Arrays - Class 1

Special class

Love Babbar • Feb 10, 2023

Arrays in C++

Instructor: Love Babbar

What is Array ?

collection of element

similar element

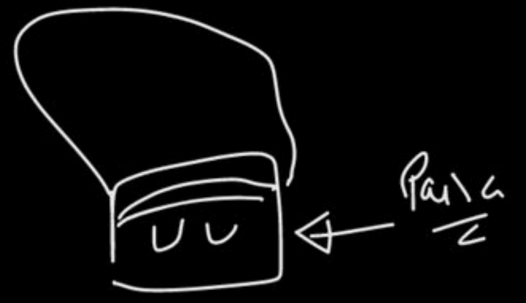
continous memory block

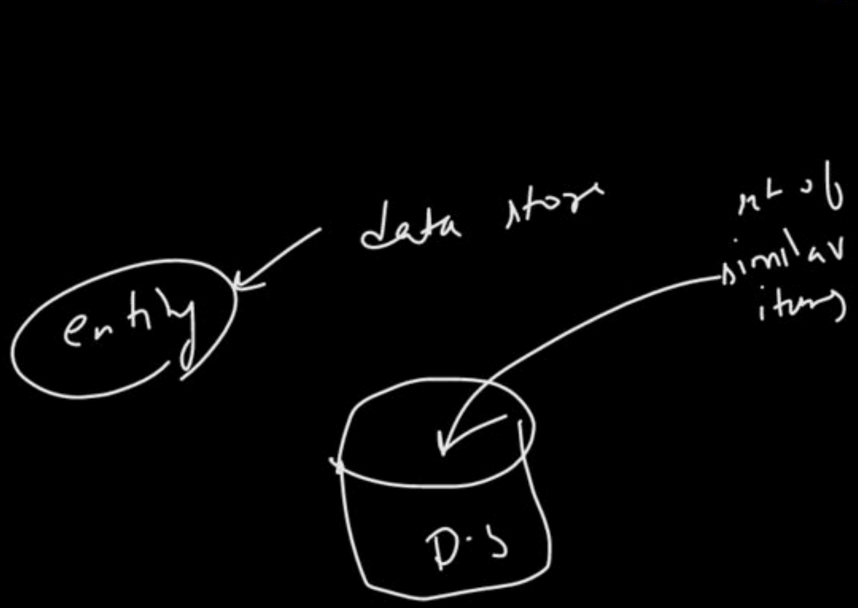
list of similar item

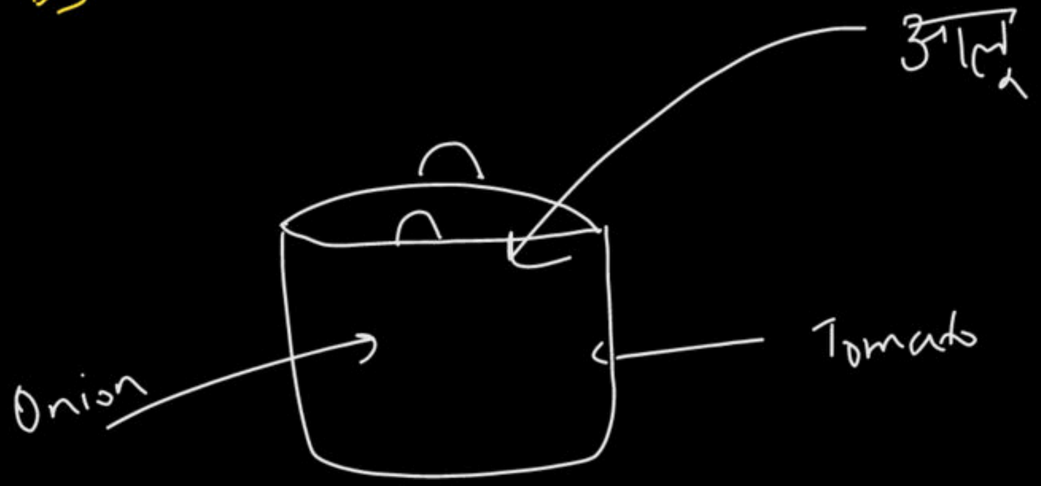
data structure

Linear DS

D.S







1 integer → 4 byte

BTS

(memory)

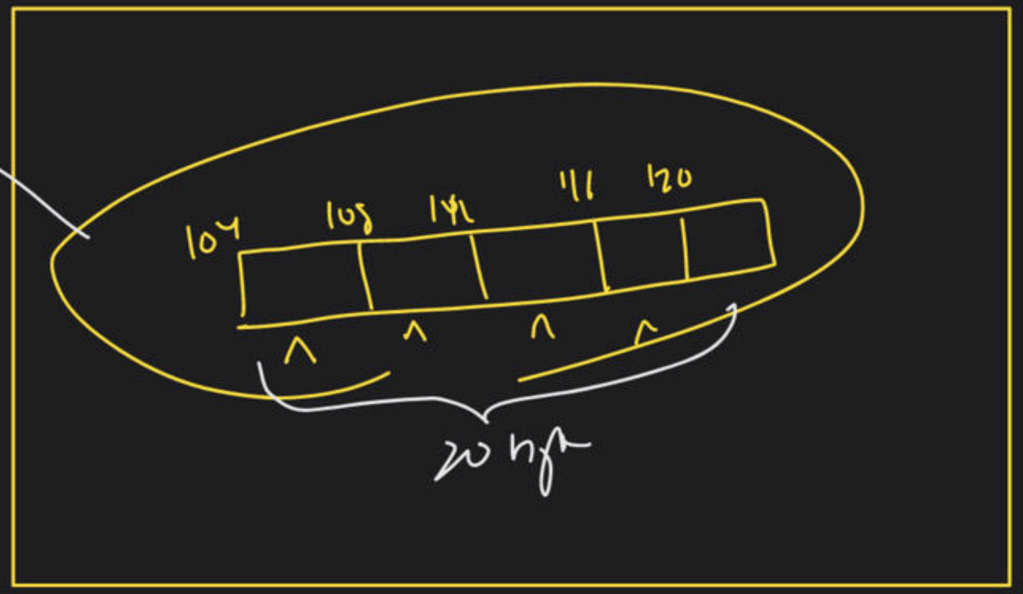
array → continous memory

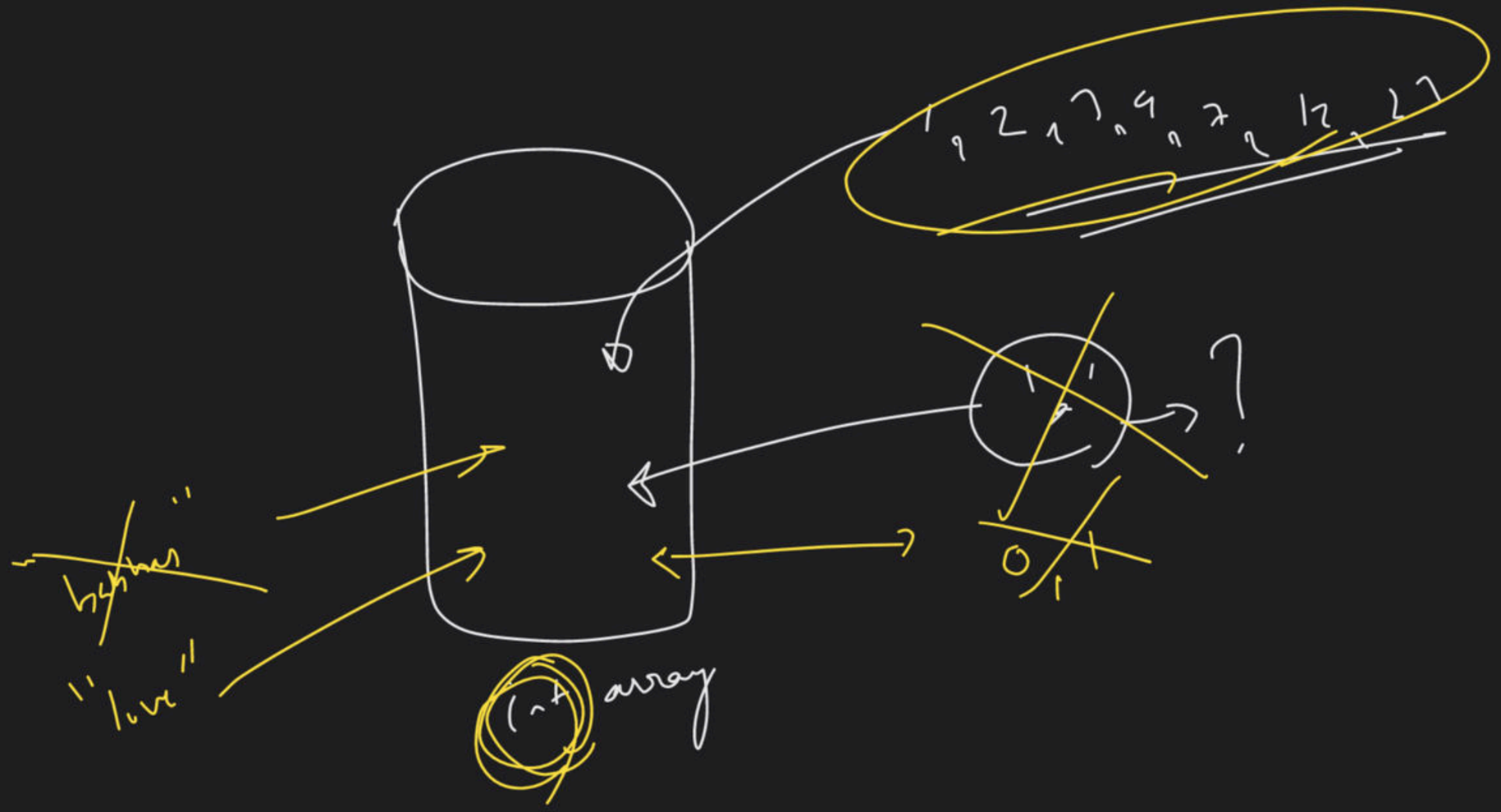
A B C D E

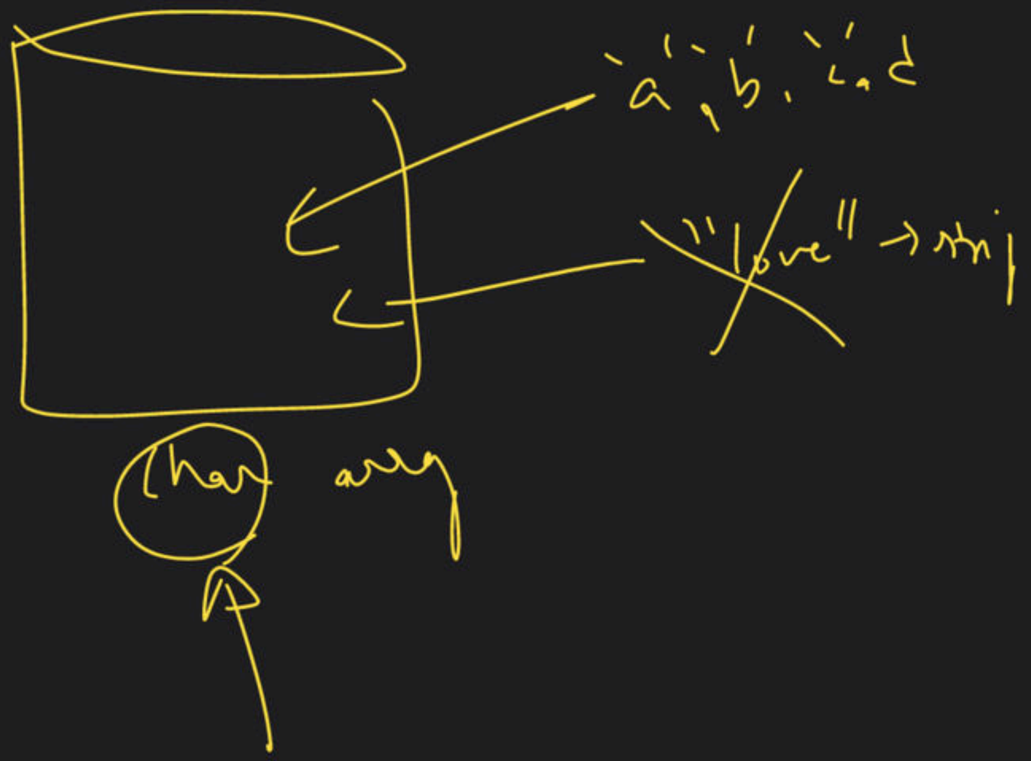
5 integer

5 X 4

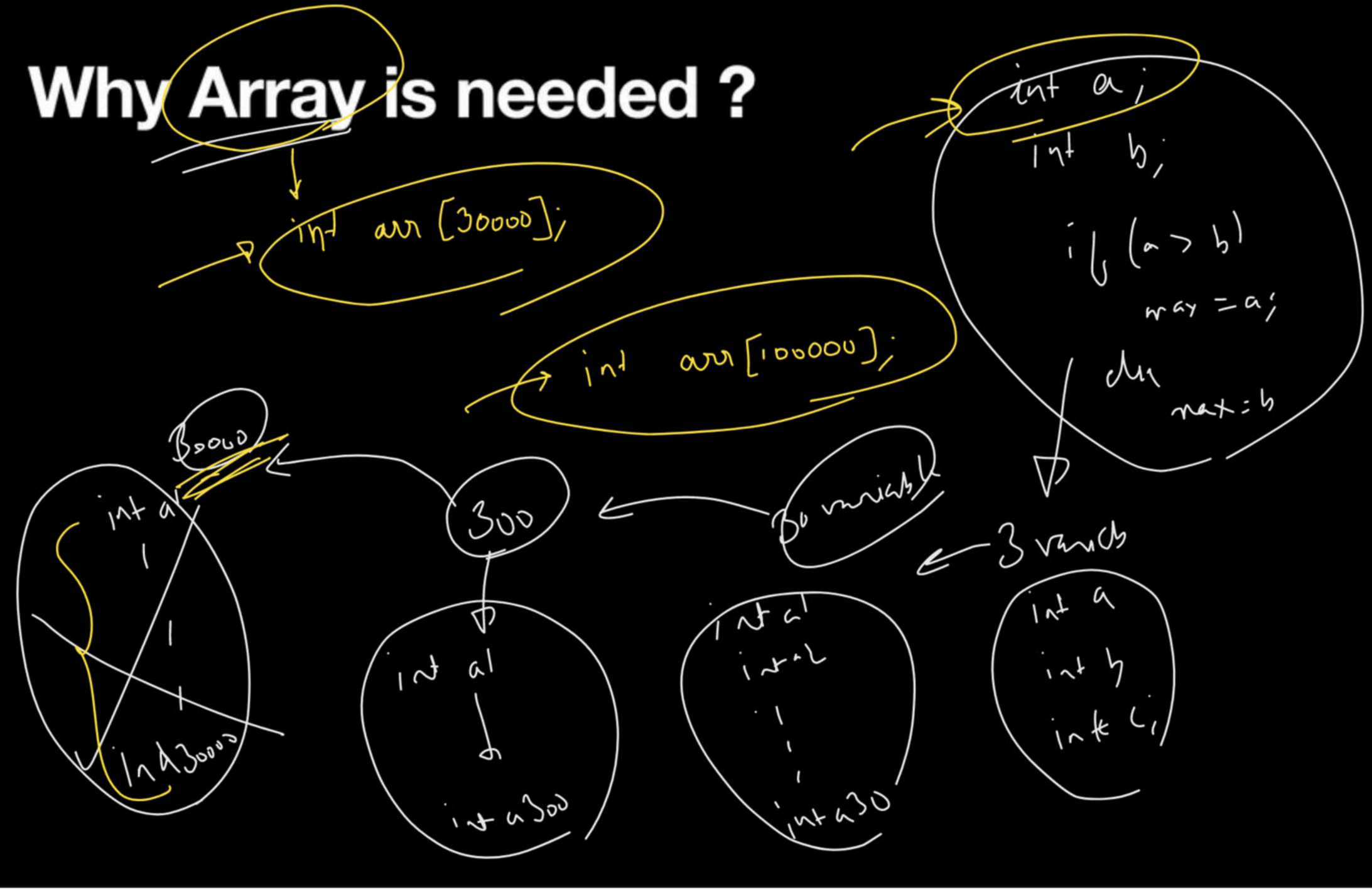
= 20 byte

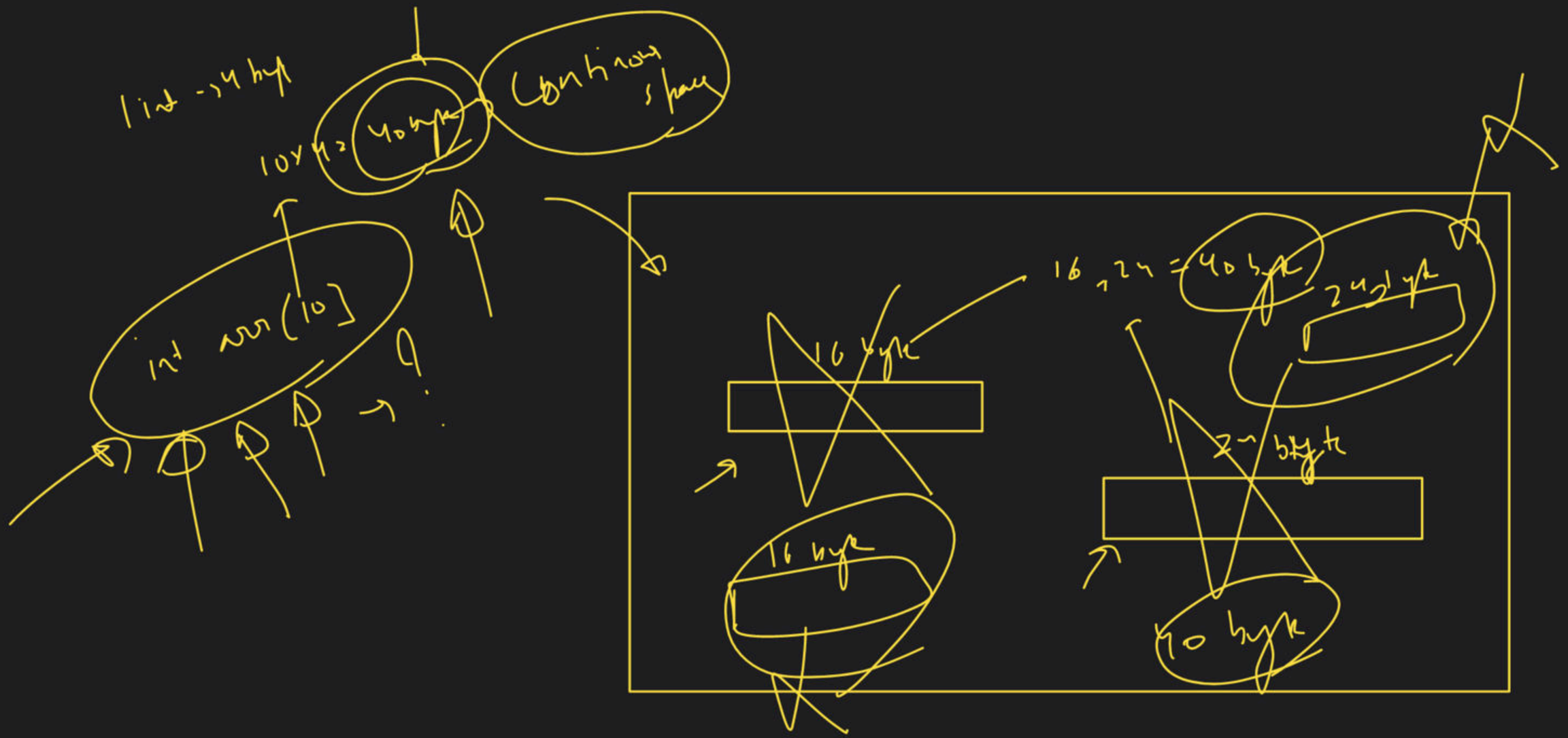



1, 2, 7, 9, 7, 12, 27
?
0
1
array


'a', 'b', 'c', d
"love" → str
char arr

Why Array is needed ?





Creating an Array:

declare

square brackets → syntax

int arr[10];

int arr[10];

int → data type

arr → variable name

10 → size of array

int a = 5;

104 | 4 byte | 5 | a

Symbol table

a → 104

Base address

104 | 10 integer | 40 byte

arr = 104

10 integer ke continuous space arr ki allocate kiya gya hai

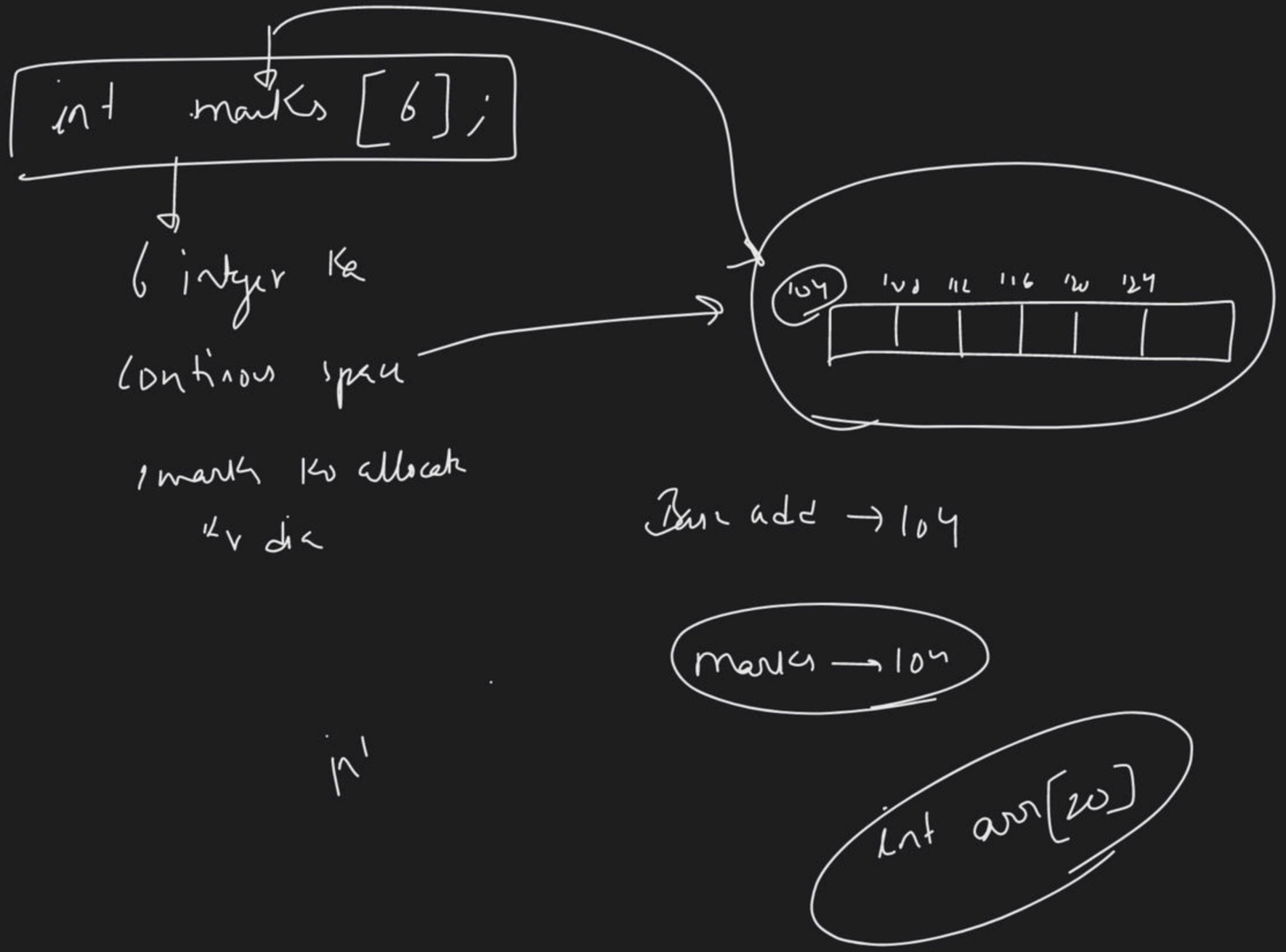
int arr = 12;

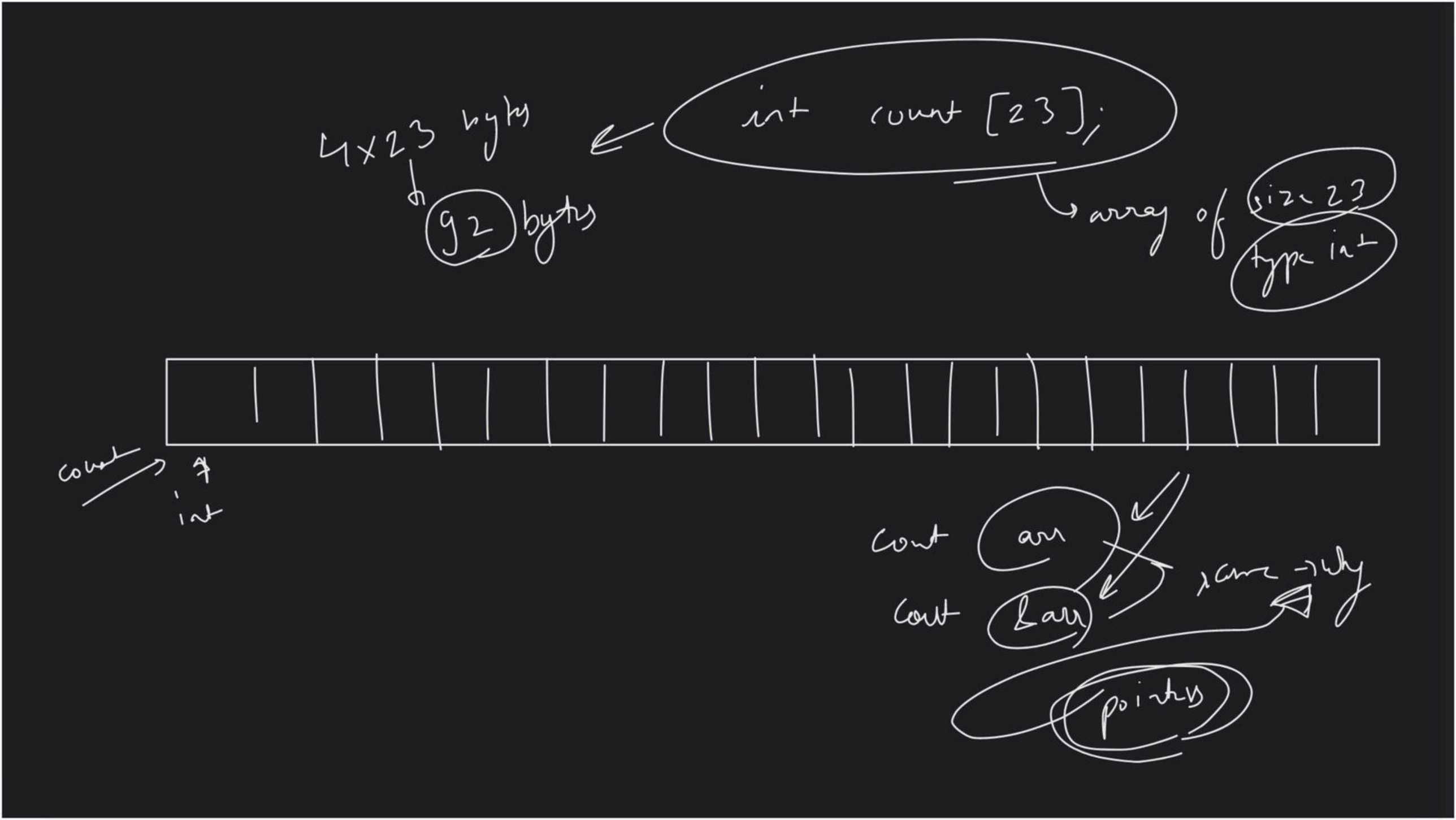
S.T

arr → 104

104 | 4 byte | 12 | arr

cout << arr;


int marks [6];
6 integer ka
continous space
marks ko allocate
kr dia
104
108
112
116
120
124
Base add → 104
marks → 104
int arr[20]

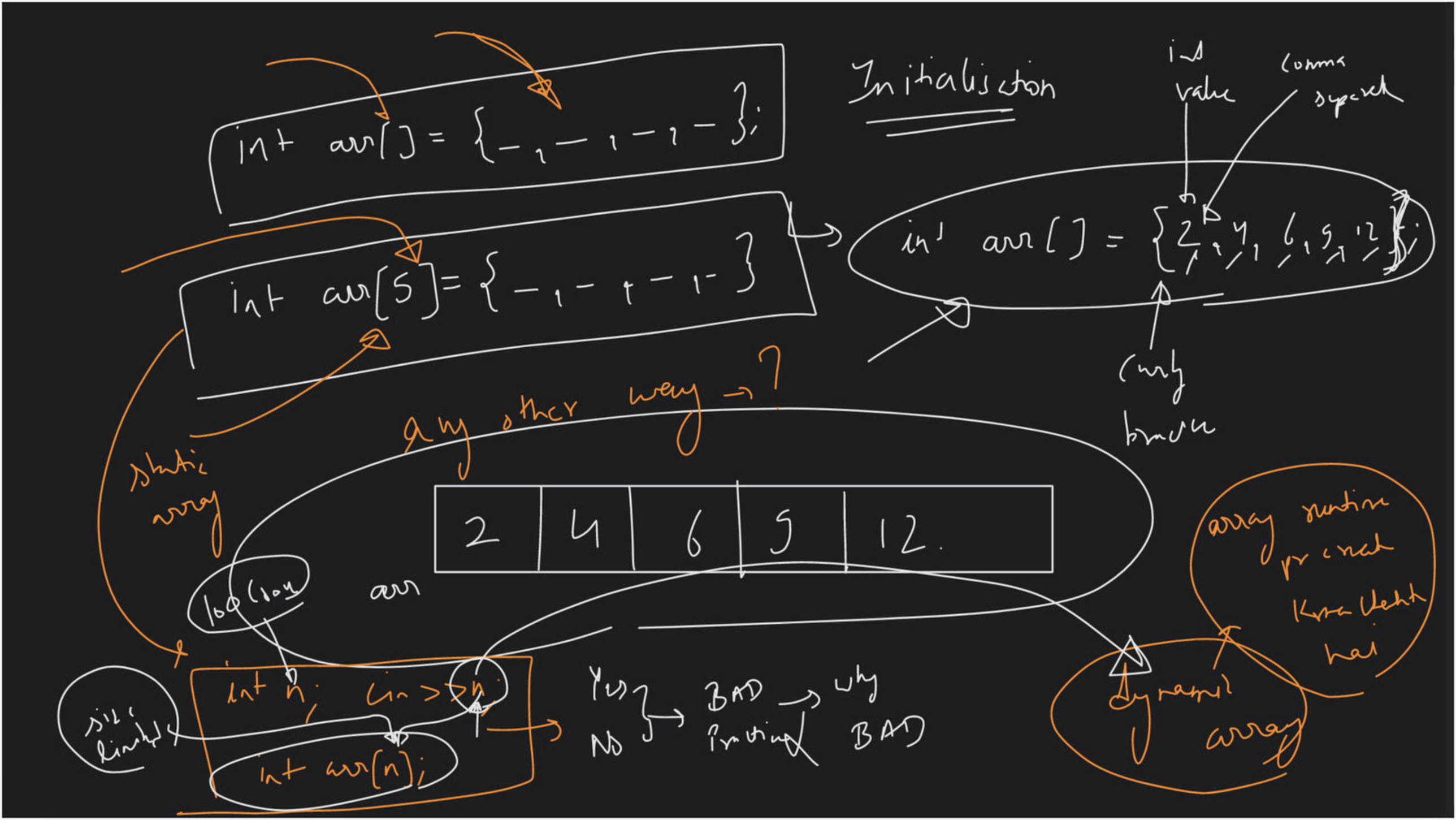

4x23 bytes
92 bytes
int count[23];
array of
size 23
type int
count
int
count
arr
count
&arr
same addy
points

(1) int array -> 57 size int a[57];

(2) char array -> 106 size-1 char a[106];

(3) bool array -> 23 size

bool arr[23];


int arr[] = {-, -, -, -, -};
int arr[5] = {-, -, -, -, -}
Initialisation
int value
comma speech
int arr[] = {2, 4, 6, 5, 12};
curly brace
static array
any other way → ?
2
4
6
5
12.
arr
log(10)
int n; cin >> n;
int arr[n];
size limit
Yes
No
BAD
why BAD
array runtime pr create karna chahta hai
dynamic array

~~int arr[n]~~

~~int arr[cin >> n;]~~

① int arr → 1, 3, 5, 7

↓

int arr[4] = {1, 3, 5, 7}

② char arry → 10 size

→ 'a' 'b' 'c'

↓

char arr[10] = {'a', 'b', 'c'}

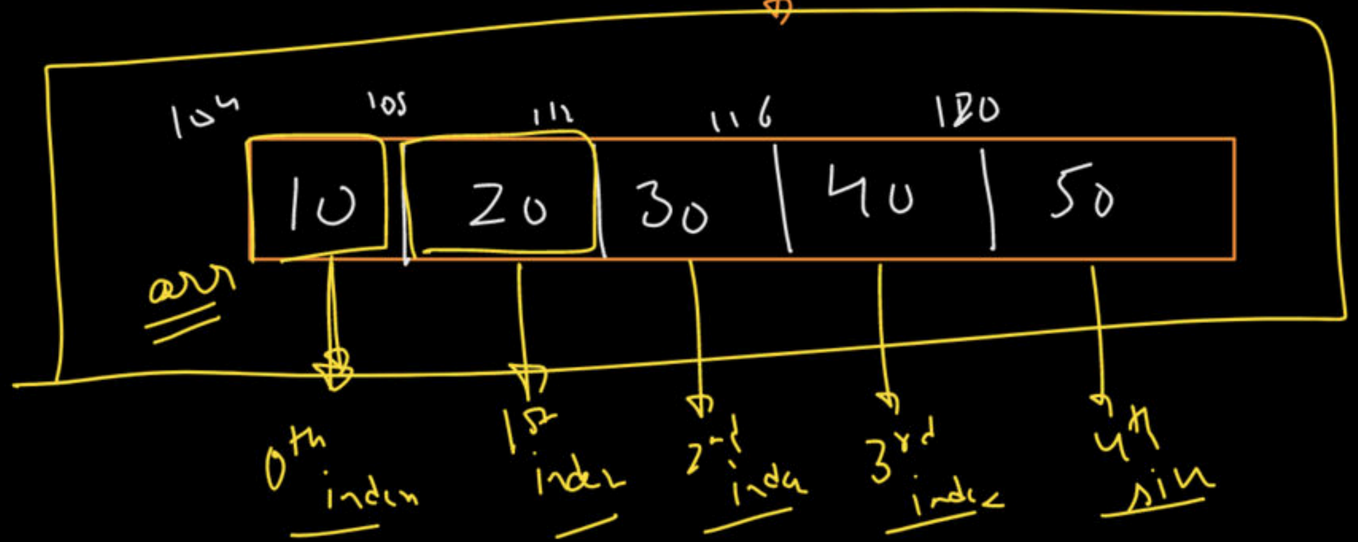
size must [illegible]!

Index && Access in Array

Indexing

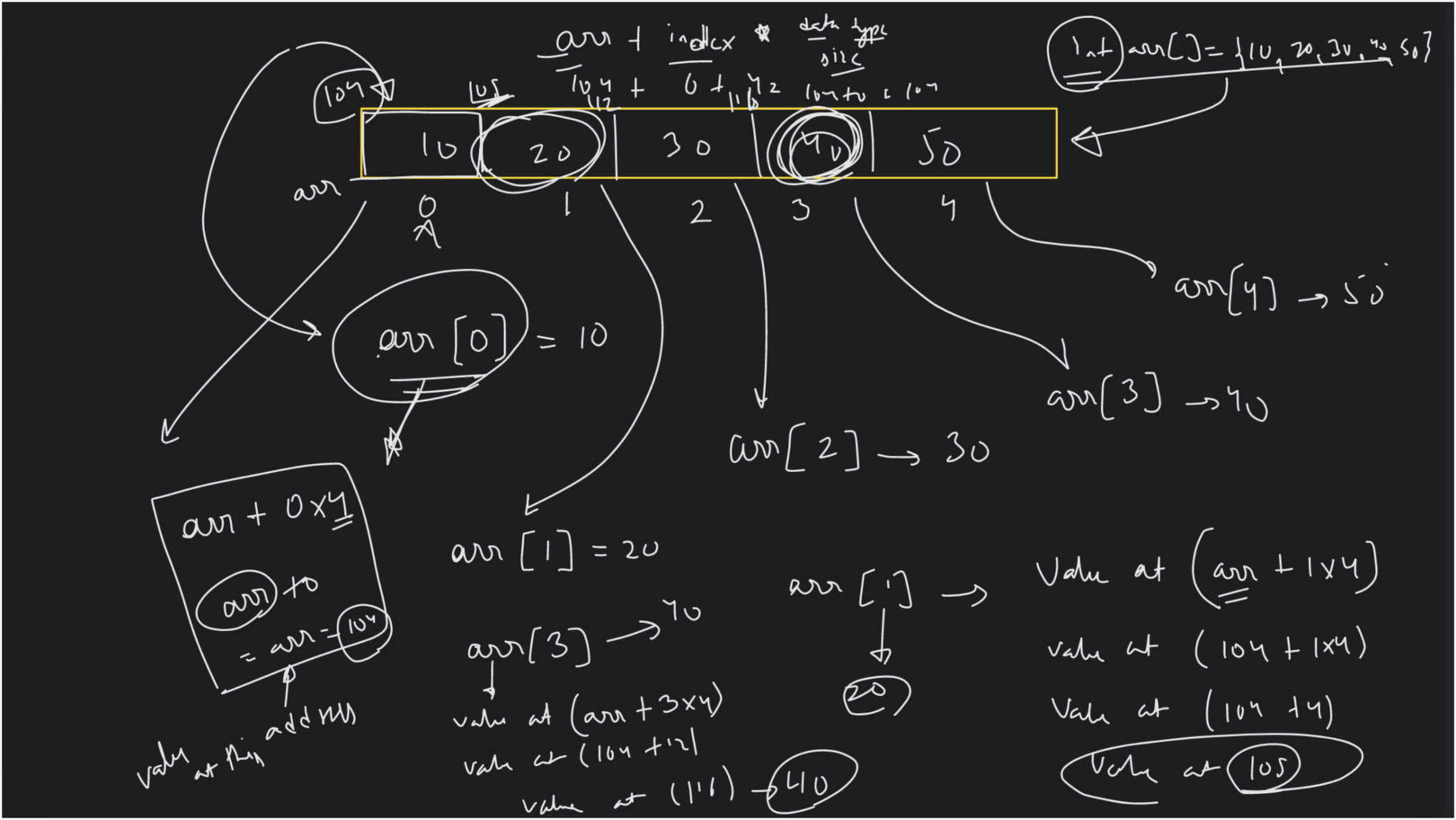
0-based indexing

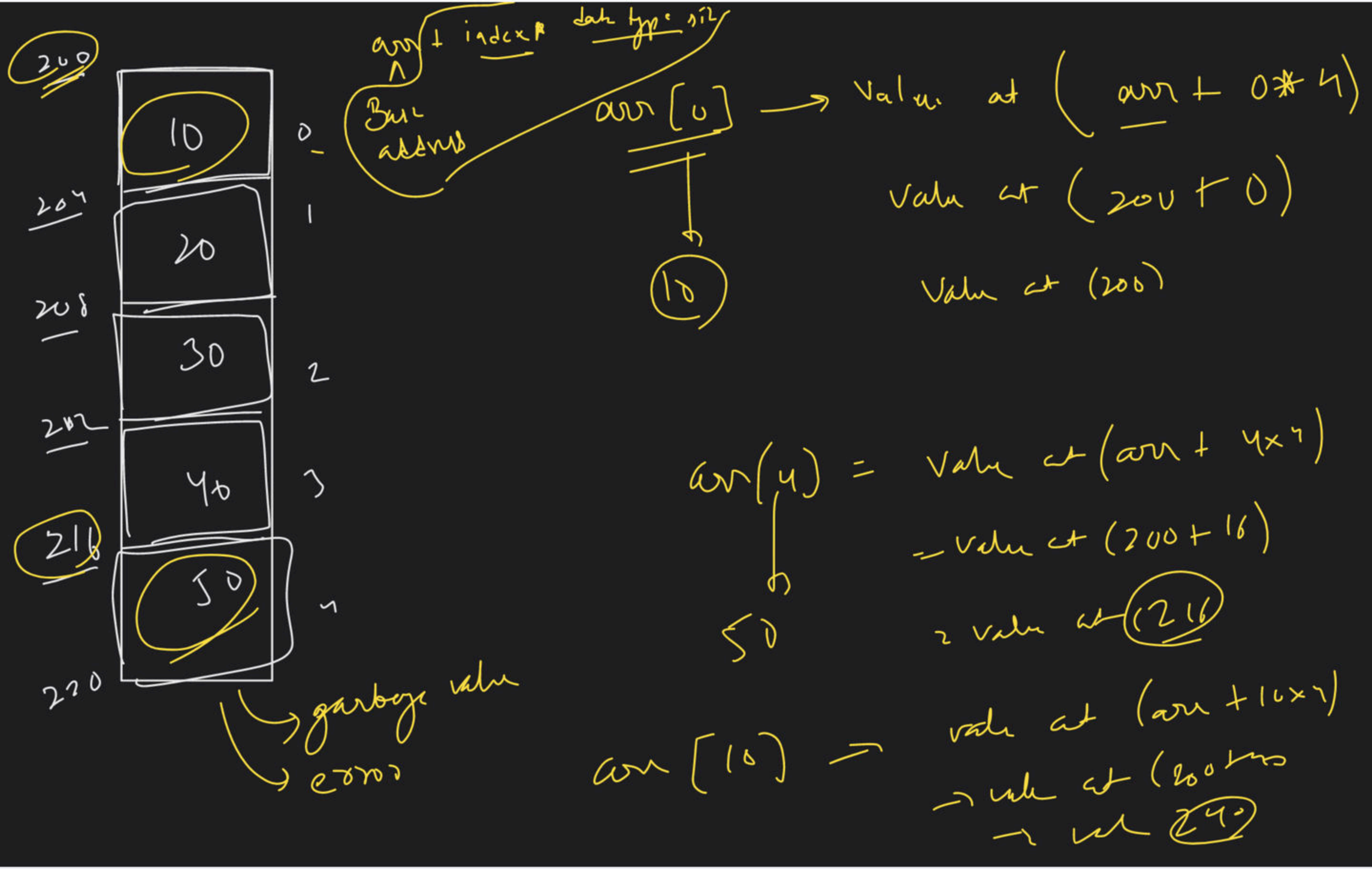
int arr[5] = {10, 20, 30, 40, 50}



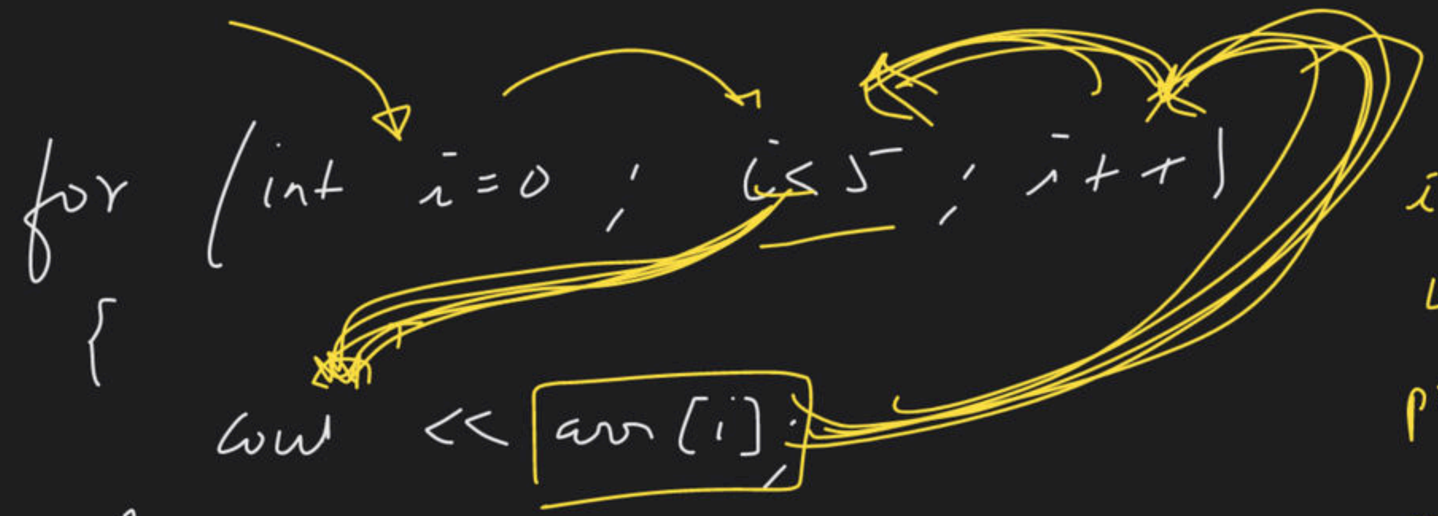
5 size array → index → [0 → 4]

n size array → index → [0 → (n-1)]


arr + index * data type size
104 + 0 * 4
104 to 107
int arr[] = {10, 20, 30, 40, 50}
104
105
112
116
10
20
30
40
50
arr
0
1
2
3
4
arr[0] = 10
arr[1] = 20
arr[2] → 30
arr[3] → 40
arr[4] → 50
arr + 0x4
arr to
= arr = 104
value at this address
arr[3] → 40
value at (arr + 3x4)
value at (104 + 12)
value at (116) → 40
arr[1] →
20
Value at (arr + 1x4)
value at (104 + 1x4)
Value at (104 + 4)
Value at 105


200
10
20
30
40
50
0
1
2
3
4
200
204
208
212
216
220
arr + index * data type size
Base address
arr[0] → Value at (arr + 0*4)
Value at (200 + 0)
Value at (200)
10
arr(4) = Value at (arr + 4x4)
= value at (200 + 16)
= value at (216)
50
arr[10] → value at (arr + 10x4)
→ value at (200 + 40)
→ value (240)
garbage value
error


for (i=0 ; i<5 ; i++)
{
}
i=0
i=1
i=2
i=3
i=4
arr[0]
arr[1]
arr[2]
arr[3]
arr[4]
i=5
i<5 is false
loop is broken



i=0
0<5 → T
print arr[0] →

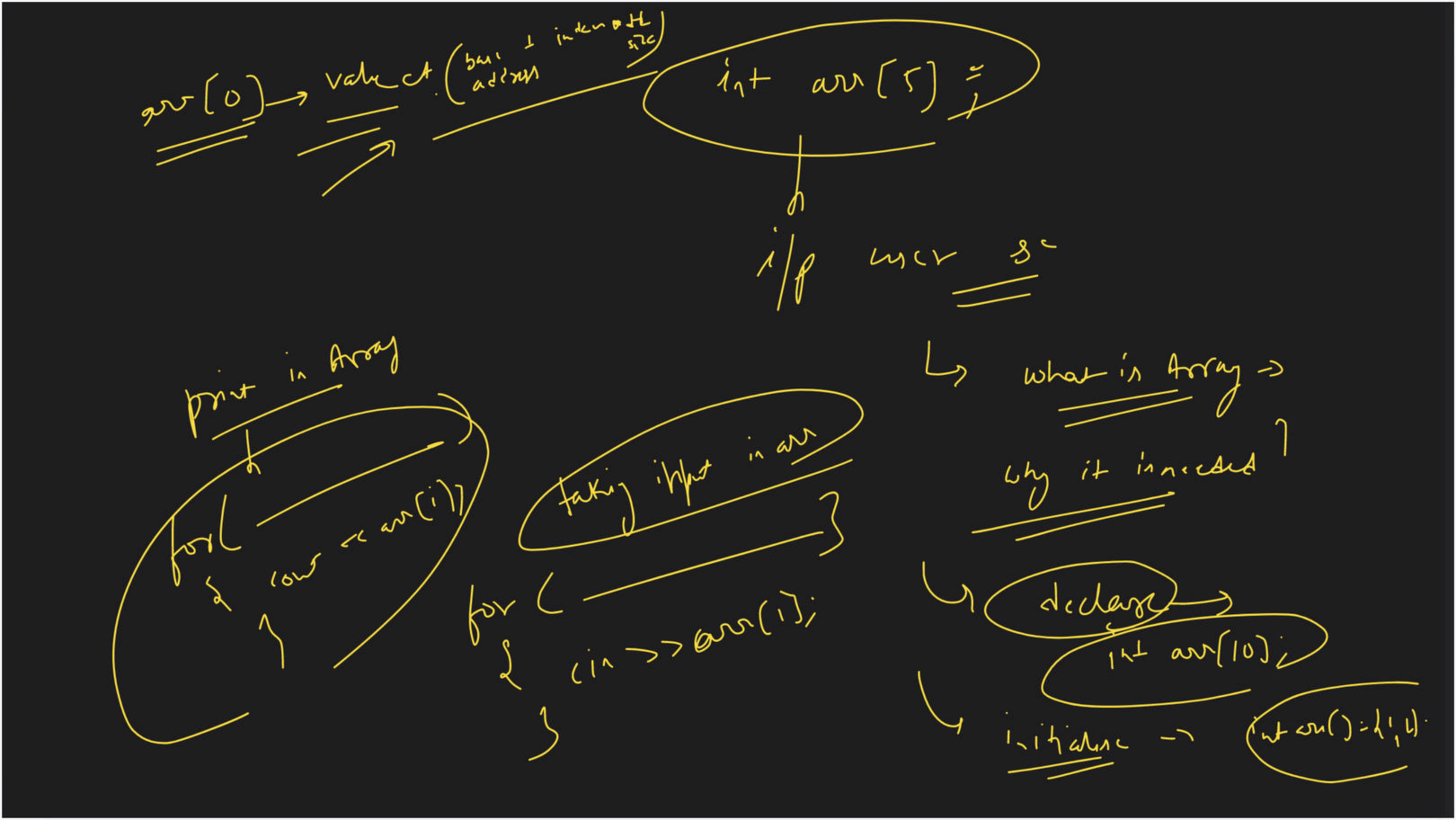
i=1
1<5 → T
print arr[1]

i=2
2<5 → T
print arr[2]

i=3
3<5 → T
print arr[3]

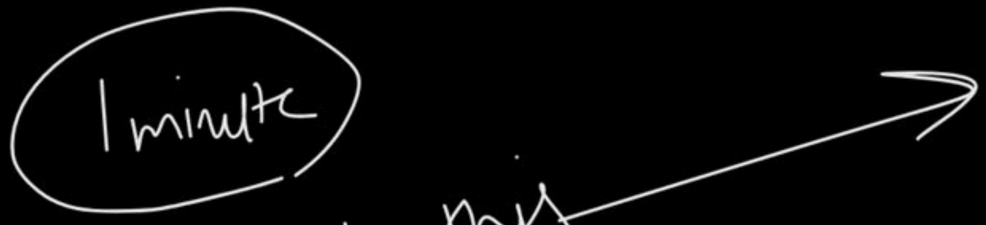
i=4
4<5 → T
print arr[4]

i=5
5<5 → F
loop se bahar


arr[0] → value at (base address + index * size)
int arr[5] = {};
i/p user se
print in Array
for(; i<sizeof(arr)/sizeof(arr[0]);)
taking input in arr
for(; ;) { cin >> arr[i]; }
what is Array ?
why it is needed ?
declare
int arr[10];
initialise → int arr[] = {1,4};

Initialising Array:

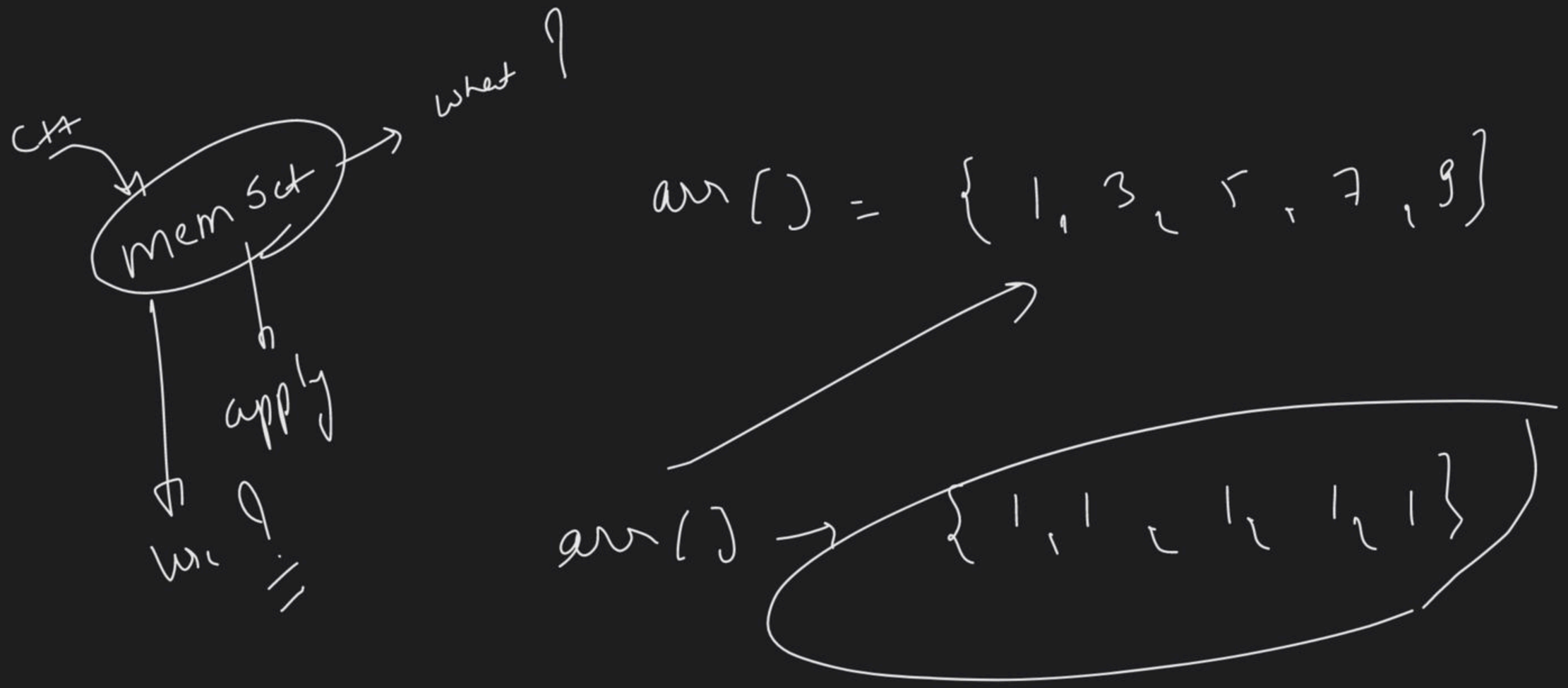
Q-1 take 5 elements i/p in array & print their doubles



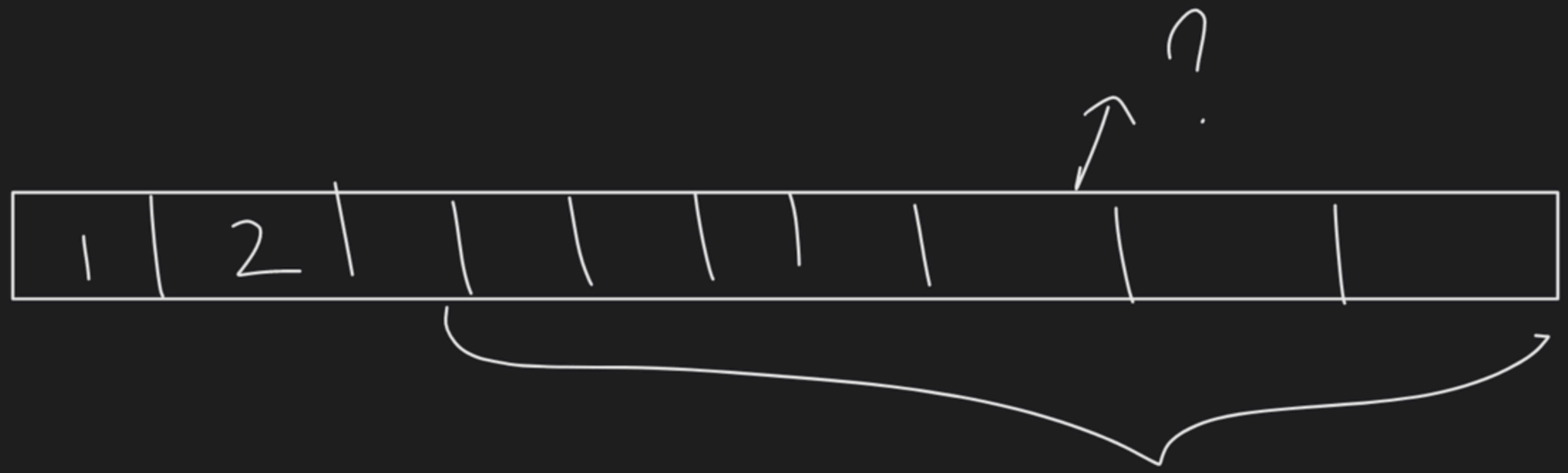
arr[] → i/p → 1, 3, 5, 7, 9

o/p → 2, 6, 10, 14, 18

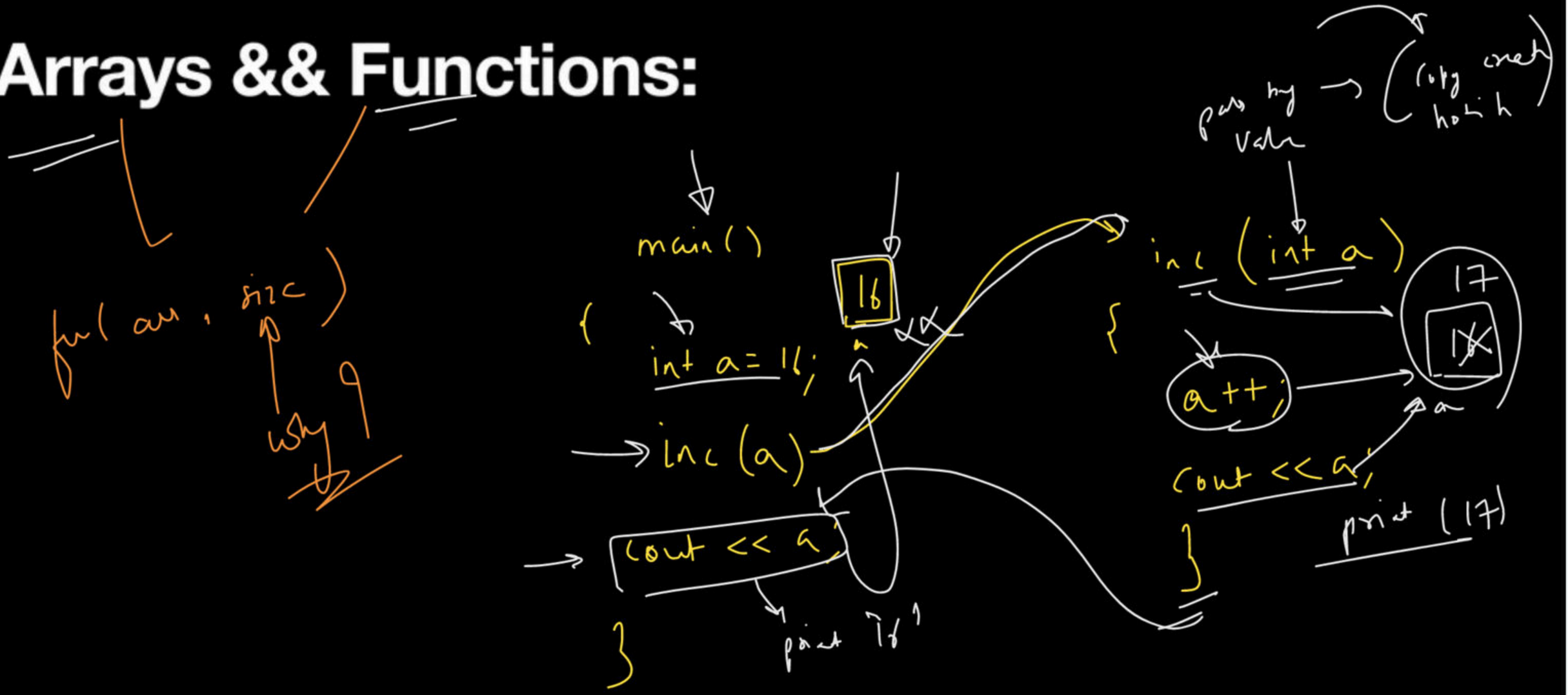
print not store

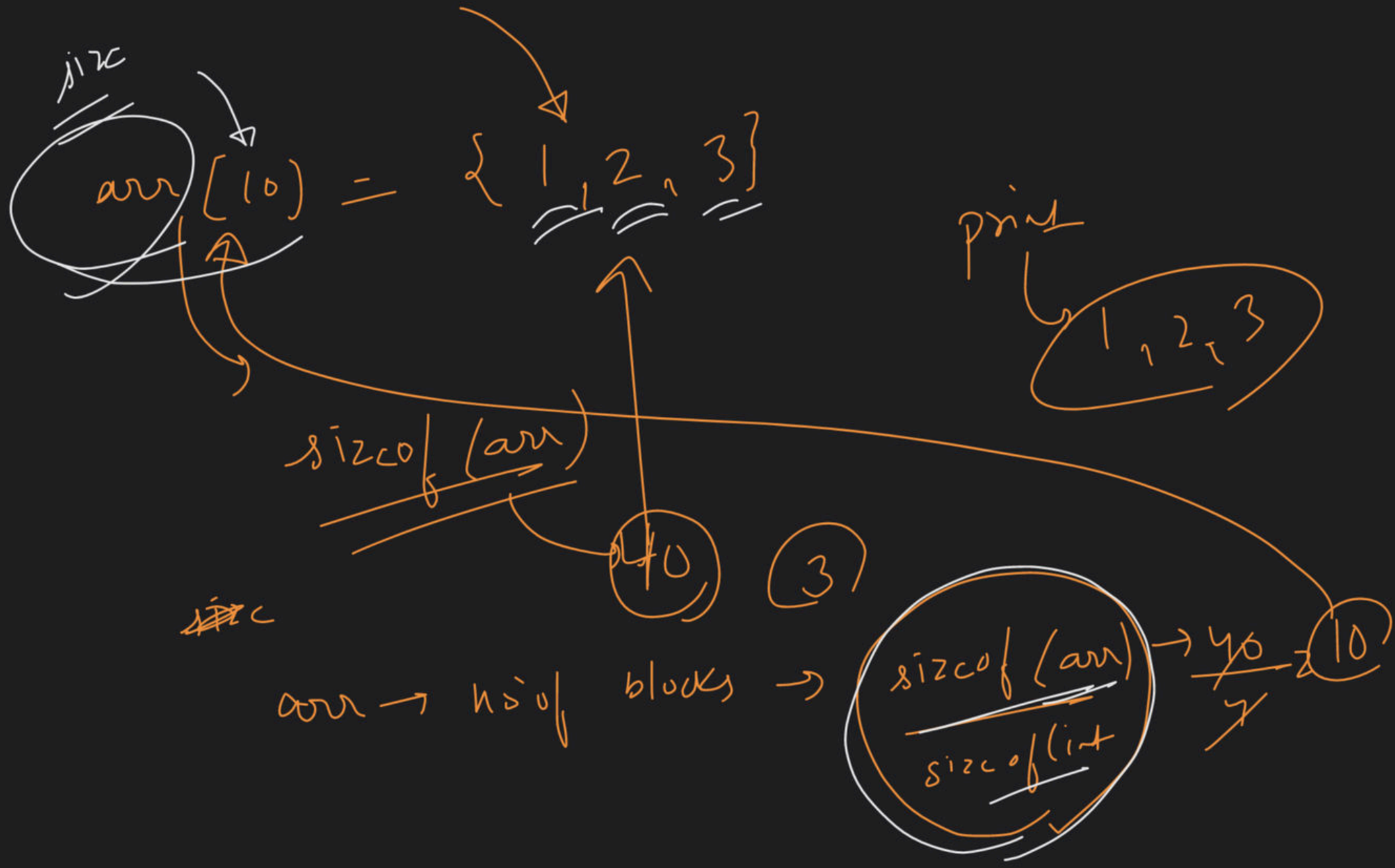

what ?
memset
apply
arr[] = {1, 3, 5, 7, 9}
arr[] → {1, 1, 1, 1, 1}

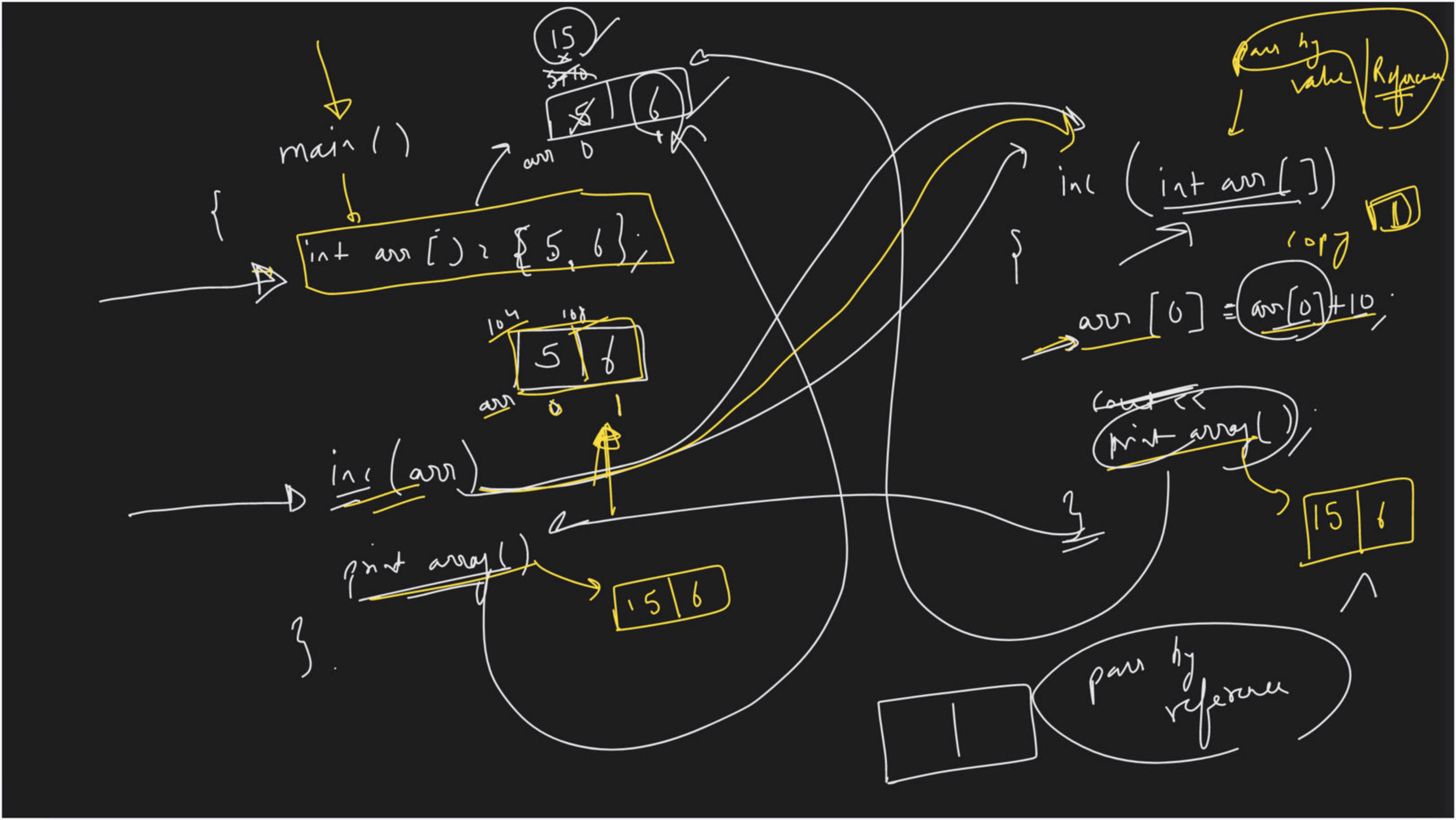
int arr[10] = {1, 2}



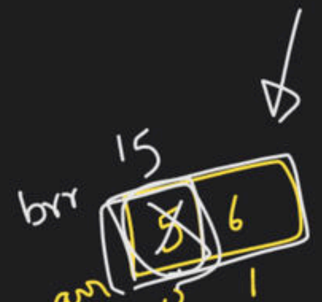
Arrays && Functions:




size
arr[10] = {1, 2, 3}
print
1, 2, 3
sizeof(arr)
40
3
sizeof
arr → no of blocks →
sizeof(arr)
sizeof(int
40
4
10


main()
{
int arr[] = {5, 6};
arr
0
15
5
6
104
108
5
6
arr
0
1
inc(arr)
print array()
15
6
}
inc (int arr[])
{
arr[0] = arr[0]+10;
}
Pass by value / Reference
1
copy
Cout <<
Print array()
15
6
pass by reference
1

```
main()
{
    int arr[] = {5, 6};
    int size = 2;
    fun(arr, size);
    printArray();
}
```



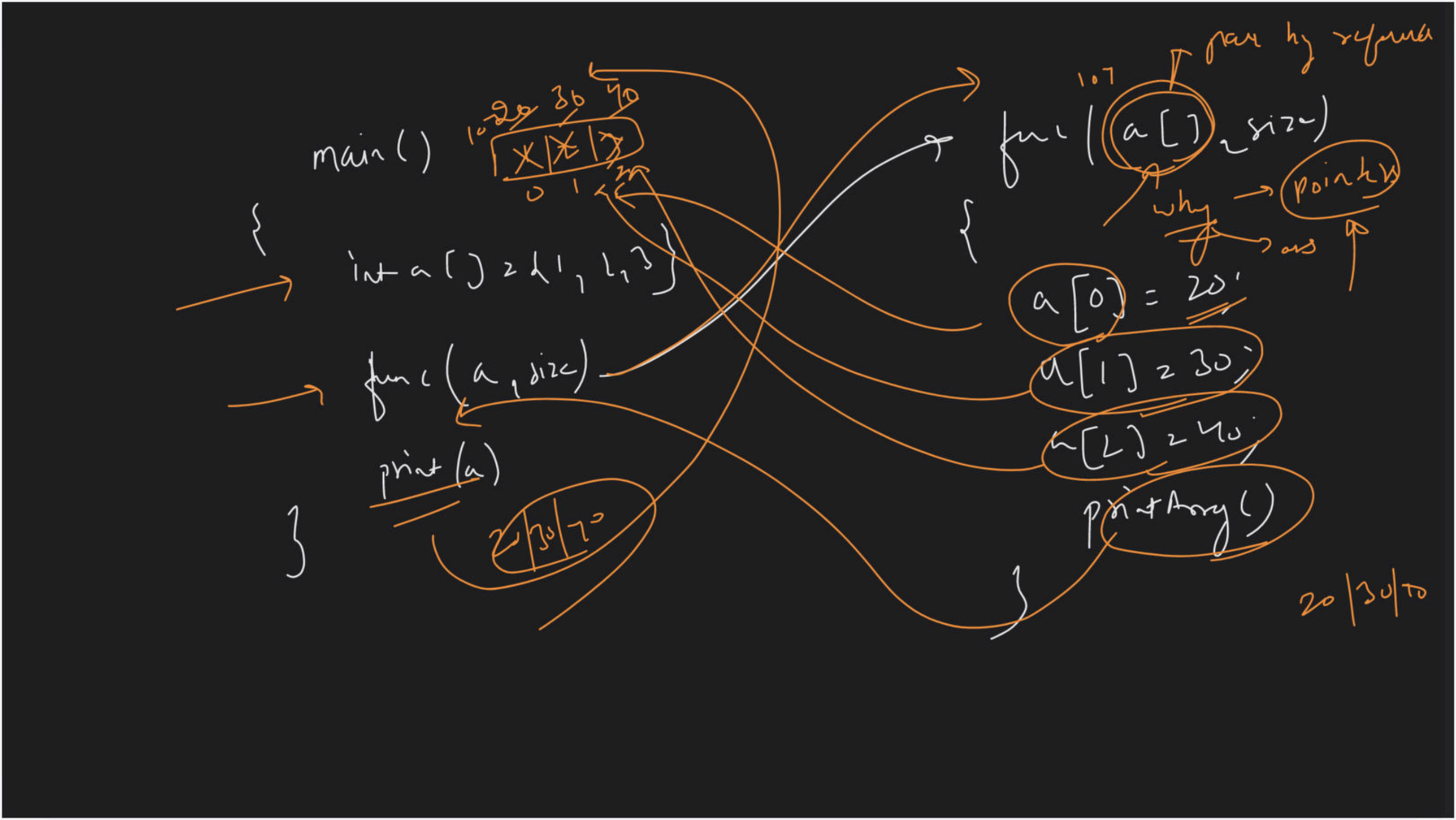

```
fun((int arr[], int size)
{
    arr[0] = arr[0] + 10;
    printArray()
}
```

Array → function

Pass by reference → address

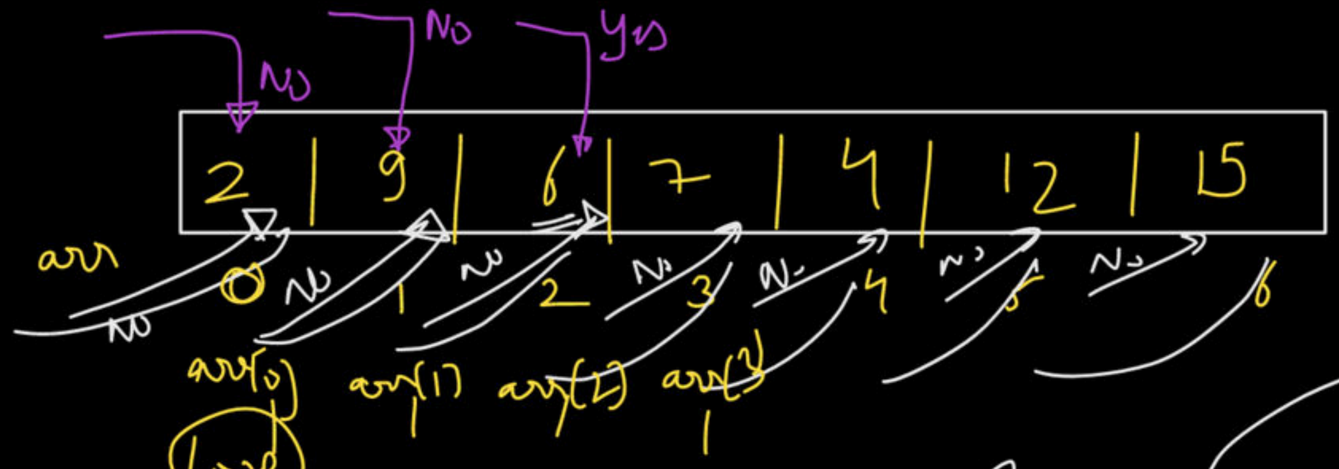
actual array me updation karna ho

copy pass hoti hai


main()
{
int a[] = {1, 2, 3}
func(a, size)
print(a)
}
10 20 30 40
0 1 2
20 30 40
func(a[], size)
{
a[0] = 20;
a[1] = 30;
a[2] = 40;
printArray()
}
pass by reference
107
why
ans
pointer
20 30 40

Linear Search in Array:

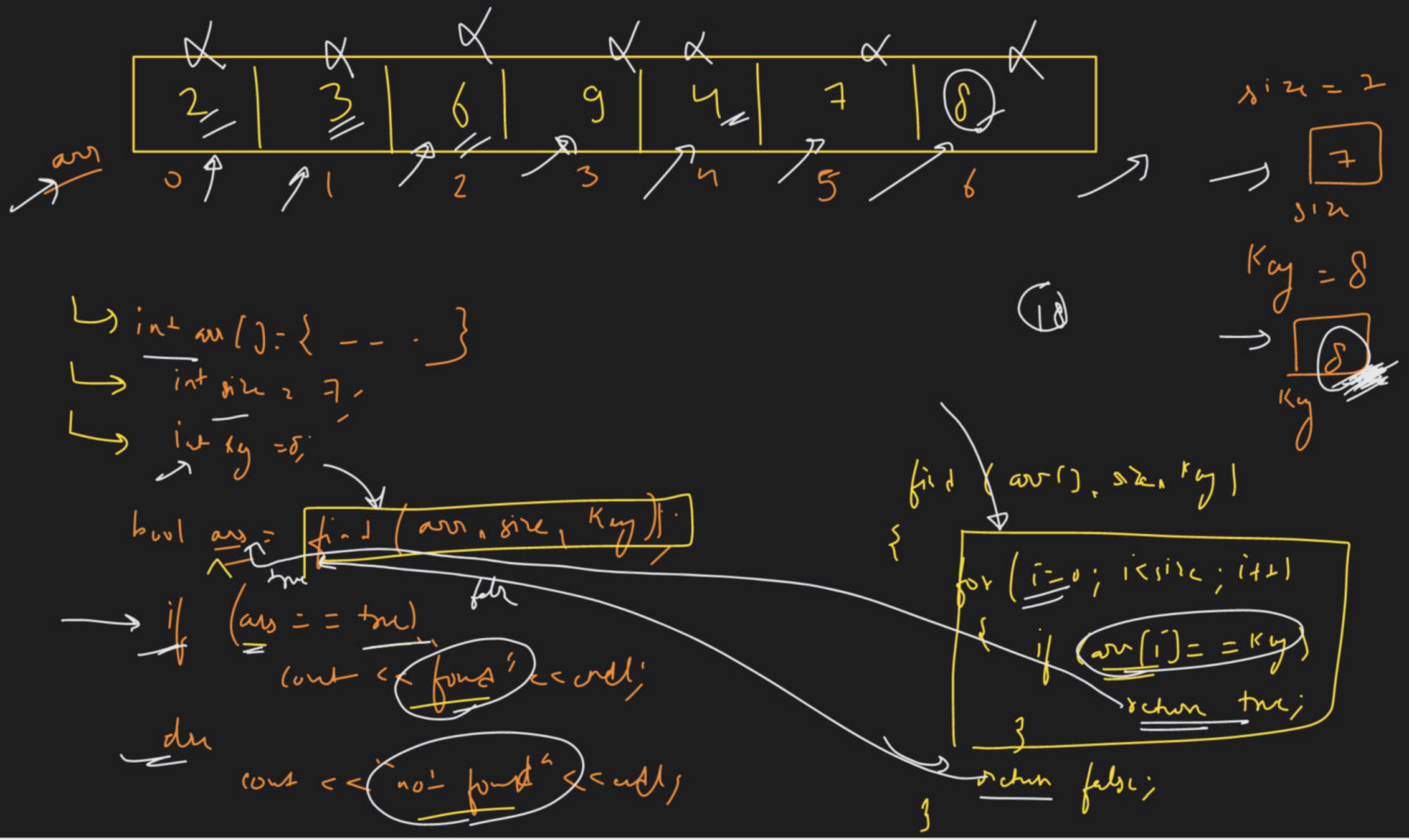
2 min Ka break



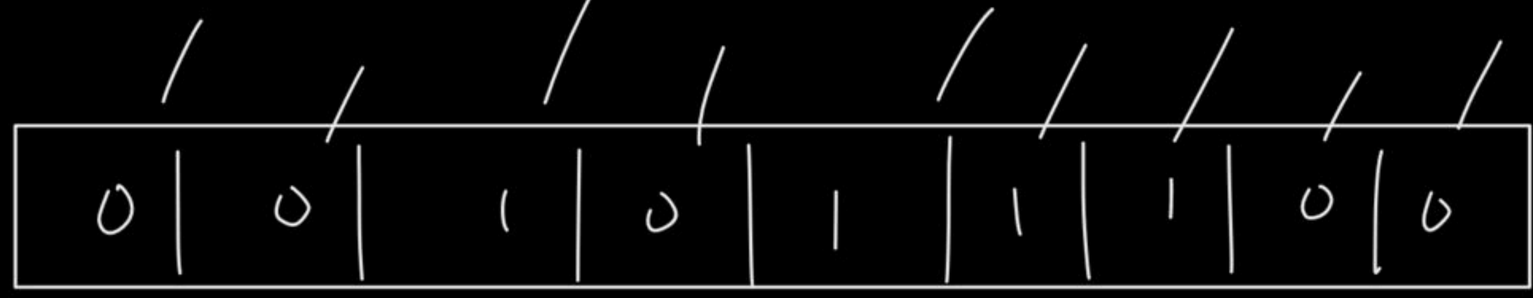
→ 6 is present or Absent in Array → Present → How do you Know

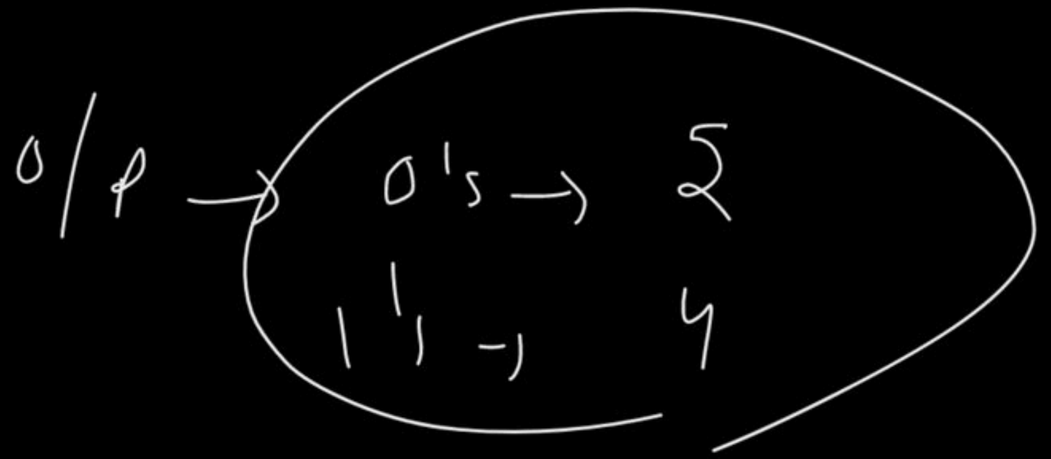
→

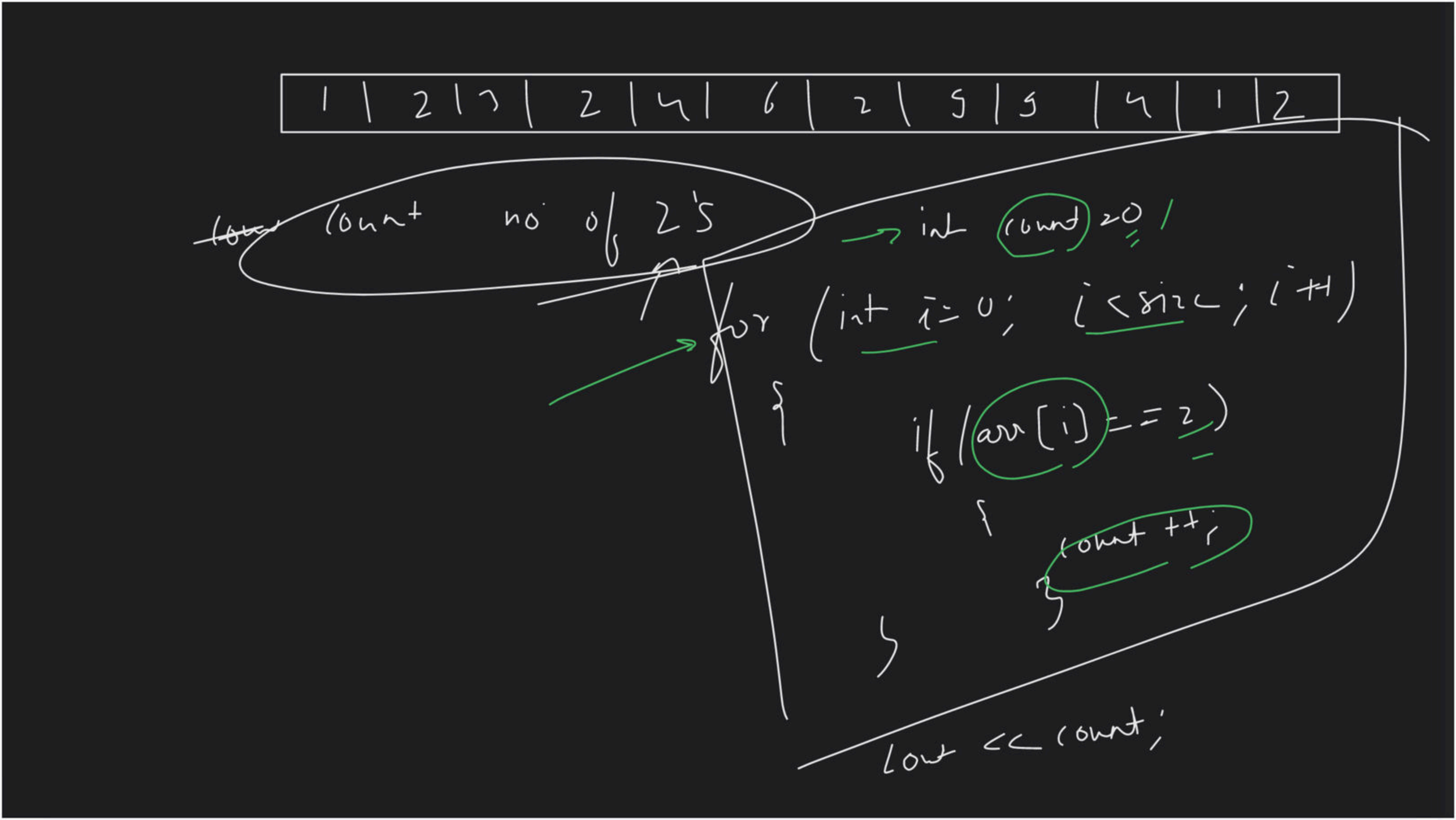
18 → P or A → Absent


2
3
6
9
4
7
8
arr
0
1
2
3
4
5
6
size = 7
size
key = 8
key
int arr[] = { ... }
int size = 7;
int key = 8;
bool ans = find (arr, size, key);
true
false
if (ans == true)
cout << "found" << endl;
else
cout << "not found" << endl;
find (arr[], size, key)
{
for (i=0; i<size; i++)
{ if (arr[i] == key)
return true;
}
return false;
}

Count 0's and 1's in Array






1 | 2 | 3 | 2 | 4 | 6 | 2 | 5 | 5 | 4 | 1 | 2
count no of 2's
int count = 0;
for (int i = 0; i < size; i++)
{
if (arr[i] == 2)
{
count++;
}
}
cout << count;

minNum → INT_MAX

Maximum number in an Array:

int → -2^{31} → $2^{31}-1$

INT_MIN

INT_MAX

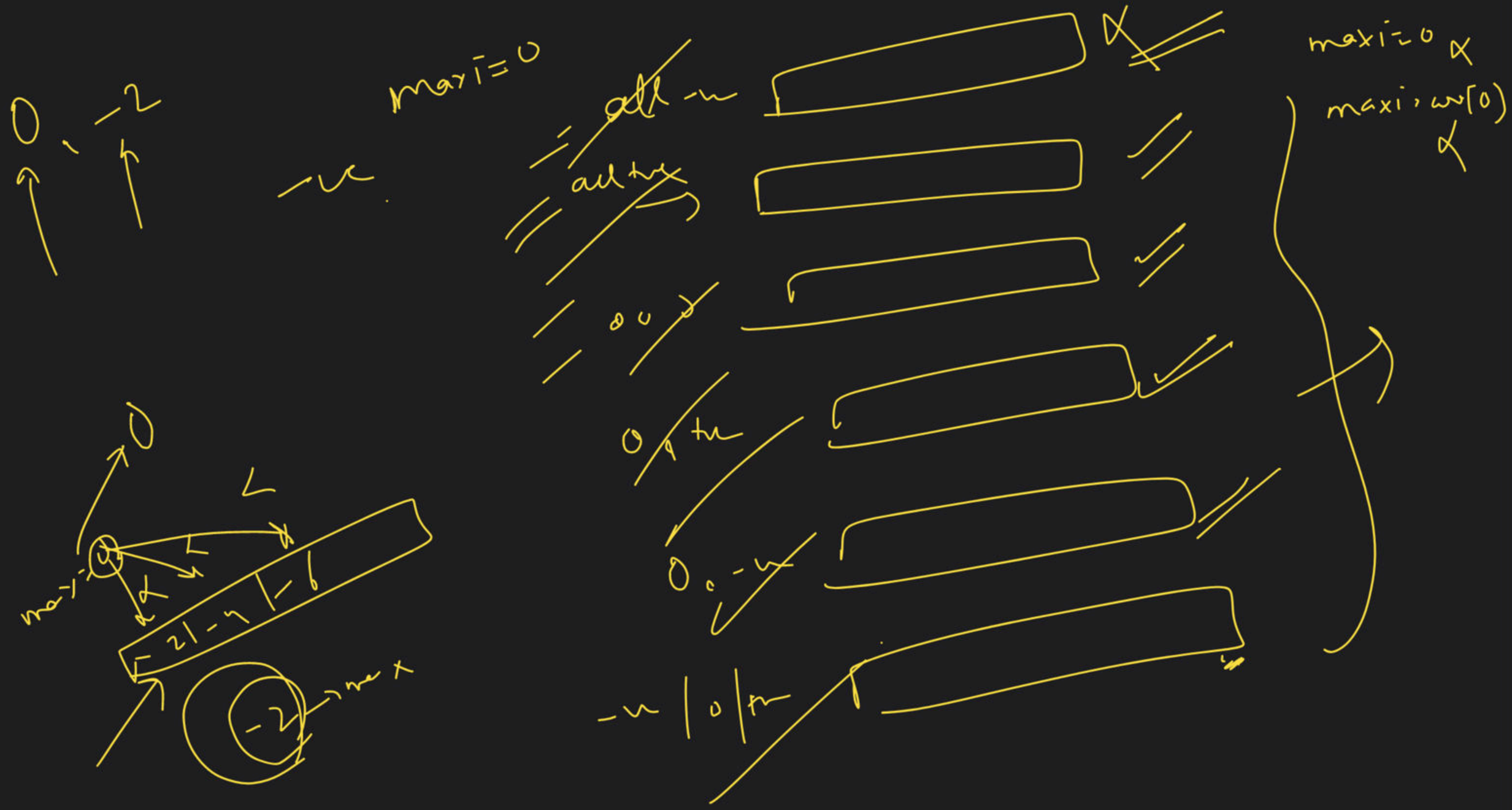
int maxNum = ~~-1~~ INT_MIN;

```
for(i=0; i<size; i++)
{   if (arr[i] > maxNum)
        maxNum = arr[i];
}
```

max → 9

But Practically

INT_MIN


maxi=0
maxi=0
maxi : ω(0)

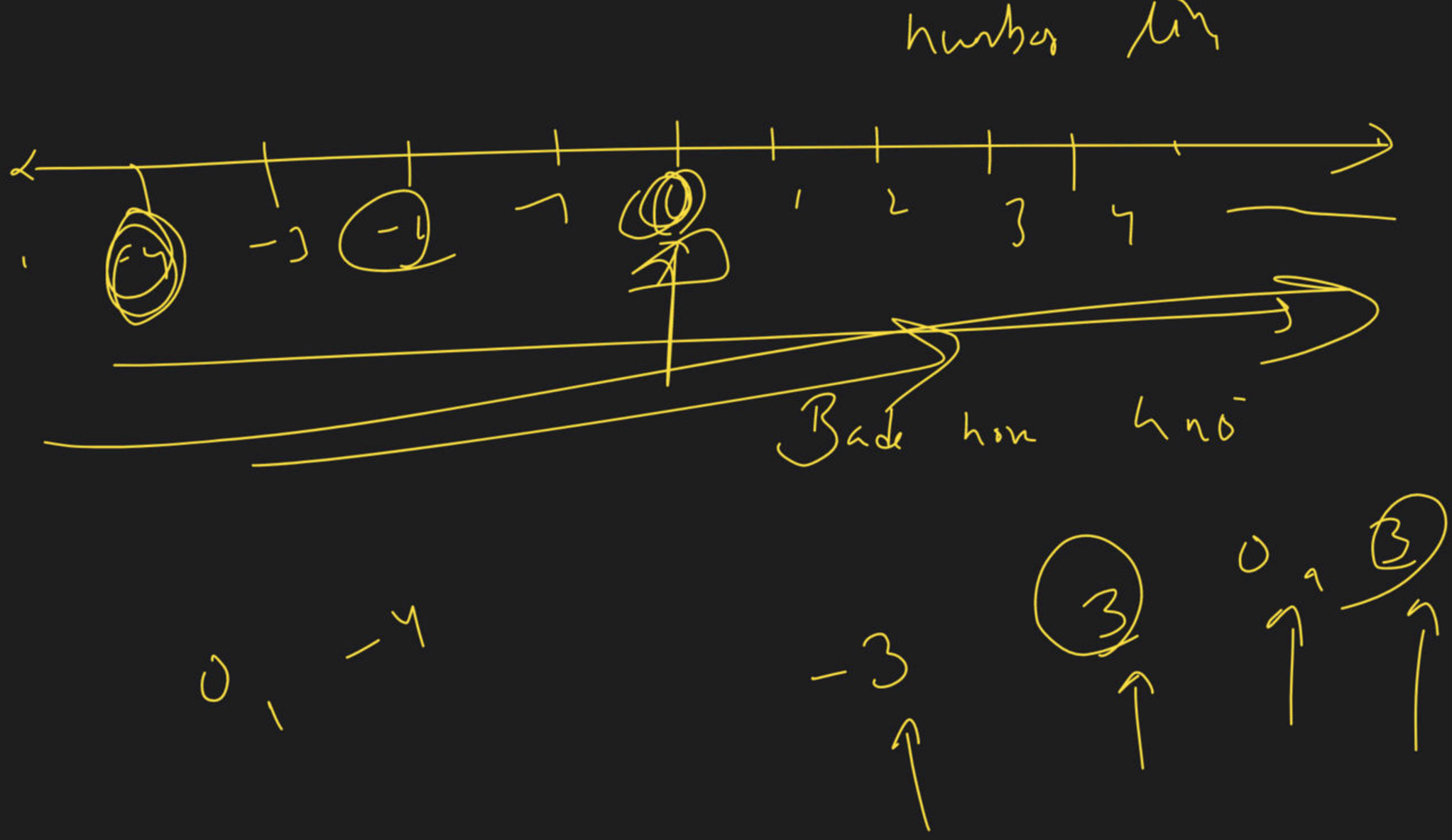
n/w → array → i/p

maxi → INT_MIN

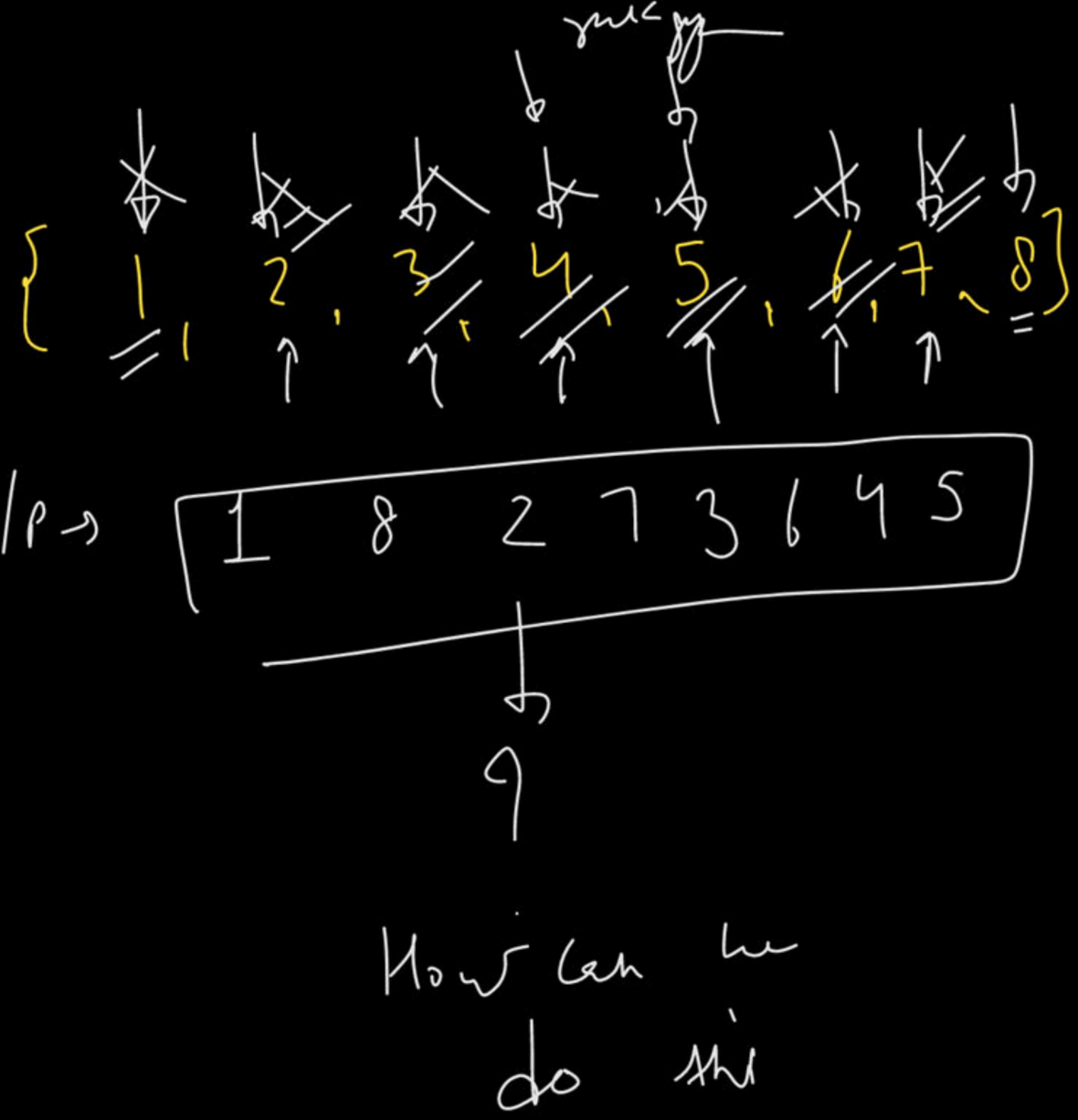
mini → INT_MAX

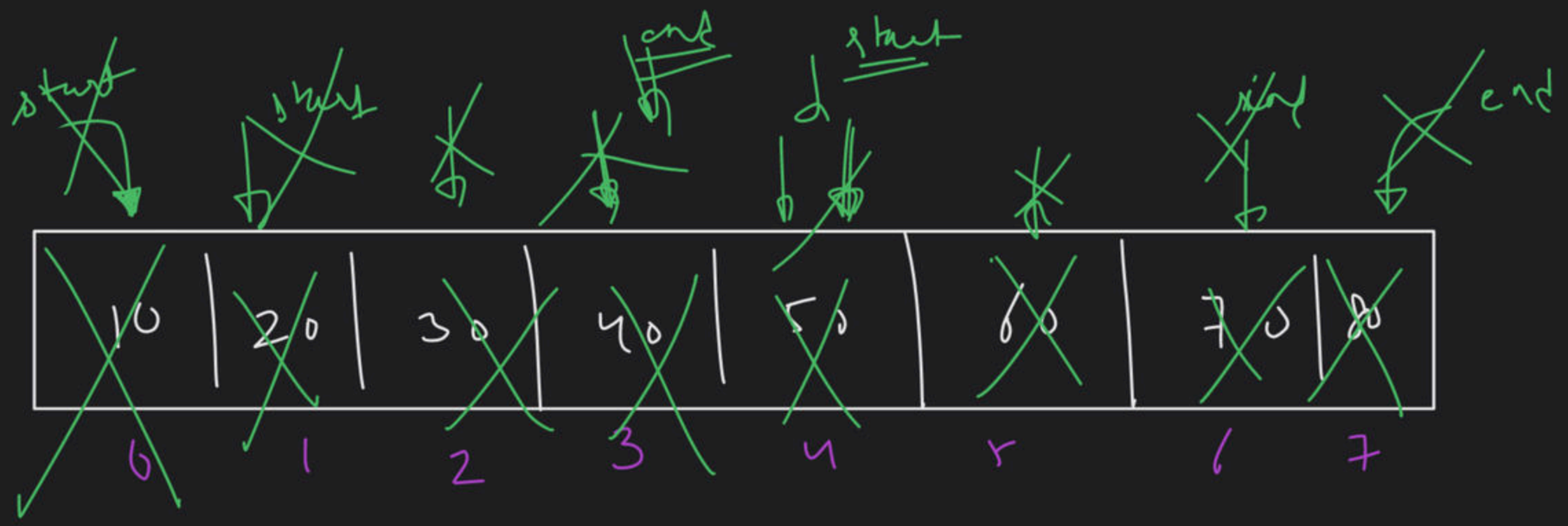
minimum nó

find out → on the spot


number line
-3
-1
0
1
2
3
4
Back hon hno
0, -4
-3
3
0, 3

Extreme Print in Array:



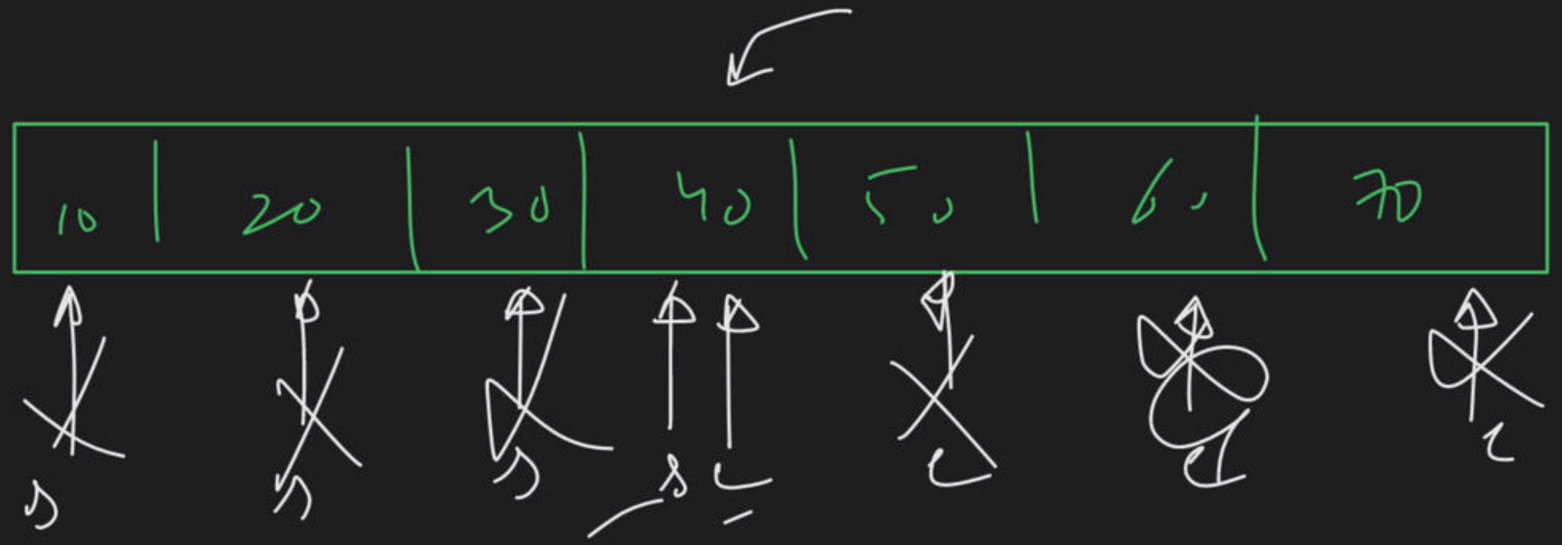


Rukna Kab ?

start > end
to
rukna hai

{
print arr [start]
print arr [end]
start ++ ;
end -- ;

loop



10 70 20 60 30 50 40 40

start > end

↓

return null

start < 2 end

↓

loop under

h

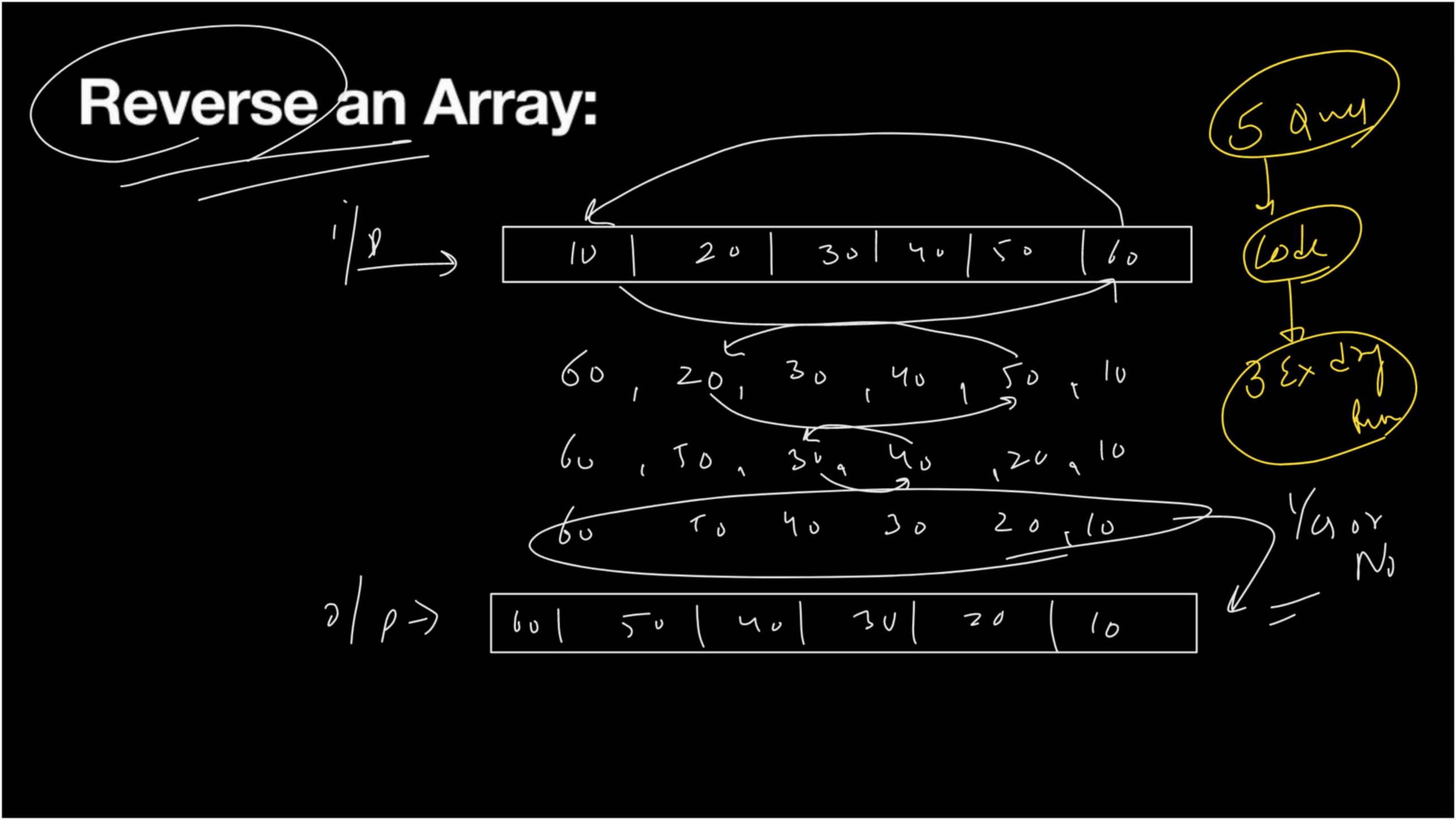
if (start == end)

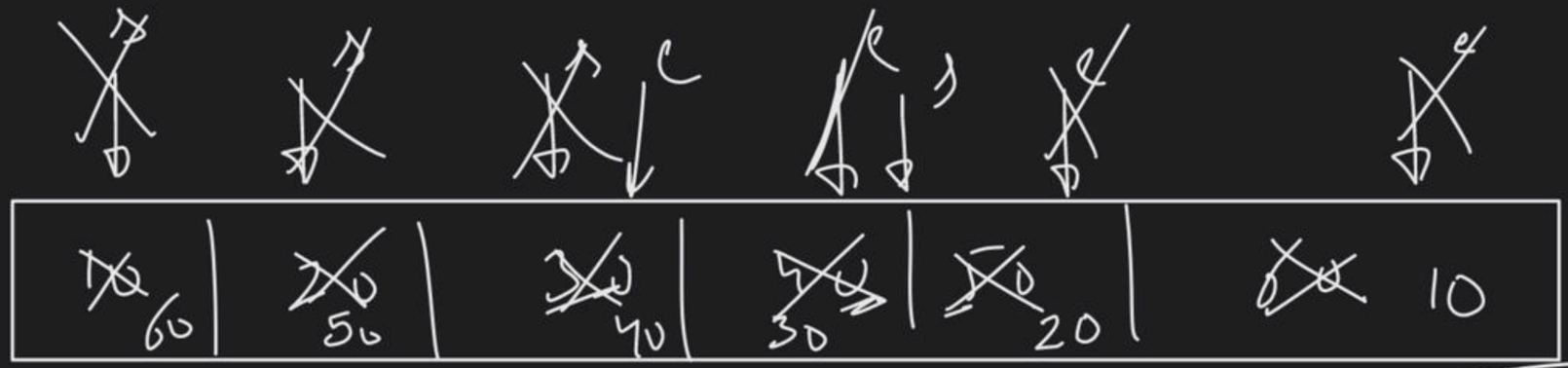
{

1 base print

}

Reverse an Array:



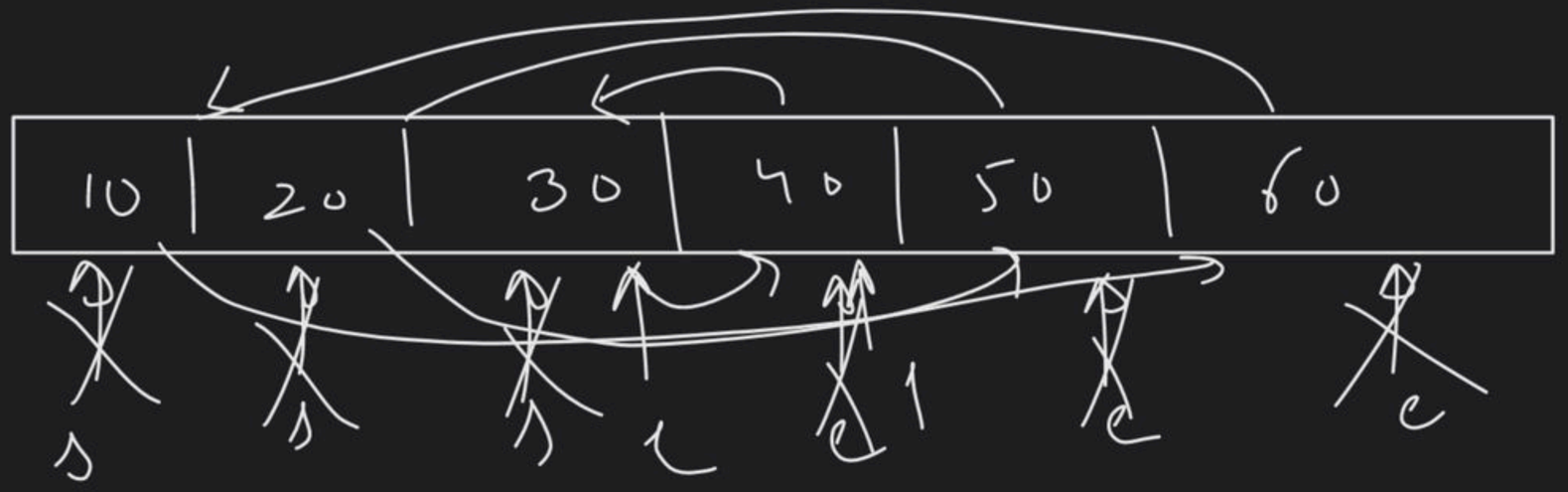


b > c → make him

2 Element swap

↳ pre-defined function → swap(a, b)

swap(arr[s], arr[e])

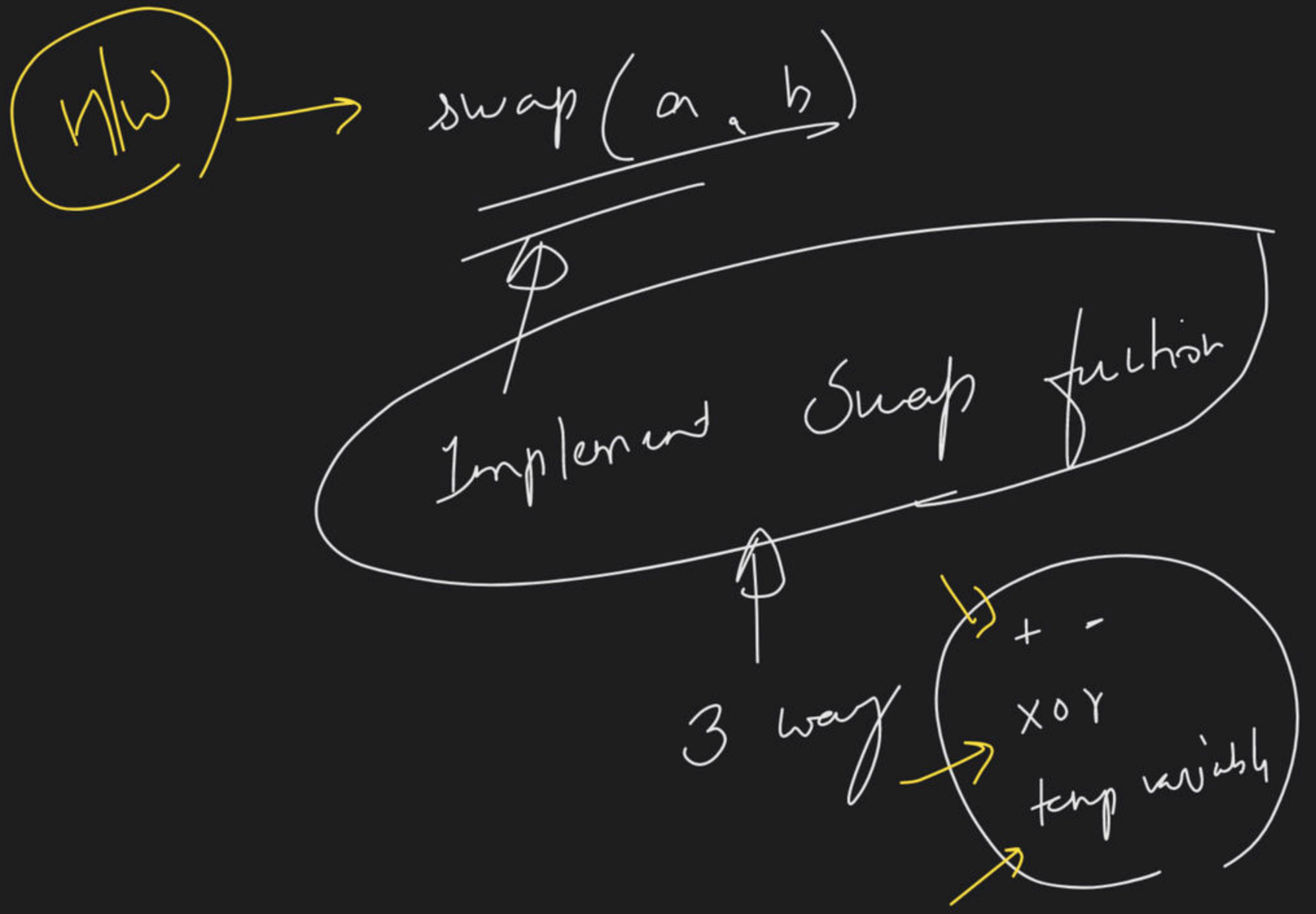


s = 0
e = n - 1

① swap (arr[s], arr[e])
② s++;
③ e--

Expect

loop use


n/w
swap(a, b)
Implement Swap function
3 way
+ -
X O Y
temp variable