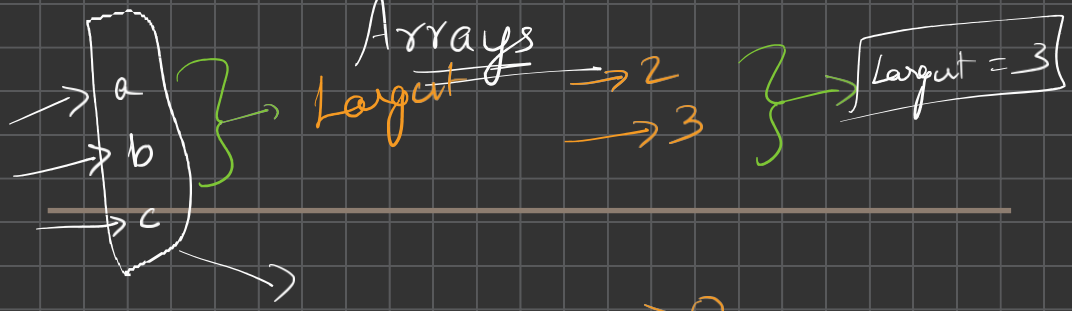


→ 0

10 values

100 values

10000 values

All elements →

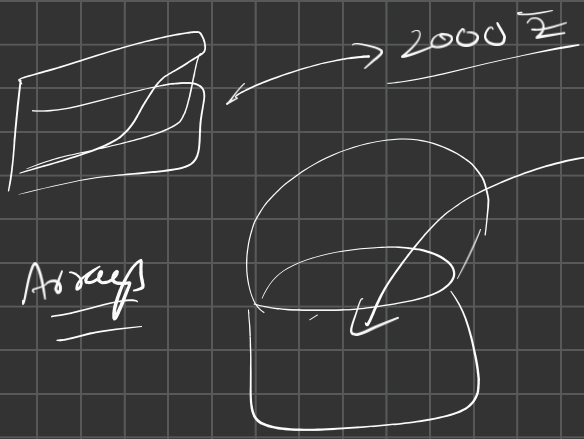
(10000 / n

O(n)

Max =

Array -

3 ft



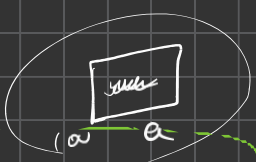
10000 int → array of int 100

1000000 char → array of char

data type → char → 1000 values

array of char

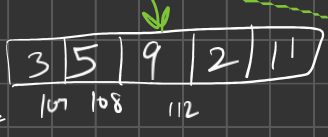
int a;



arrays → similar type of Item

→ integer

→ contiguous location



→ Index → access

why ?

10000 values → 10000 variables

1 variable

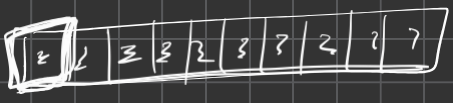
→ what is array ?

→ why ?

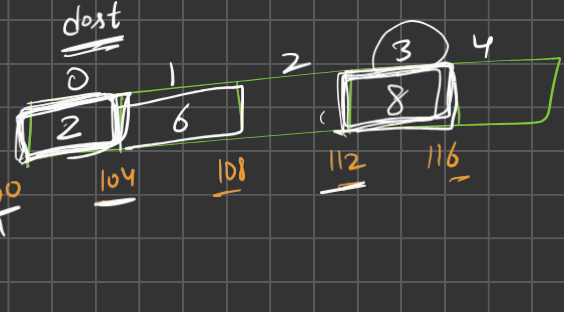
→ Implementation

Declaration → int a;

int dost [10];



int v[5]



cout << v[3] → 8

100 + 3x4 = 112

v[0] → 1st location → 2

cout << v[0]; → 2

v[1] → 100 + 1x4 = 104

cout << v[1] → 6

→ Declaration

→ Access → int dost[10] → 0-9

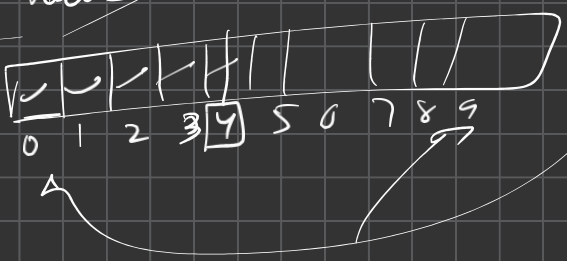
array (n size)

↓

Index

0 - (n-1)

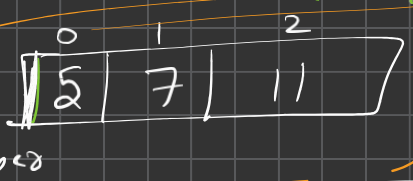
5th location value → dost[4]



→ Initialisation

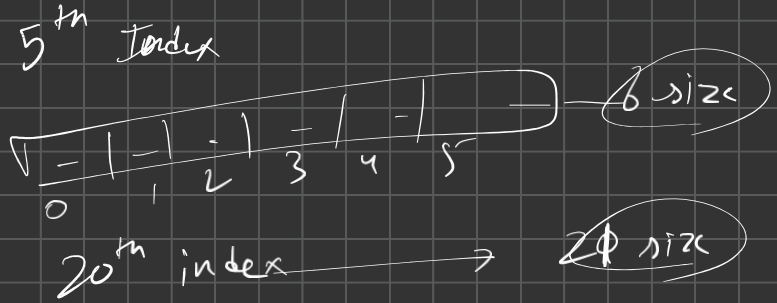
int a=5;

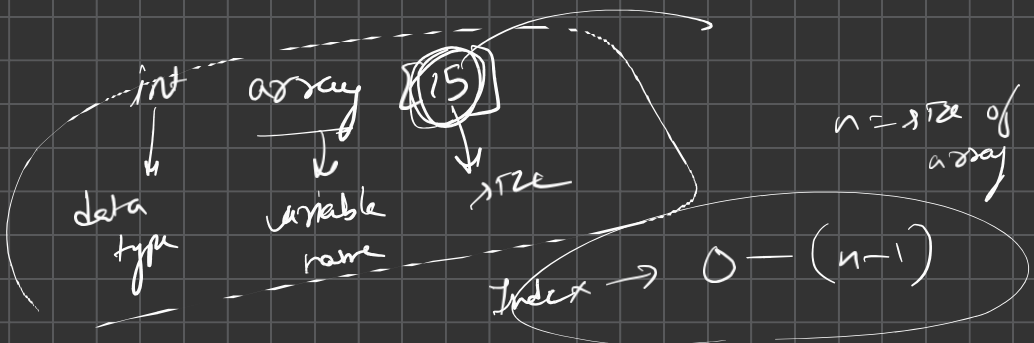
int number[3] = {5, 7, 11}



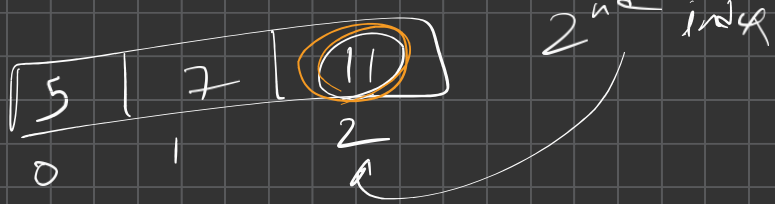
int array[100000] = {1};

H/w → entire array ko kisi bhi value se initialise kaise karu?





second[3] = {5, 7, 11}



Access

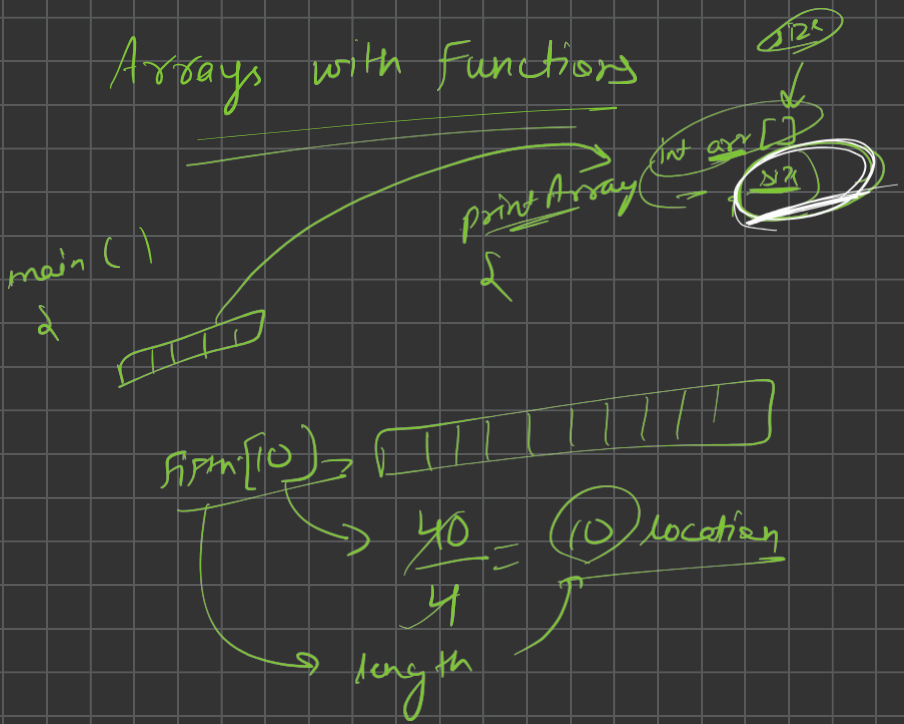
array_name[i];

index

i/p

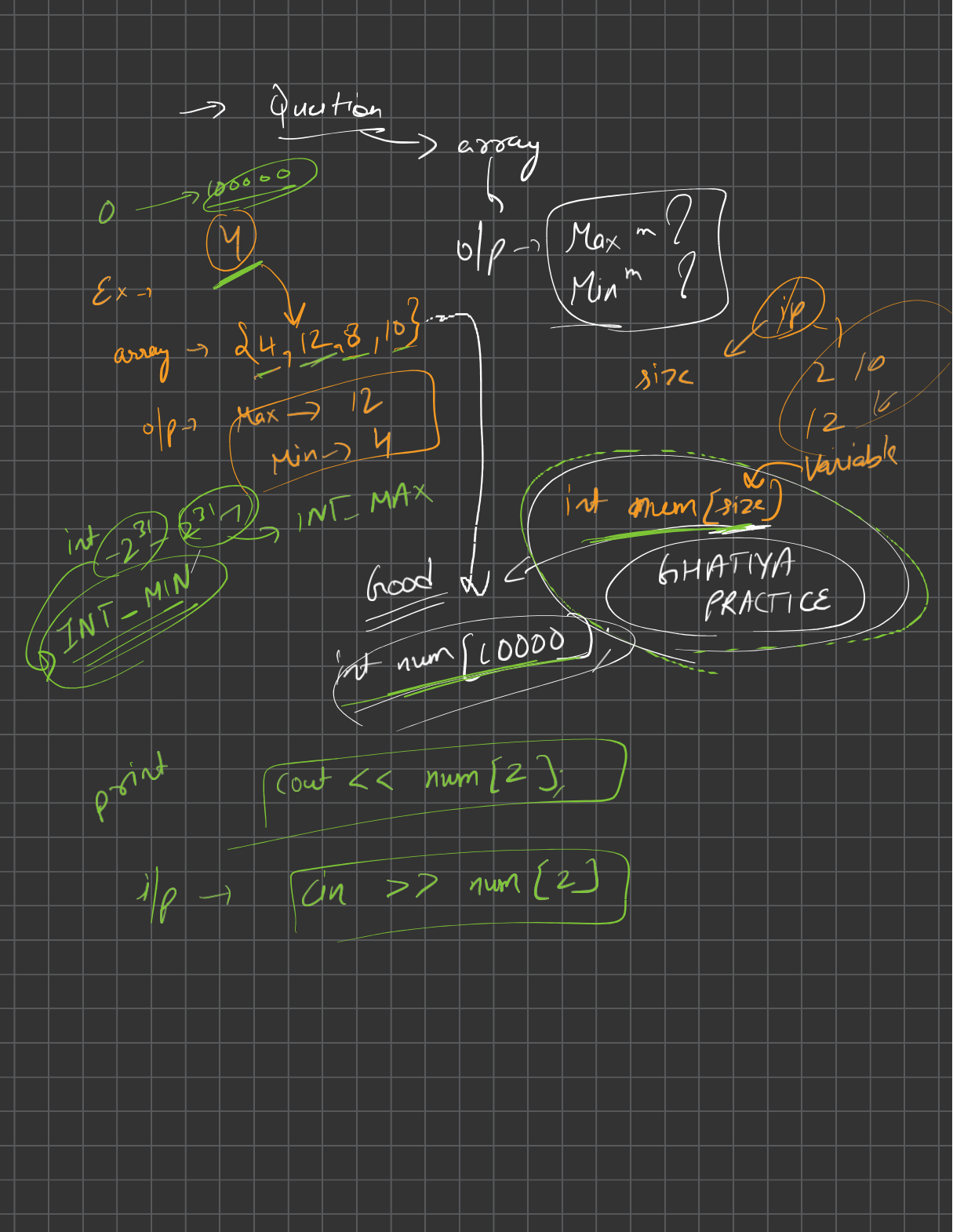
0 - (n-1)

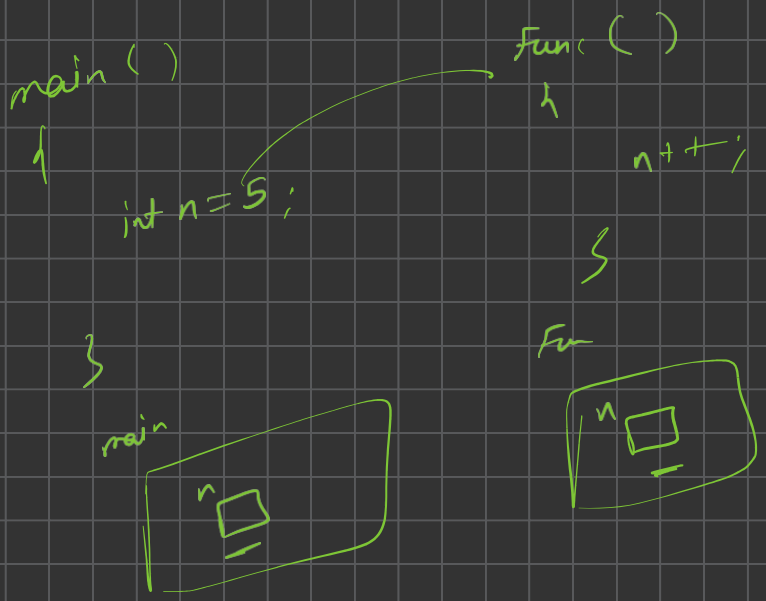
Arrays with Functions

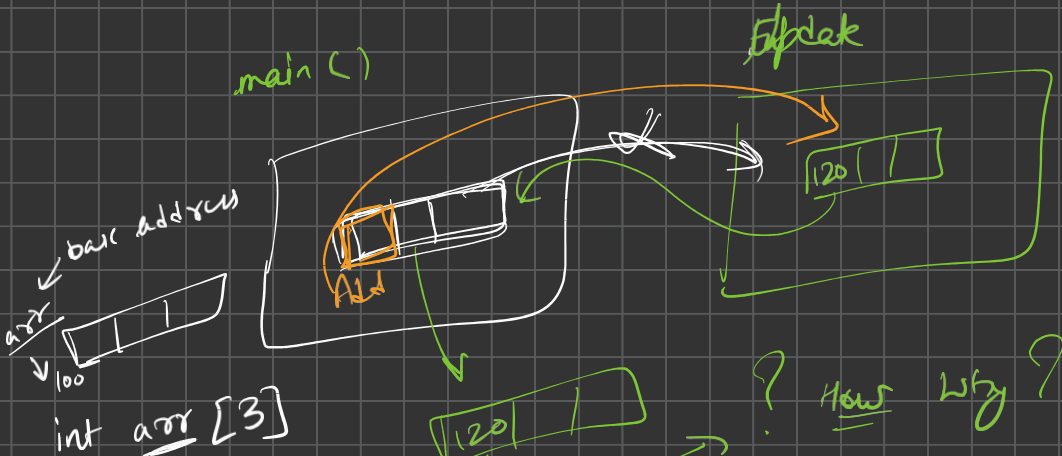


fifth[10] → 40 / 4 = 10 location

length

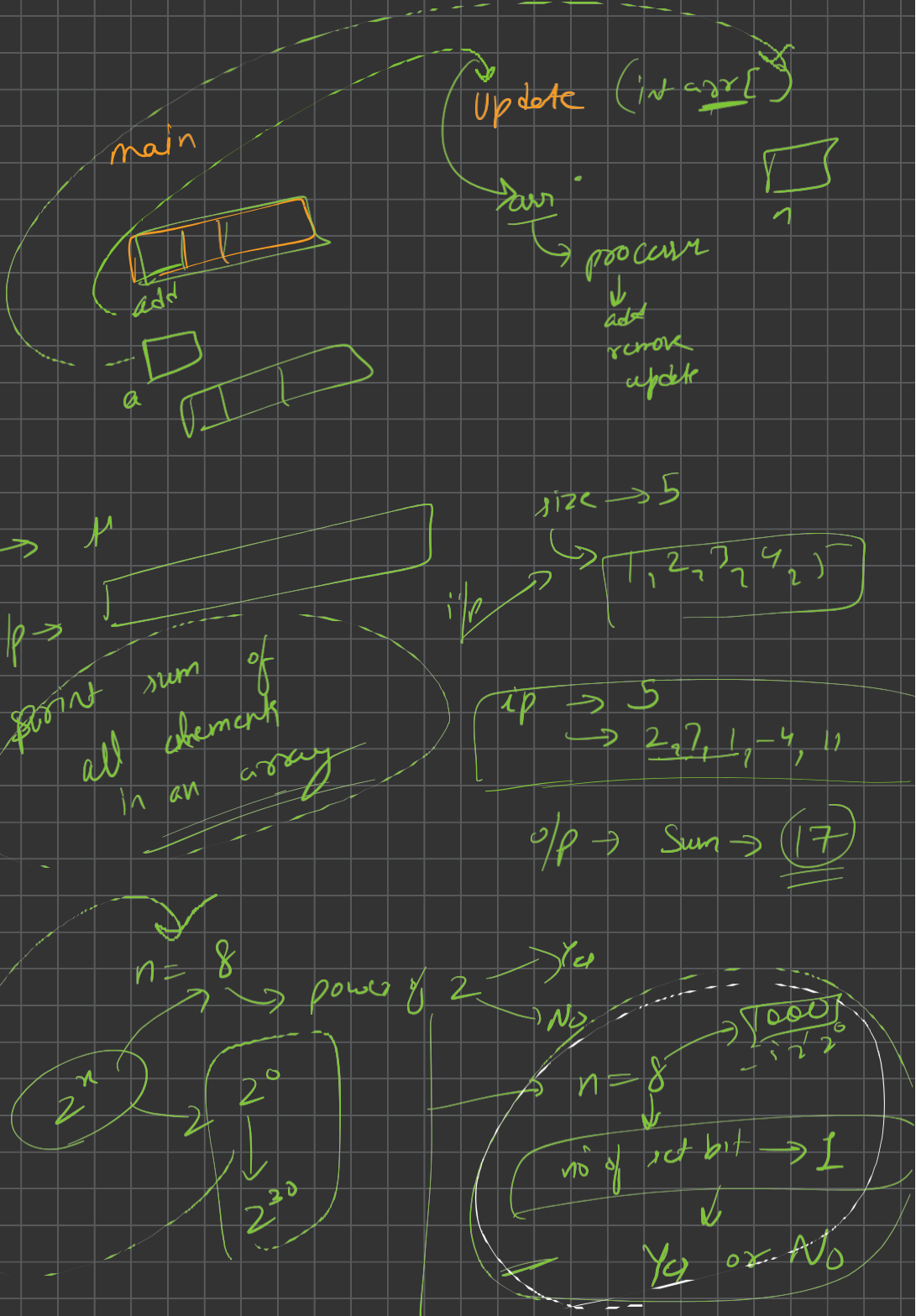

Question
array
o/p → Max = ? Min = ?
0
100000
N
Ex →
array → {4, 12, 8, 10}
o/p → Max → 12
Min → 4
i/p
size
1
2
10
12
16
variable
int num[size]
GHATIYA PRACTICE
int
-2^{31}
$2^{31}-1$
INT_MIN
INT_MAX
Good
int num[10000]
print
cout << num[2];
i/p →
cin >> num[2]





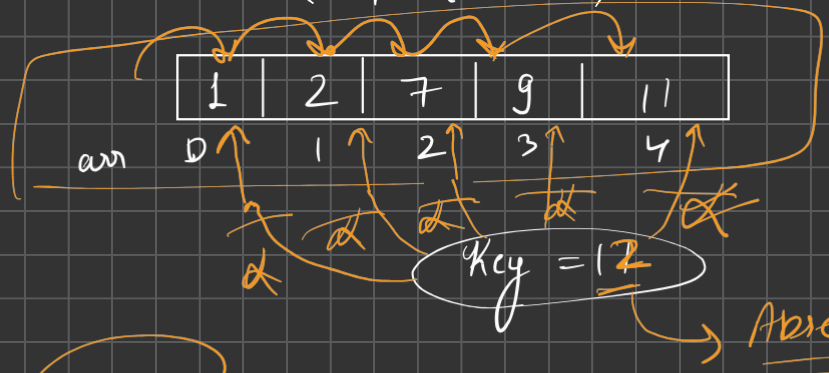
sig^{n} ⟶ Update (int arr[ ], int n)

Call ⟶ update (arr , 3)


main
add
a
Update (int arr[])
arr
process
add
remove
update
1
size → 5
1 2 7 7 4
i/p
i/p → n
print sum of all elements in an array
i/p → 5
→ 2, 7, 1, -4, 11
o/p → Sum → 17
n = 8
power of 2
Yes
No
2^n
2^0
2^{30}
1000
n = 8
no of set bit → 1
Yes or No

Linear Search

arr [5] → {1, 2, 7, 9, 11}



==

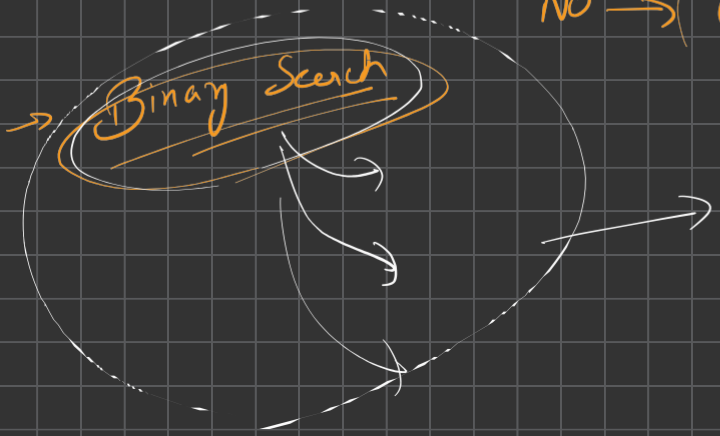
arr [i] == Key

2 4

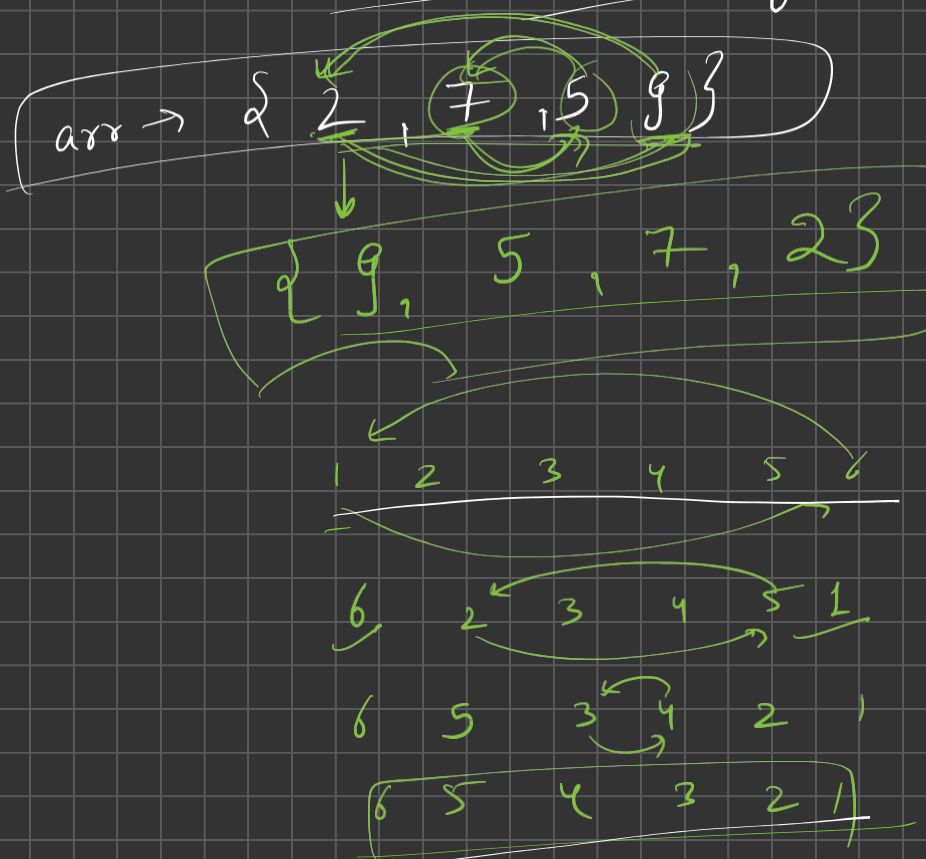
2 == 4

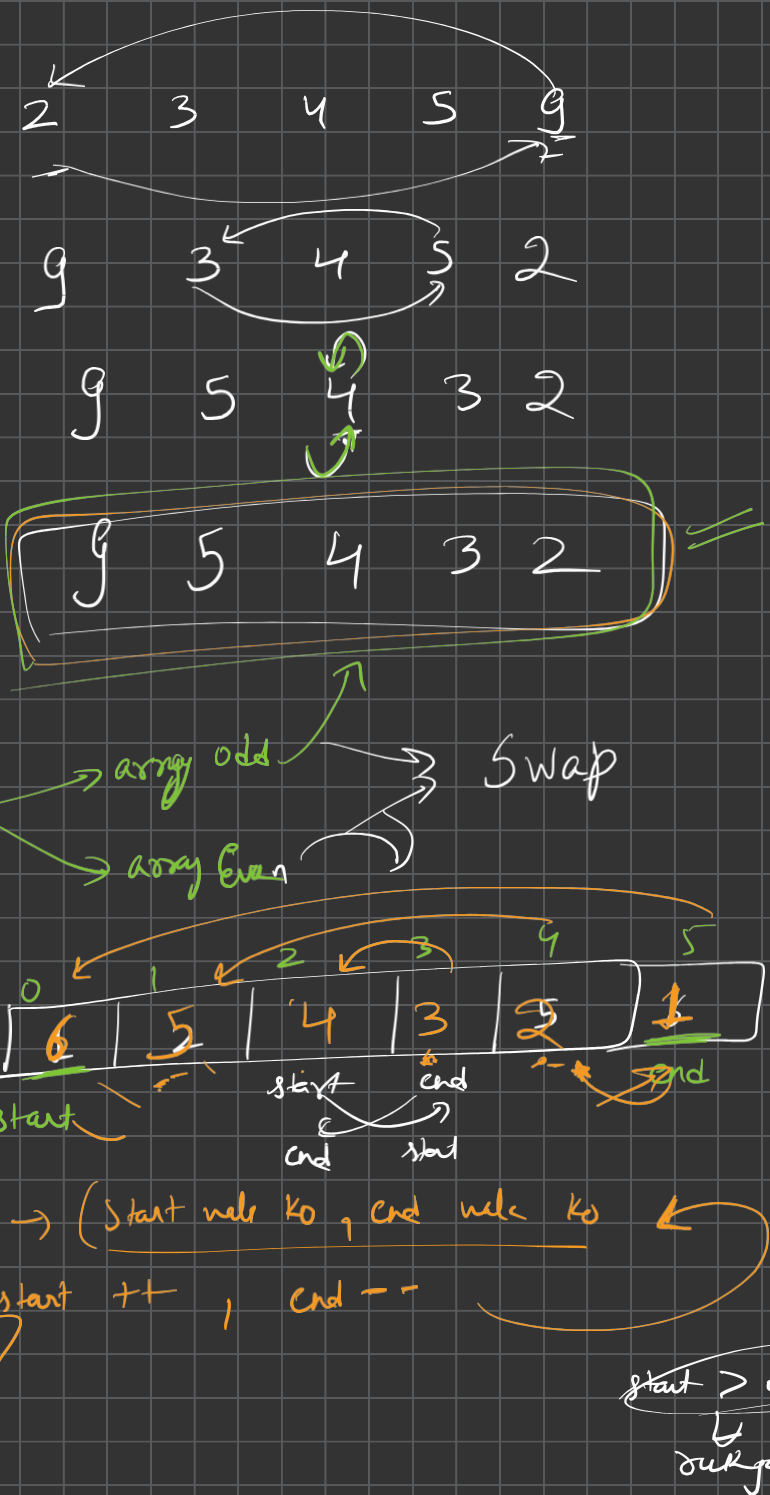
Kya ye Equal hai

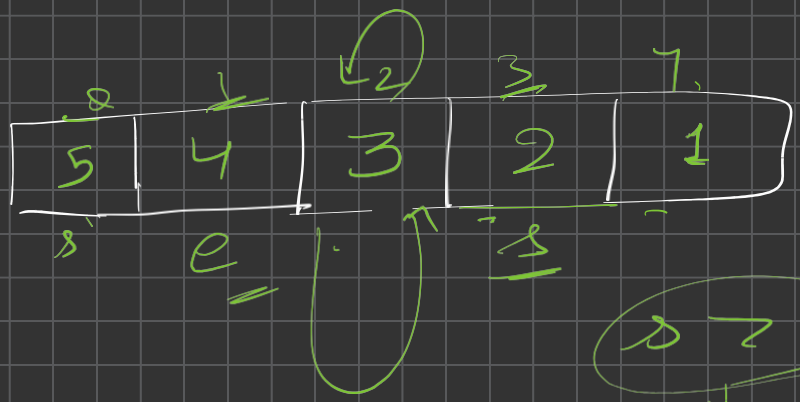
NO → 0

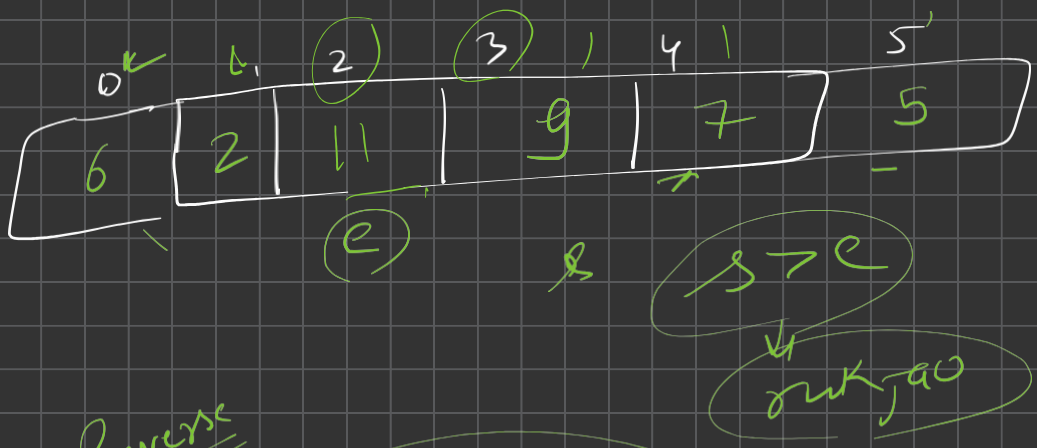


Reverse an array :-

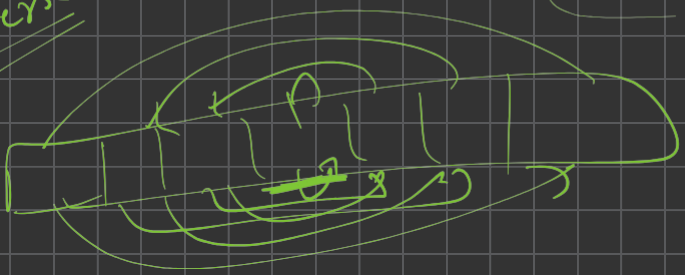



2 3 4 5 9
9 3 4 5 2
9 5 4 3 2
9 5 4 3 2
Obs:
array odd
array Even
Swap
Index → 0 1 2 3 4 5
6 5 4 3 2 1
start
start
end
end
start
end
Algo
→ swap → (start wale ko , end wale ko
→ start ++ , end --
start > end
↳ dusra gana





Reverse



→ swap alternate {1, 2, 3, 4, 5, 6}

o/p {2, 1, 4, 3, 6, 5}

→ find unique

→ find duplicate

→ Array intersection → {1, 2, 3}

{2, 4, 6, 8}

o/p → {2, 4}

→ pair sum

→ Triplet sum

→ sort 0's & 1's

Important Notes :

- 1 - Jb bhi hume Min nikalna hoo interger ke case me to , Min=INT_MAX LIKHENGE

- 2 - Array ke case me hum array ka address dete hai means agr array ki kisi value ko hum kisi function me update krte hai to , wo original me hoga , because waha array ko humne uska original address dia hua hai

- But in case of single variable , hum kisi function me us variable ki Copy send krte hai , means agr wo function me update krta hai to wo original me nhi hoga

- Array ke case me size nikalenge size operator se , But in case of string hum size niklaenge via length operator se

- Array me indexing Humesha 0 se start hoga to array ko access hum 0 to n-1 tk kr skte hai