

In this Video, we are going to learn about 2D Arrays and solve the following questions:

- Row-wise Input/Column wise input
 - Linear Search over 2D arrays
 - Row-wise Sum/Column wise Sum
 - Largest Row sum
 - Wave Print
 - Spiral Print
 - Rotate a Matrix by 90 degrees
 - Search in a 2D Matrix I LeetCode
 - Search in a 2D Matrix II LeetCode
-

There is a lot to learn, Keep in mind “ Mnn boot karega k chor yrr apne se nahi yoga ya maza nahi para, Just ask 1 question “ Why I started ? ”

Visit Relevel: <https://relevl.co/prwb>

Stack Overflow Link: <https://stackoverflow.com/questions/1...>

Discord Server Link: <https://discord.gg/feSQvVXMrd>

Course Flow: <https://whimsical.com/dsa-4-placement...>

Homework: timestamps added below

Notes Link: <https://drive.google.com/file/d/1Vng3...>

Code Links: <https://github.com/loveBabbar/CodeHel...>

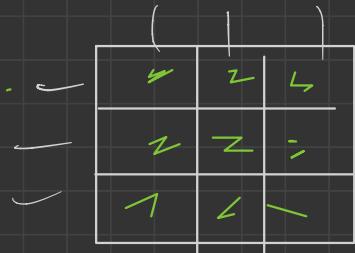
Question Links:

- Wave Print: <https://bit.ly/329Le3K>
- Spiral Print: <https://leetcode.com/problems/spiral-...>
- Rotate a Matrix by 90 degrees: <https://leetcode.com/problems/rotate-...>
- Search in a 2D Matrix I LeetCode: <https://leetcode.com/problems/search-...>
- Search in a 2D Matrix II LeetCode: <https://leetcode.com/problems/search-...>

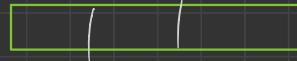
Do provide you feedback in the comments, we are going to make it best collectively.



2-D Arrays



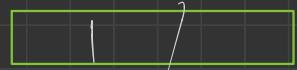
arr1



arr2

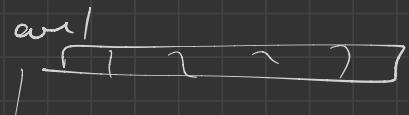


arr3



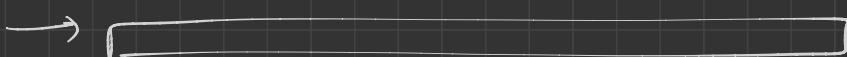
Ques:-

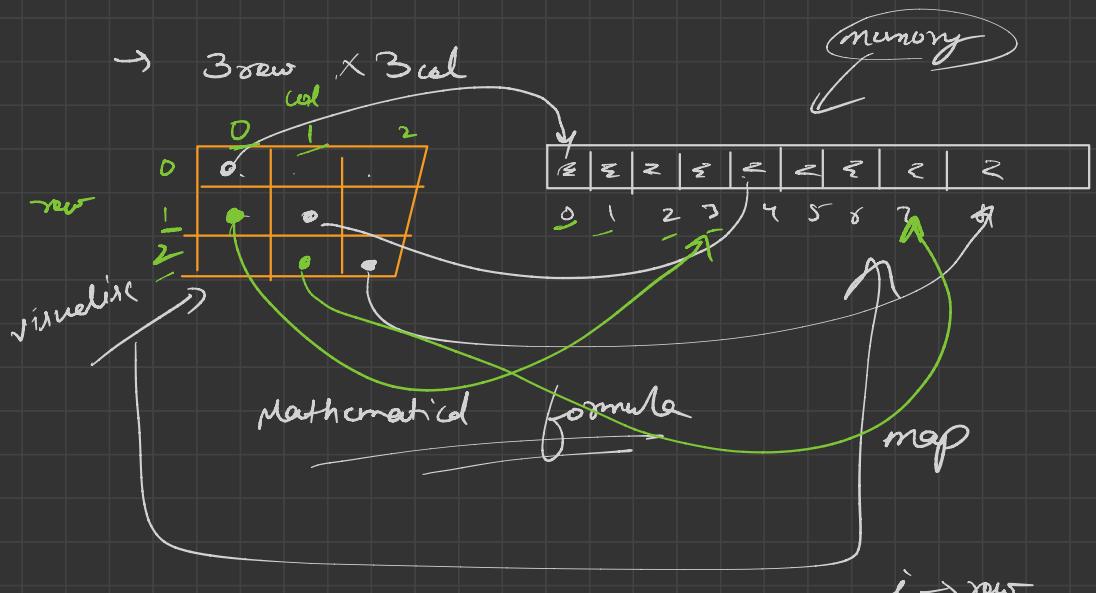
10 rows, 10 col



arr10

arr
1000 rows x 1000 col





$$= c \times \underline{i} + \underline{j} \quad j \Rightarrow \text{col}$$

$$\begin{aligned}
 L^{\text{ext}} & \quad \text{row} \quad | \quad 0^{\text{in}} \quad \text{col} \\
 & = 3 \times 1 + 0 \\
 & = \boxed{3}
 \end{aligned}$$

$$\begin{aligned}
 2^{\text{nd}} \quad \text{row} \quad | \quad \text{col} & \Rightarrow c \times i + j
 \end{aligned}$$

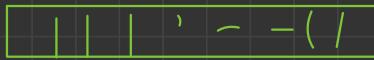
$$\begin{aligned}
 & \Rightarrow 3 \times 2 + 1 \\
 & = \boxed{7}
 \end{aligned}$$

①

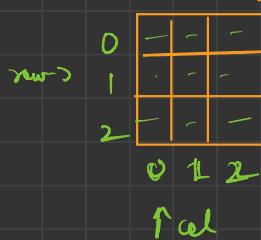
Create 2D array

1D →

int arr[10];



int arr [3][3];



input → ?

1D array
→ cin >> arr[i]

2D array

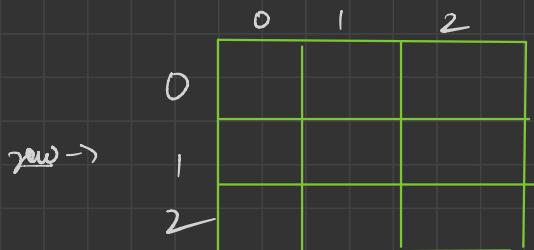
→ cin >> arr[i][j]

Output ?

1D
→ cout >> arr[i] col

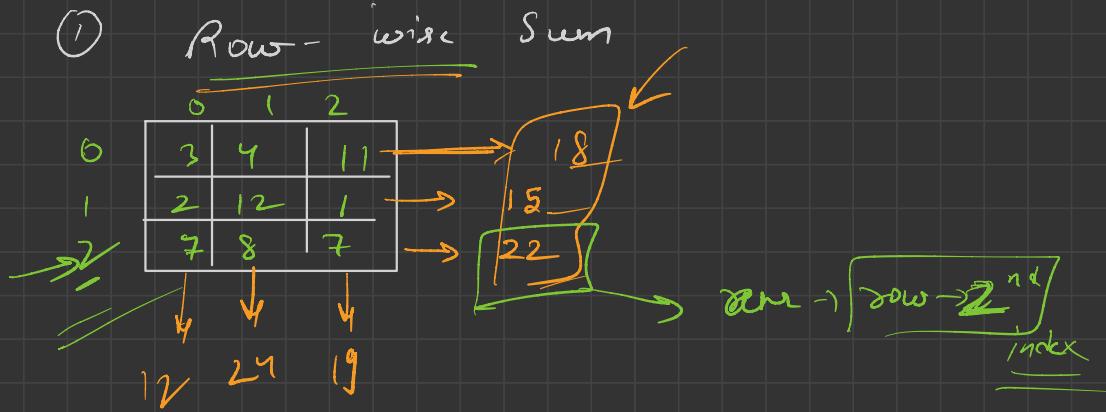
2D

→ cout << arr[i][j]

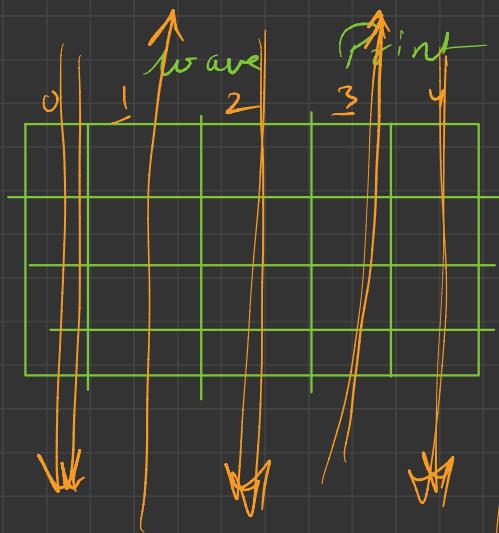


arr[2][1]

arr[2][2]



② Largest Row Sum → ?



arr[5][1]

Obj -

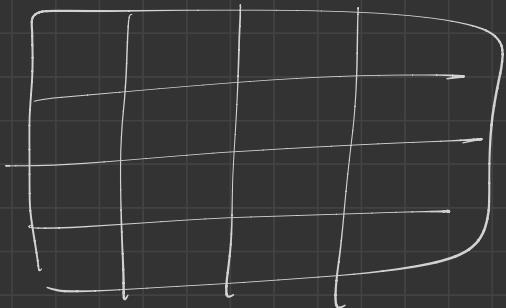
col Index → Odd
Bottom to Top

col Index → even or 0
Top to Bottom

Spiral Print

start row
start col
end row
end col

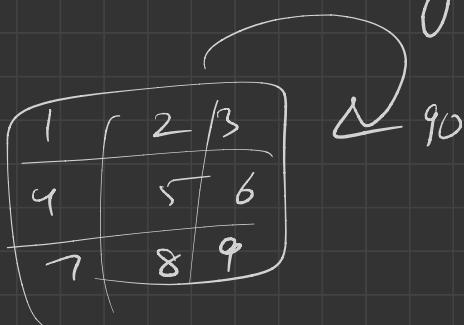
1	2	3
4	5	6
7	8	9



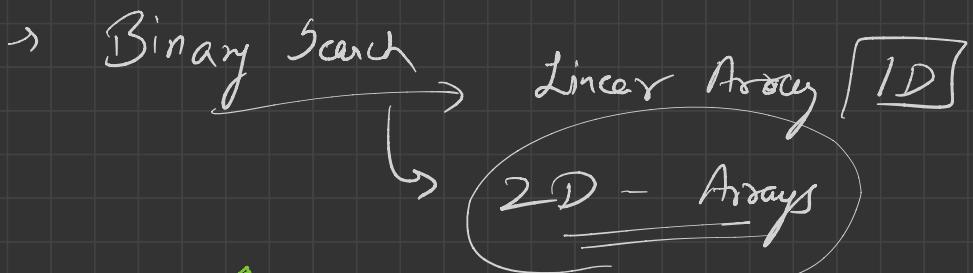
Approach:-

- Print starting row Row
- Ending column print Col
- Ending row point Rows
- starting col point Cols

① Rotate matrix by 90 degrees



Homework



			col $\leftarrow n$	
row $\leftarrow m$	0	1	2	3
1	1	3	5	7
2	10	11	16	20
3	23	30	34	66

$$11/4 = 2 \quad n = \text{row} \times \text{col}$$

$$O(\log(n))$$

$$\boxed{1 \ 3 \ 5 \ 7 \ | \ 10 \ 11 \ 16 \ 20 \ 23 \ 30 \ 34 / 66}$$

$$0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11$$

$$start = 0$$

$$end = \text{row} \times \text{col} - 1$$

$$6/4 = 1 \rightarrow$$

$$O(\log(mn))$$

$$(mn \times col - 1)$$

$$(3 \times 4 - 1)$$

$$(12 - 1)$$

$$11$$

mid

↓

Case 1: $\text{arr}[\text{mid}] \geq \text{target}$
↳ found → return

(Case 2) $\rightarrow \text{arr}[\text{mid}] < \text{target}$
 $s = \text{mid} + 1$

(Case 3) $\rightarrow \text{arr}[\text{mid}] > \text{target}$
 $e = \text{mid} - 1$

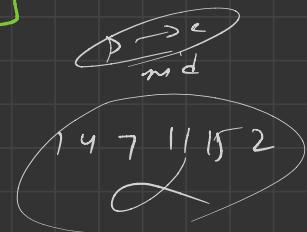
row index col Index

mid

matrix [] []

Search 2D matrix \rightarrow II

Case 1: $\text{arr}[\underline{\text{mid}}] = \underline{\text{target}}$
 \rightarrow return ans :-



$$\text{target} = 10$$

Case 2 $\text{arr}[\underline{\text{mid}}] < \underline{\text{target}}$

\rightarrow row++

$$9 < 10$$

Case 3 $\text{arr}[\underline{\text{mid}}] > \underline{\text{target}}$

\rightarrow col--

$$17 > 10$$