



Functions & Arrays

Foundation Course on Data Structures & Algorithm - Part I

Questions:

Linear Search in an array

Reverse an Array

find max and min element in an array

Swap Alternates in an array

Sort an Array of 0, 1 and 2

Move all negative number to one side of array

Find union and intersection of 2 sorted arrays

Program to cyclically rotate an array by one

find duplicate in an array of $N+1$ integers

find the pair that sum to a given value

find the triplet that sum to a given value

Check whether an array is palindrome or not

Minimum swaps required bring elements less equal k together

Unique Number of Occurrences - Leetcode

Kadane's Algo

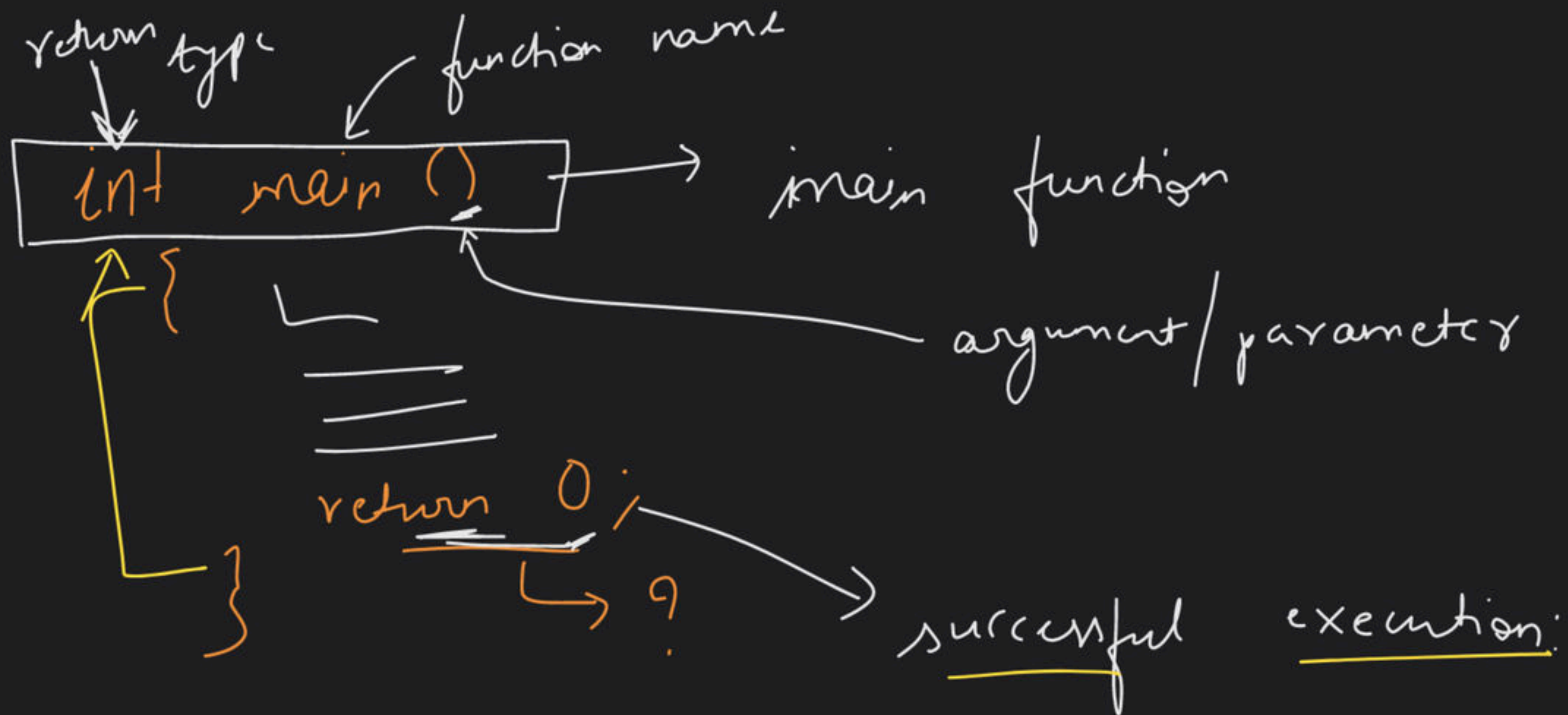
array
↳ creation
↳ input
↳ pointer
↳ ques
//

find max & min
CS
7/11

LB
//
→

→ Functions → what?

↳ Block of code → that performs some operation



Kuch return nahi krge
→ void xyz()

}

int return type function name (arg./parameter)

} return - ;

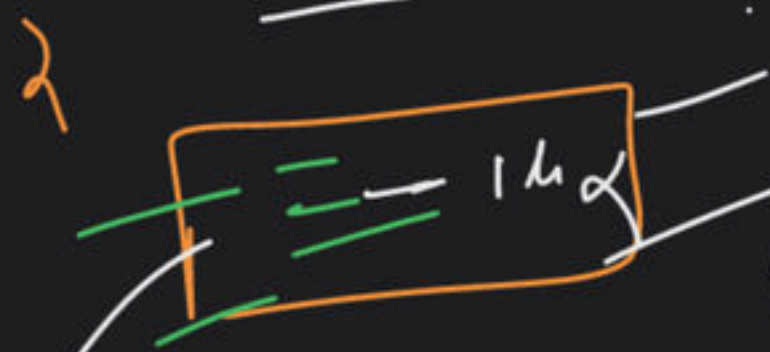
void xyz ()
{

~~return~~ ==
return; ==
}

why?



int main()



300 LOC



3

report =

readability =

reuse =

modular approach

save time

code clear

maintainability

easy to understand

testable

redundancy

→ Bulky

→ Buggy

→ Readability

→

```
input()
int main()
```

↓

input()

print()

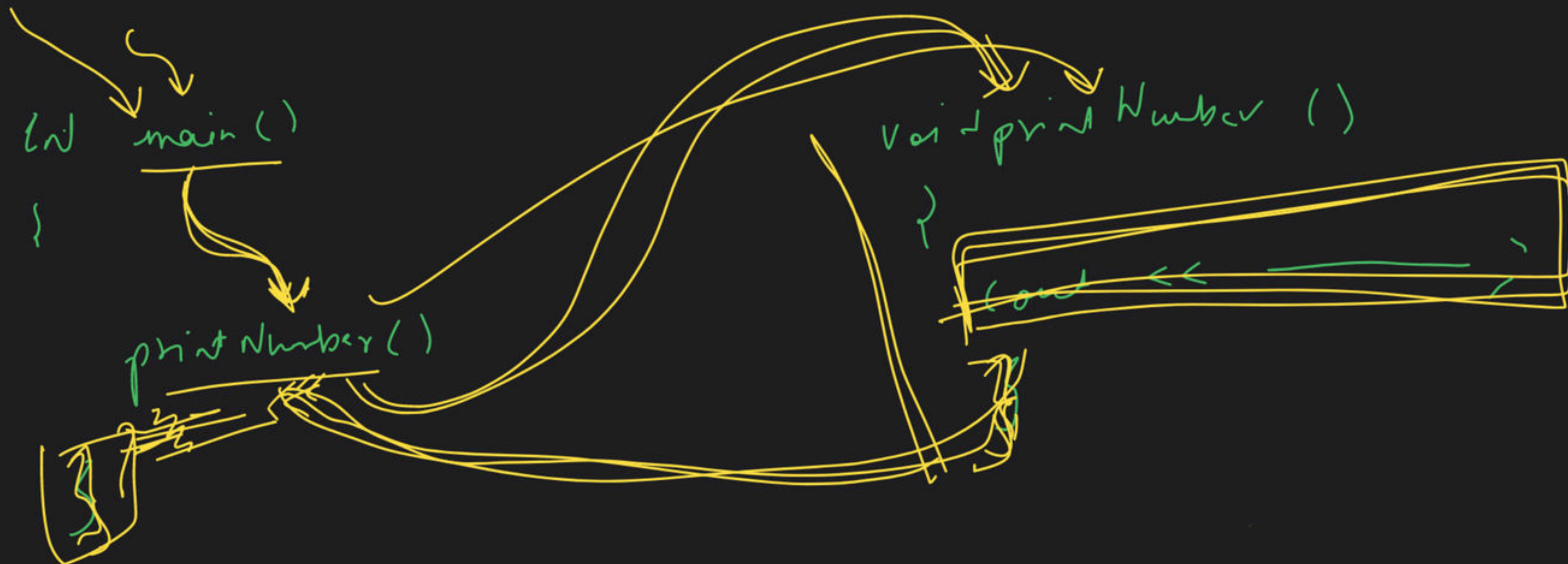
}

print()

}

}

Klaine — 9



O/p → ~~~~~

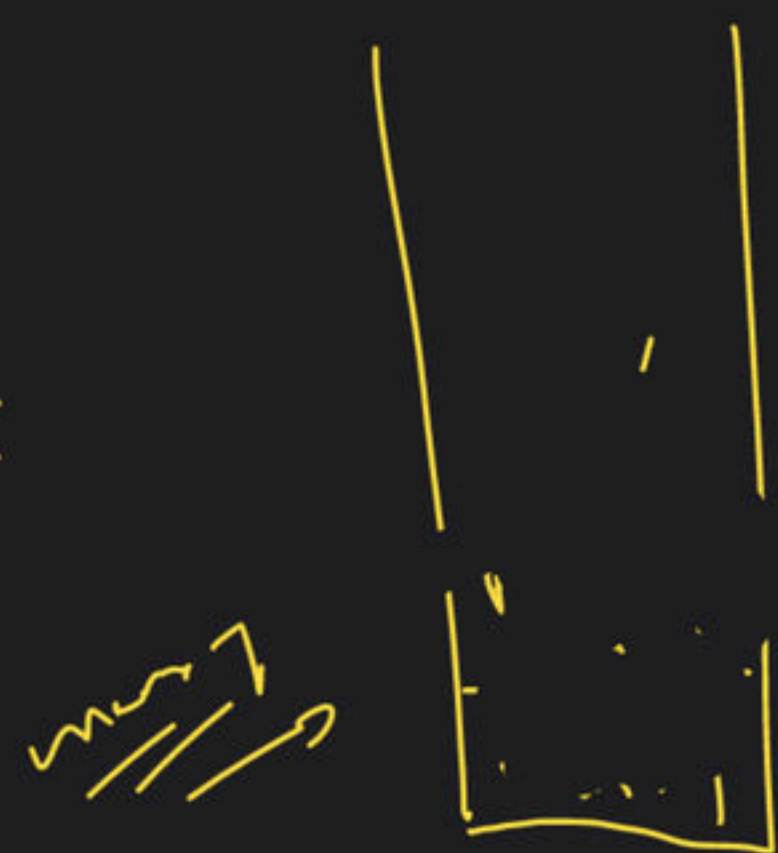
stack → ?

input → 1, 2, 3, 4

output → 4, 3, 2, 1

LIFO order

function call stack



main()

{

printNumber(num)

}

arg/param

arg

i/p param

void

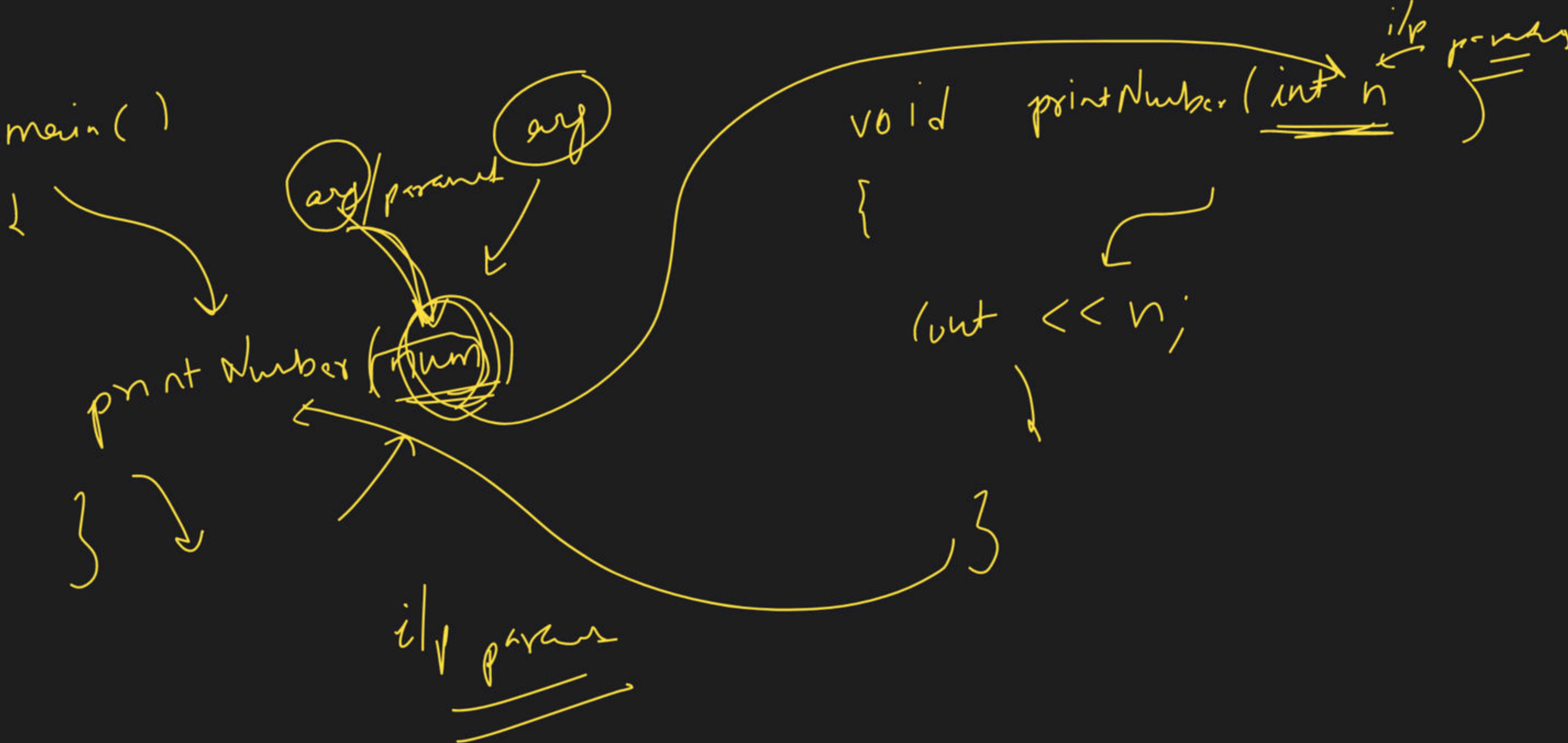
printNumber(int n)

i/p param

{

cout << n;

}



Pass by Value

main ()

{ int num = 5

print (num)

return 0;

}



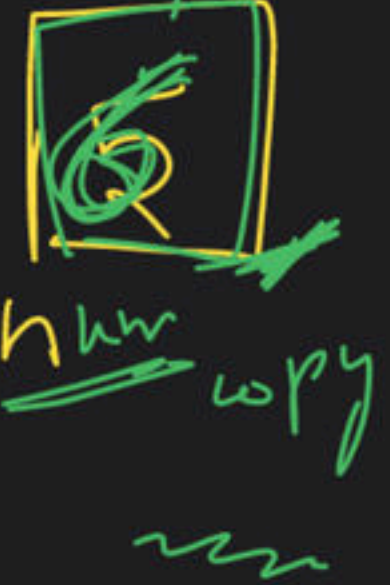
|||

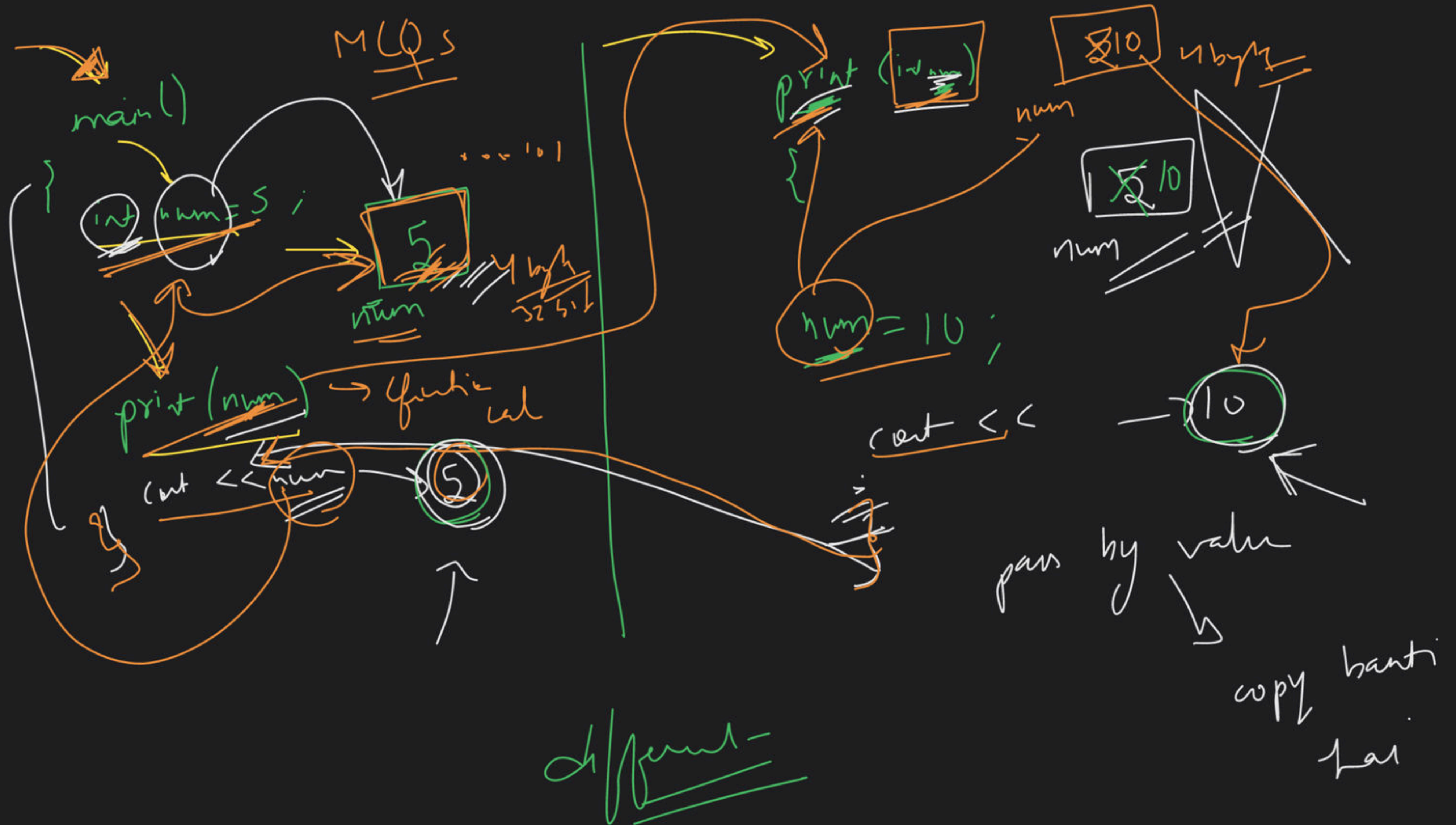
print (int num)

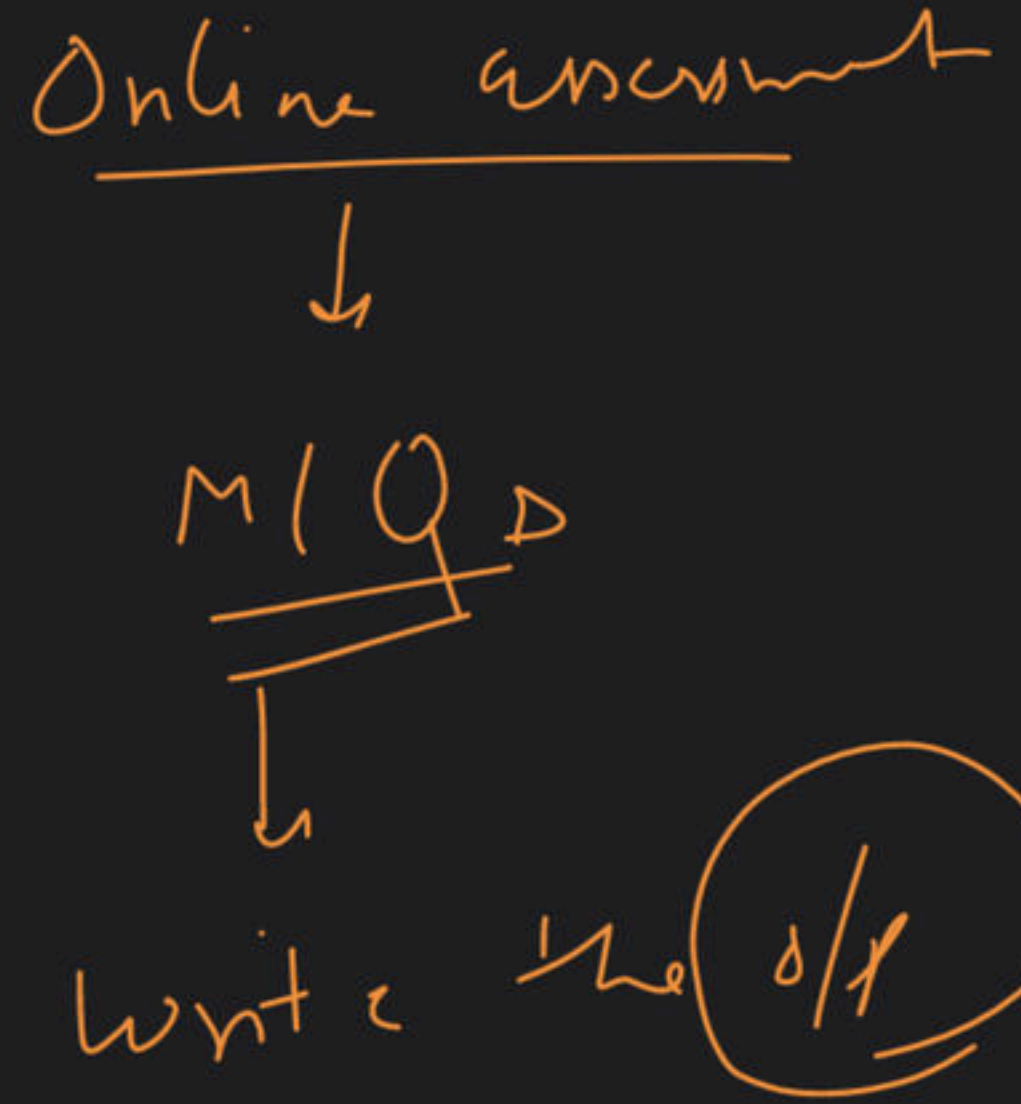
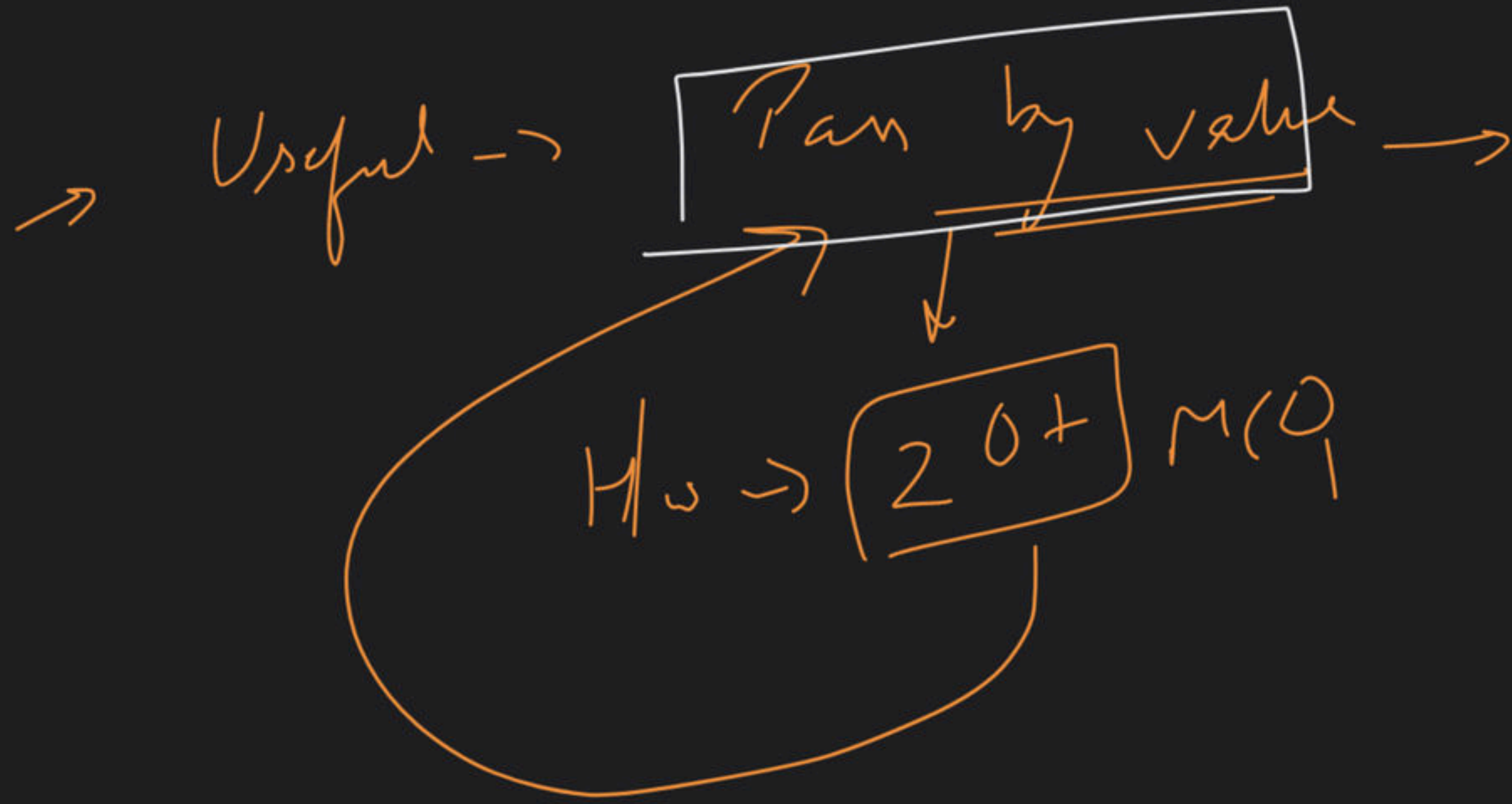
{

cout < - -
num++

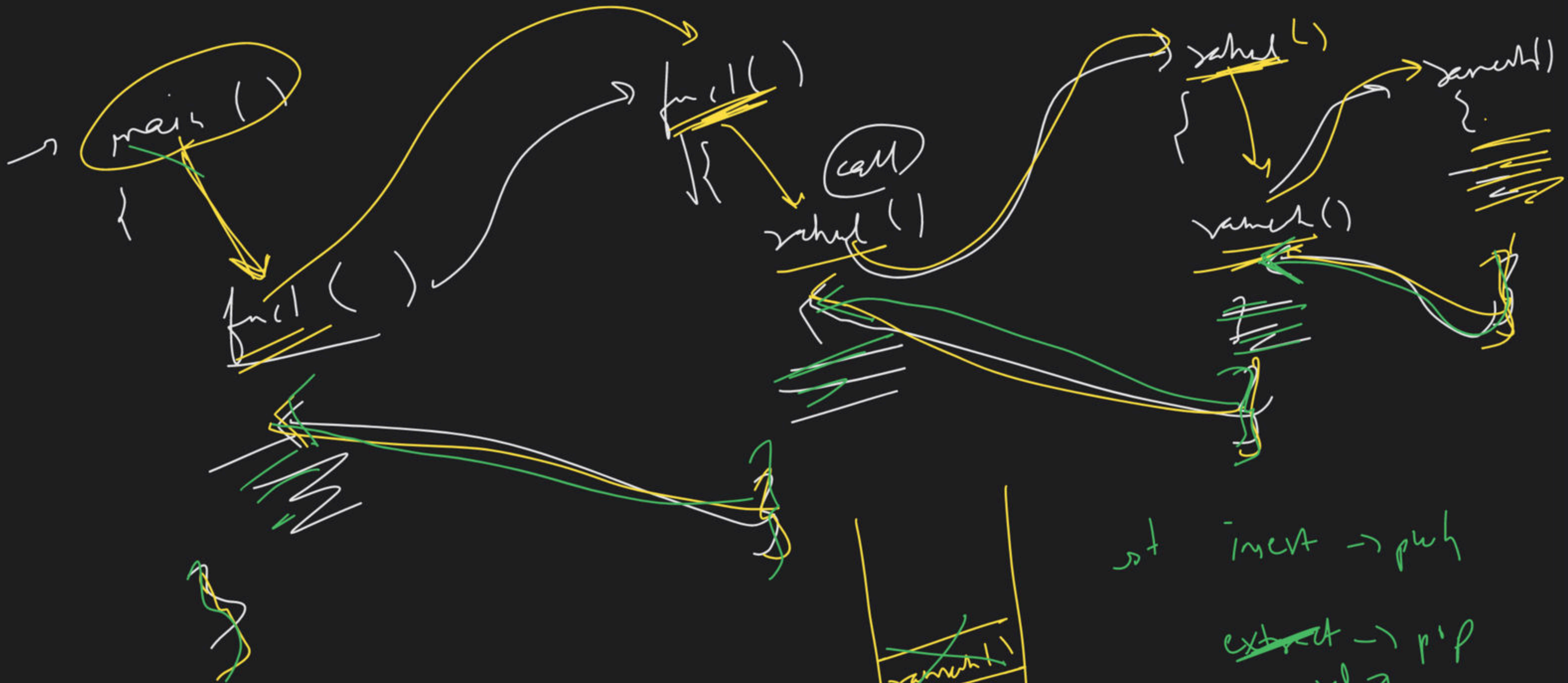
}





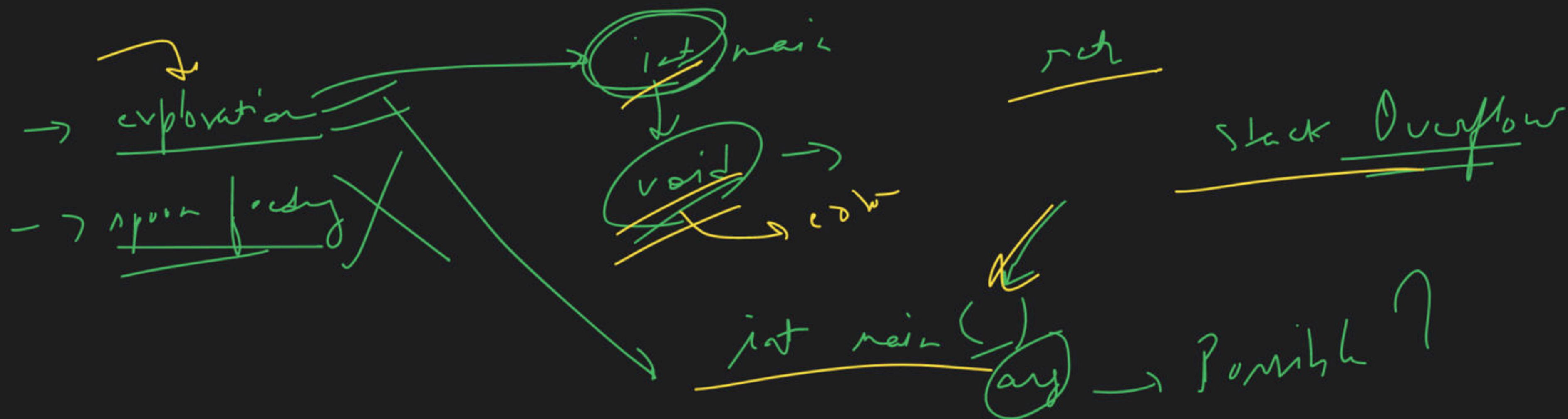


→ h+L / (S / 2B)



recur()
recur()
func()
main()

st insert → push
 extract → pop
 removed →



```
void arg1()
{
    // 
    // 
    // 
    return i
}
```

```
int main()
{
    // 
    // 
    // 
    return 1, 2, 3, 4
}
```

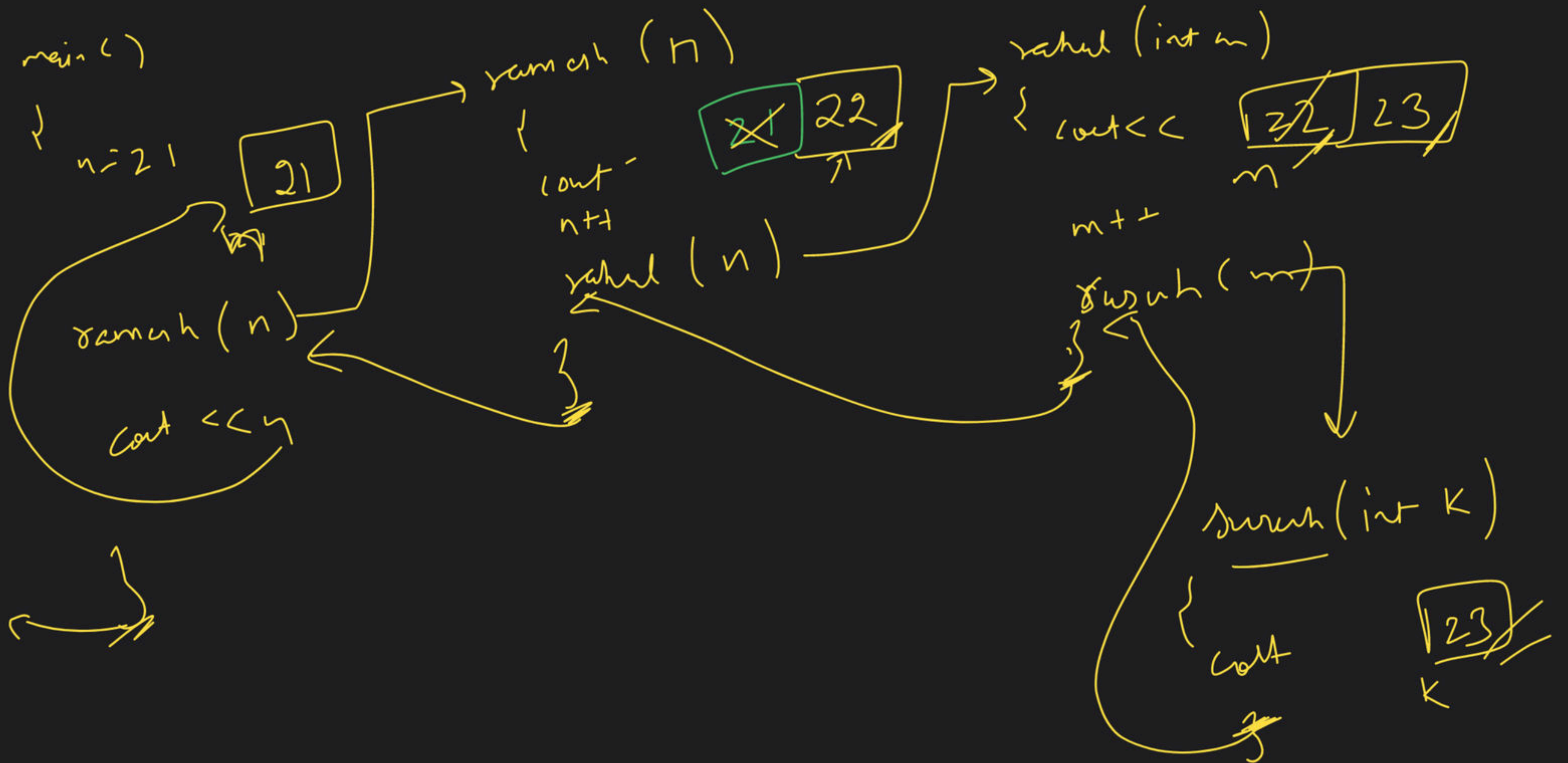
main()

{ n = 21

21
n

Study

→ help @ unacademy.com



21, 22, 23, 21

→ i/p → ~~find~~ n = 15 int n = 15

print all even no till n

↳ Wkne 16th

i/o: 2



n Ans

→ for (int i = 2; i <= n ; i = i + 2 or i += 2)
{
 cout << i;
}

→ for (int i = 1; i <= n; i++)

{

if (i % 2 == 0) or if (i % 2 != 0)

 count << i count << i

}

odd

↓

→ i/p → $n = 6$ →

$4, 4, 9, 16, 25, 36$

Don't

Print all squares

till

$1^2, 2^2, 3^2, 4^2, 5^2, 6^2$

→ $4+1+1$

RD Man
Trigno

$n = 6$

→ size → NDA

one

$1^2, 2^2, 3^2, 4^2, 5^2$
 $1, 1, 1, 1, 1$
 $1, 1, 1, 1, 1$

$1^2, 2^2, 3^2, 4^2, 5^2, 6^2$
 $1, 1, 1, 1, 1, 1$

$1^2, 2^2, 3^2, 4^2, 5^2, 6^2$
 $1, 1, 1, 1, 1, 1$

→ 7^{th} → good

rather

→ n

→ side

$= \frac{n \times n}{2} = \frac{6 \times 6}{2} = 18$

Practice ← method

→ ~~i/p~~ n i/p → $n = 5$

o/p → $5!$ → 120

$$5! = 5 \times 4 \times 3 \times 2 \times 1$$

factorial

flowchart

→ code

1 → 12

$[-2^{31}, 2^{31} - 1]$ ^{Int overflow}

int
↓
long

int

3, 14, 15

wrong

long long error

DSA

Dev
2/4/2

Custom Library

C++

Optimii

→ function Yes/No

i/p $\boxed{n=10}$
→ f → print Name

that you want

→ i/p → $\boxed{n=10}$

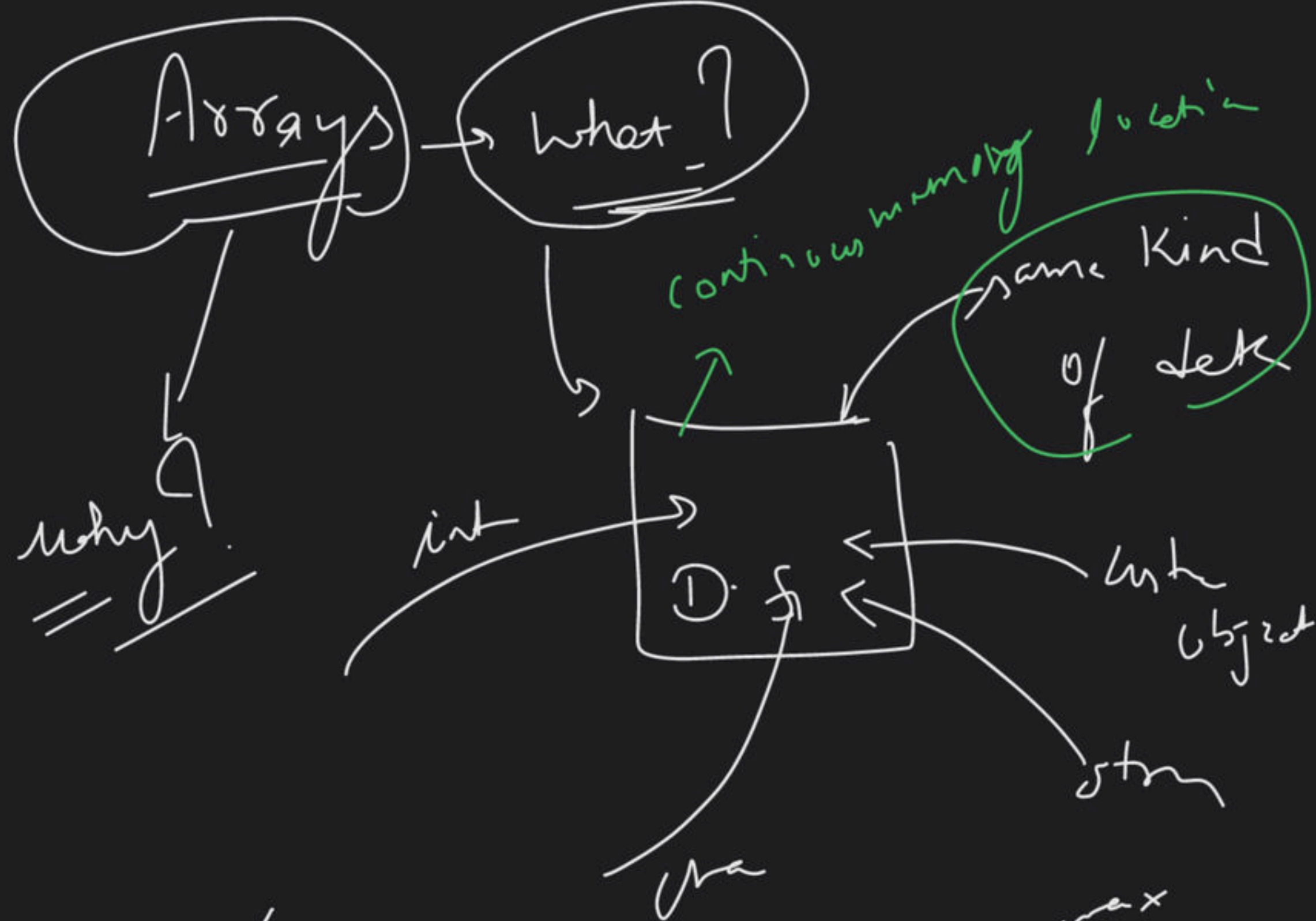
10 times →

babbar
babbar
babbar
babbar

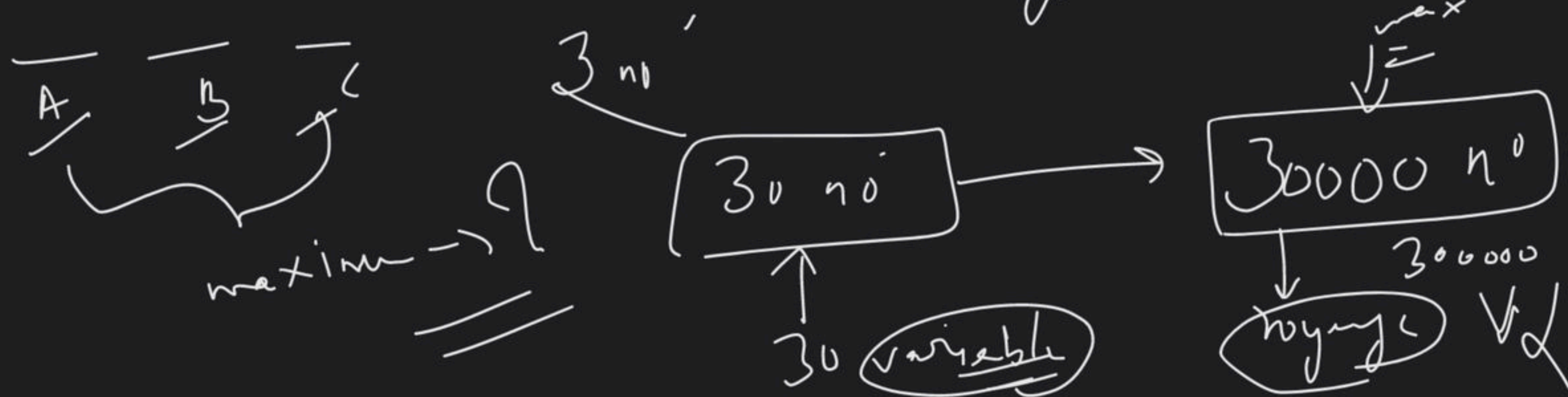
10 times
2

CBK
→

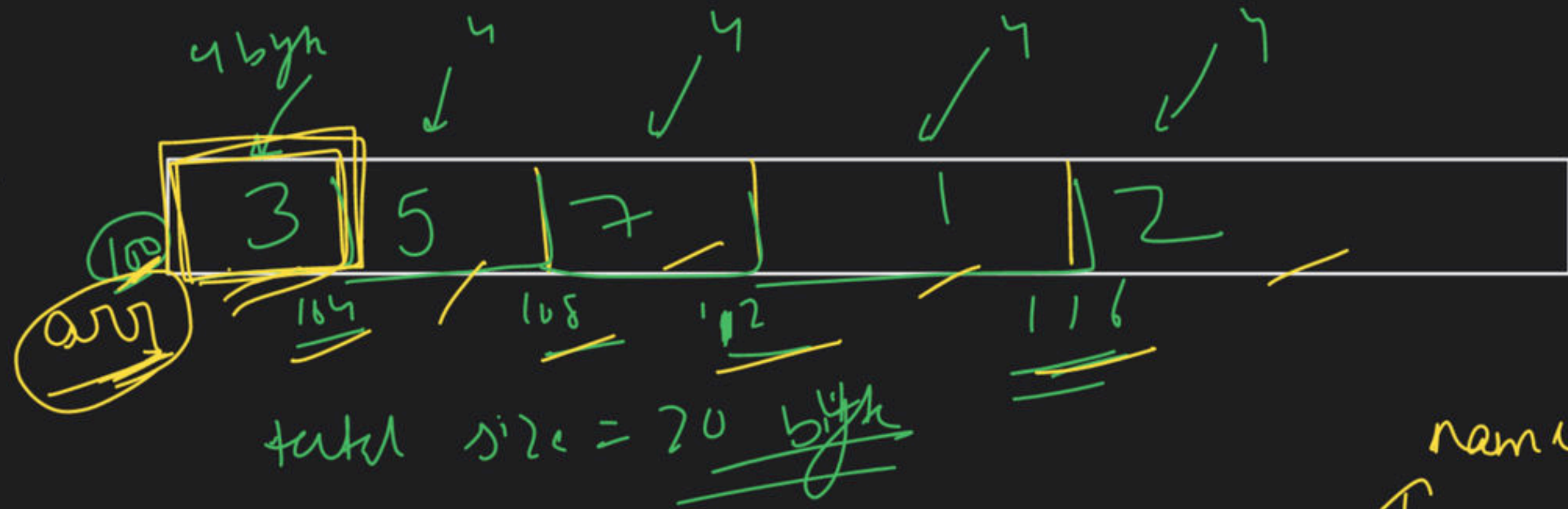
future → what
 → why
 → how?



Ex → P.S



Arrays



→ x
→ rml

int a = 5

100

arr

declare

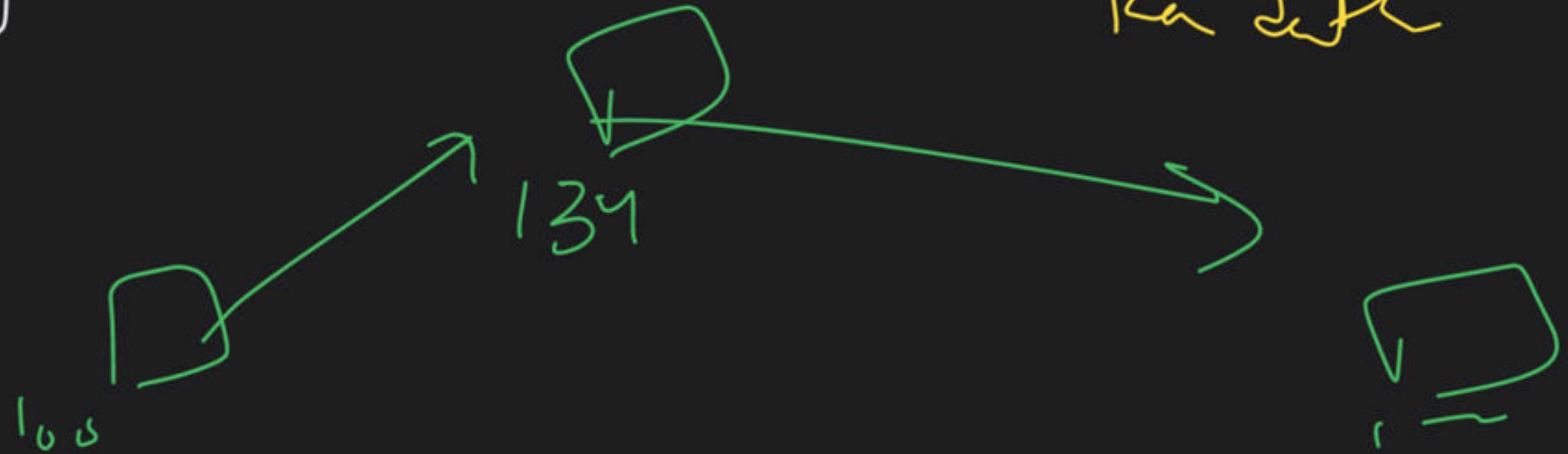
int arr [5];

int type
Ka dete

arr 100

arr
3m- address

5
a



→ 1 2 3 4 5

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

✓ initialization

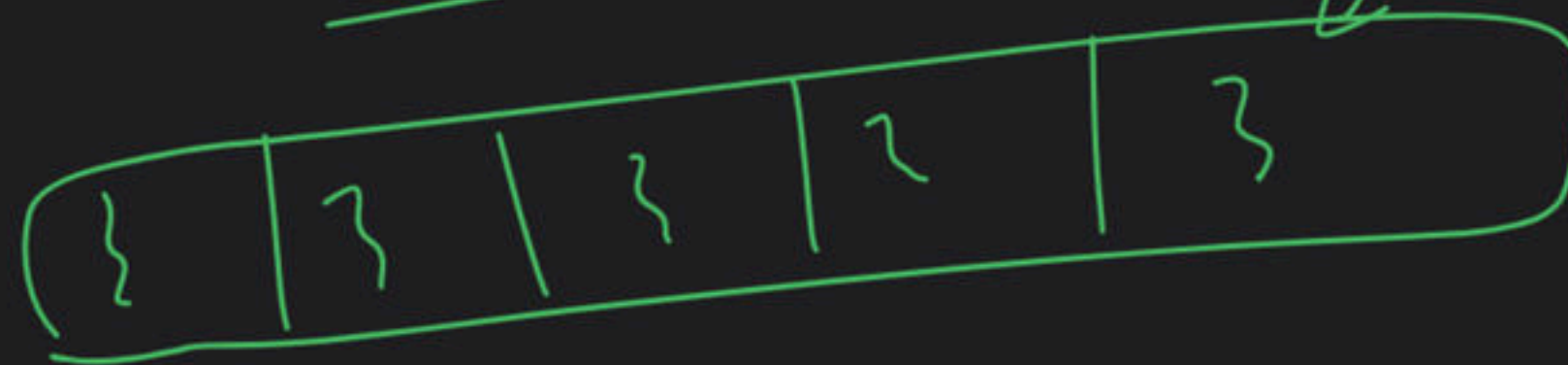


→ `int arr[] = {1, 2, 3, 4, 5}`

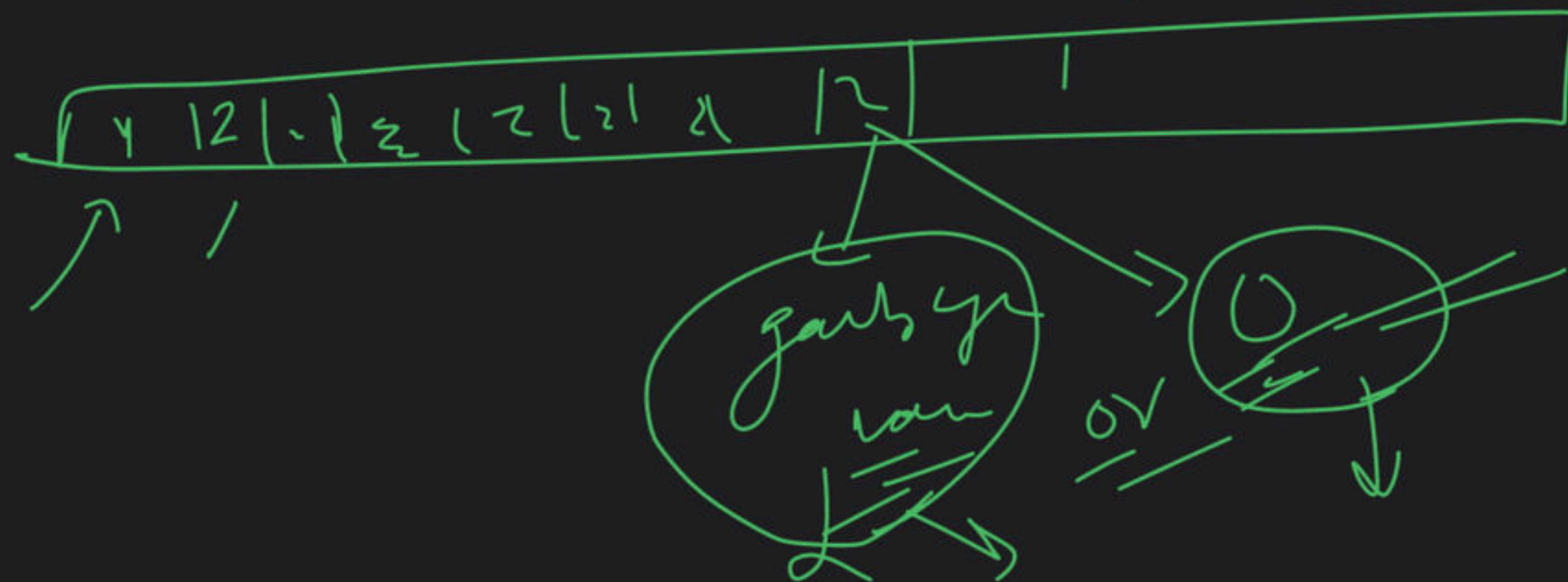
creation

`int arr[5];`

garbage value



int arr[15] = {1, 2}



1 | 0 | 0 | 0 | 0 | 0

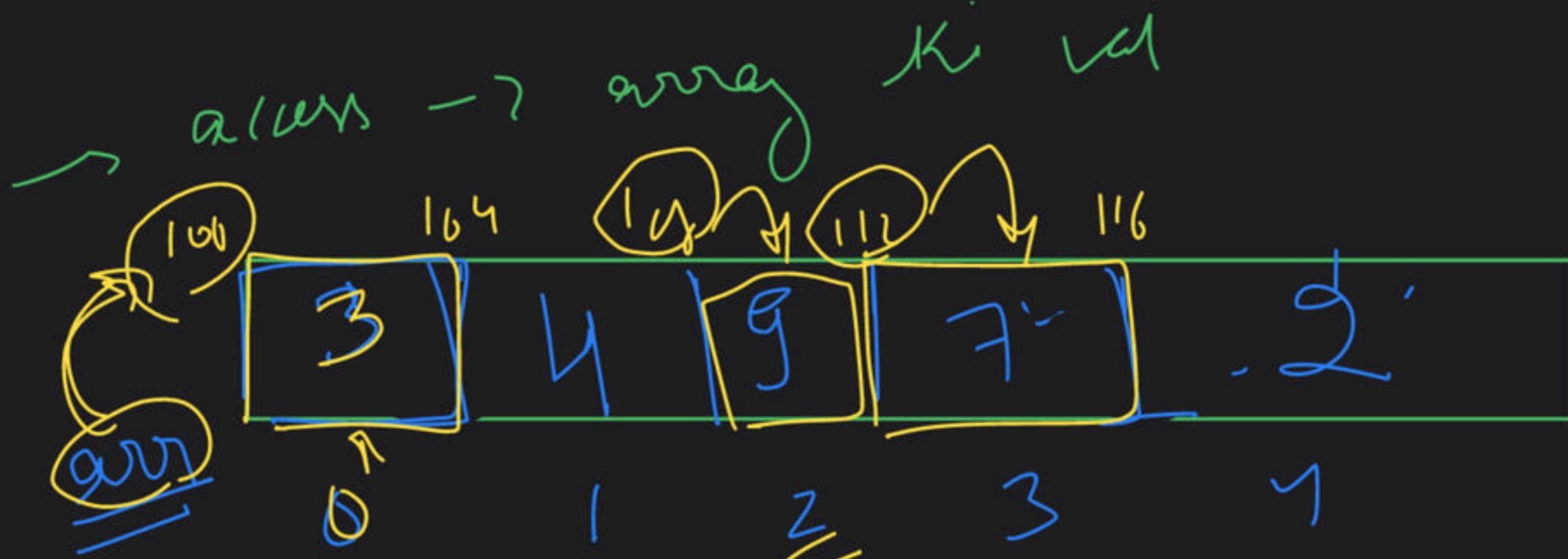
int arr[30] = {1}

1/w

→ initiale → right val.

int arr[25] = {0}

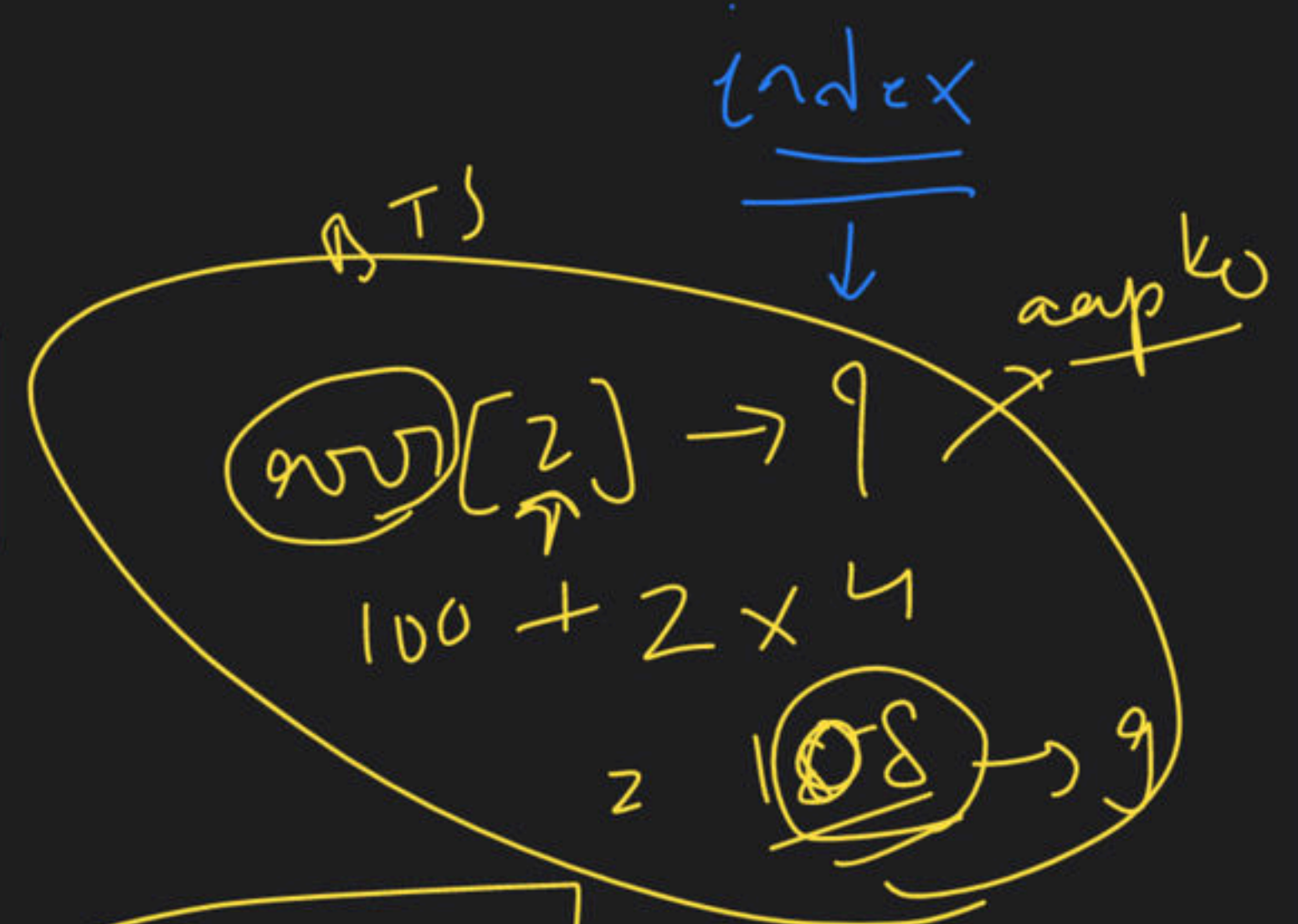
0 so initialis



first element → arr[0] → 3

arr[3] → 7

arr[i] = $\text{base address} + \text{index of data}$



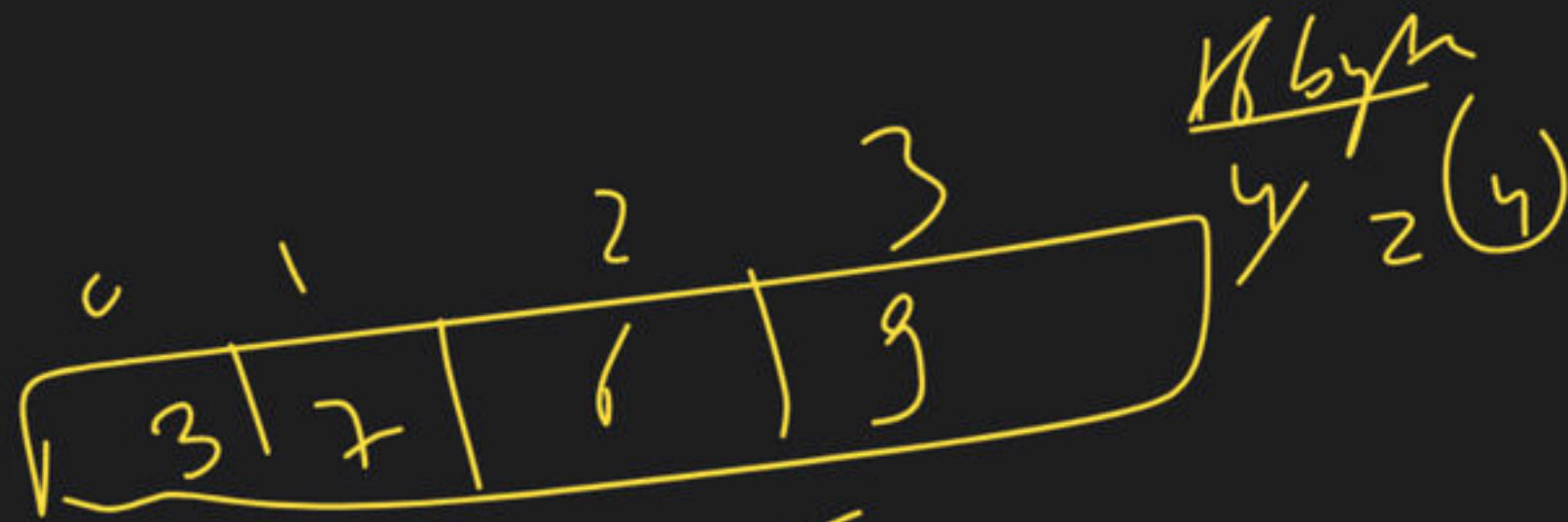
arr[0]

= 100 + 0 × 4 = 100

↓

3

arr[3] = 100 + 3 × 4 = 112 → 7



arr[0] → 3

arr[1] → 7

arr[2] → 6

arr[3] → 9

function → print element of array

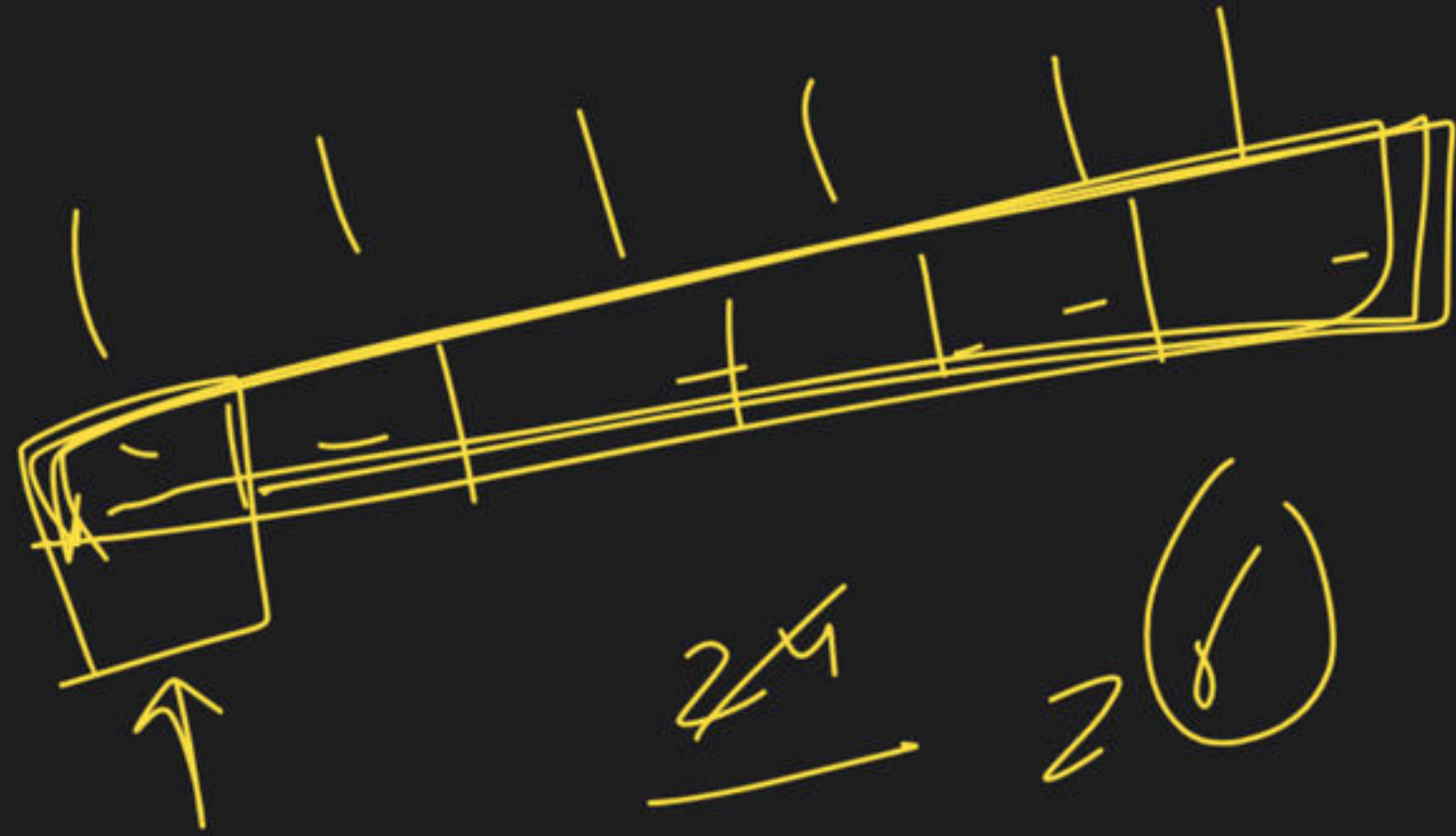
sizeof

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

int nro = $\frac{\text{sizeof(arr)}}{\text{sizeof(int)}}$

Yes No

Row → ?



$$\frac{2^4}{4} = 2^3$$

under

int arr[30] = { 1, 2, 3 }

$$\text{int size} = \frac{\text{sizeof(arr)}}{\text{sizeof(int)}}$$

2 30

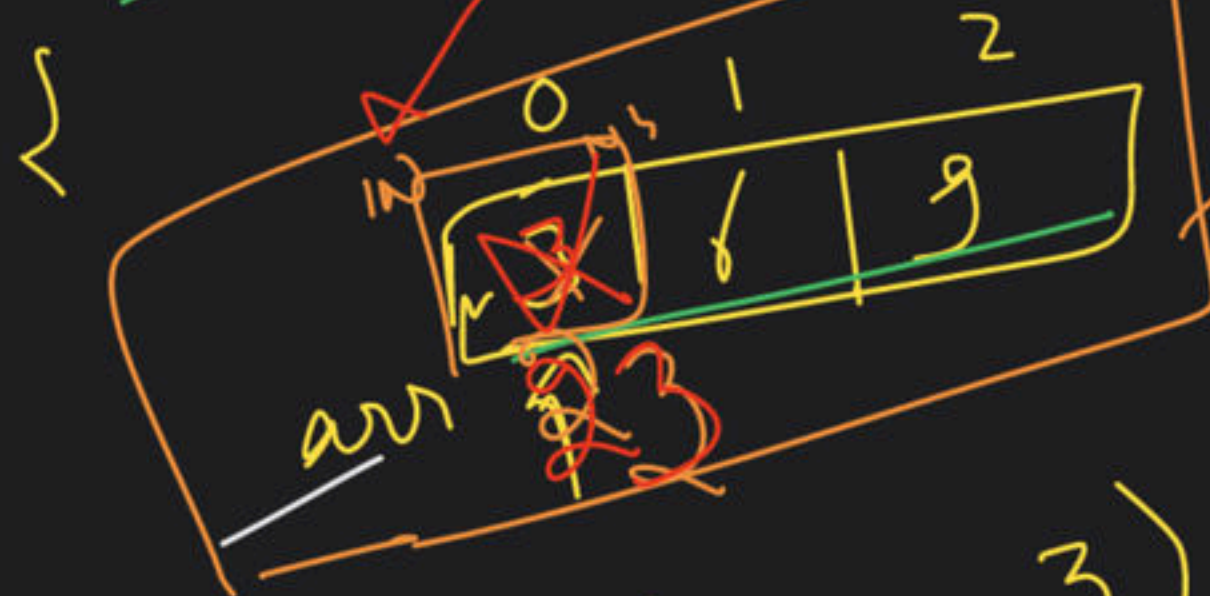
3

→ Array location ? → Y or N

→ Array access → index //

→ Not array with function

main()



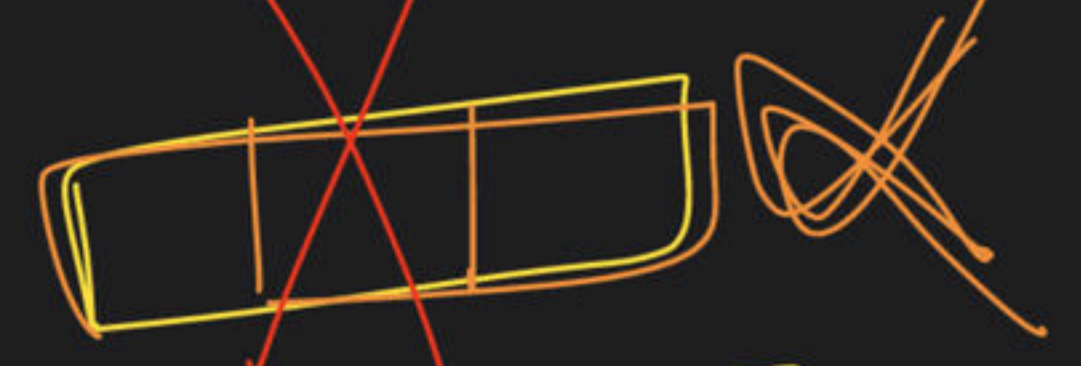
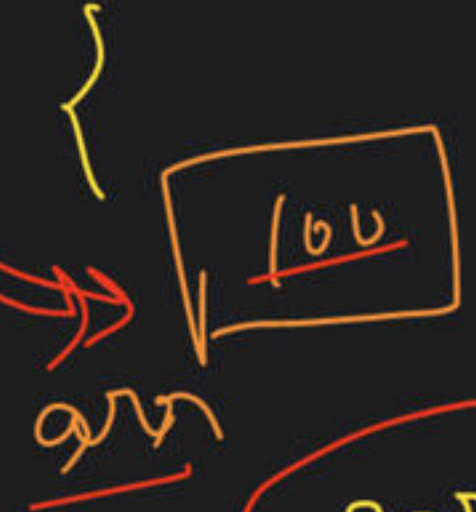
xyz(arr, 3)

print Array()



Call

xyz (int arr[], int s)



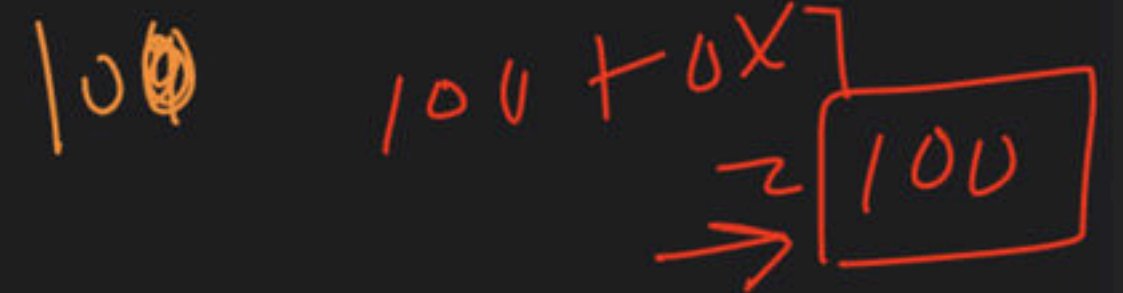
arr[0] = 23

$100 + 0 \times 4 = 100$

print Array() → o/p

arr[0]

Same or differ



main()

5

n

call

print(int n)

A - 1/1

kanal
nyau

5

n

copy

arr[-]

↓

add num

100

100

unique

100

100

→ explore

main

arr

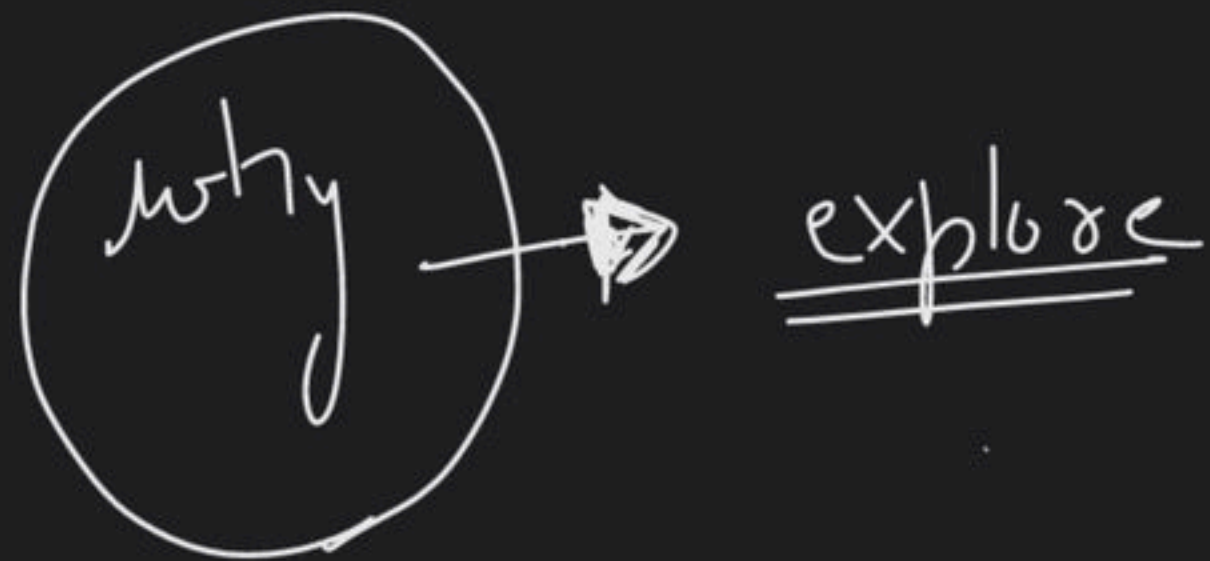
my 2 ()
{

arr
value
Update

}

Change surtain
for jayah
}

→ Doubt 9



→ Linear Search:-

arr size → 7

target = 4

arr	0	1	2	3	4	5	6
	3	6	7	12	2	4	1













