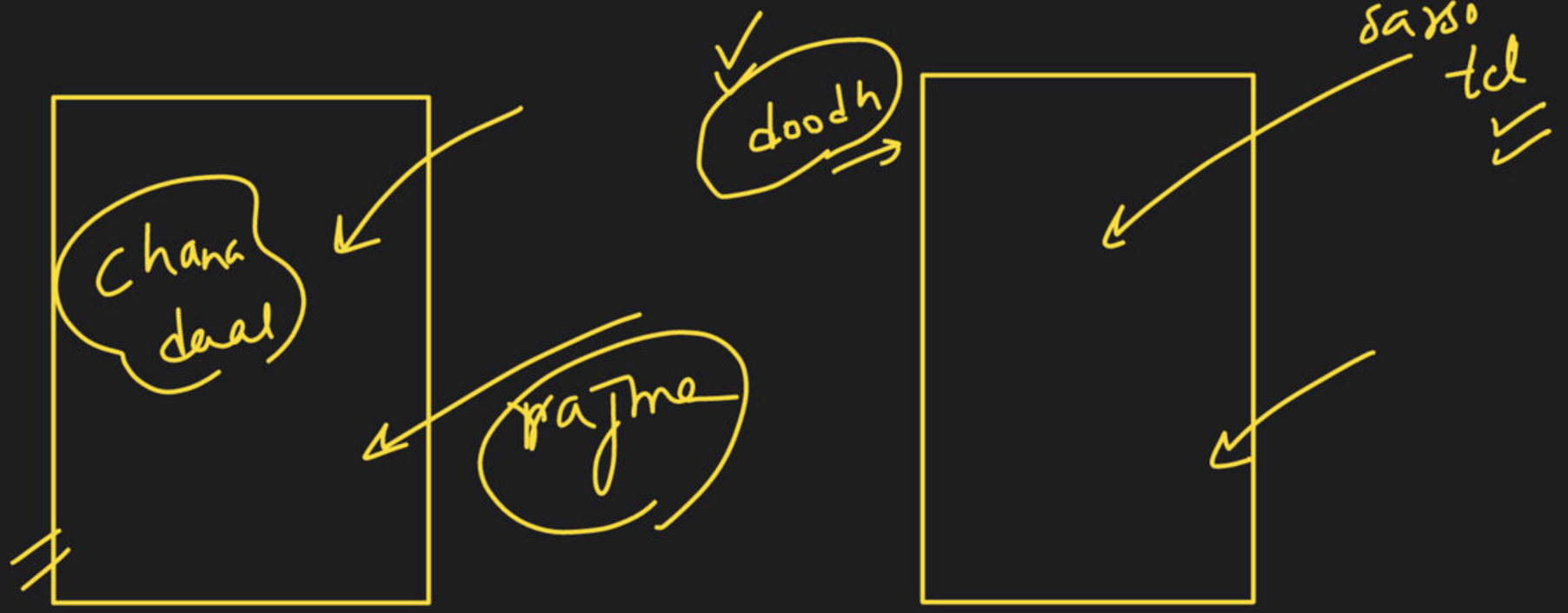
Arrays → packet / container / Bucket

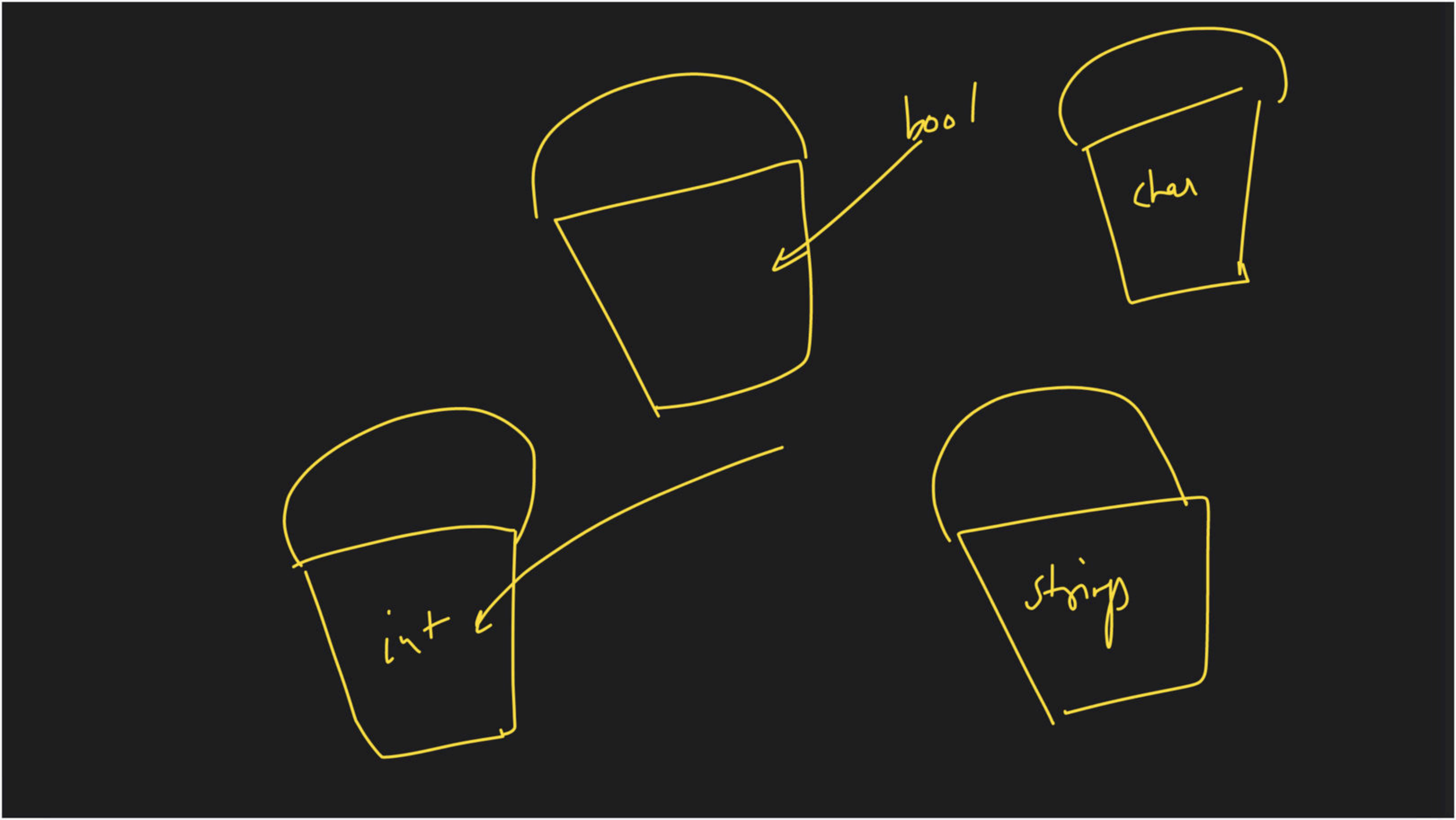
what?

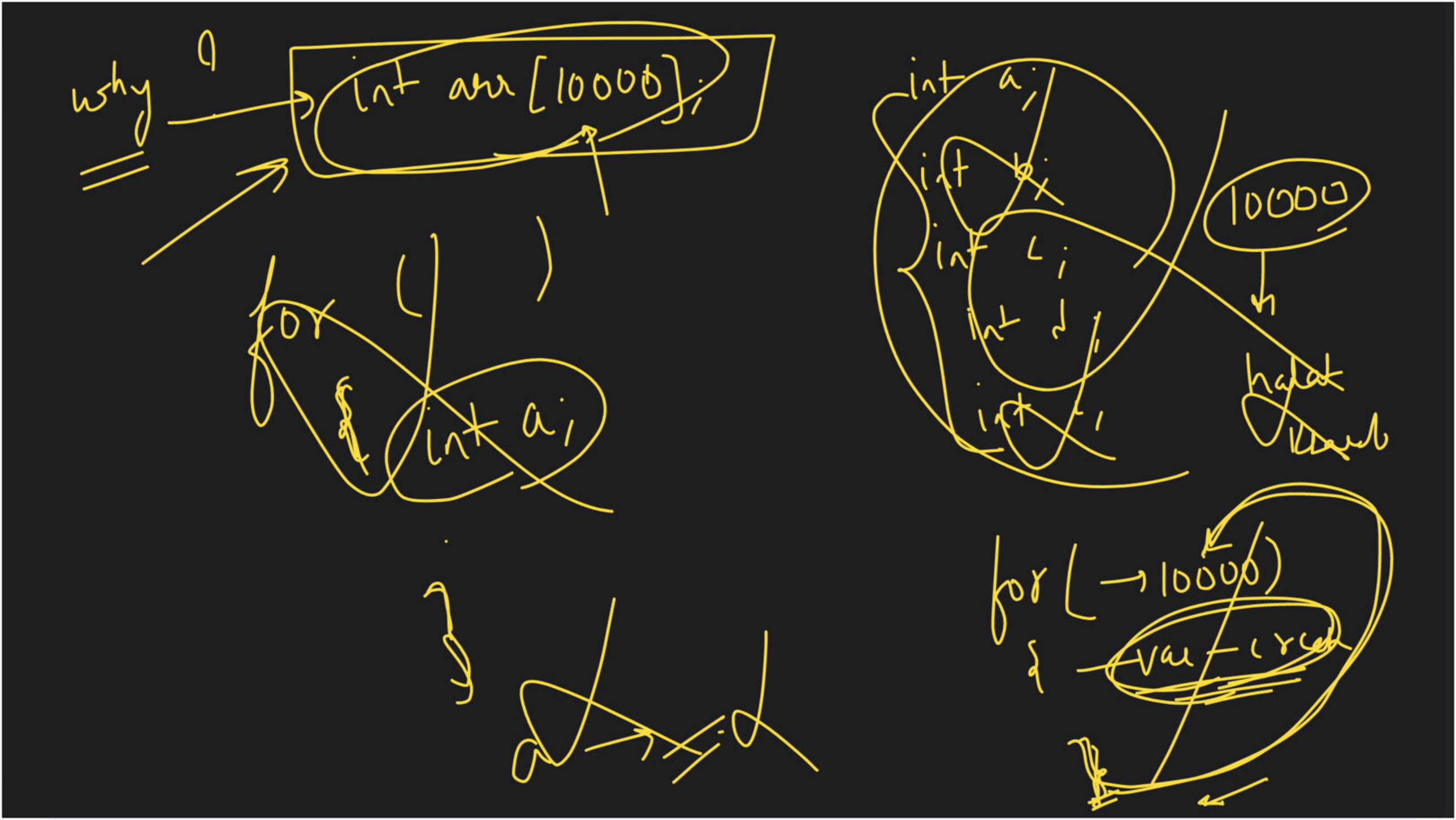
same type items

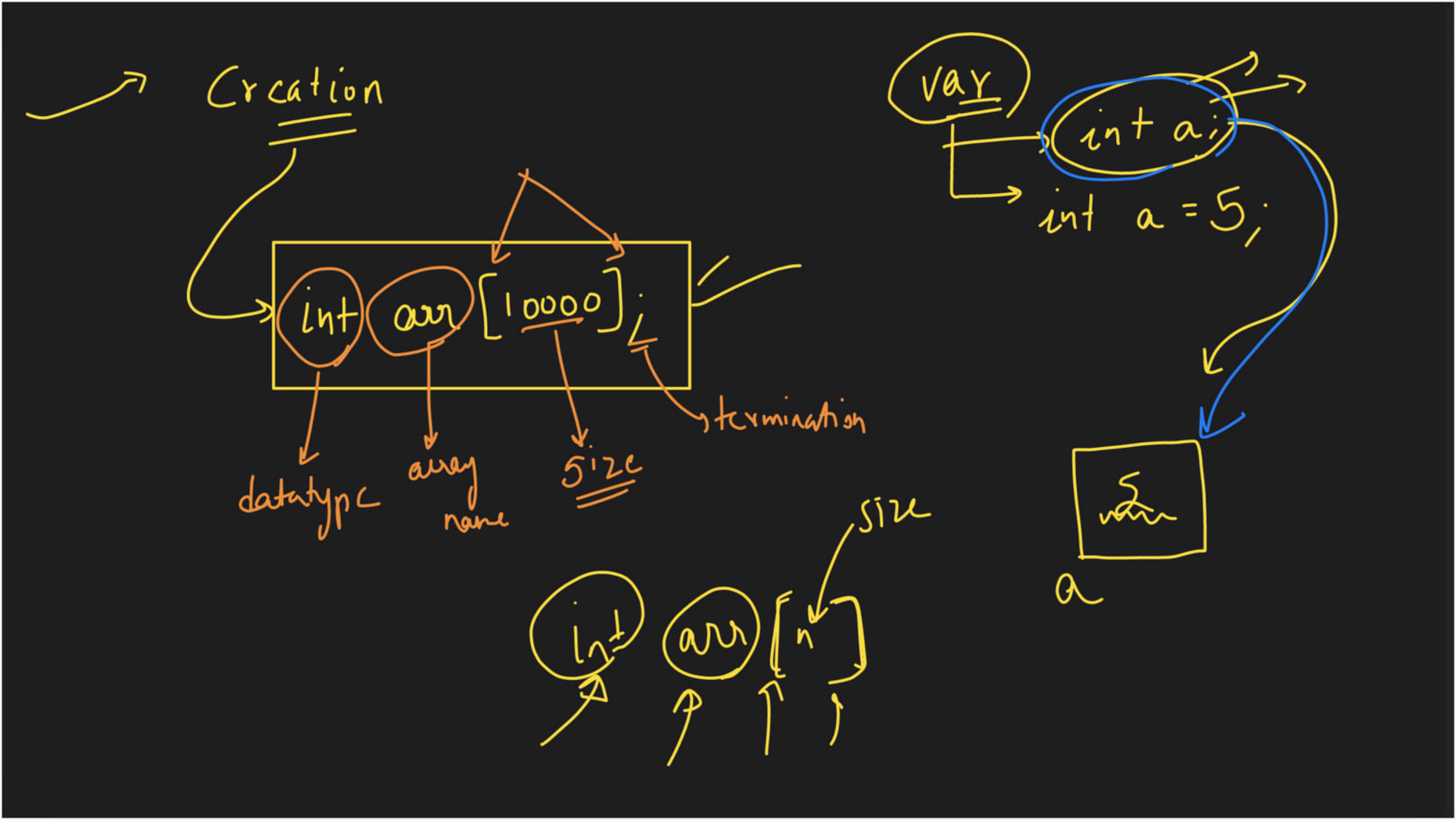
store

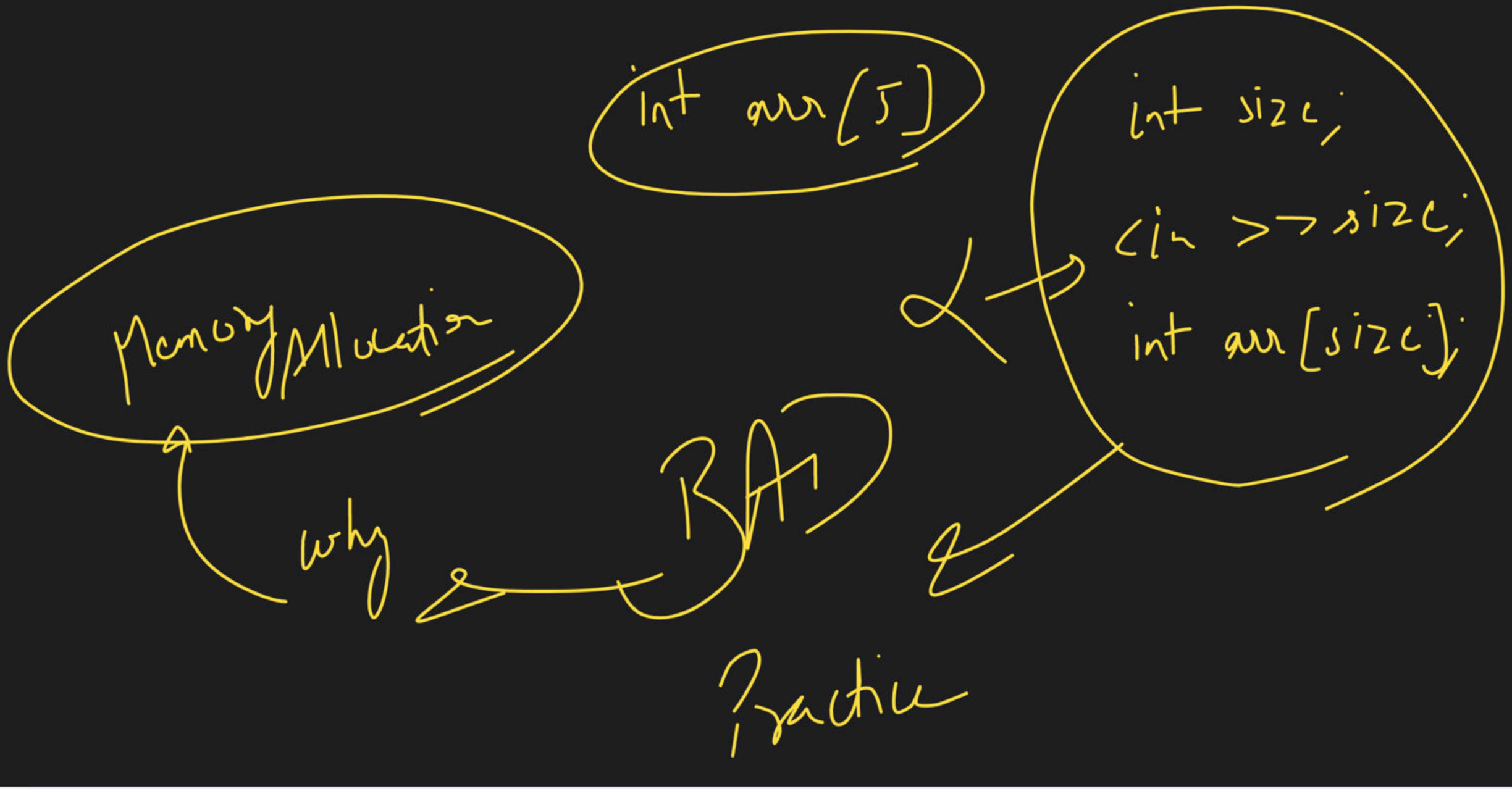
Lag
- ycloud
- net connection



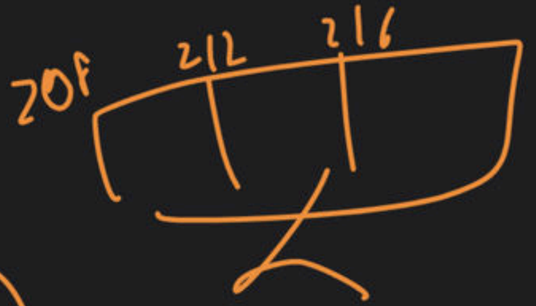

bool
char
int
string


why ?
int arr[10000];
for ( )
{
int a;
}
int a;
int b;
int c;
int d;
int e;
10000
halt
for (→10000)
{
val = (rand


Creation
int arr [10000];
datatype
array name
size
termination
var
int a;
int a = 5;
5
a
int arr [n];
size

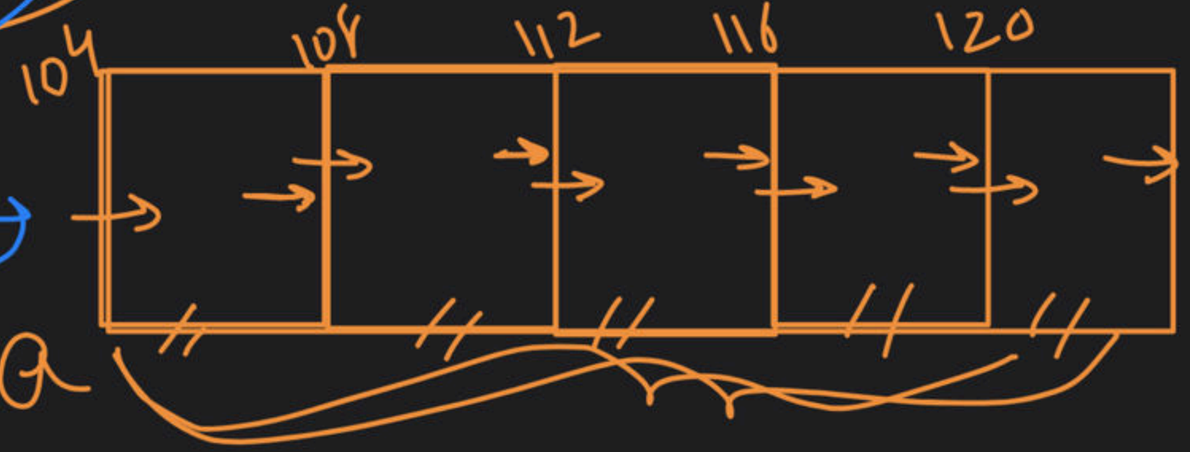

int arr[5]
int size;
cin >> size;
int arr[size];
Memory Allocation
why BAD
Practice

int a;

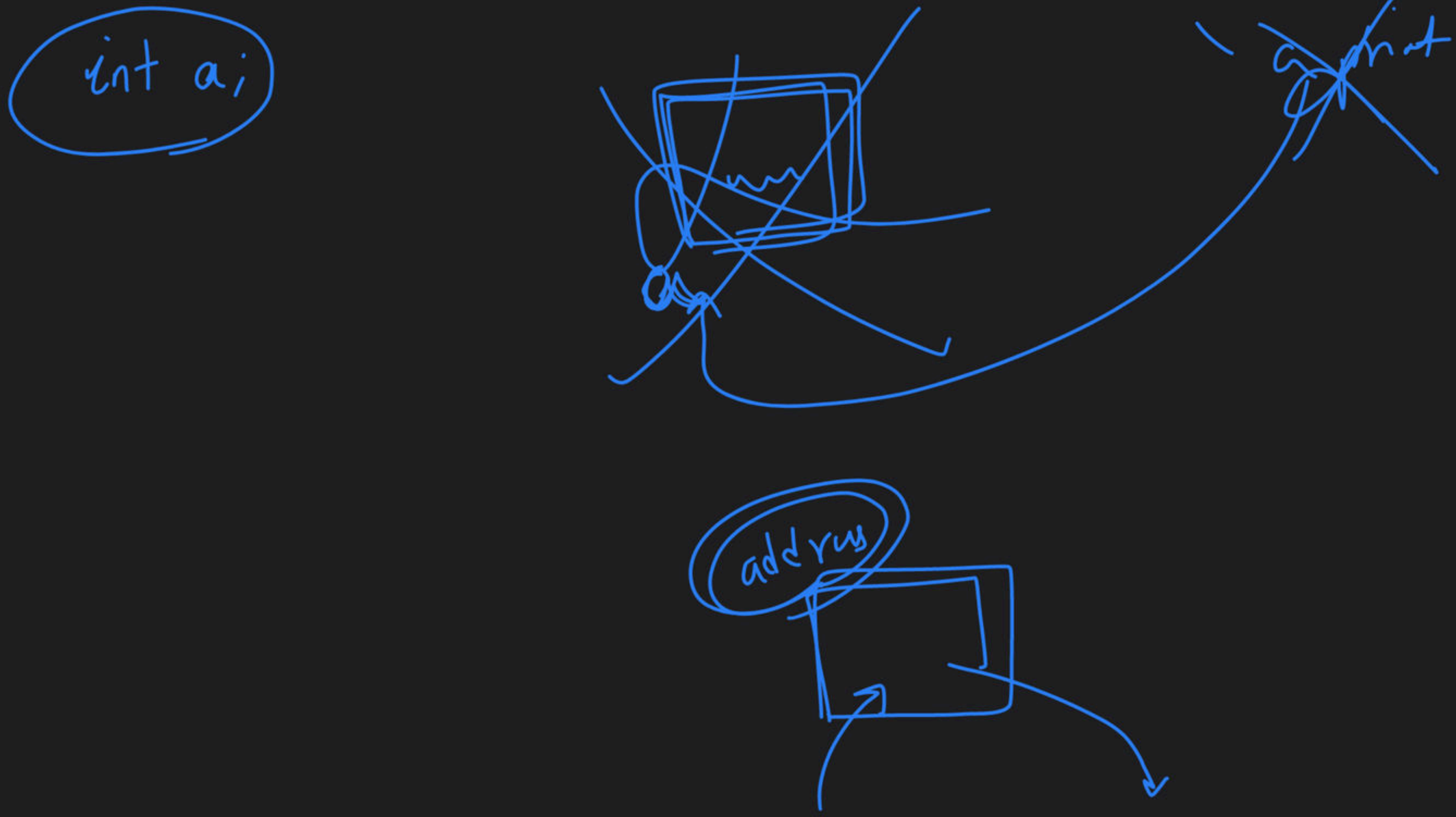


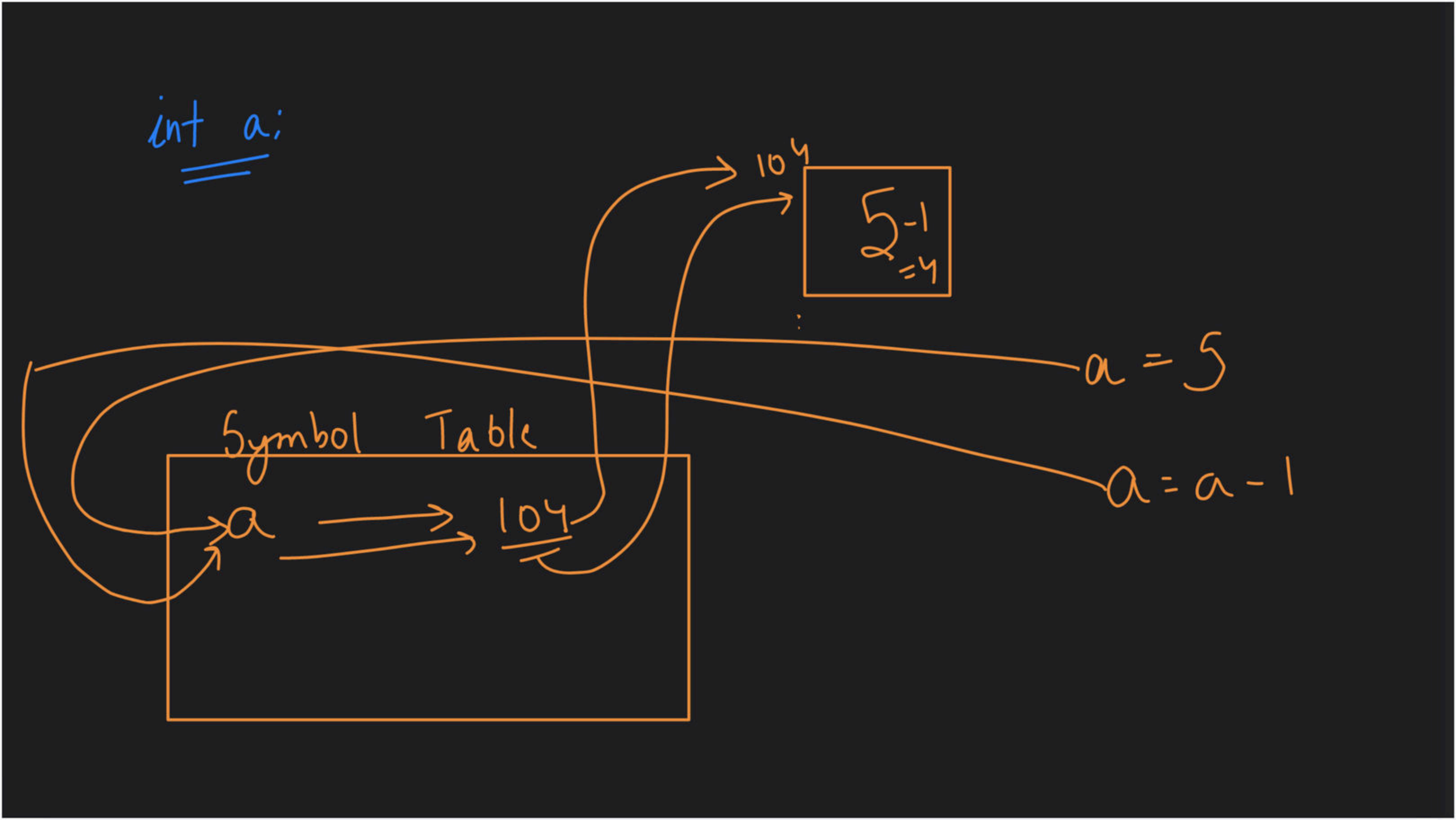
int a[5];

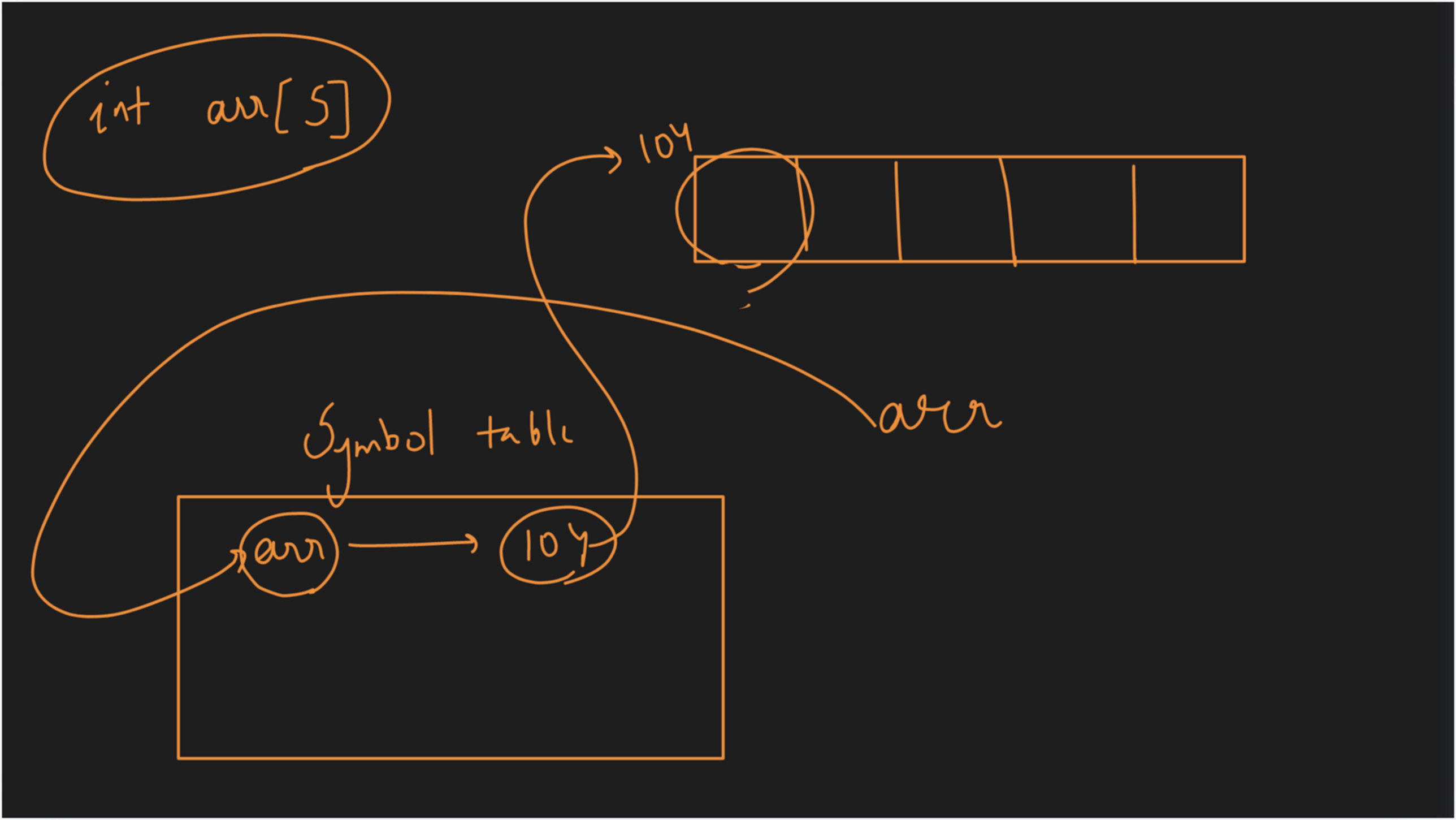
Continous

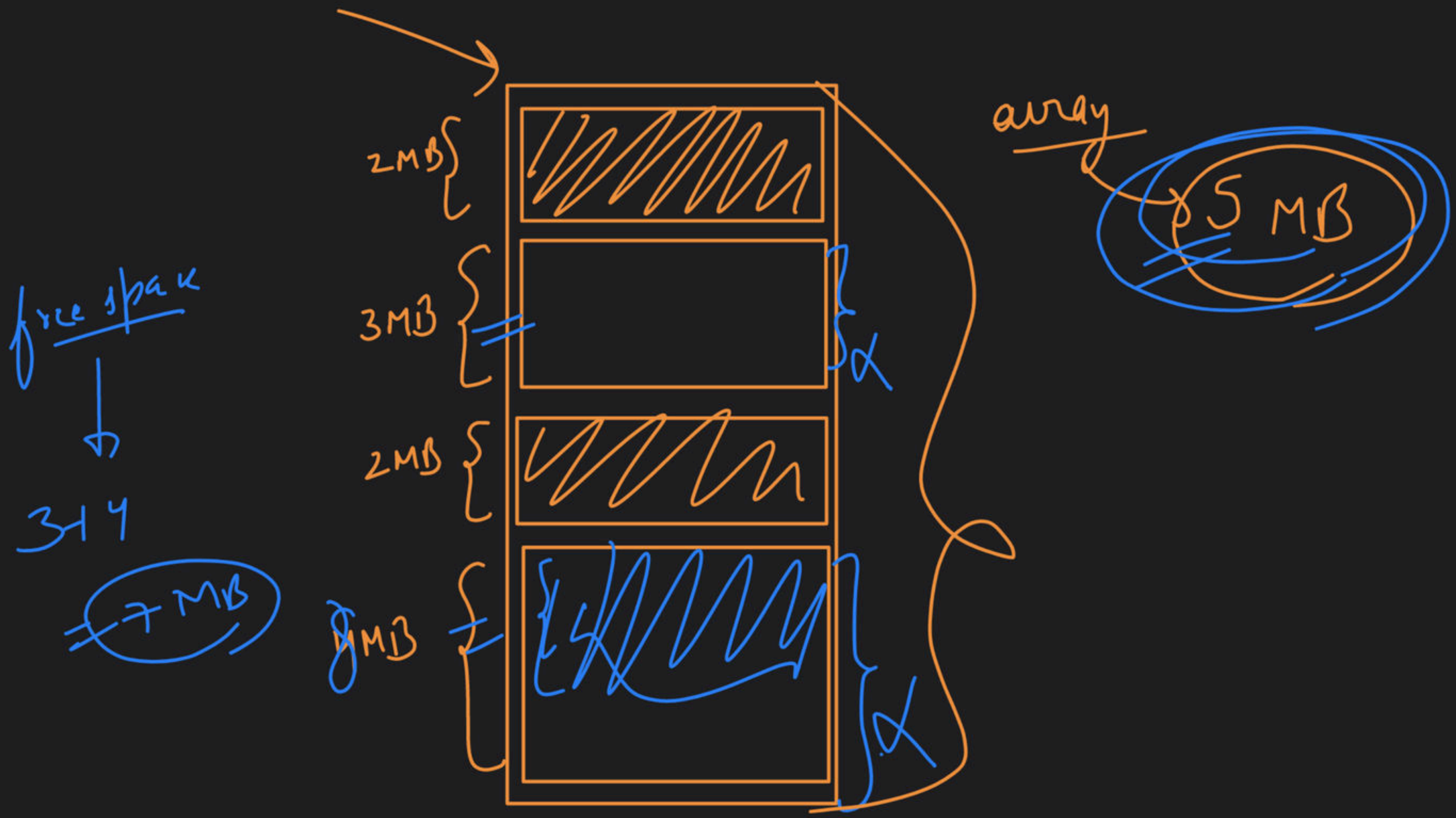


galat

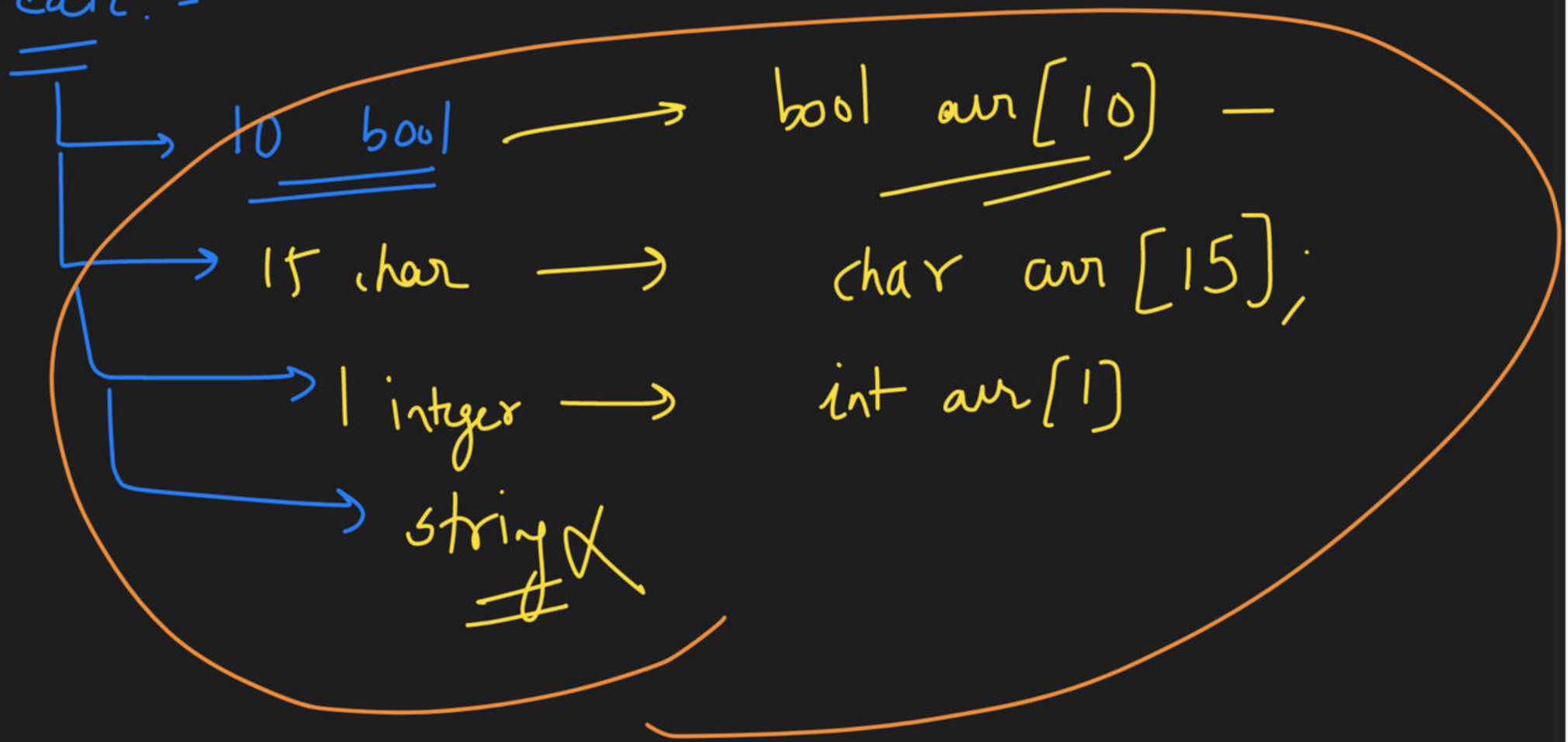

int a;
addrus

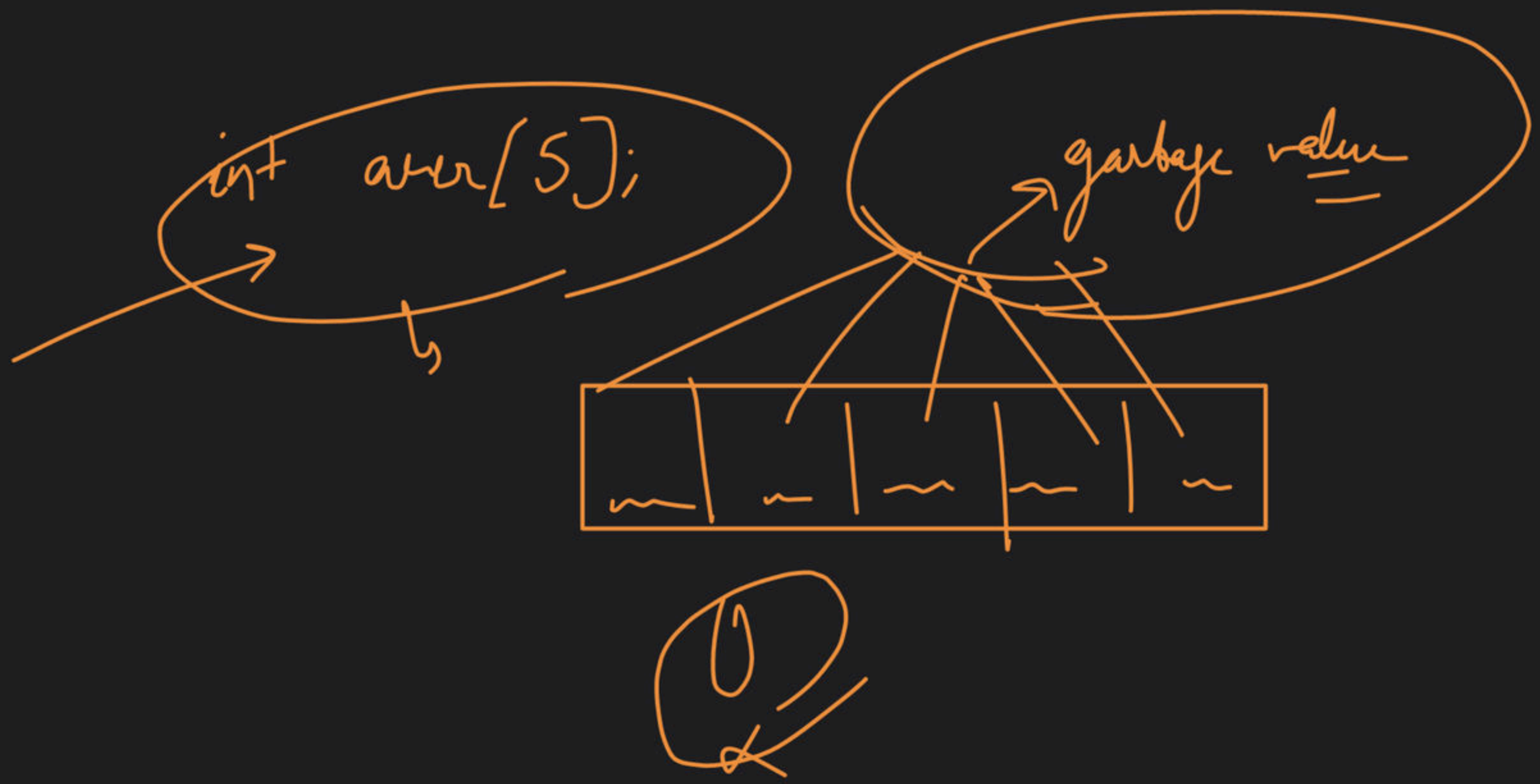

int a;
104
5
-1
=4
a = 5
a = a - 1
Symbol Table
a
104

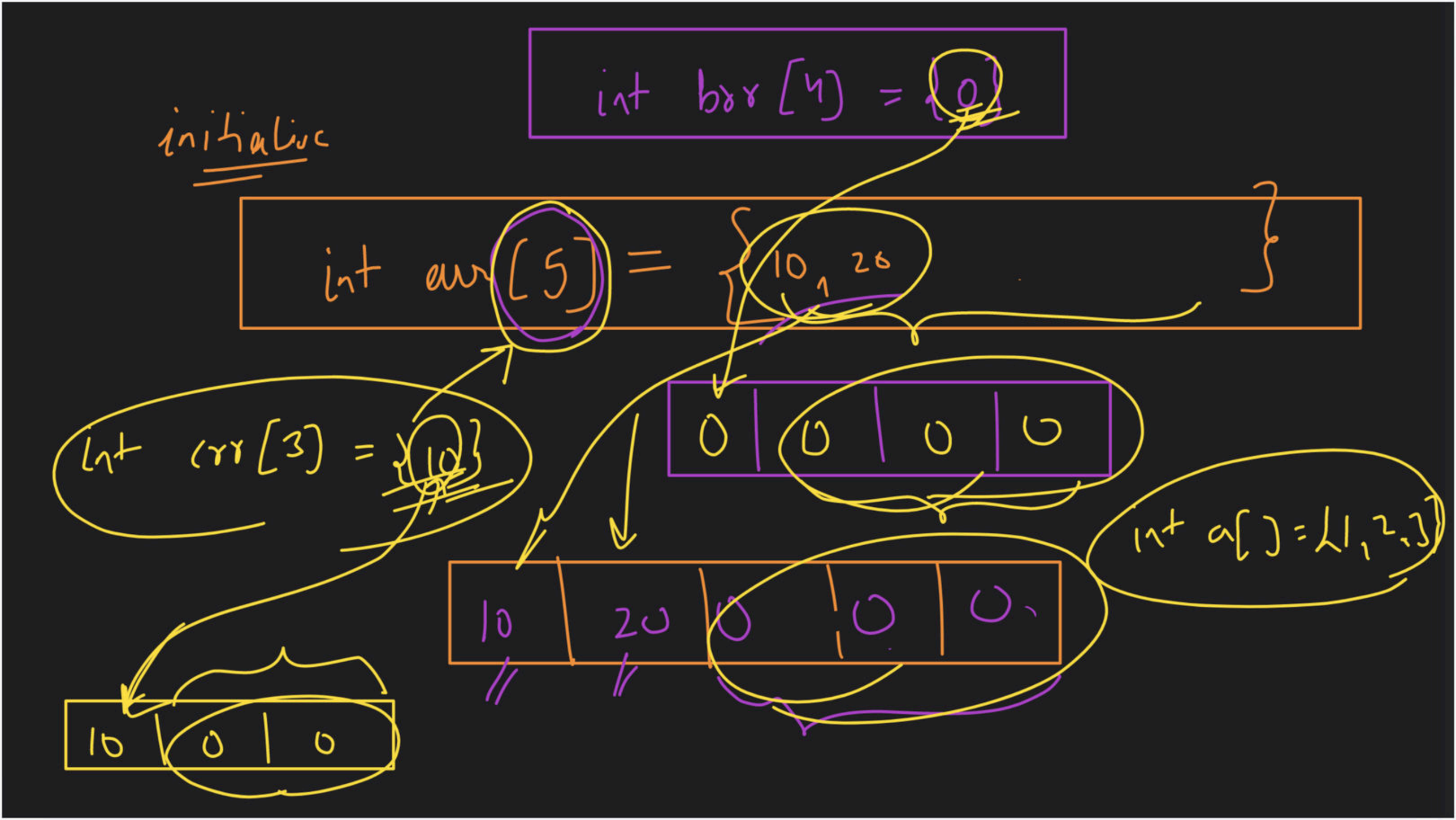

int arr[5]
104
arr
Symbol table
arr
104

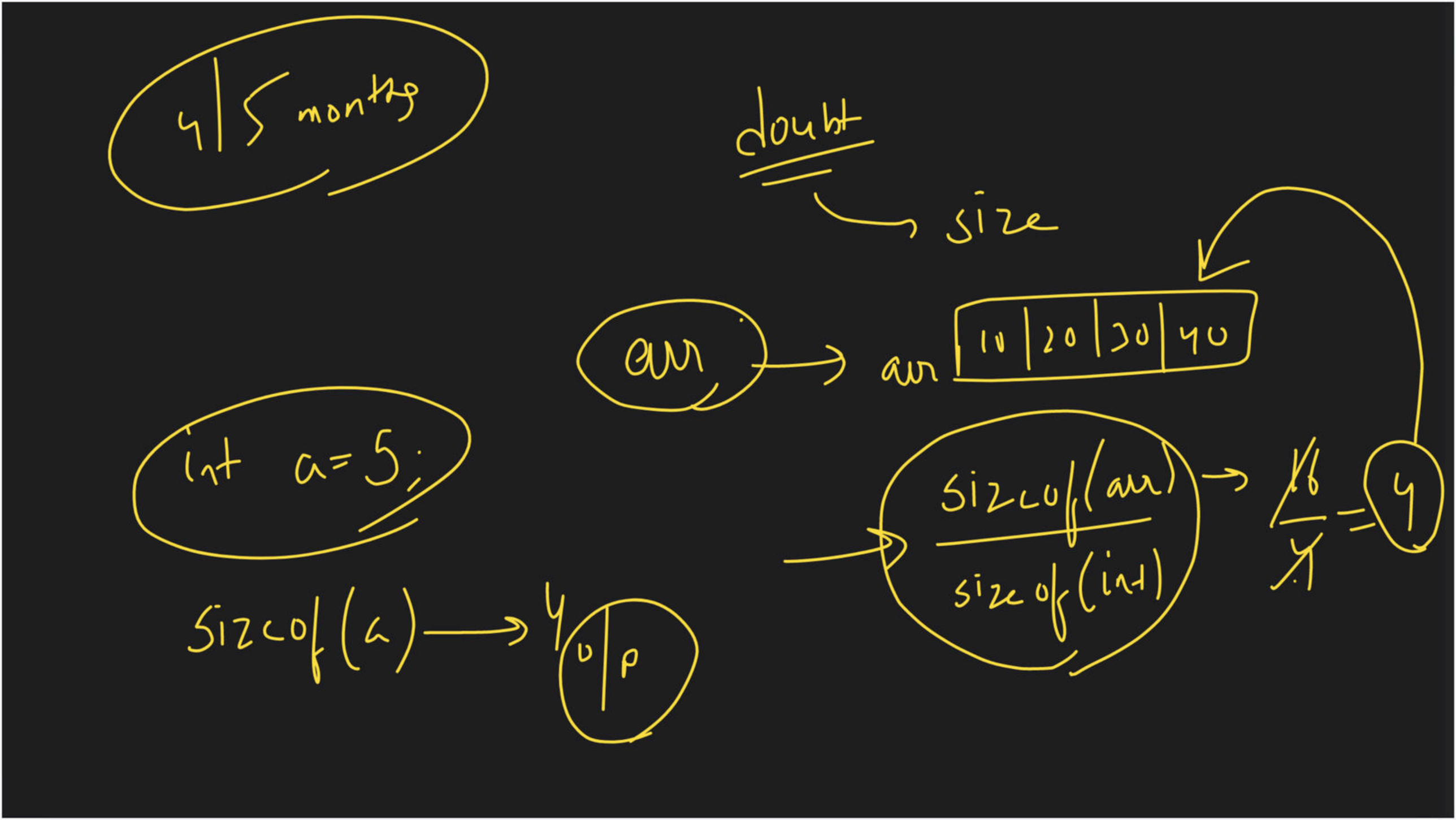

2MB
3MB
2MB
8MB
array
5 MB
free space
314
7 MB

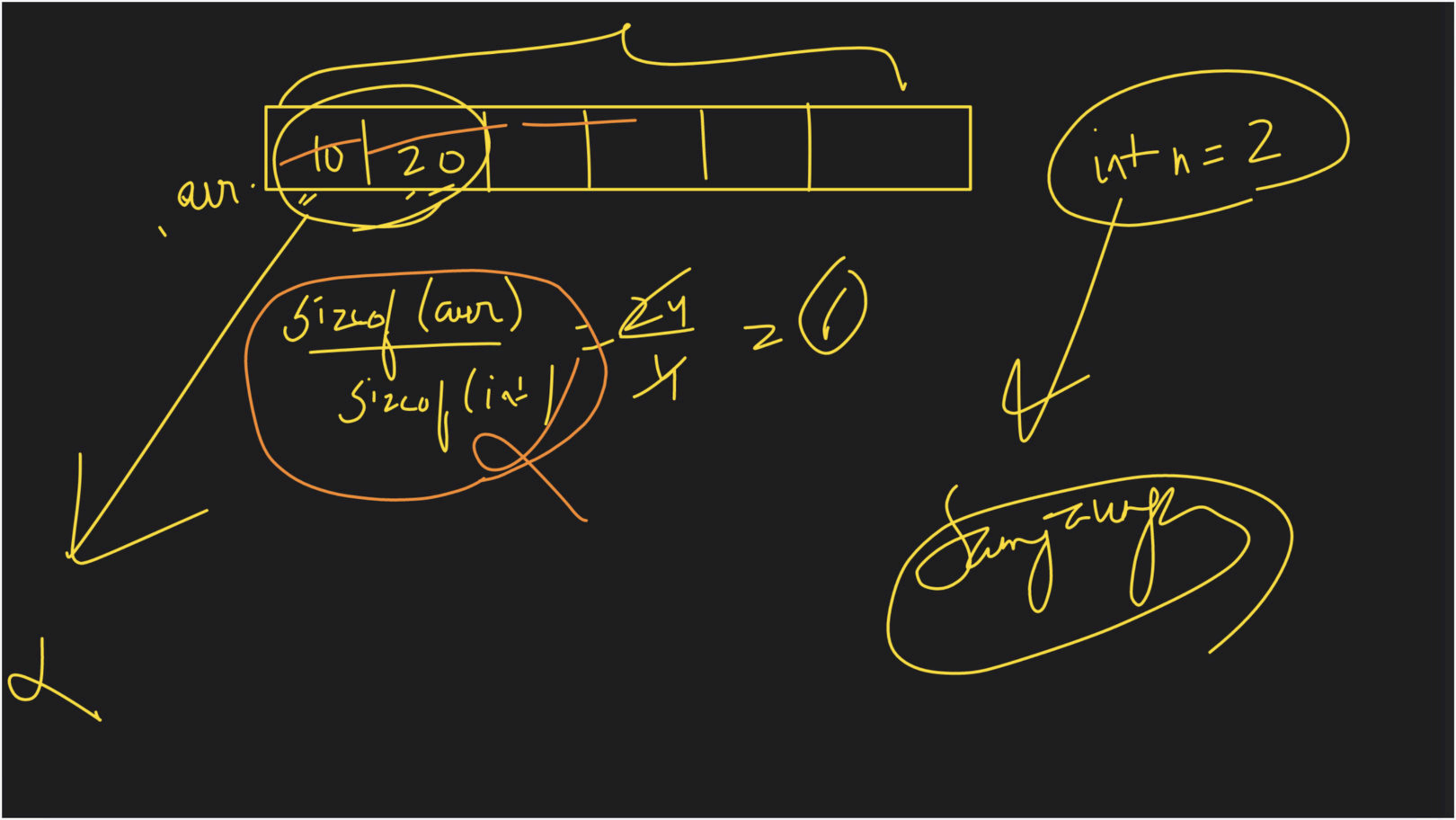
Create :-

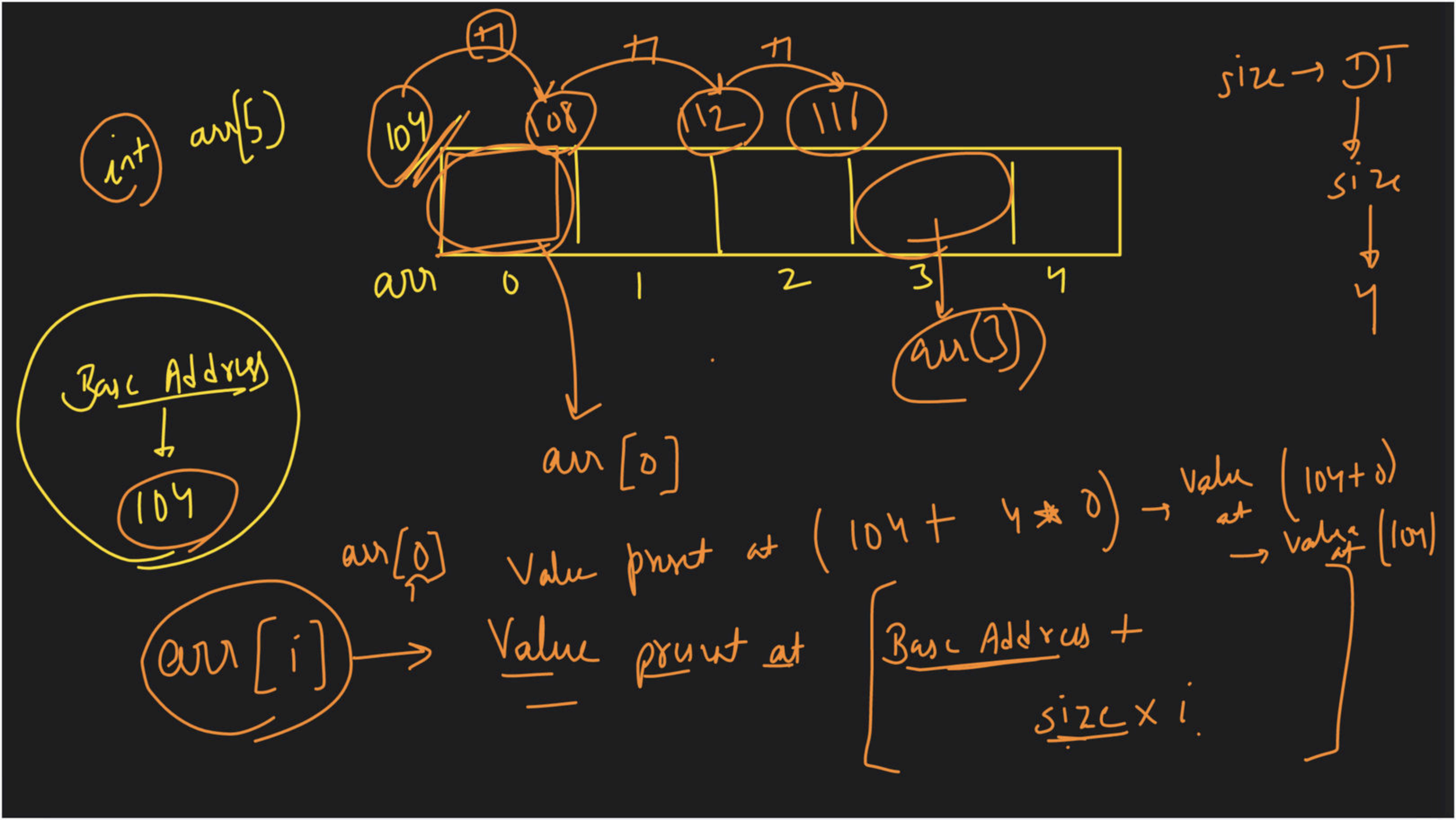

10 bool
bool arr[10] ;
If char
char arr[15];
1 integer
int arr[1]
string ✗

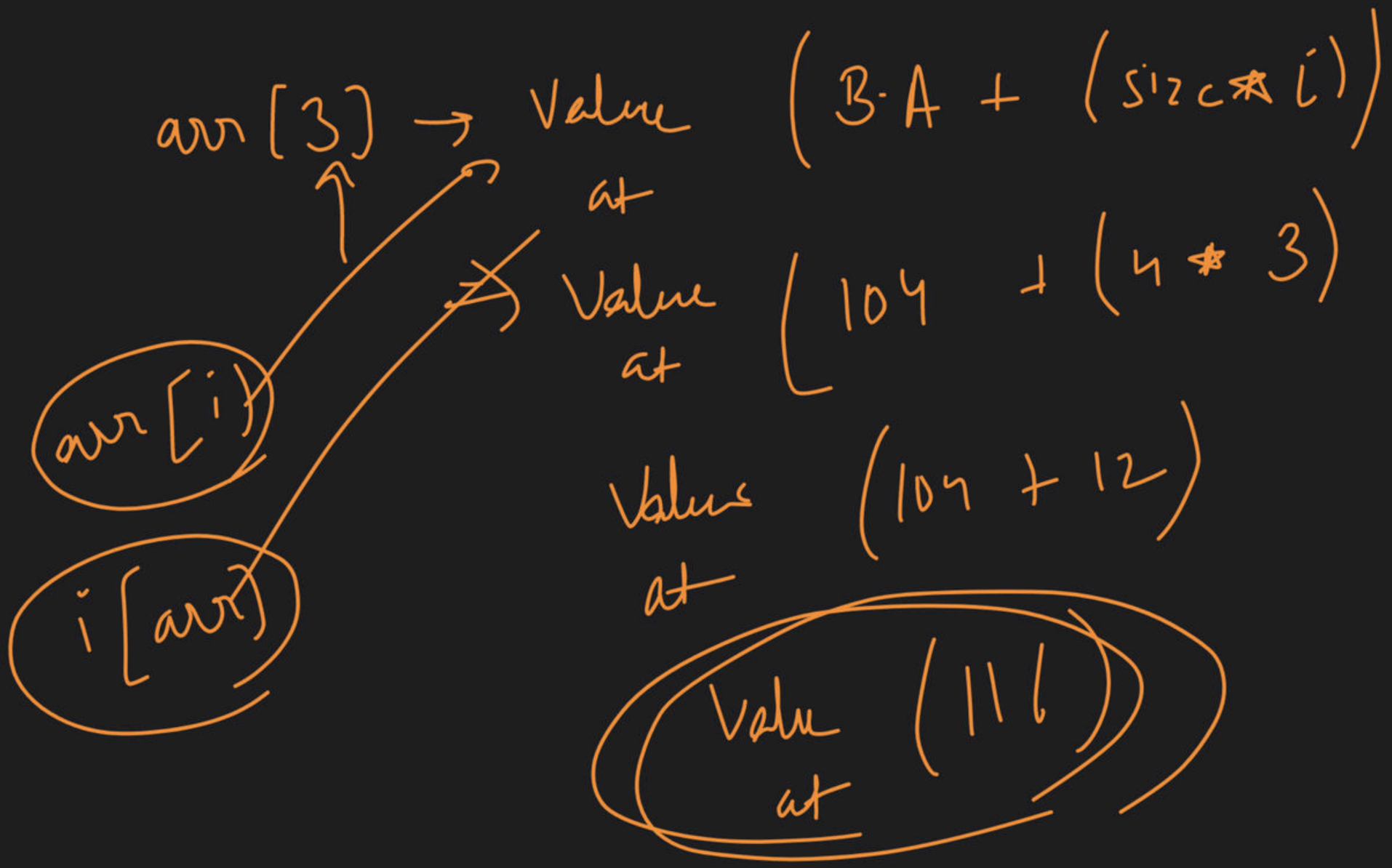

int arr[5];
garbage value


initialize
int bar[4] = {0};
int arr[5] = { 10, 20 };
int arr[3] = {10};
0 0 0 0
10 20 0 0 0
10 0 0
int a[] = {1, 2, 3};


4/5 months
doubt
size
arr
arr
10
20
30
40
int a=5;
sizeof(arr)
sizeof(int)
16
4
4
sizeof(a)
4
0
0


10
20
arr
sizeof(arr) / sizeof(int)
= 24 / 4
= 6
int n = 2

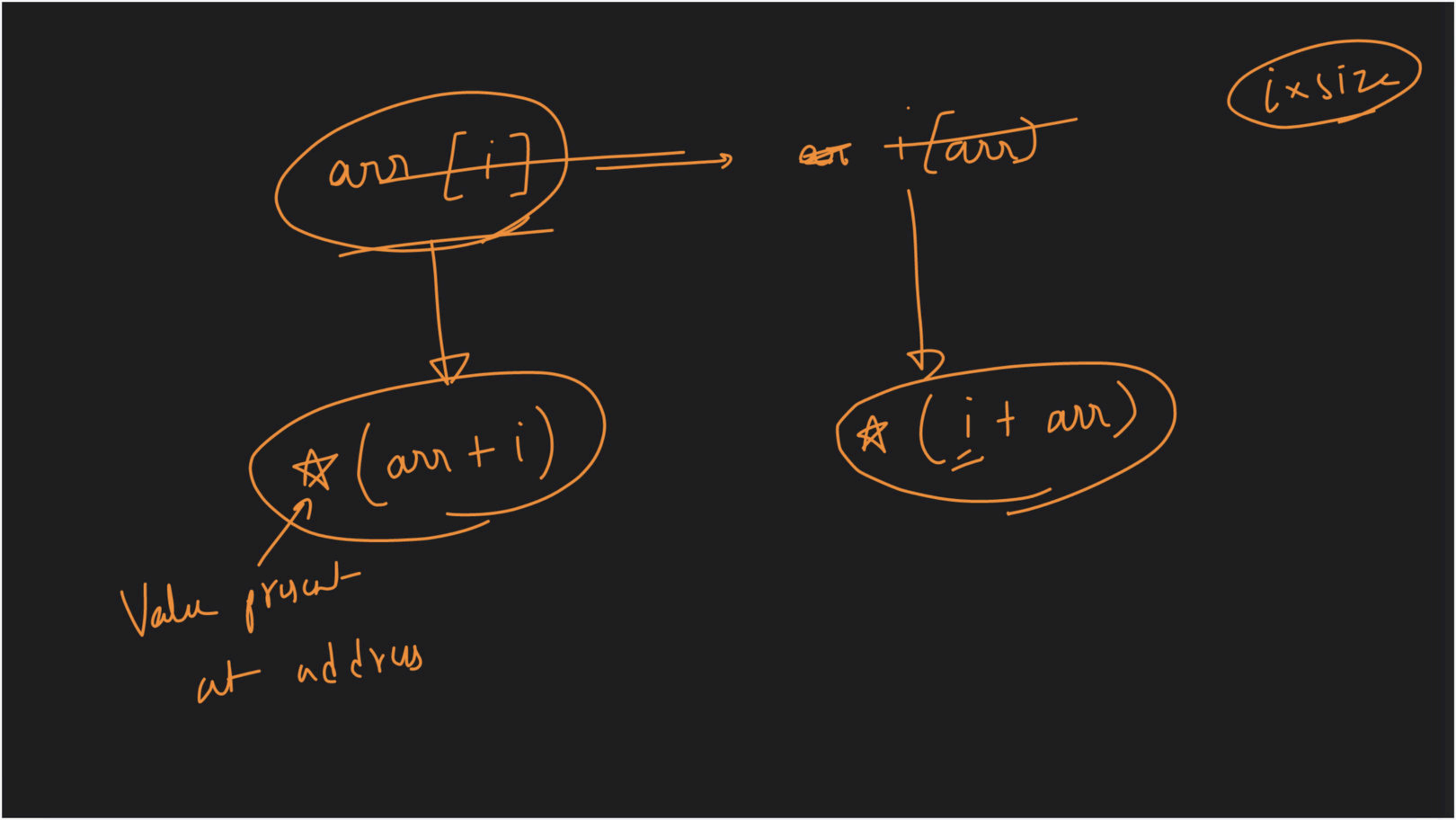

int arr[5]
104
108
112
116
arr 0 1 2 3 4
arr[3]
arr[0]
size → DT
size
4
Base Address
104
arr[i] → Value present at [Base Address + size x i]
arr[0] Value present at (104 + 4 * 0) → Value at (104+0) → Value at (104)

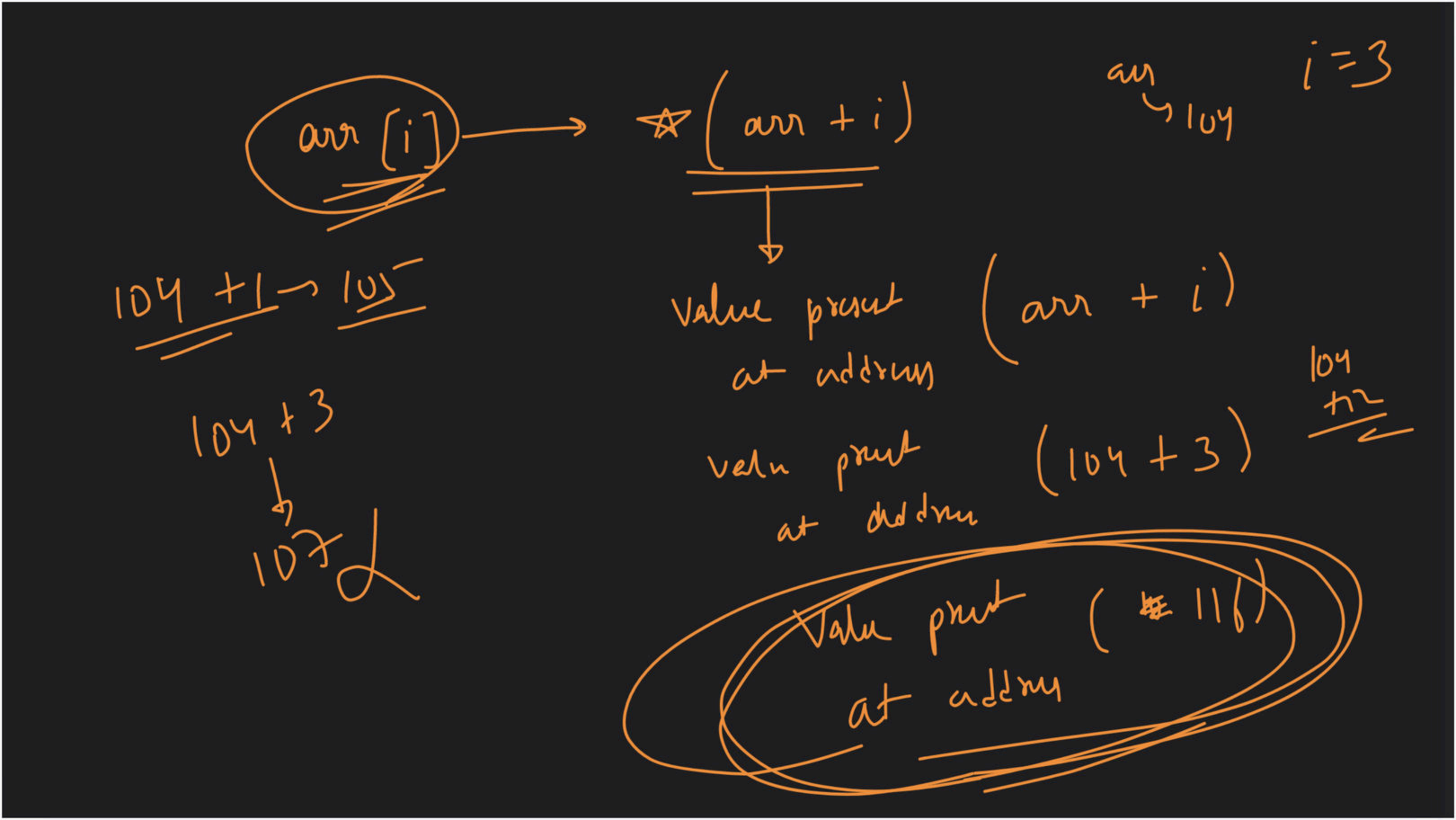

arr[3] → Value at (B.A + (size * i))
Value at (104 + (4 * 3))
Value at (104 + 12)
Value at (116)
arr[i]
i[arr]

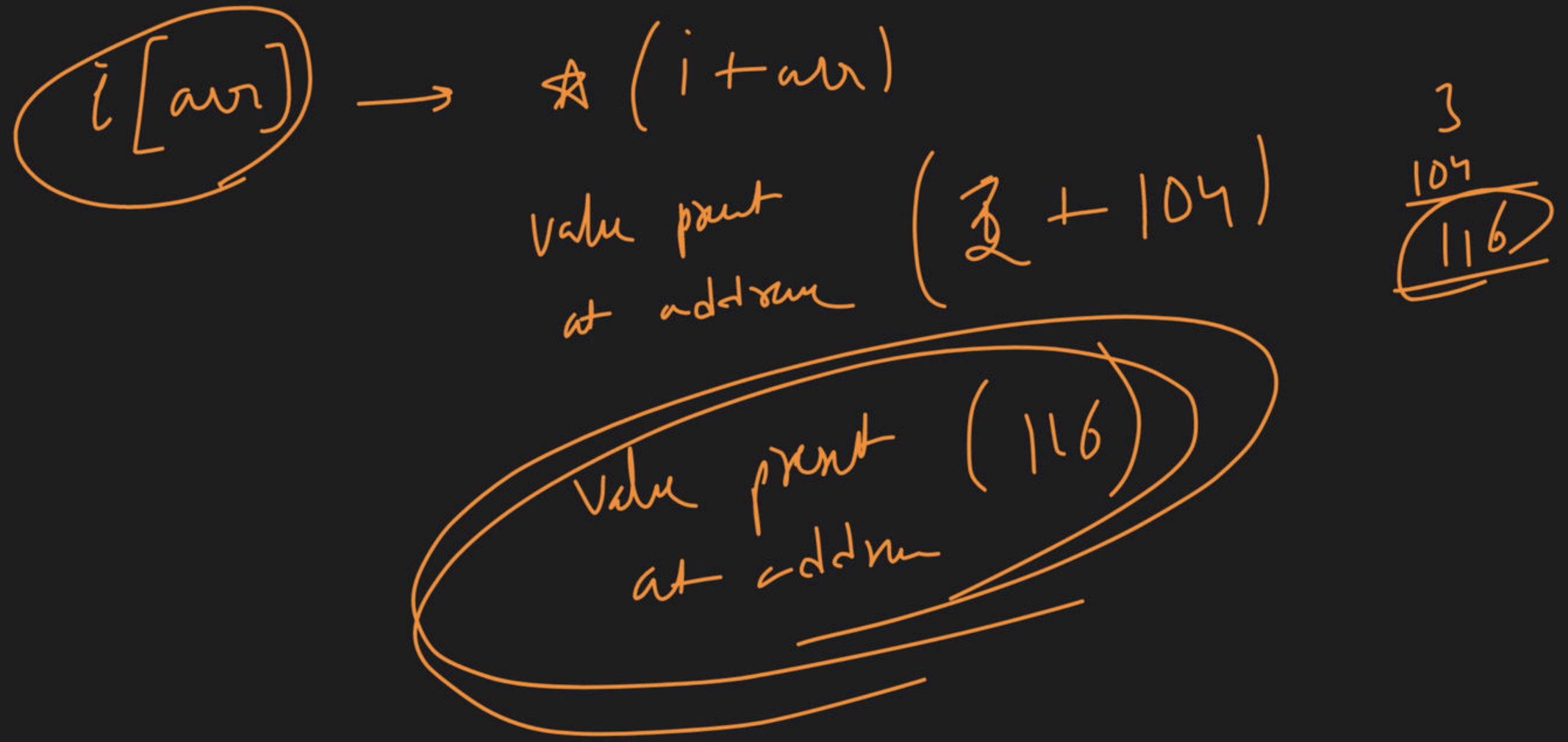
arr[i] → Value at (arr + (size * i))

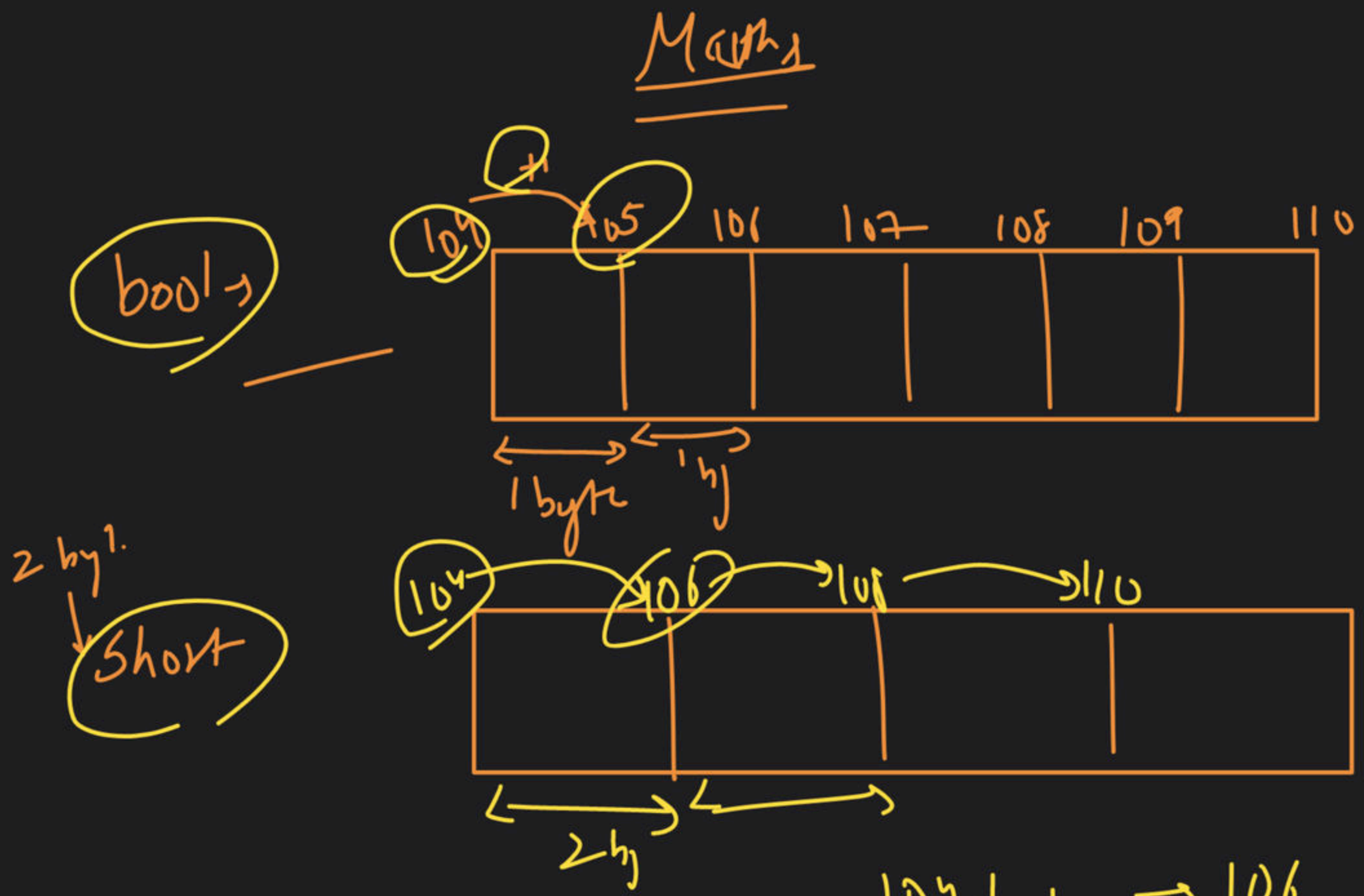
i[arr]

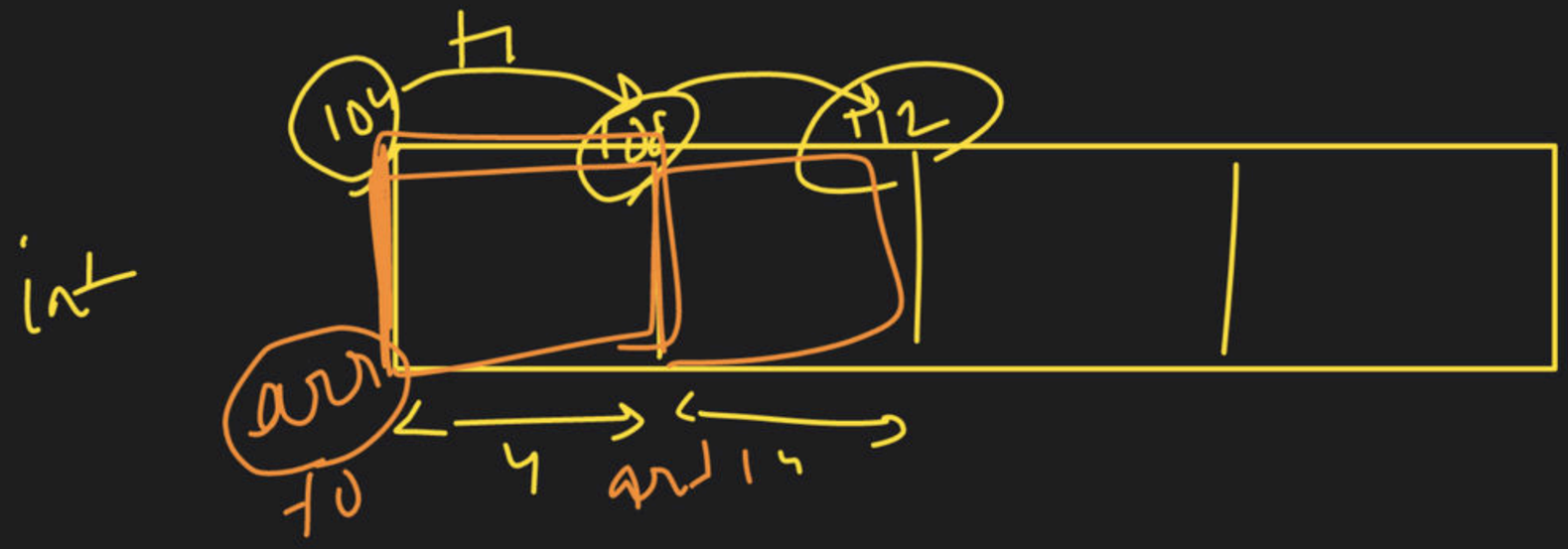
arr[i] → i[arr]


arr[i]
*(arr + i)
Value present at address
*(i + arr)
i x size

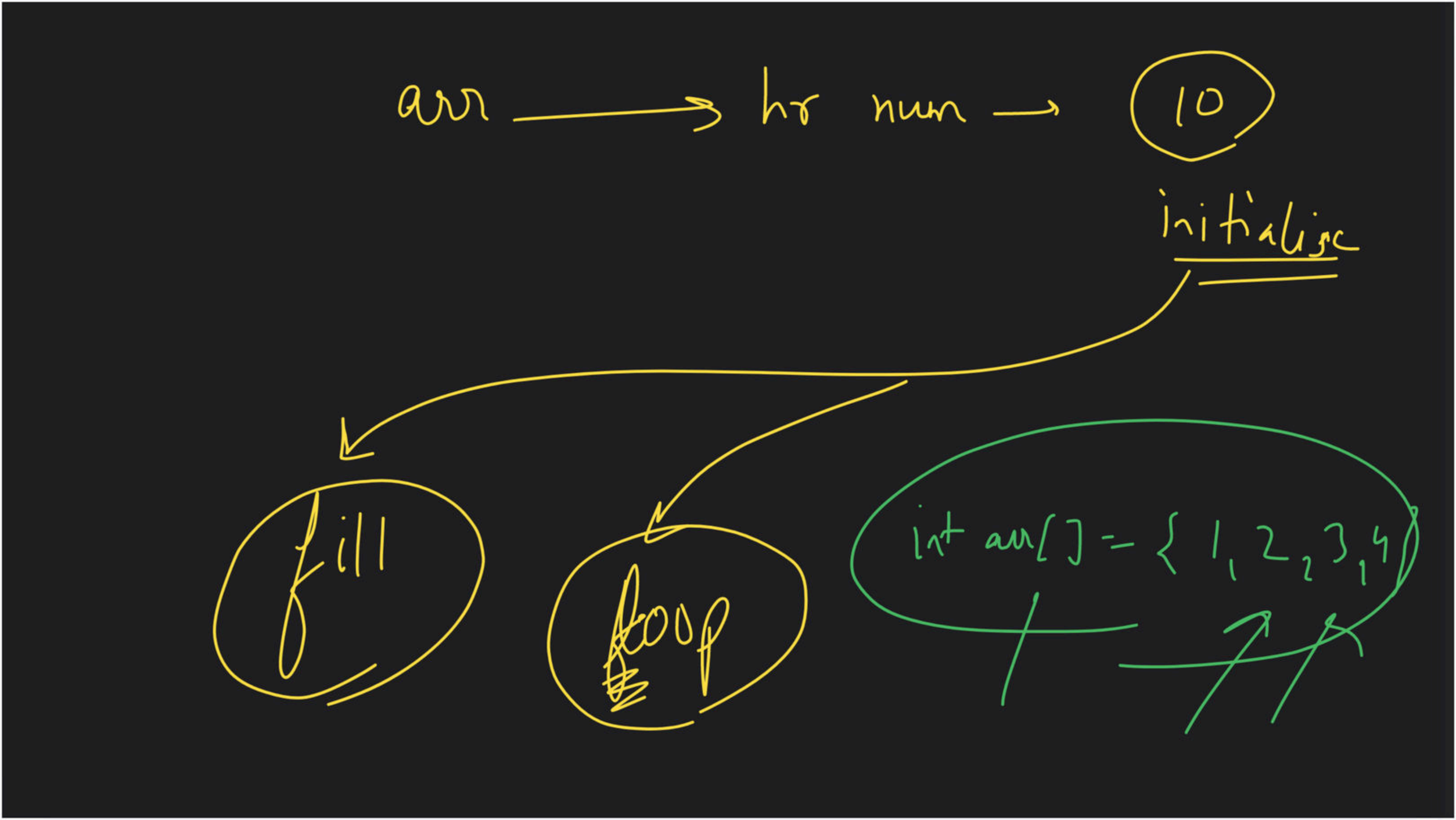

arr [i]
*(arr + i)
arr → 104
i = 3
104 + 1 → 105
104 + 3 → 107
Value present at address (arr + i)
Value present at address (104 + 3)
104 + 2
Value present at address (* 107)


i[arr] → *(i+arr)
value present at address (3+104)
3
104
116
value present (116) at address


Main
bool
105
101
107
108
109
110
1 byte
1 byte
2 byte
short
104
106
108
110
2 byte
104 + 1 → 106



104 → 108


arr
hr
num
10
initialize
fill
loop
int arr[] = { 1, 2, 3, 4 }

pattern

i -> 1 2

i = 0

create

initialize

fill

Indexing

taking input

Indexing

1 -> n

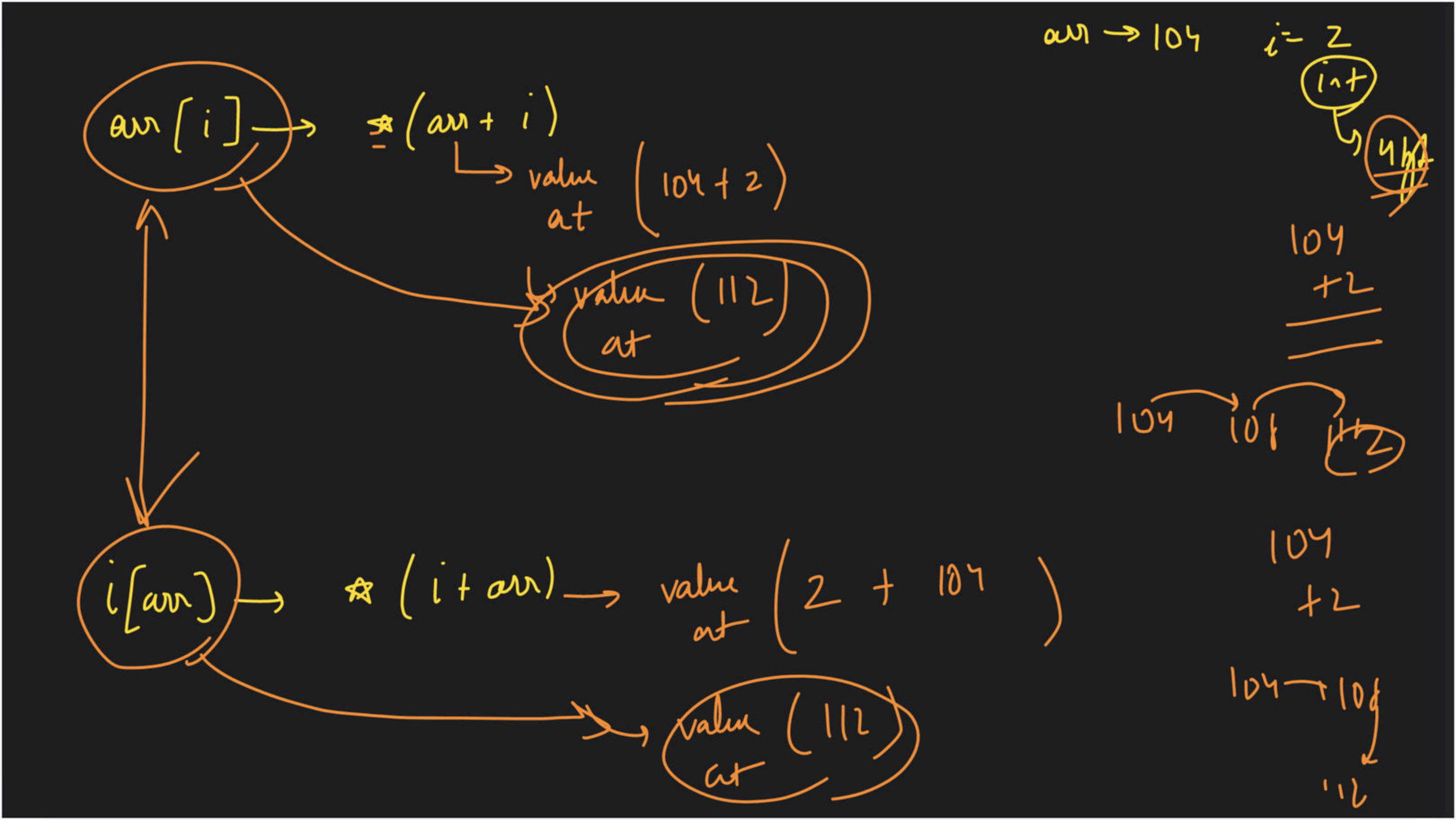
n no.

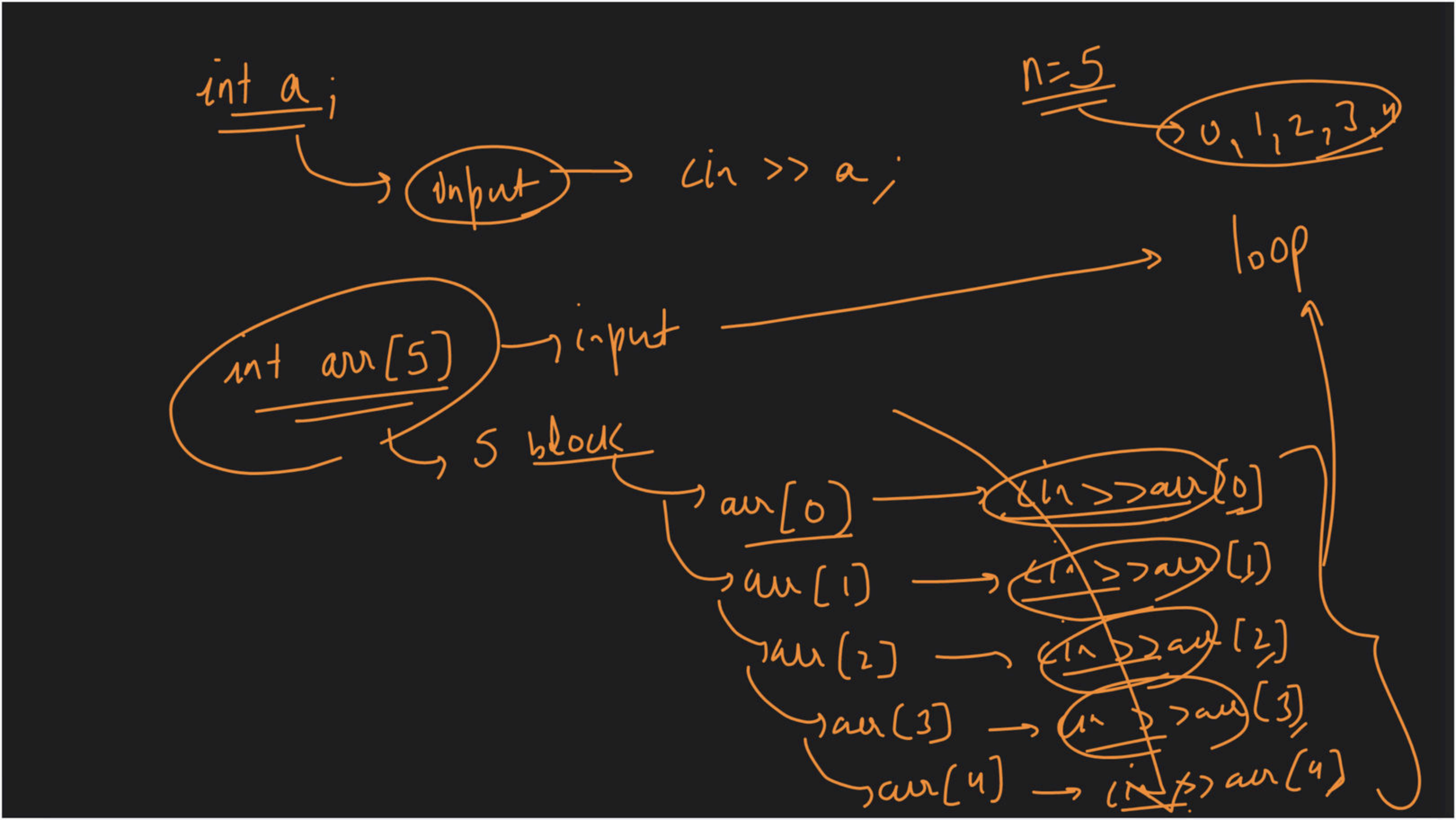
1 -> 5

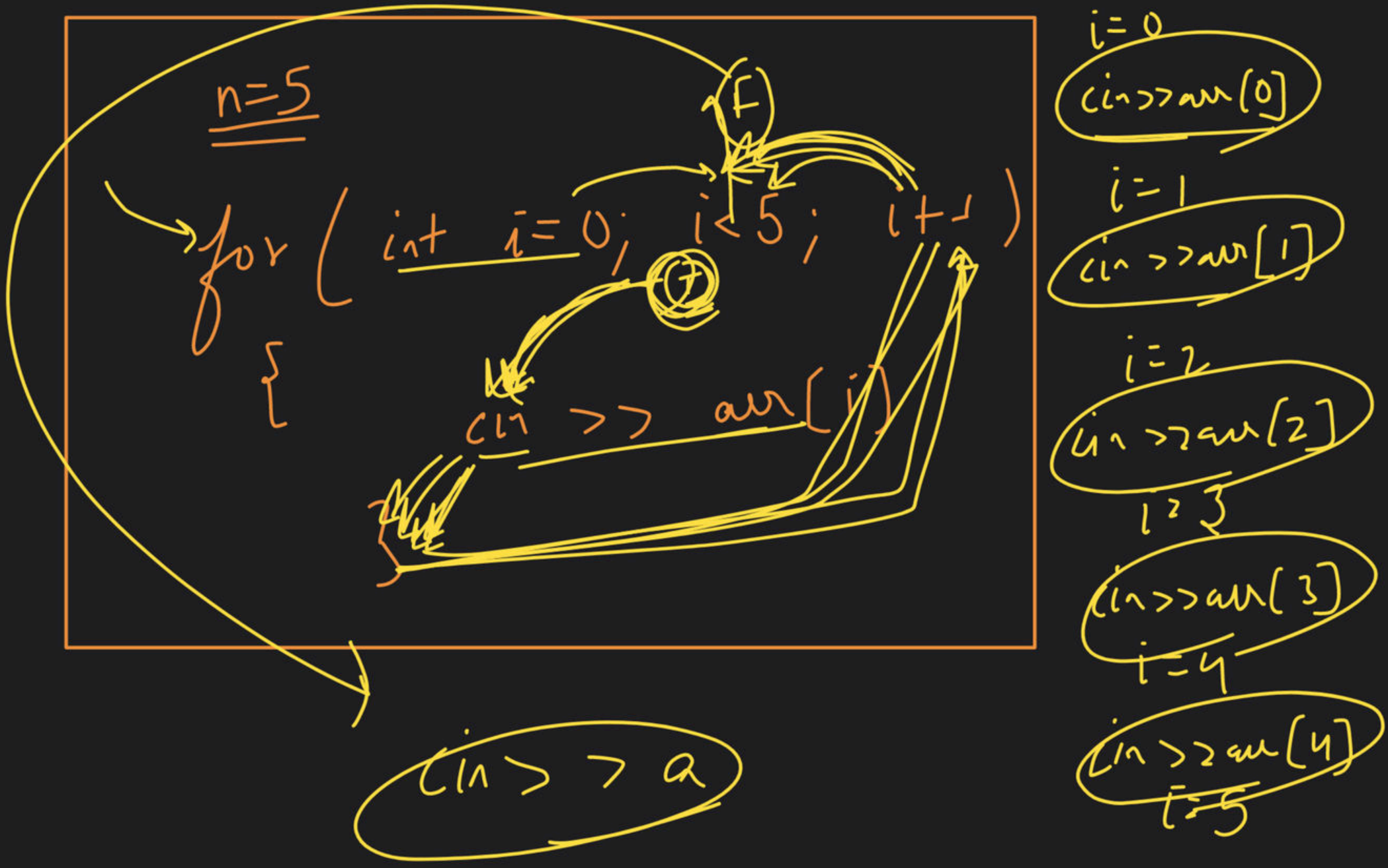
1 -> n

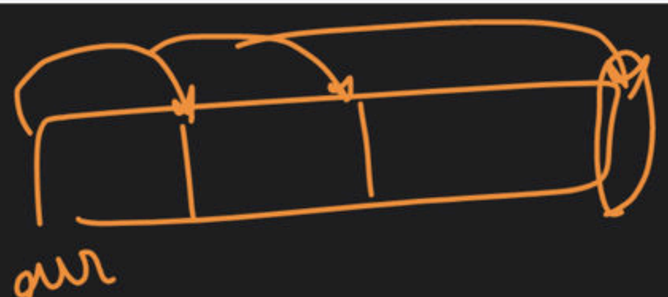
(-1) (-1)

0 -> n-1 -> n terms


arr → 104
i = 2
int
arr[i] → *(arr + i)
value at (104 + 2)
value (112)
at
104
+2
104 → 106
i[arr] → *(i + arr) → value at (2 + 104)
value (112)
at
104
+2
104 → 106
112


int a;
input
cin >> a;
n=5
0, 1, 2, 3, 4
loop
int arr[5]
input
5 block
arr[0]
cin>>arr[0]
arr[1]
cin>>arr[1]
arr[2]
cin>>arr[2]
arr[3]
cin>>arr[3]
arr[4]
cin>>arr[4]


n=5
for (int i=0 ; i<5 ; i++)
{
cin >> arr[i]
F
T
cin >> a
i=0
cin>>arr[0]
i=1
cin>>arr[1]
i=2
cin>>arr[2]
i=3
cin>>arr[3]
i=4
cin>>arr[4]
i=5



fill

arr → 104

int arr[3] = {27 27 27}

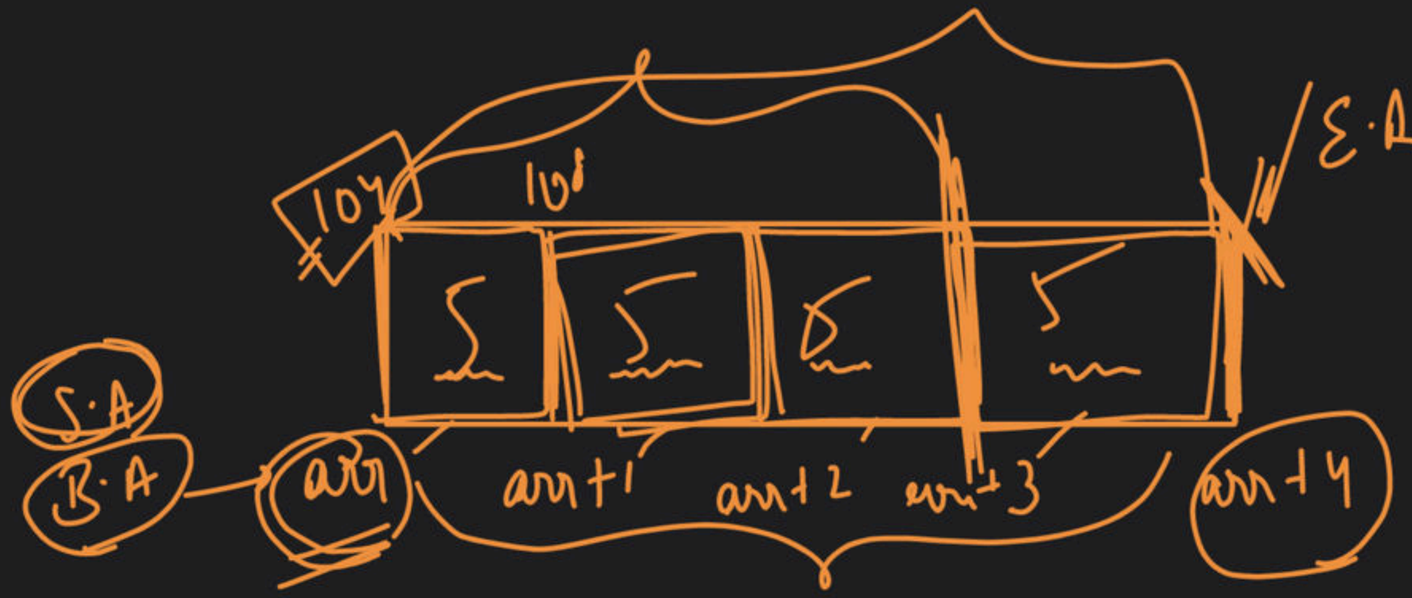
23

fill (arr , arr+3 , 23)

S.A

Val

int arr[4]



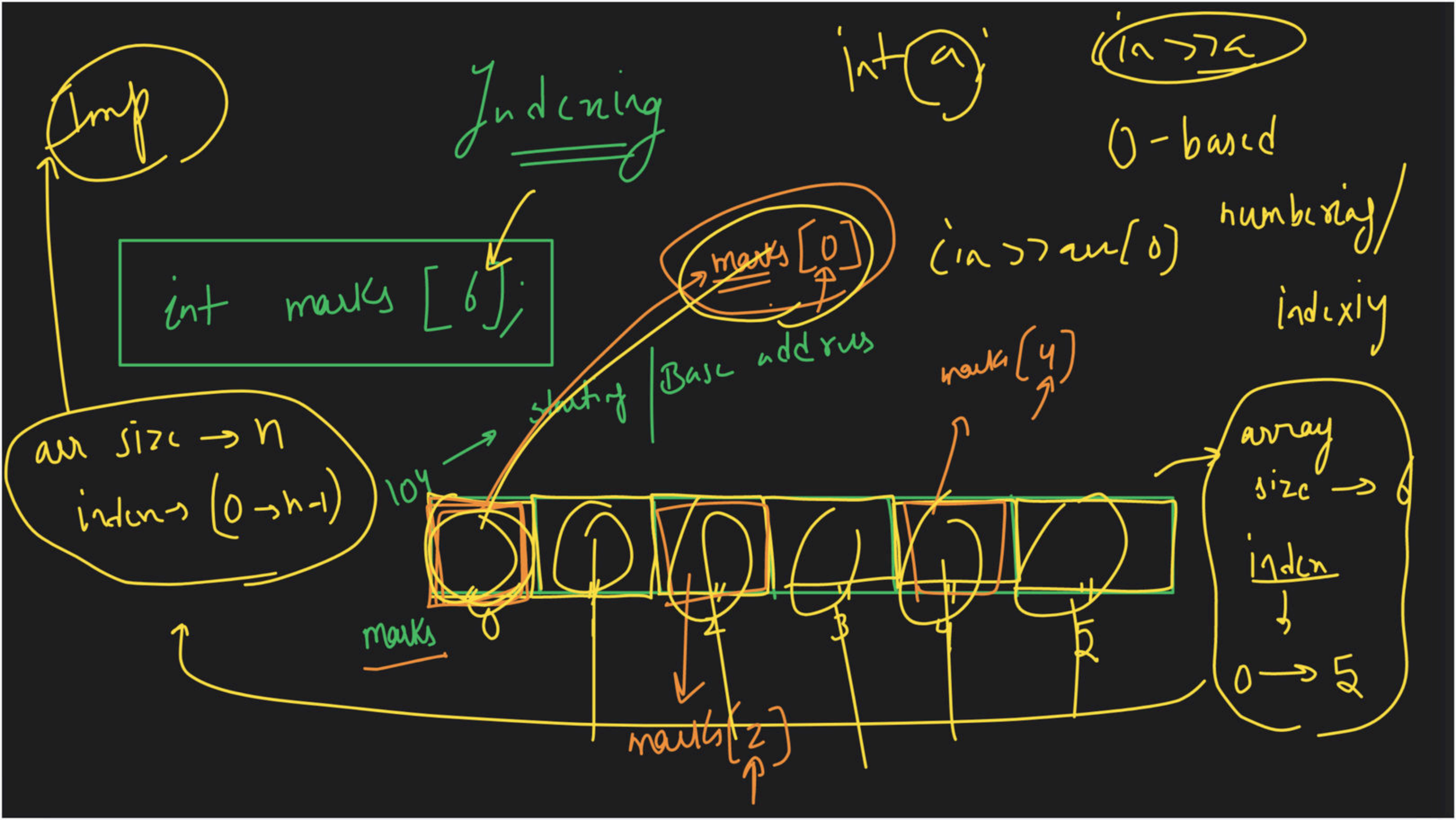
5 se initialize hr block ko

fill(arr , arr+4 , 5)

arr → S·A

arr+4 → E·A

5 → Val


Imp
Indexing
int a;
cin >> a
0-based
numbering/
indexing
int marks [6];
marks [0]
cin >> arr[0]
Base address
starting
arr size → n
index → (0 → n-1)
104
marks [4]
array
size →
index
0 → 5
marks
0
1
2
3
4
5
marks [2]

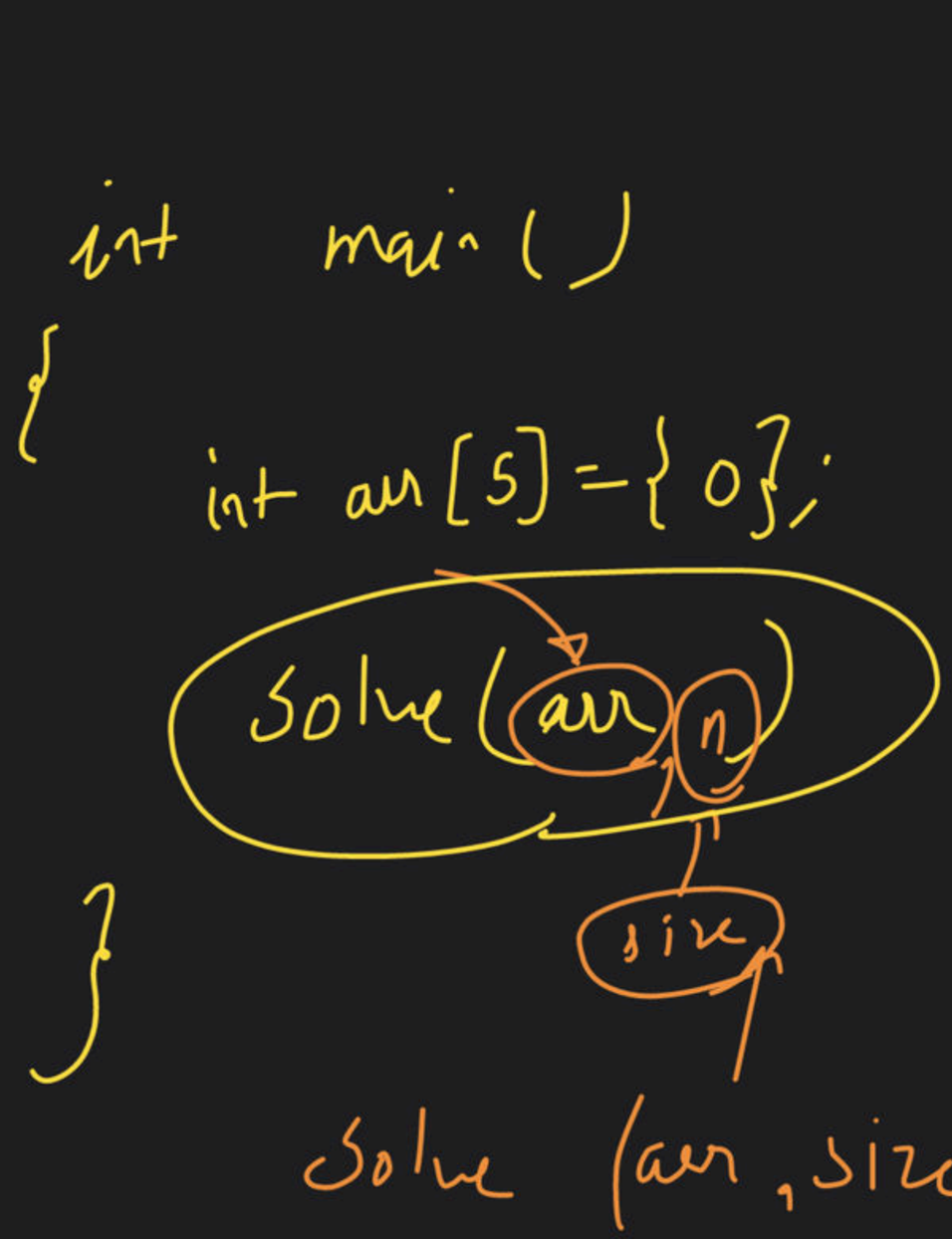

// print
2 min
Break

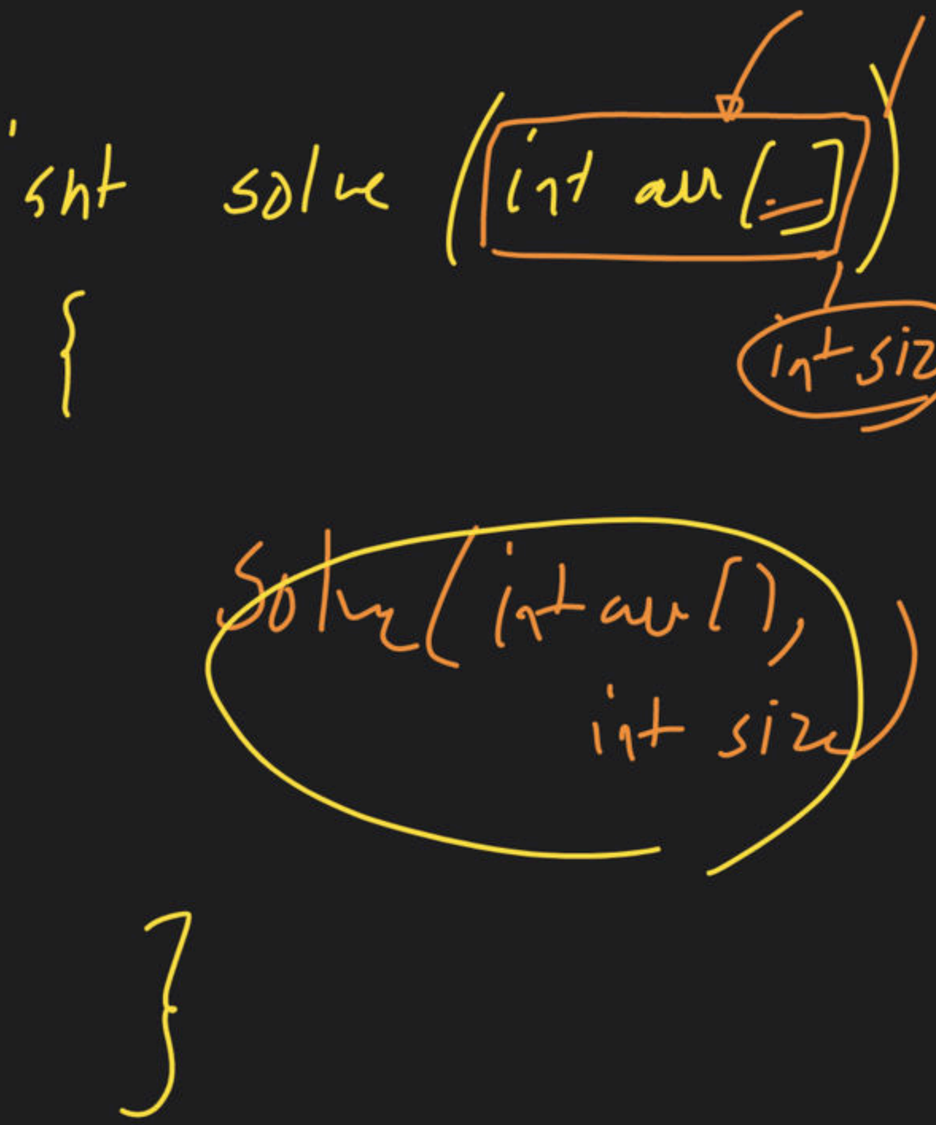
Functions with Array

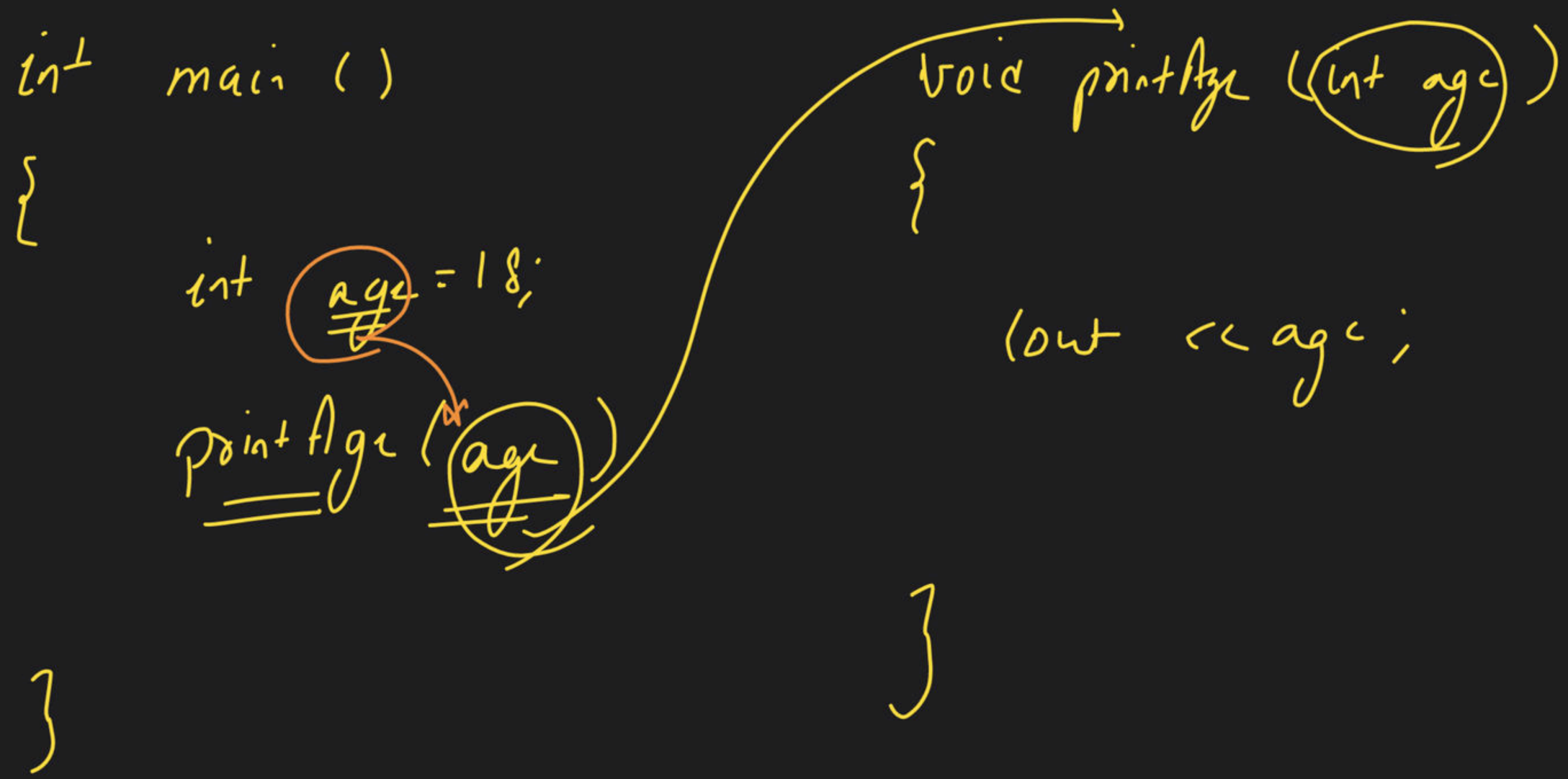
```
int main()
{
    int arr[5];
    int a=5;
    solve(a);
}

solve(arr);
```

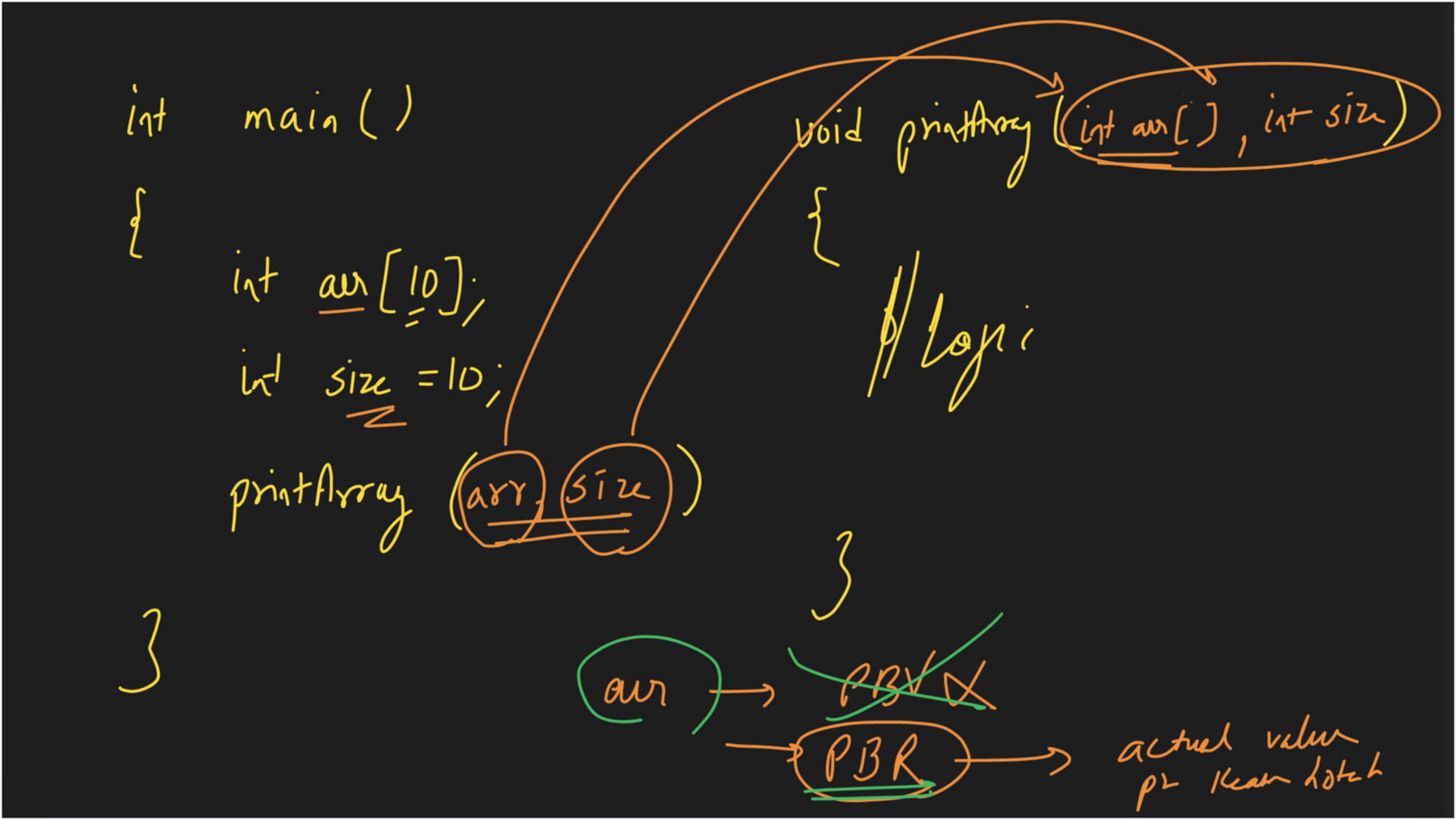
```
int solve(int a)
{

}
```

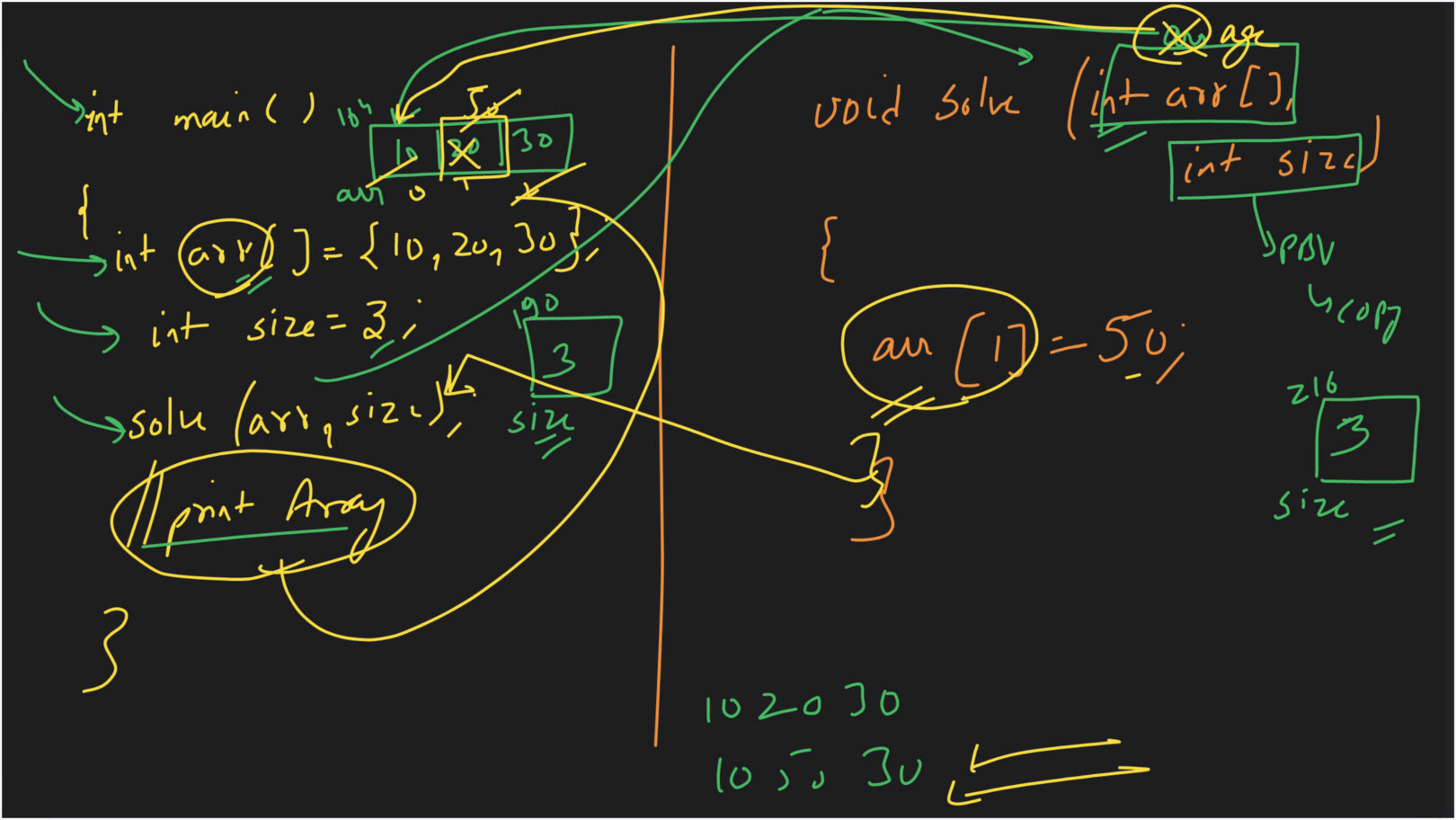
int arr[]


int main()
{
int arr[5]={0};
Solve(arr,n);
}
size
Solve(arr,size)


int solve(int arr[])
{
int size
Solve(int arr[],
int size)
}


int main ()
{
int age = 18;
printAge (age);
}
void printAge (int age)
{
cout << age;
}

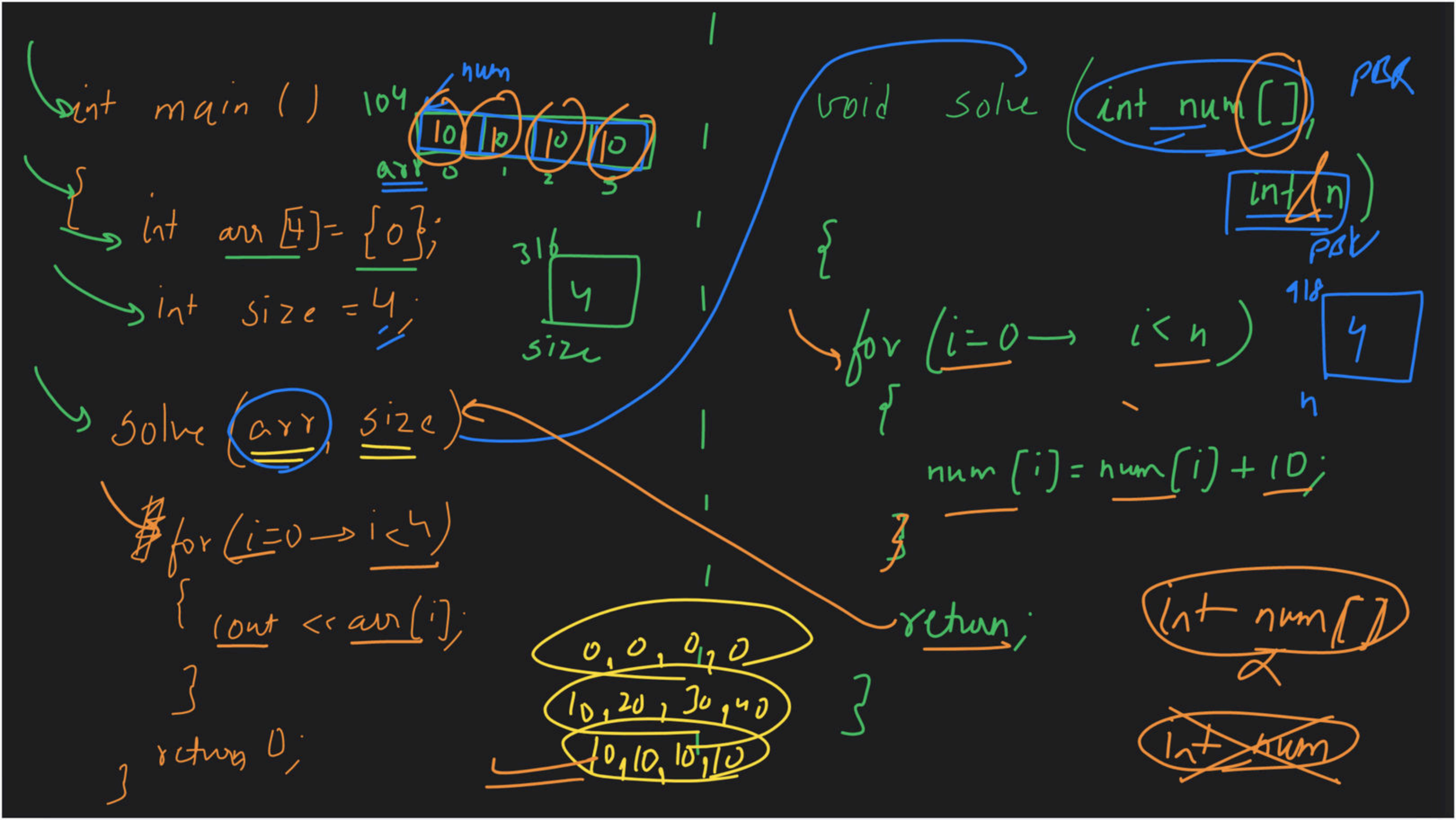

int main()
{
int arr[10];
int size = 10;
printArray(arr, size)
}
void printArray(int arr[], int size)
{
// Logic
}
arr
PBV
PBR
actual value
pe kaam hota h


int main ( )
{
int arr[ ] = {10, 20, 30};
int size = 3;
Solu (arr, size);
print Array
}
100
10
20
30
arr 0 1
size
190
3
void Solu (int arr[],
int size)
{
arr[1] = 50;
}
arr
PBV
copy
216
3
size
10 20 30
10 50 30

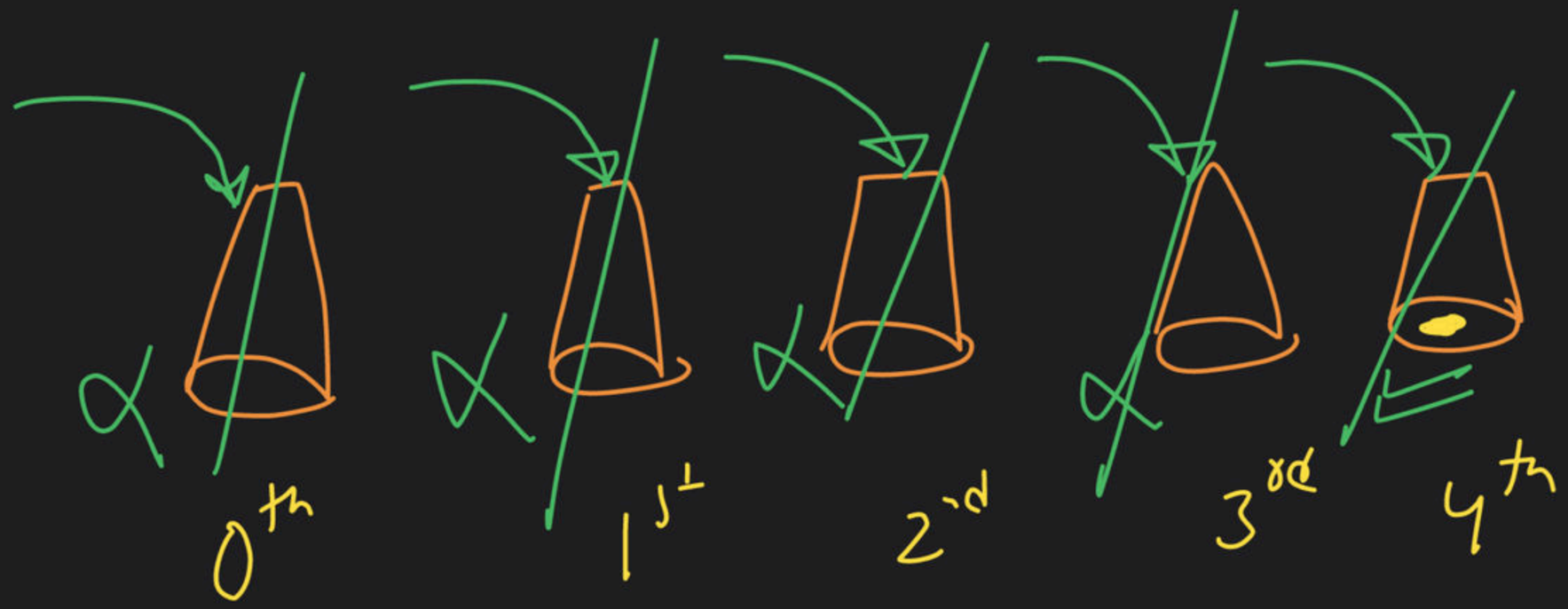
Pass by Reference

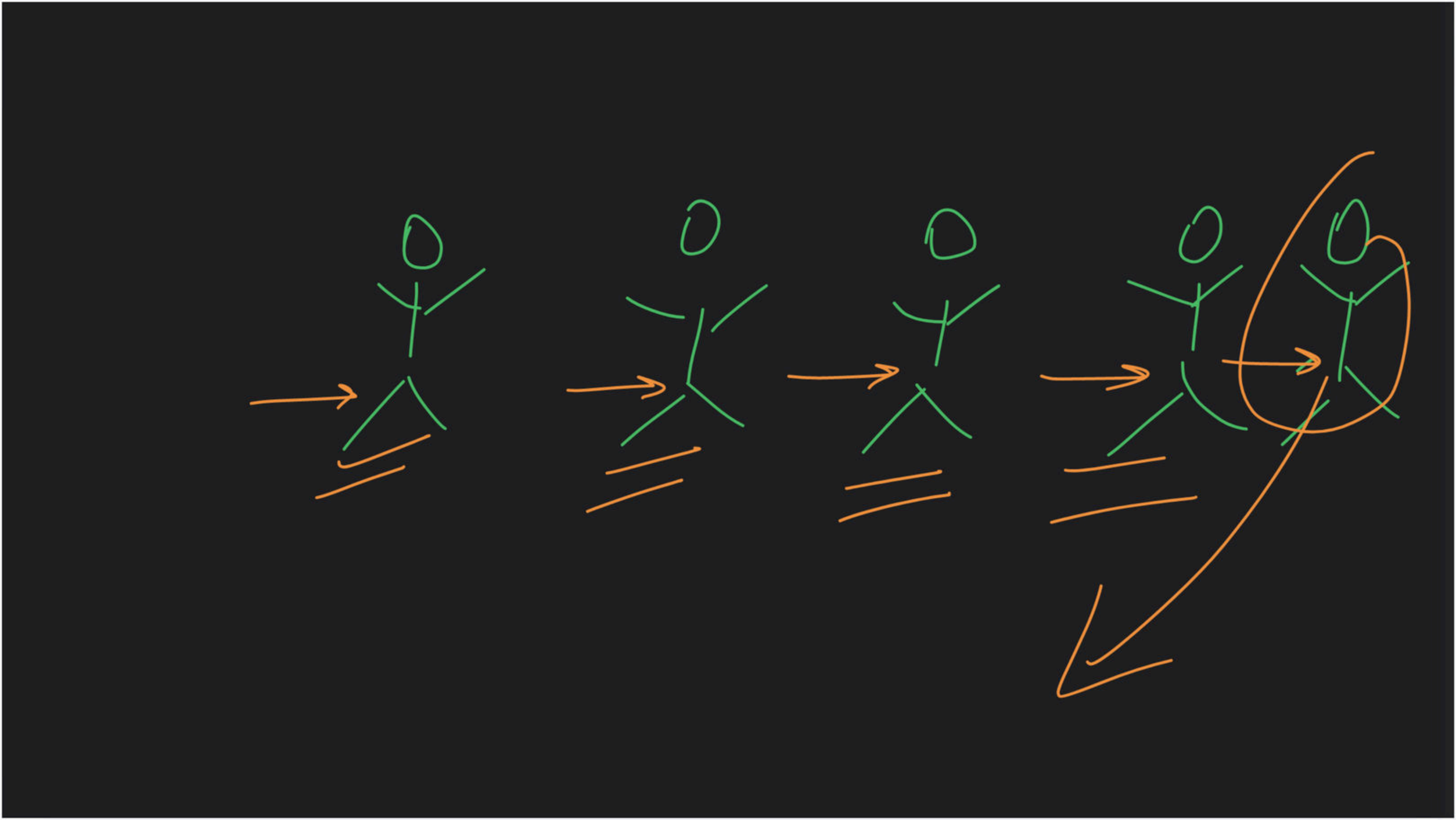
Last Class

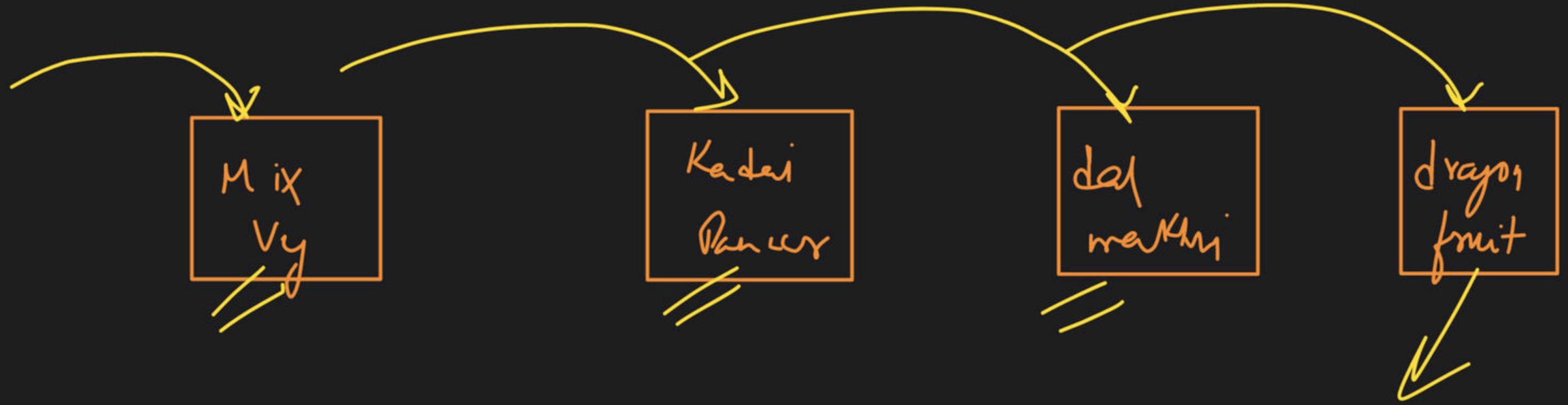
Re-visit

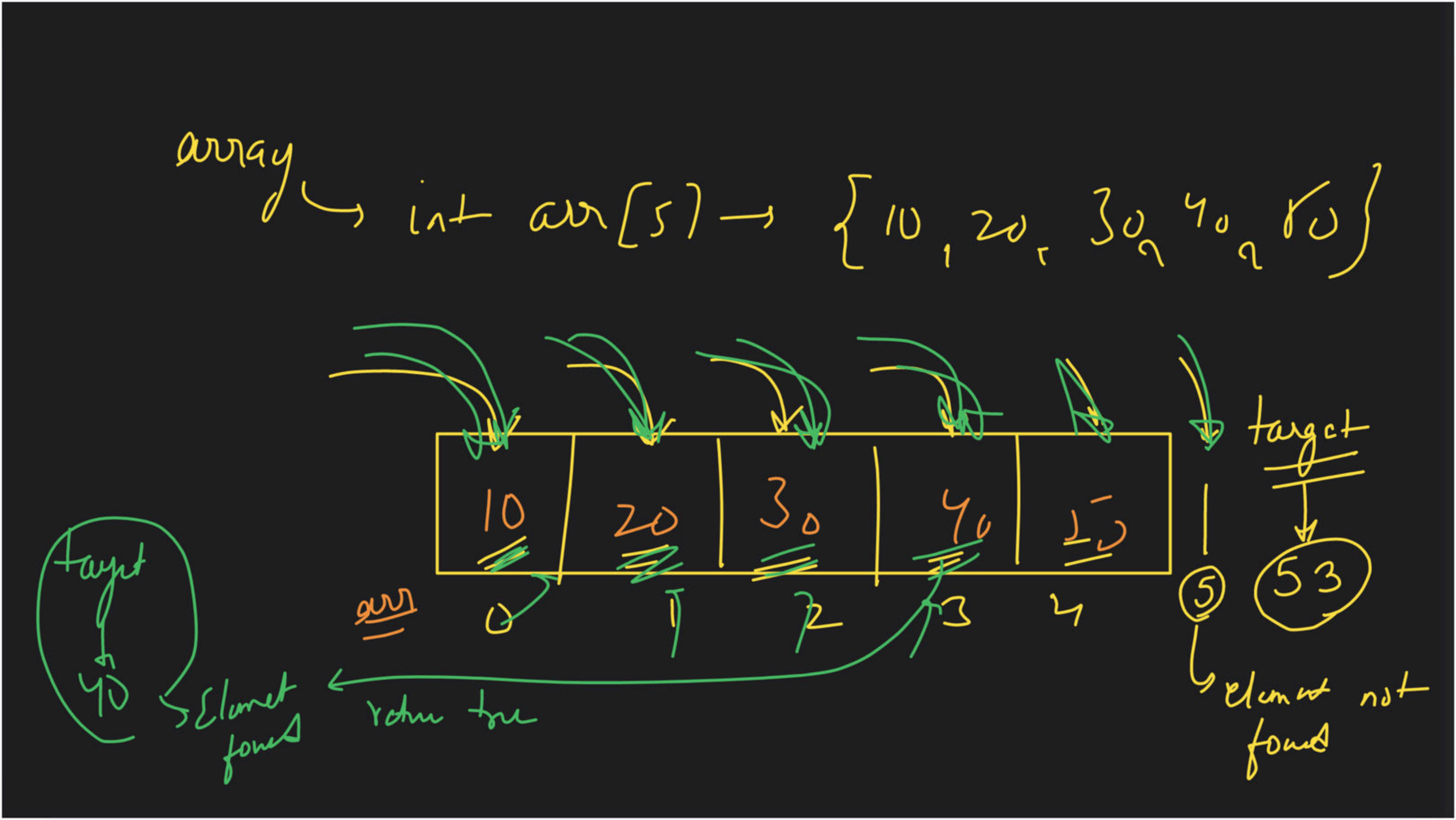

int main ()
{
int arr[4]={0};
int size=4;
Solve(arr, size)
for(i=0→i<4)
{
cout<<arr[i];
}
return 0;
}
104
num
10
10
10
10
arr 0 1 2 3
316
4
size
void Solve(int num[], int n)
{
for(i=0→ i<n)
{
num[i]=num[i]+10;
}
return;
}
pbr
pbr
n
n
0, 0, 0, 0
10, 20, 30, 40
10, 10, 10, 10
int num[]
int num

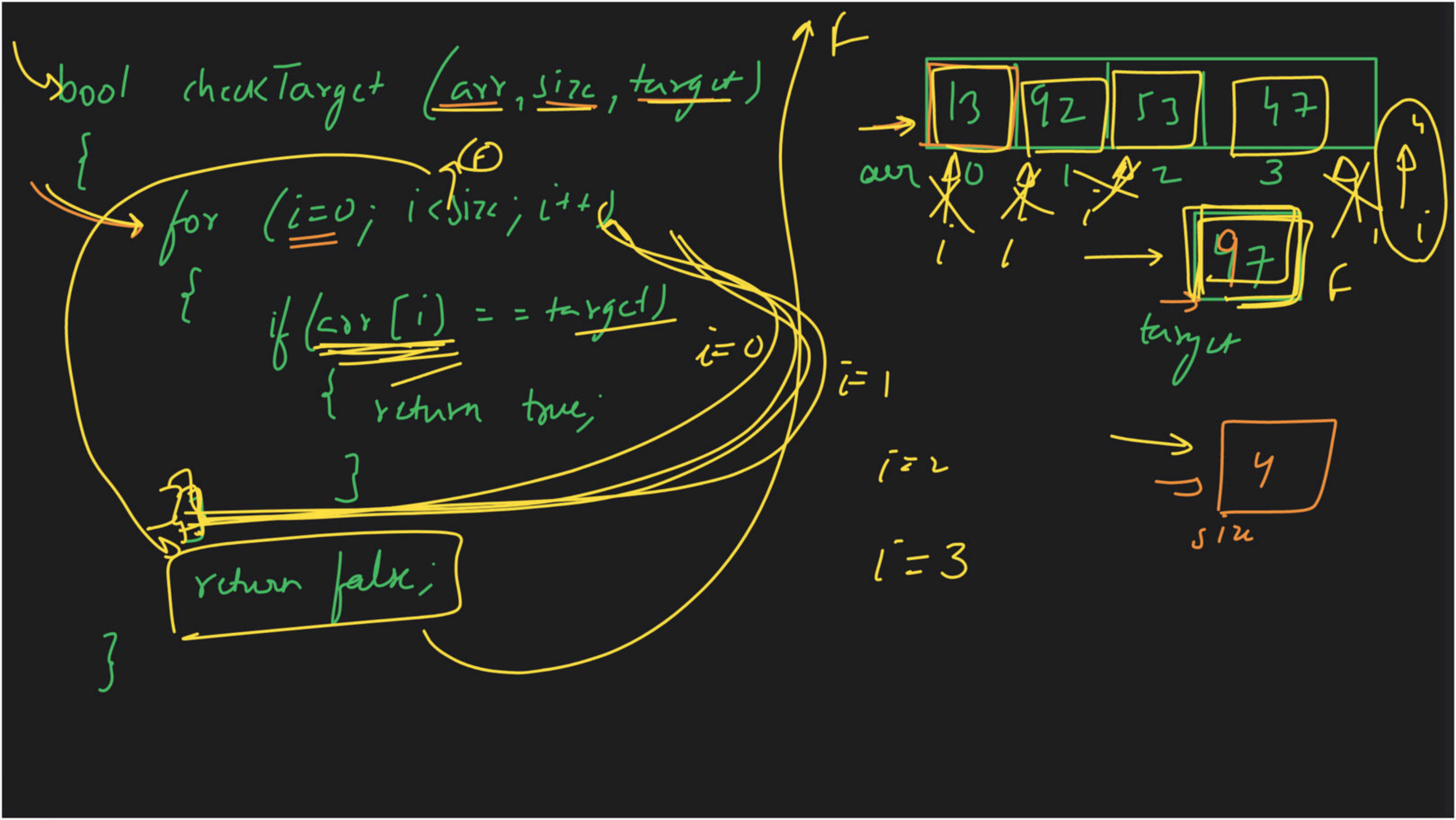
① Algorithm → Linear Search

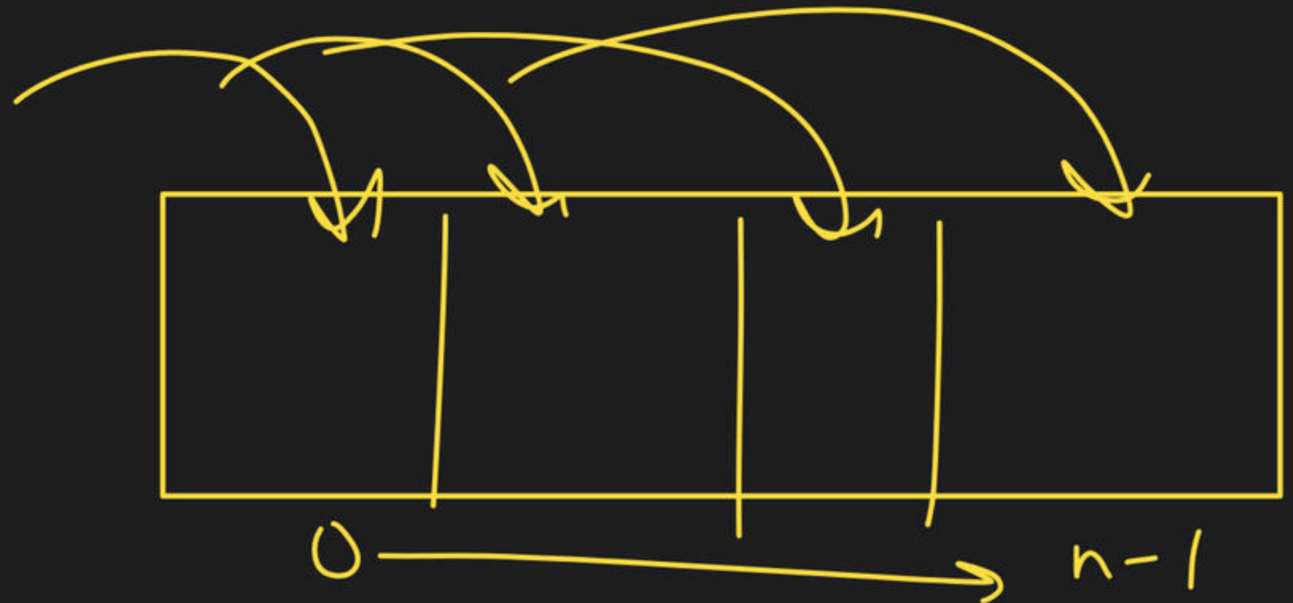


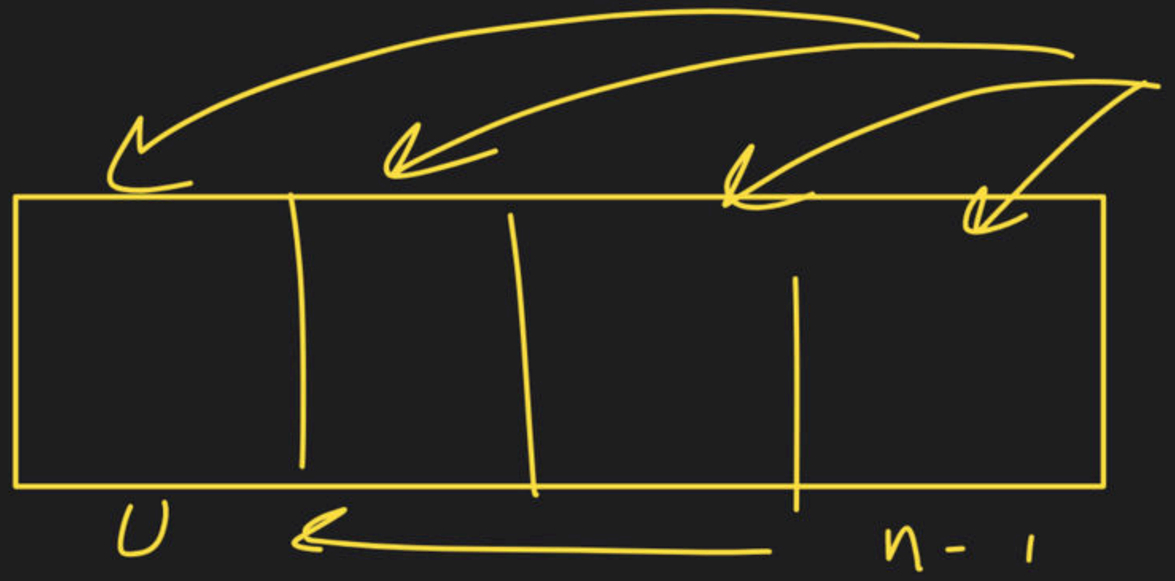


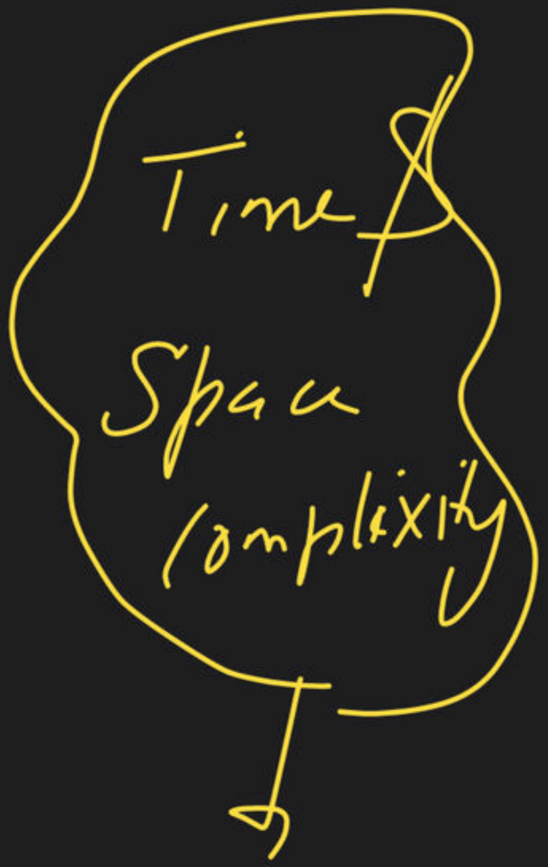

Mix
Veg
Kadai
Paneer
dal
makhni
dragon
fruit


array
int arr[5] → {10, 20, 30, 40, 50}
10
20
30
40
50
arr
0
1
2
3
4
target
53
5
target
40
Element found
return true
element not found


bool checkTarget (arr, size, target)
{
for (i=0; i<size; i++)
{
if (arr[i] == target)
{ return true;
}
return false;
}
F
i=0
i=1
i=2
i=3
13
92
53
47
arr
0
1
2
3
i
i
97
target
F
4
size

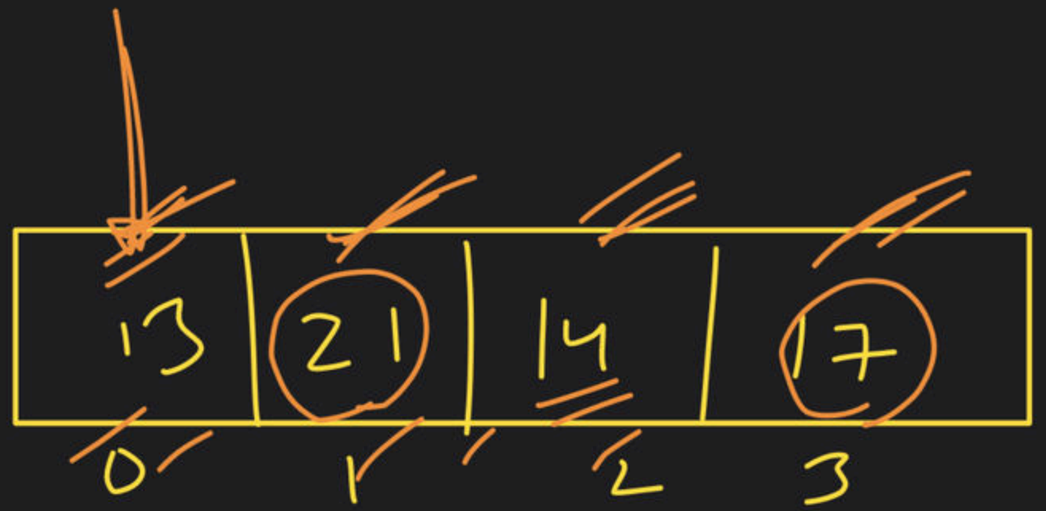

0
n-1


0
n-1


Time
Space
complexity

max

arr



int

Range

-2^{31} → $2^{31}-1$

INT_MIN

INT_MAX

max

arr[0]

ans = INT_MIN

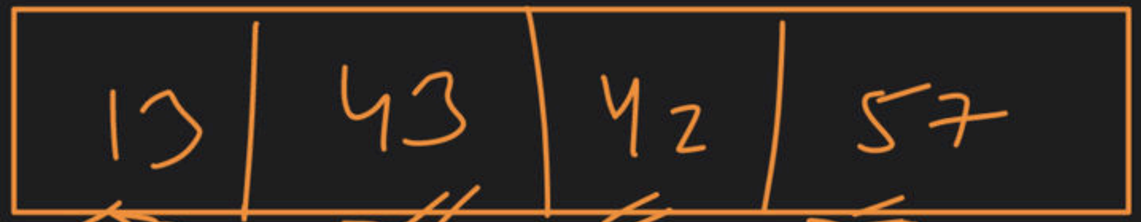
21 → final Ans

max no → ans → INT_MIN

min ho → ans → INT_MAX

arr[i] > ans ⟶ ans = arr[i]

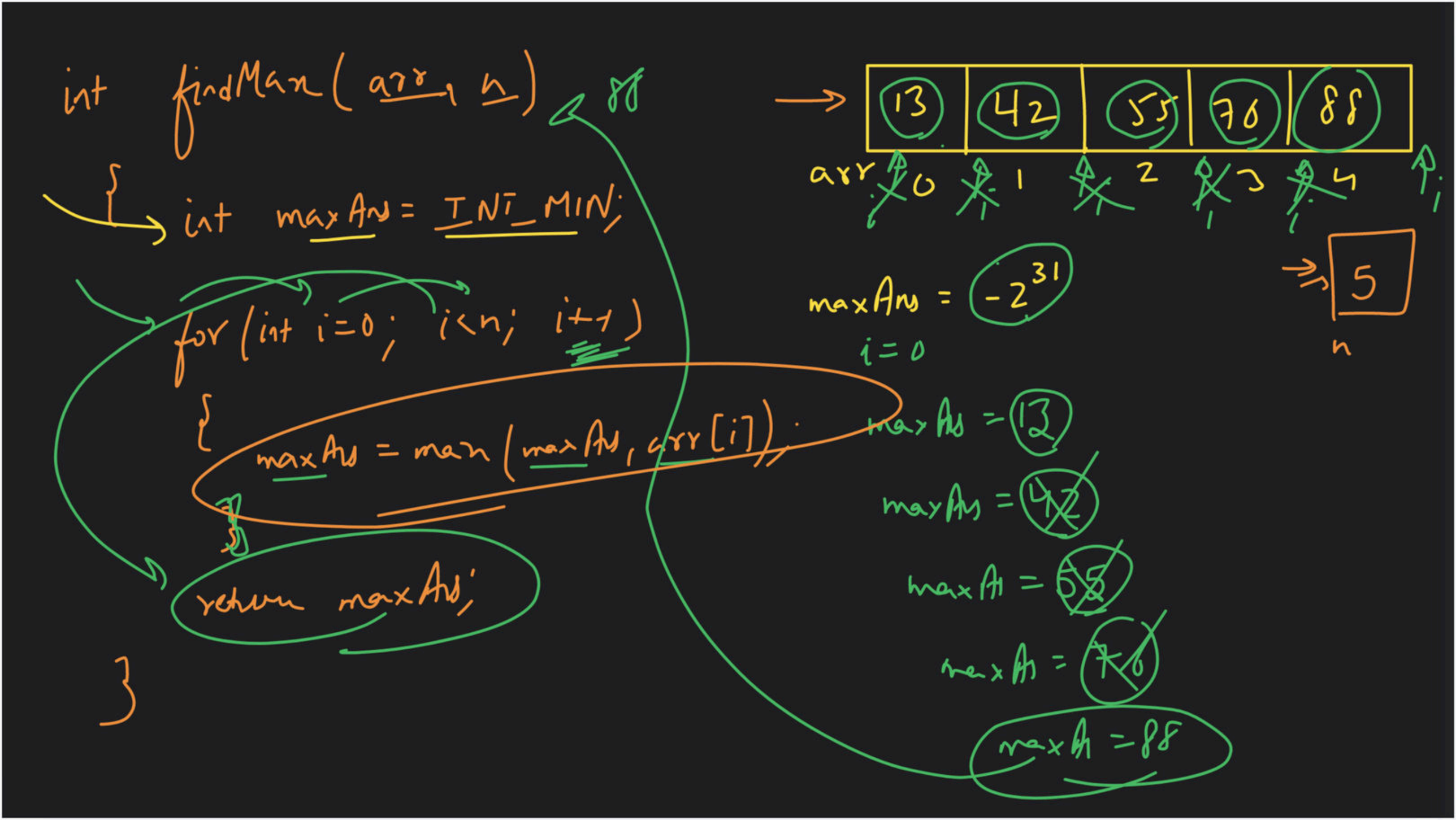
arr[i] < ans ⟶ ignore



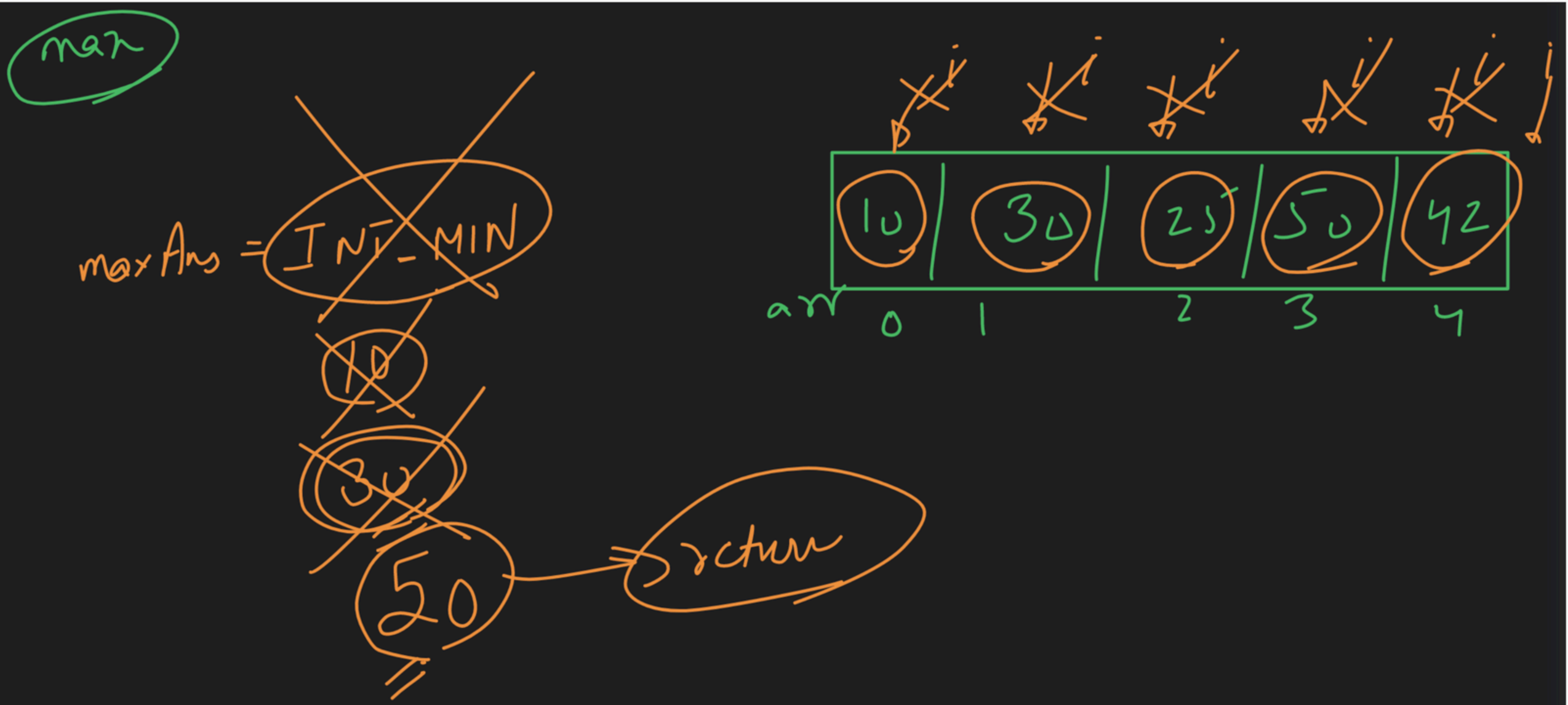
max

ans = INT_MIN

57 → final ans


int findMax(arr, n)
{
int maxAns = INT_MIN;
for(int i=0; i<n; i++)
{
maxAns = max(maxAns, arr[i]);
}
return maxAns;
}
13
42
55
70
88
arr
0
1
2
3
4
i
n
5
maxAns = -2^31
i = 0
maxAns = 13
maxAns = 42
maxAns = 55
maxAns = 70
maxAns = 88

```
int findMin(arr, n)
{
    int minAns = INT_MAX;
    for (i=0; i<n; i++)
    {
        minAns = min(minAns, arr[i]);
    }
    return minAns;
}
```


max
maxAns = INT_MIN
10
30
50
return
arr
0
1
2
3
4

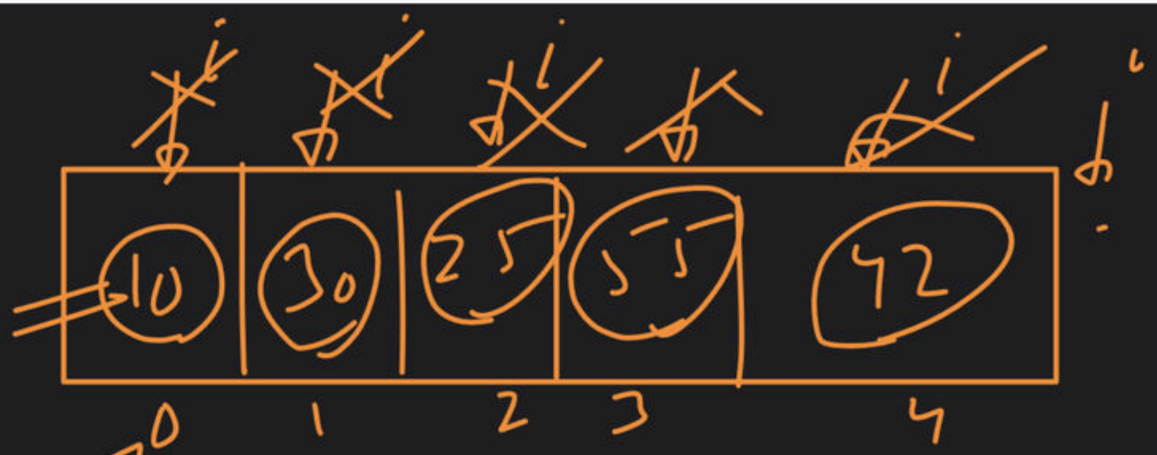
maxAns = arr[0]

maxAns = 10

30

55

arr



-ve

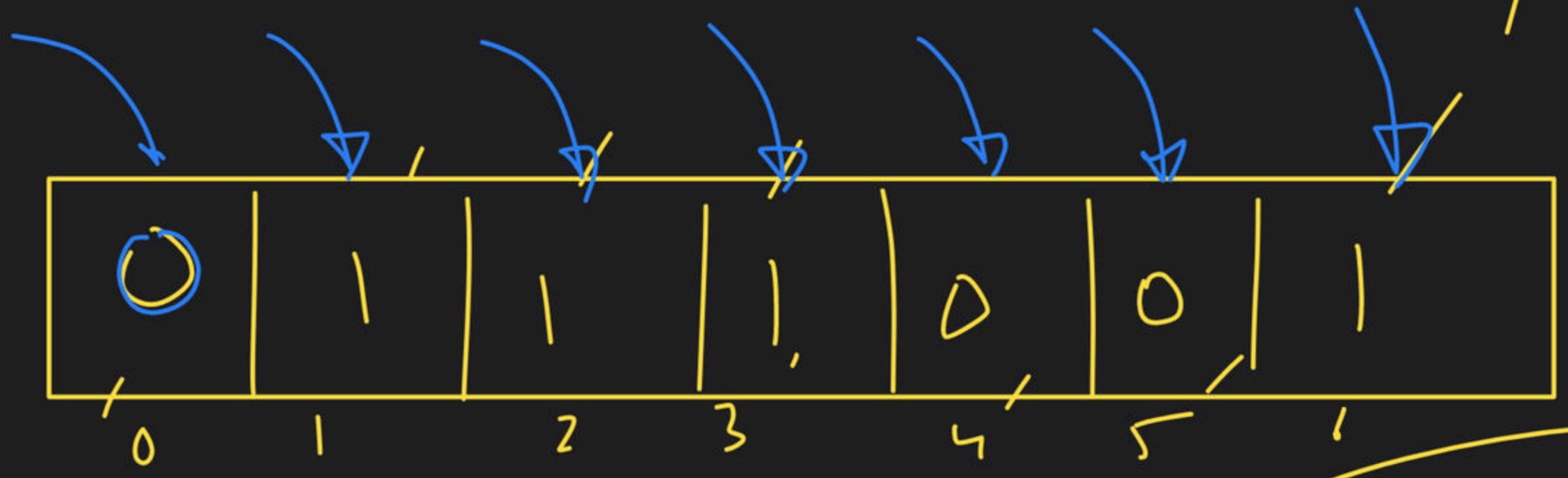
+ve

-ve


Baatein
Pattun

→ i/p → array

count → 0's / 1's

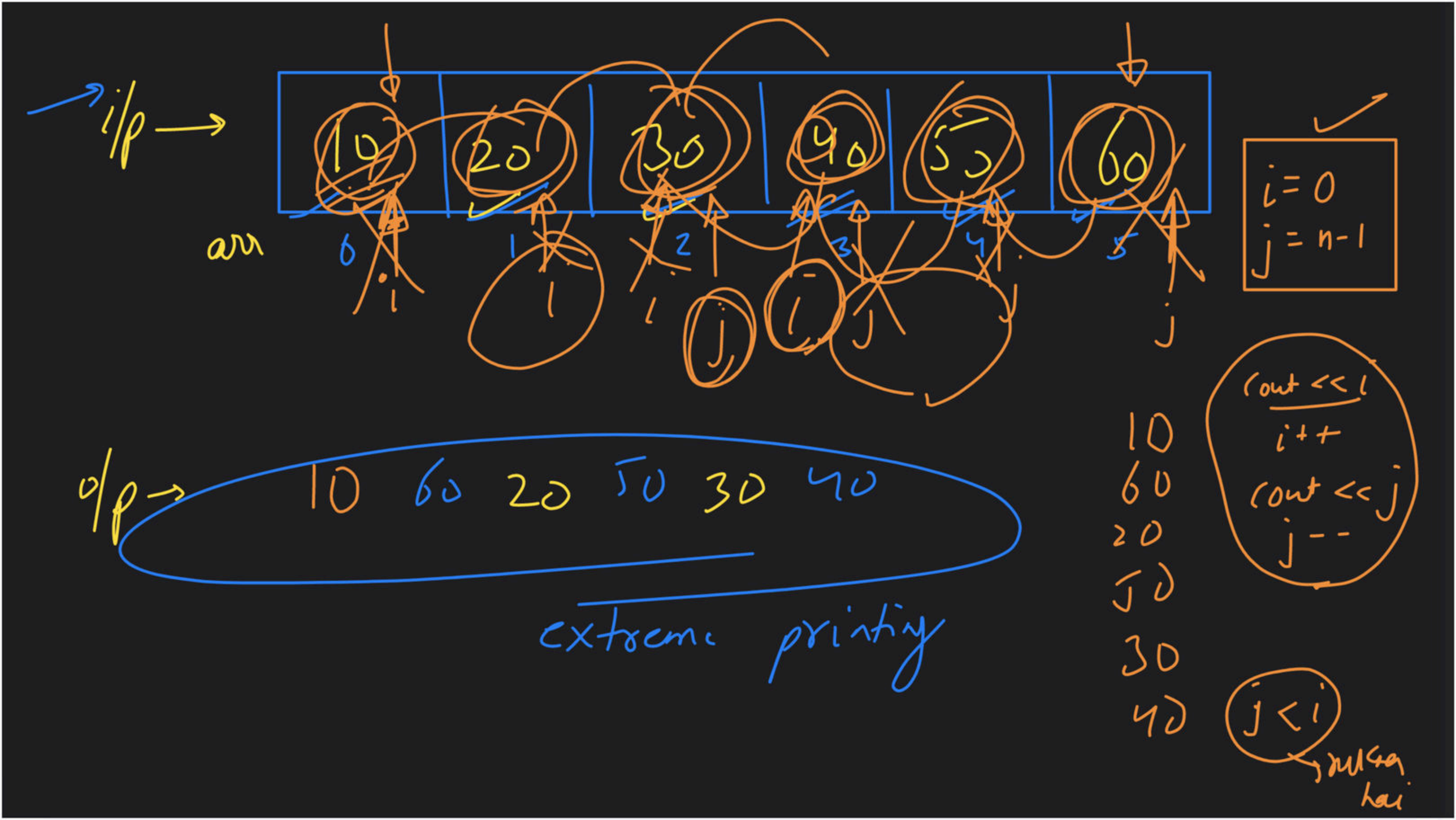


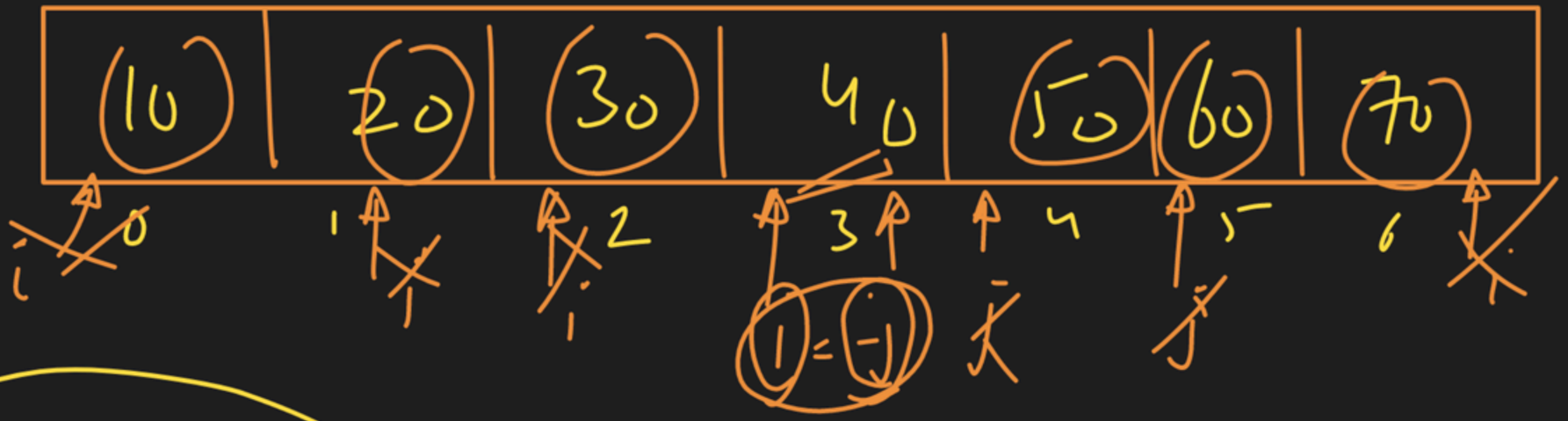
zeroCount = 0+1+1+1 → 3

oneCount = 0+1+1+1+1 → 4

totalZero → 3

totalOne → 4


i/p →
10
20
30
40
50
60
arr
0
1
2
3
4
5
i
j
i = 0
j = n-1
o/p →
10 60 20 50 30 40
extreme printing
cout << i
i++
cout << j
j--
10
60
20
50
30
40
j < i



while (i > j)

{

}

10

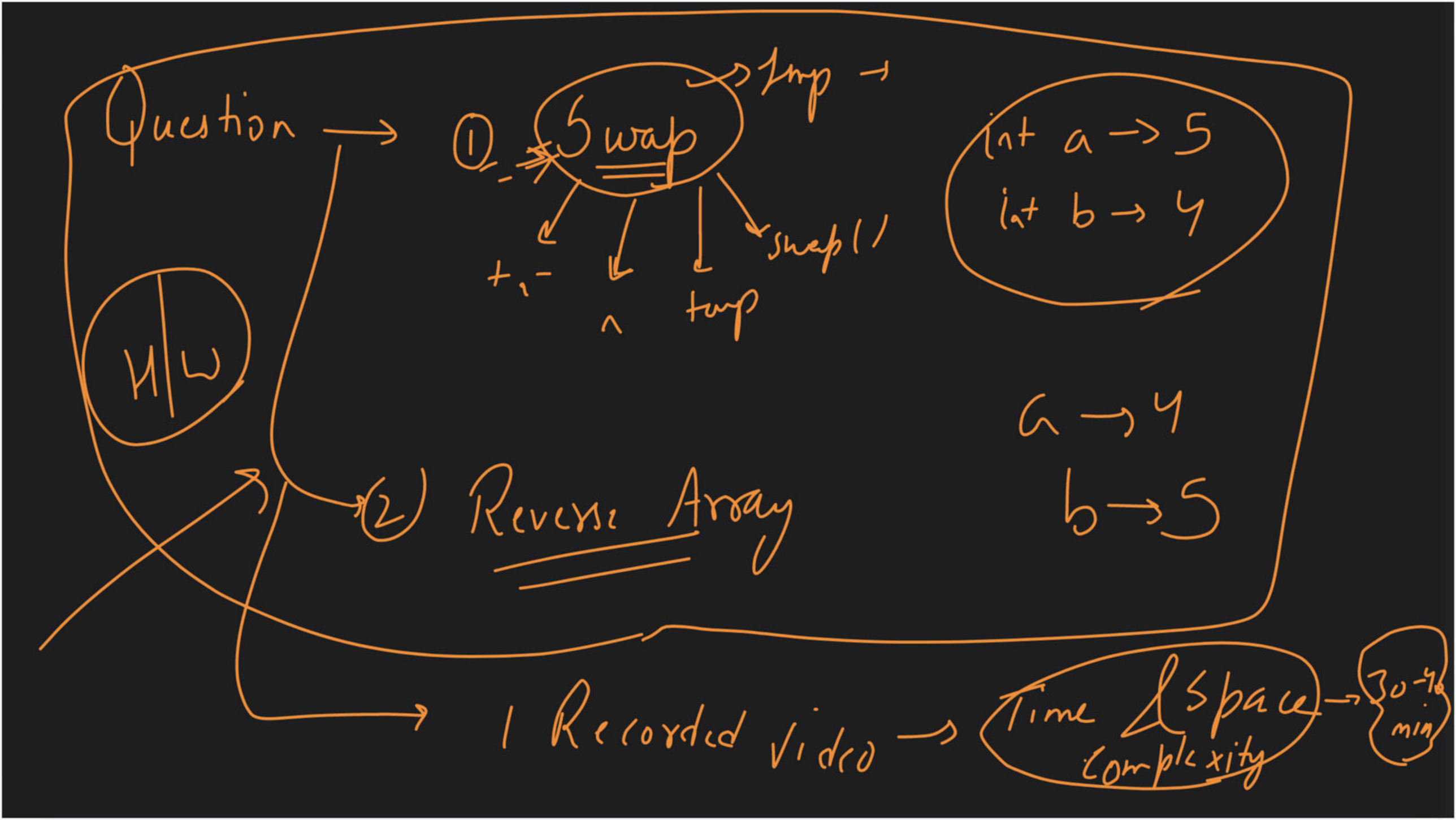
70

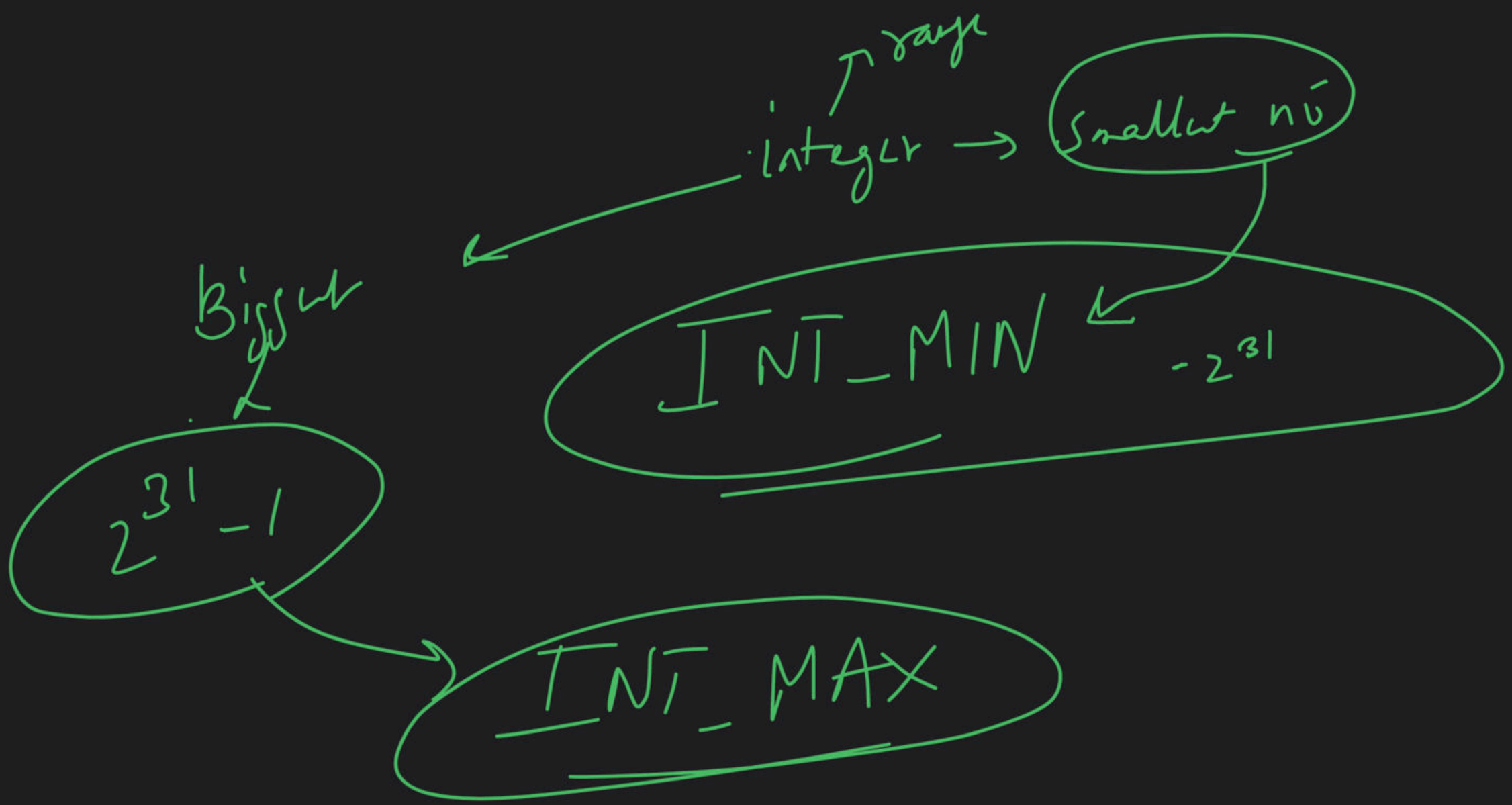
20

60

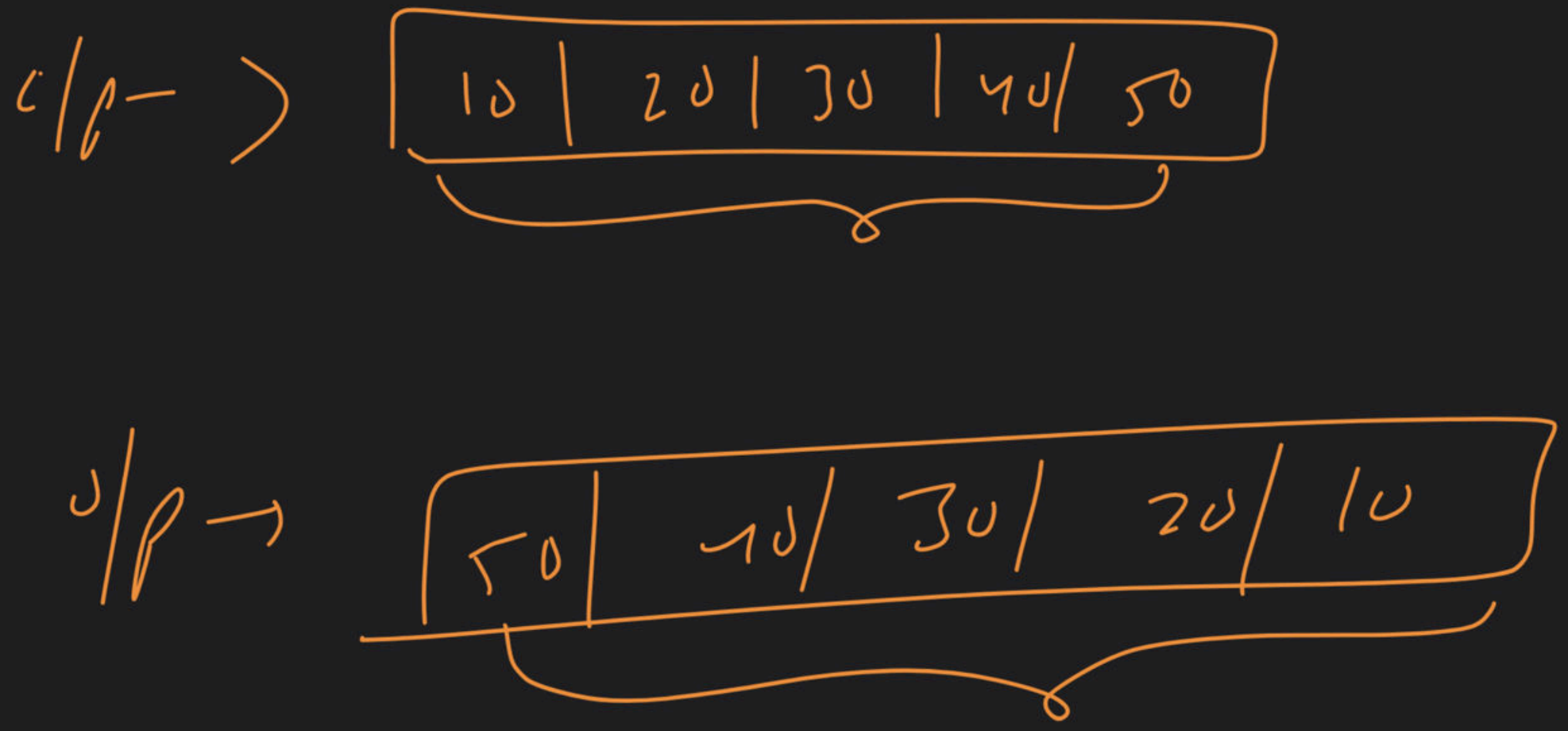
30

50


Question
1
Swap
temp
int a → 5
int b → 4
swap(1
temp
H/W
a → 4
b → 5
2
Reverse Array
1 Recorded Video
Time & Space
complexity
30-40 min


range
integer
smallest no
INT_MIN
-2^{31}
Biggest
$2^{31}-1$
INT_MAX

```
int findMax(arr, n)
{
    int maxAns = INT_MIN;
    for (i=0; i<n; i++)
    {
        if (arr[i] > maxAns)
        { maxAns = arr[i];
        }
    }
    return maxAns;
}
```


i/p →
10 | 20 | 30 | 40 | 50
o/p →
50 | 40 | 30 | 20 | 10