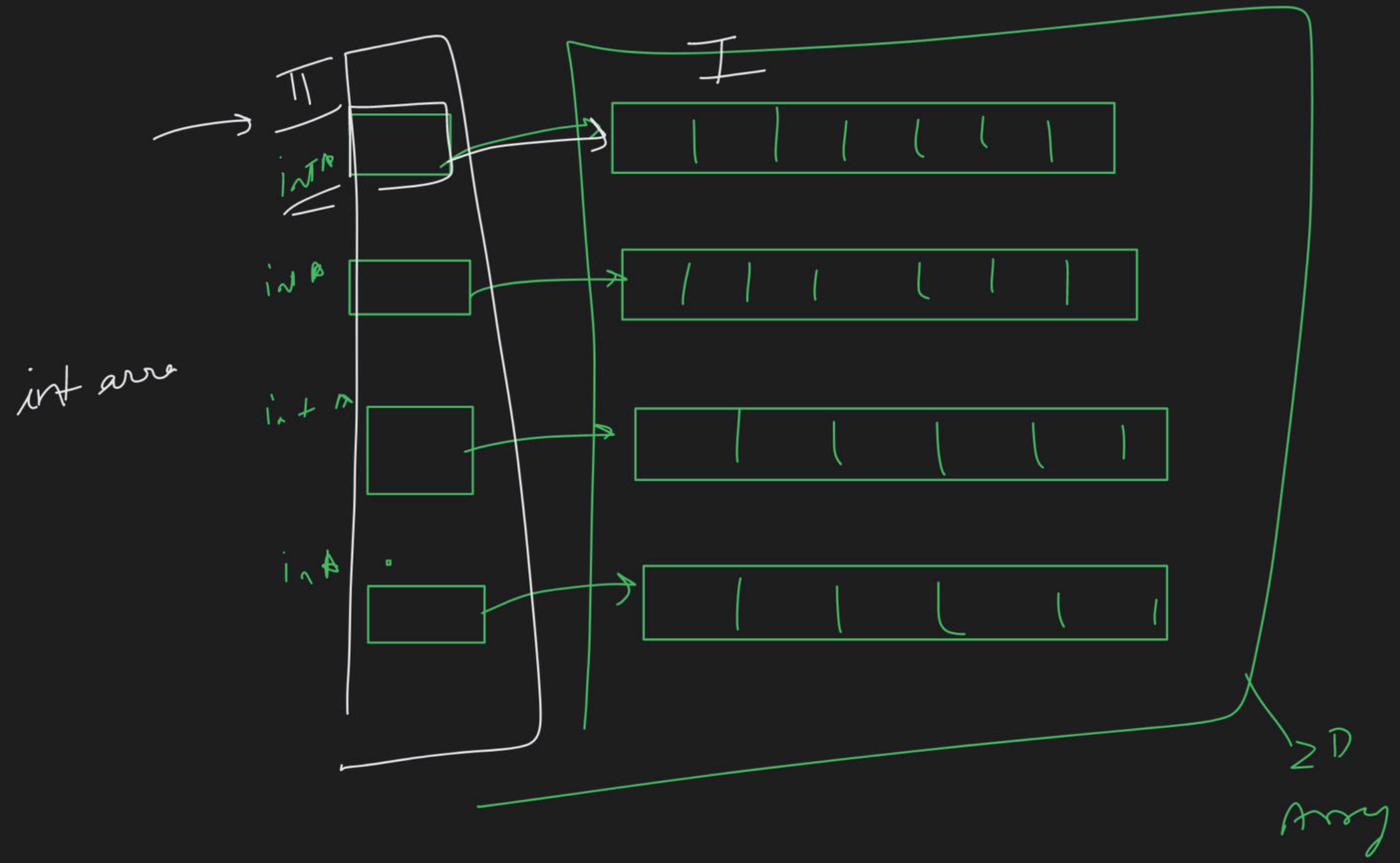
2 min

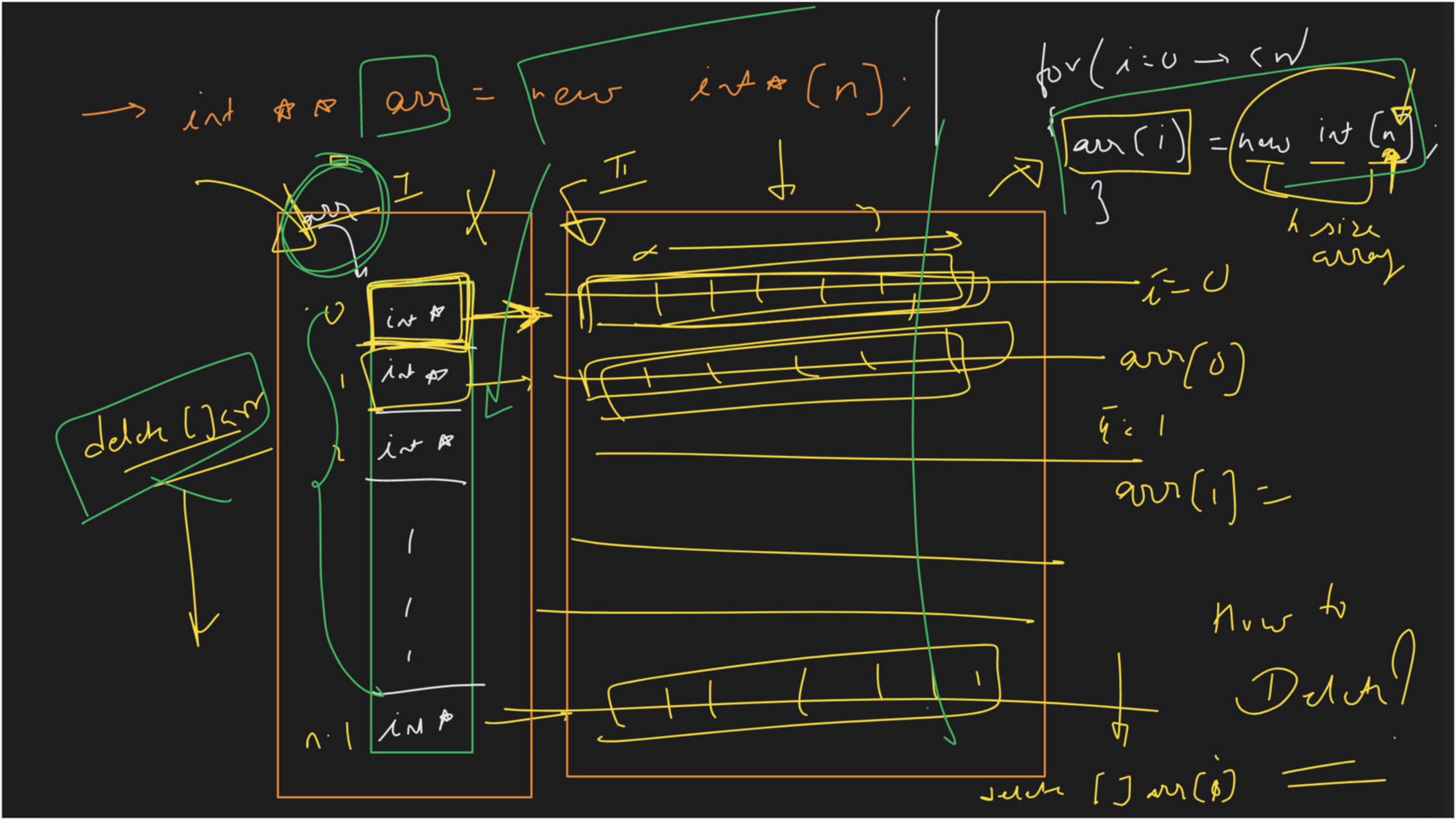



int ** arr = new int*[m];
array of pointer
n x n
arr
int*
int*
int*
int*
int*
int*
for (i = 0 ; i < m ; i++)
{
arr[i] = new int[n]
2D array


int *
ptr
ptr = new int [n];
array


int *
ptr
int * ptr = new int [n]
int ** arr = new int* [n]

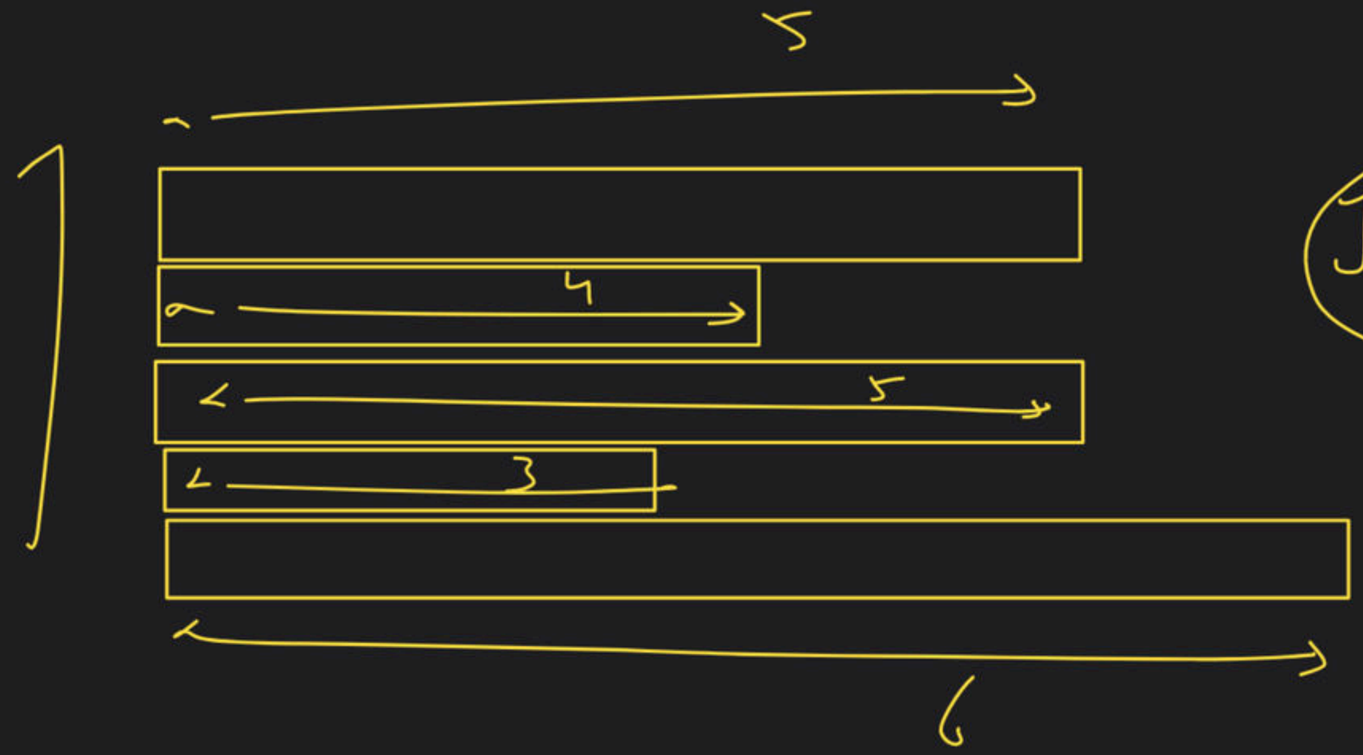

II
I
int arr
2D
Array


int ** arr = new int*[n];
for(i=0 → <n)
arr[i] = new int[n];
n size array
delete[]arr
int*
int*
int*
int*
n-1
i=0
arr[0]
i=1
arr[1] =
How to
Delete?
delete[] arr[0]


delete [] ptri
ptr

size

5
7
5
3
6



Jagged array

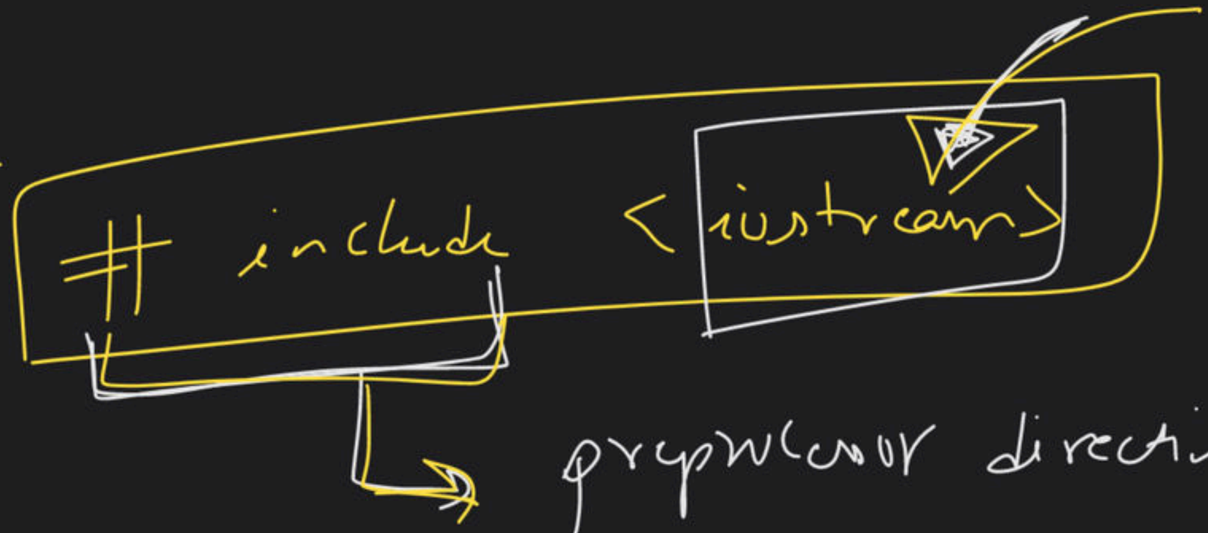
```
int** arr2 = new int*[n];

for(int i->0 ; <n )
{
    arr[i] = new int [size[i]];
}
```

simple ?

Repeat

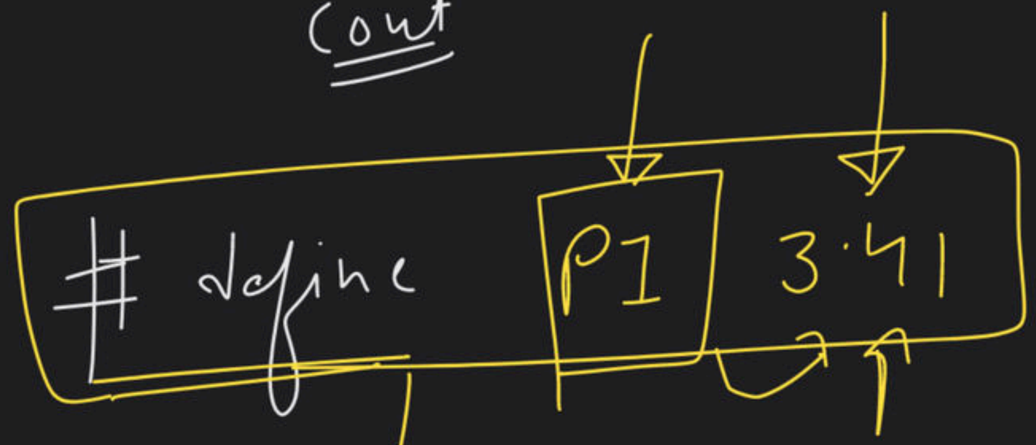
UUID

→ Macro :-



Directive :-

cout



replacement

int area

= PI x r x r

3.141

space

a piece of code

in p/g that

is replaced by

value of ——

3.41

PI = 3.14

double pi = 3.14 3.41

1. space ley raha hai

double pi = 3.41

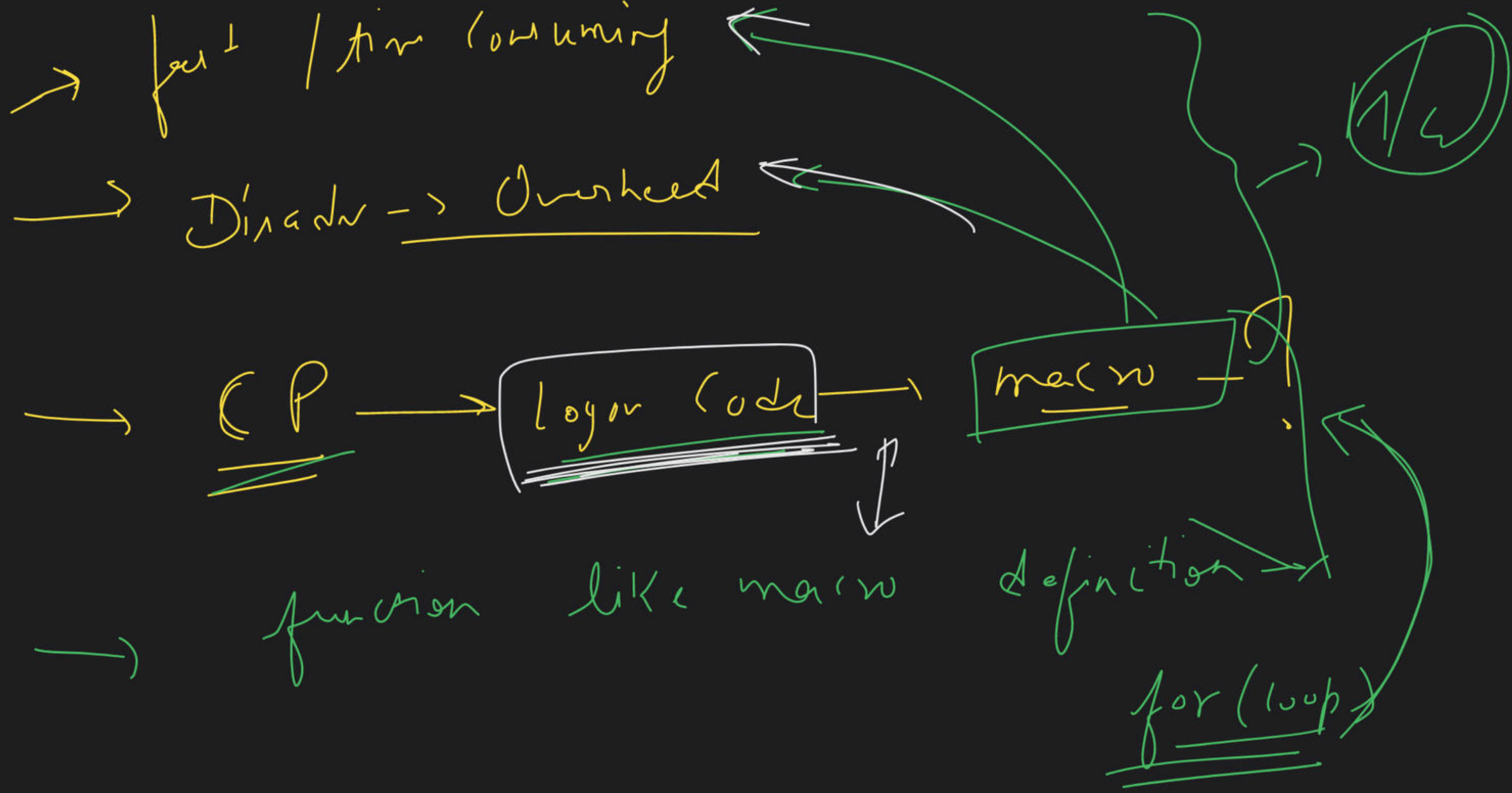
100

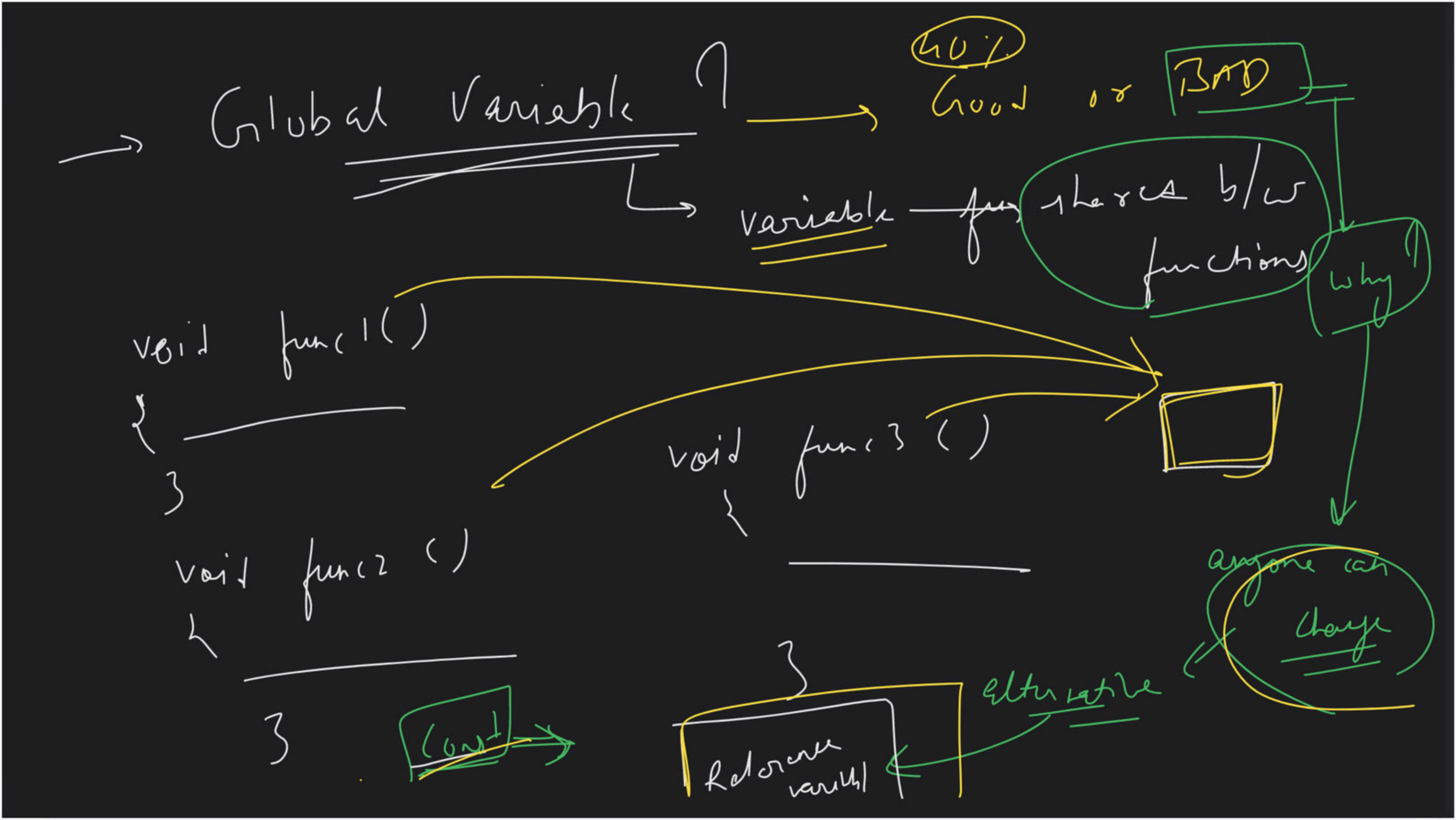
3.14 or 3.41

PI

3.41

pi 3.41 → yes or No


fast / time consuming
Disadv -> Overhead
CP
Logic Code
macro
function like macro definition
for (loop)


Global Variable ?
Good or Bad
Variable — share b/w functions
Why ?
void func1()
{
}
void func2 ()
{
}
void func3 ()
{
}
Anyone can change
Alternative
Reference variable
Copy

→ const → Constant Variable

final

```
const int a = 5;
```

```
a++;
```

→ Error

inline functions → ?

{ - 2 3

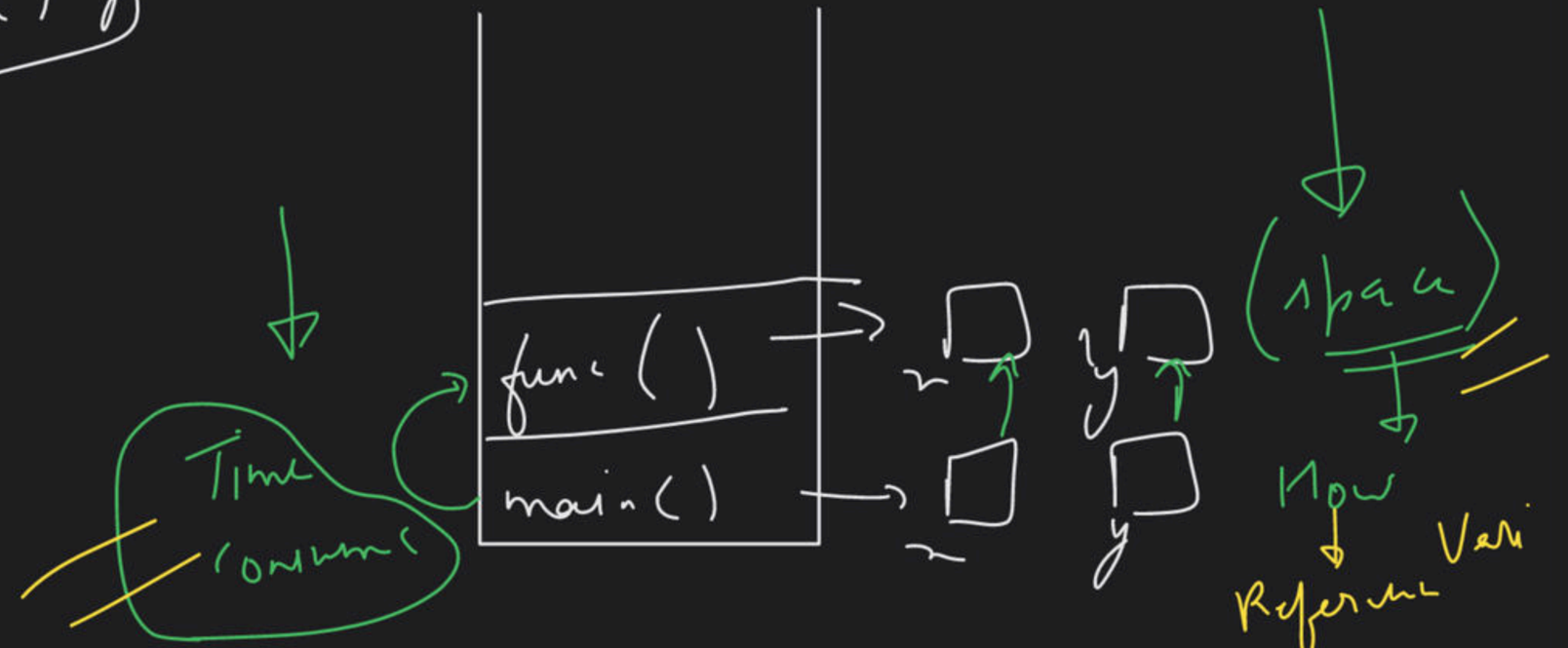
used to reduce function call overhead

main()

fun(x, y)

x, y

}



inline int func (int x, int y)

{

}

can not be applied on every fun

RM

Compiler

① line → inline allowed

2-3 line → Compiler

>3 line → [Straight NO]

inline → what → ?

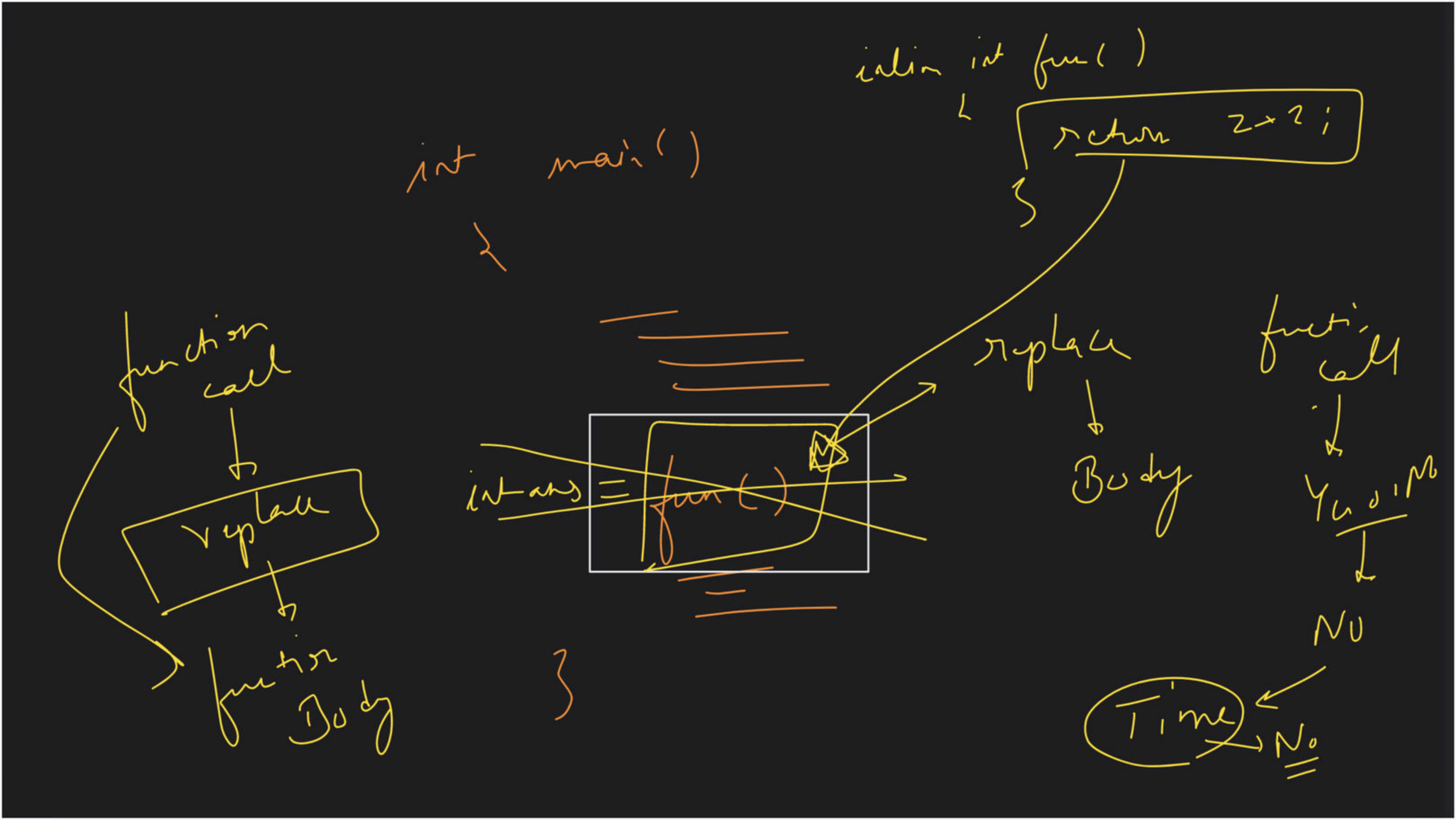
fun()

{

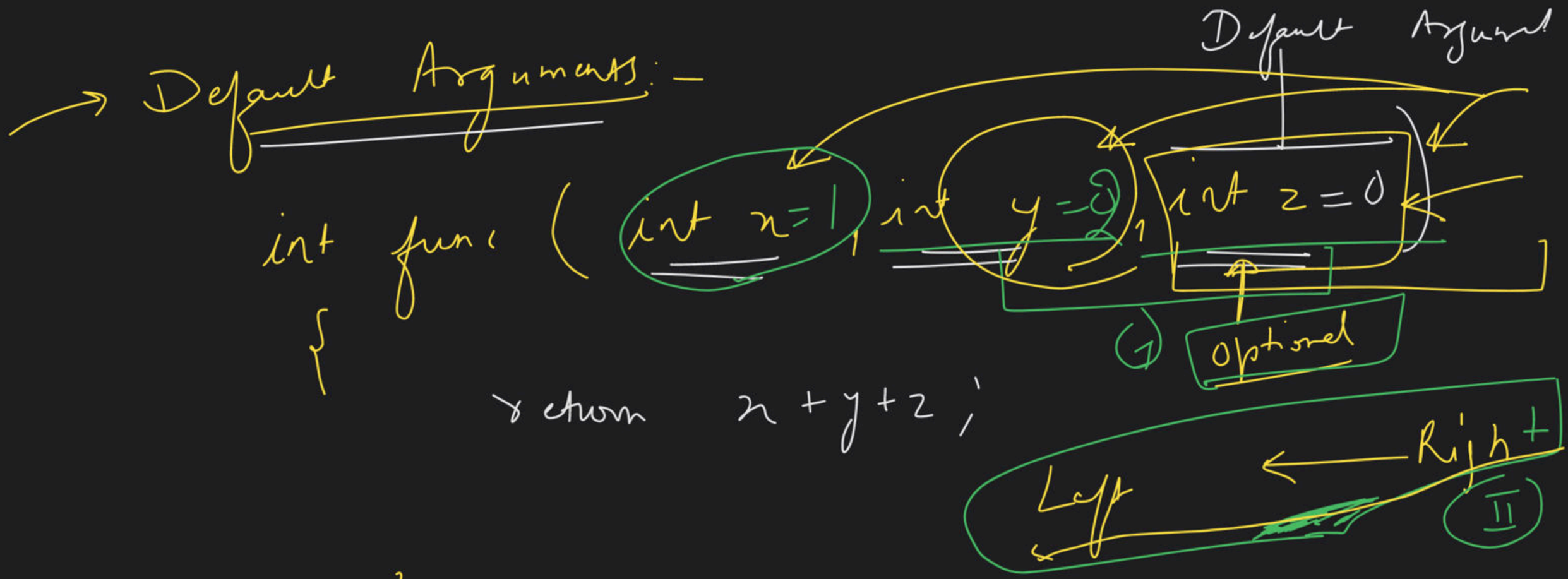
}

How ?

check


int main()
{
}
int fun()
{
return 2+2;
}
int ans = fun();
function call
Value
function Body
replace
Body
function call
Value
NU
Time
No

Default Arguments :-



write o/p of this p/f

→

```
int func ( int a = 1  int b  int c )
{
    return a+b+c;
}
```

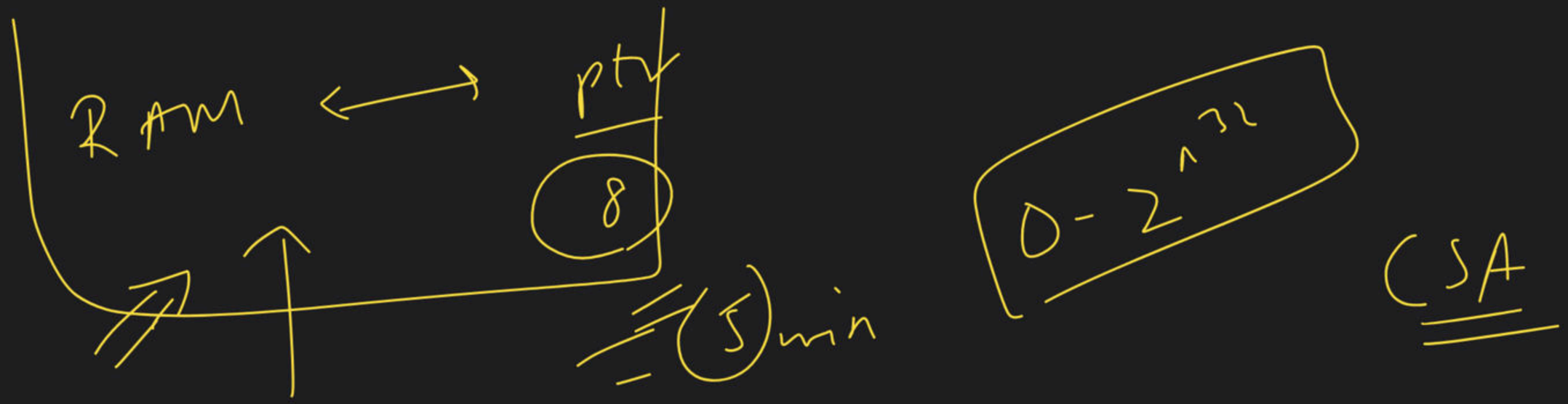
error

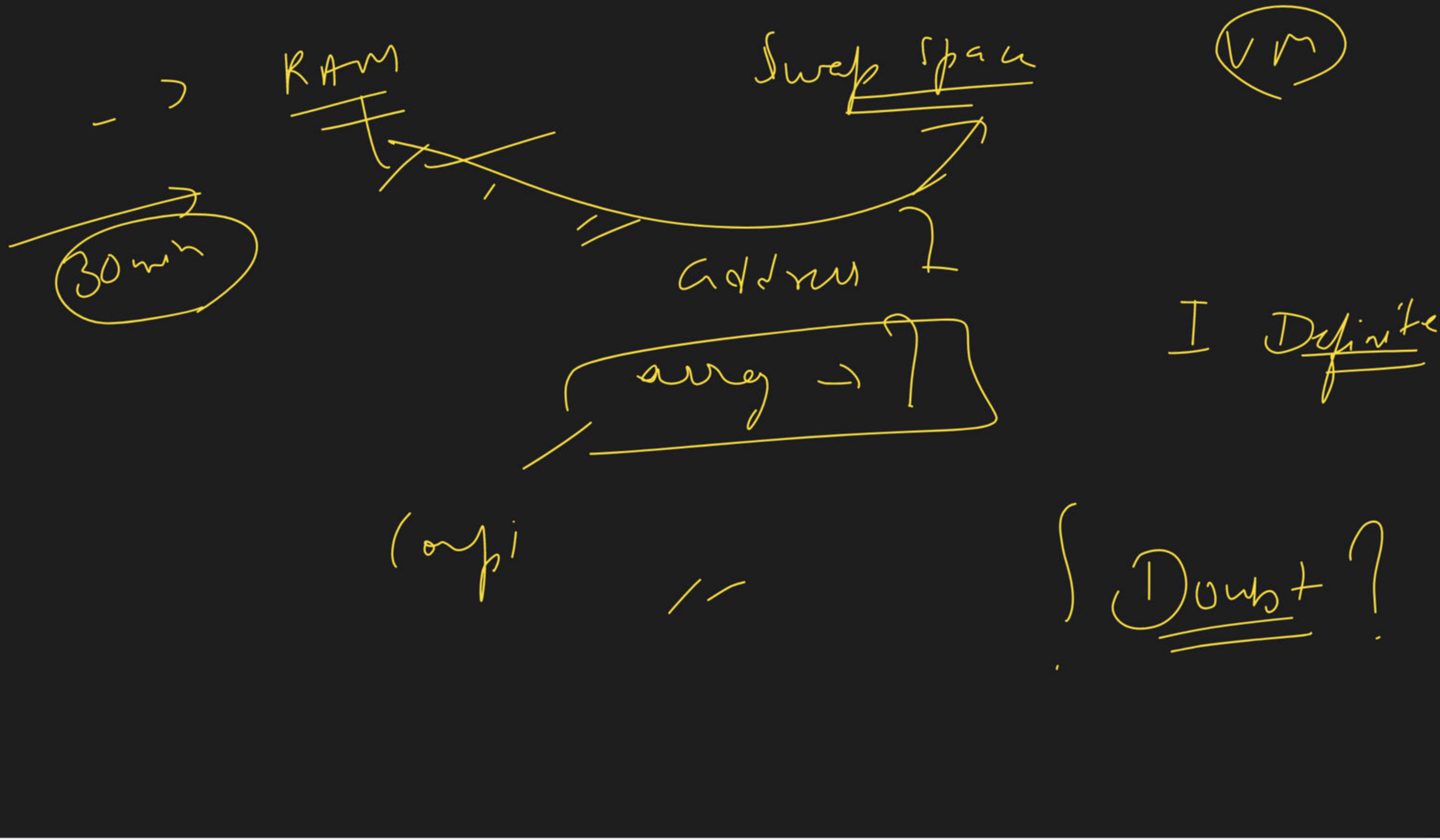
error

func (2 , 3)

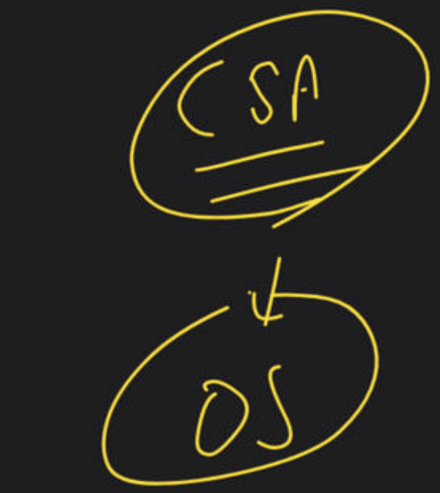
out → Error

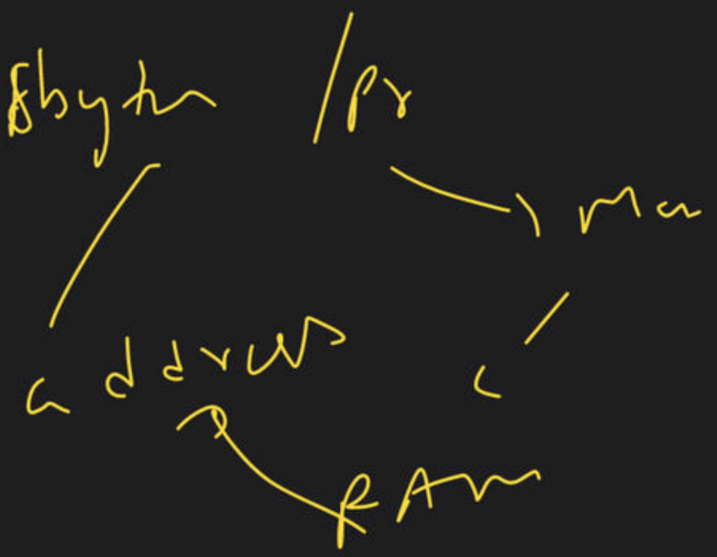
why ← why ?


RAM
ptr
8
5 min
0 - 2^32
CSA


RAM
Swap Space
VM
30 mb
address
arr
I Definite
Compi
Doubt ?

→ Doubt





Non-stop