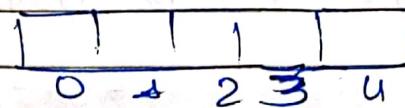


12/9/23

Class 5 Array Level -> 5

APCQ
Date: 156

1-D Array \rightarrow $\text{int arr}[5] = \{1, 2, 3, 4, 5\};$



2-D Array \rightarrow $\text{int arr}[2][5] = \begin{matrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 & 7 \end{matrix}$

Matrix $\left[\begin{matrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 & 7 \end{matrix} \right]$

max Row x Col
5x5

Initialization \rightarrow 1. $\text{int arr}[5];$

2. $\text{int arr}[5][10];$

5 Rows 10 Columns

2 rows 4 columns

$\text{int arr}[2][4] = \{$

$\{10, 20, 30, 40\},$

$\{1, 2, 3, 4, 5\},$

$\{0\}$

$\text{int arr}[4][3] = \{$

- $\{1, 2, 3\},$
- $\{4, 5, 6\},$
- $\{7, 8, 9\},$
- $\{1, 2, 3\},$

12/9/23

APCO
Date: 457

Access \rightarrow arr [i][j]

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

arr[0][0] \rightarrow 1
arr[0][1] \rightarrow 2
arr[0][2] \rightarrow 3
arr[1][0] \rightarrow 4
arr[1][1] \rightarrow 5
arr[1][2] \rightarrow 6
arr[2][0] \rightarrow 7
arr[2][1] \rightarrow 8
arr[2][2] \rightarrow 9

What / How getting stored in Memory \rightarrow

int arr [3][3]

	0	1	2
0			
1			
2			

	0	1	2	3	4	5	6	7
0								
1								
2								

arr[1][1] \rightarrow How to get
know which

formula \rightarrow C * i + j

\downarrow
No of columns

12/9/23

APCO
Date: 158

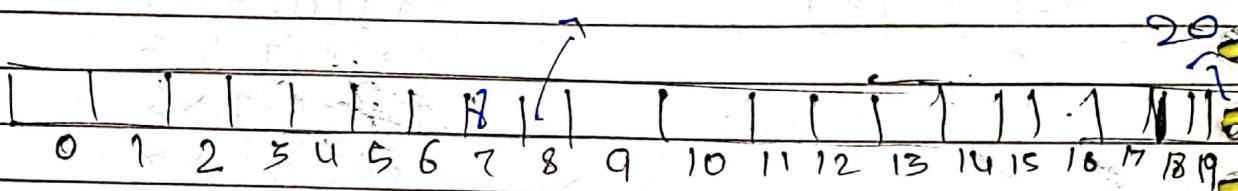
	0	1	2	3	
0	1	2	3	4	
1	5	6	7	8	
2	9	10	11	12	
3	13	14	15	16	
4	17	18	19	20	

$$C = 4$$

$$\rightarrow C \leftarrow i + j$$

$$4 * (1+3) = 16$$

$$4 * 4 + 3 = 19$$



Note \rightarrow At least column size to Initialize 2-D Array, because it also in function we need $i + j$ no of column to conversion

Point row first then column

1	2	3
4	5	6
7	8	9

$\rightarrow 1 2 3 4 5 6 7 8 9$

for ($i=0$; $i < \text{row}$; $i++$) {

 for ($j=0$; $j < \text{col}$; $j++$) {

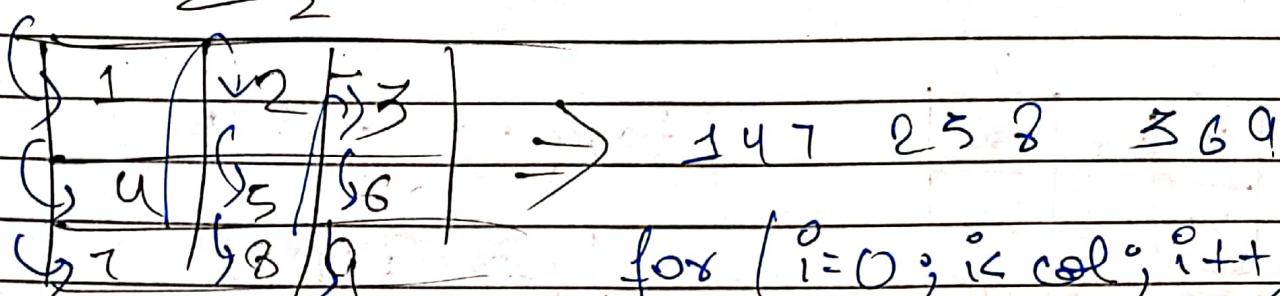
 cout << arr[i][j];

}

12/01/23

A.P.E.O.
Date: 159

Col wise access \rightarrow



cout << arr[j][i]
 }
}

Input \rightarrow for ($i = 0; i < \text{row}; i++$) {
 for ($j = 0; j < \text{col}; j++$) {
 cin >> arr[i][j];
 }
}

Row wise \rightarrow for ($i = 0; i < \text{col}; i++$) {
 for ($j = 0; j < \text{row}; j++$) {
 cout << arr[i][j];
 }
}

Col wise \rightarrow for ($i = 0; i < \text{col}; i++$) {
 for ($j = 0; j < \text{row}; j++$) {
 cout >> arr[j][i];
 }
}

32/01/23

APCO
Date: 16/01/23

Linear Search in 2-D Array

	0	1	2
0	10	20	30
1	40	50	60
2	70	80	90

target = 70 $i \in [0, 2]$

$o/p \rightarrow T/P$

```
for (i=0; i<row; i++)  
    for (j=0; j<col; j++)
```

if target == arr[i][j]
 return true;

}

return false.

(S3)

Max. No in an array

int maxAns = INT_MIN;

no > maxAns

maxAns = no

No.
 $\rightarrow arr[i][j]$

102 \rightarrow Min No

minAns = INT_MAX;

no < minAns

minAns = no

Q. →

Row of Row

	0	1	2	3	Col
0	10	20	5	7	Sum
1	2	4	6	8	Sum
2	10	15	25	10	Sum

Row-Wise Sum
42
20
50

(S-1) → Access Row-wise
(S-2) → Then add

for ($i \rightarrow i < r$)
int sum = 0

for ($j \rightarrow j < c$)

sum += arr [i] [j]

cout << sum;

} y

ii) Col-Wise Sum

	0	1	2	3
0	10	20	5	7
1	2	4	6	8
2	10	15	25	10

Plan

(S-1) → Access Column wise

(S-2) → Then Add

for ($i \rightarrow i < c$)

int sum = 0

for ($j \rightarrow j < r$)

sum += arr [j] [i]

} y

cout << sum;

} y

12/9/23 (Square pattern)

③ \Rightarrow Diagonal-wise sum \rightarrow

	0	1	2	3	4
0	1	6	11	16	21
1	2	7	12	17	22
2	3	8	13	18	23
3	4	9	14	19	24
4	5	10	15	20	25

$$arr[0][0] + arr[1][1] + arr[2][2] \\ + arr[3][3] + arr[4][4]$$

\Rightarrow for ($i=0$; $i < \text{row/col}$; $i++$)

$$\text{sum} = arr[i][i];$$

H/W \rightarrow Q3

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

$$arr[3][0] + arr[2][1] + arr[1][2] + arr[0][3]$$

$$3+0, 2+1, 1+2, 0+3 = 3$$

If ($i+j = n-1$)

$$\text{sum} = arr[i][j]$$

12/9/23

A B C D
Date: 161

① \rightarrow Transpose of a Matrix

	0	1	2
0	1	4	2
1	4	5	6
2	7	8	9

Transpose \rightarrow

	0	1	2
0	1	4	7
1	4	5	8
2	3	6	9

$i, j \leftrightarrow j, i$

(1)

$$\text{arr}[0][0] \rightarrow [0][0]$$

$$(2) \text{arr}[0][1] \rightarrow [1][0] \rightarrow s-1$$

$$\text{arr}[0][2] \rightarrow [2][0]$$

$$\text{arr}[1][0] \rightarrow [0][1] \rightarrow s-2$$

$\star \rightarrow$ Agar ek bar swap hogya Tolu kro

$$\hookrightarrow \text{arr}[0][0] \rightarrow [0][0]$$

$(0, 1)$

$$\rightarrow \text{arr}[0][1] \leftrightarrow [1][0]$$

$(0, 2)$

Right

Half Triangle

$$\rightarrow \text{arr}[0][1] \leftrightarrow [2][0]$$

$(1, 1)$

$$\rightarrow \text{arr}[1][1] \leftrightarrow [1][1]$$

$(1, 2)$

$$\rightarrow \text{arr}[1][2] \rightarrow [2][1]$$

$(1, 3)$

$$\rightarrow \text{arr}[2][2] \leftrightarrow [2][2] \rightarrow$$

12/9/23

APCO

Date: 16/2

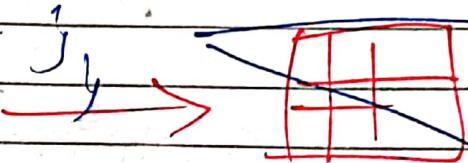
for ($i = 0$; $i < r$; $i++$)

for ($j = i$; $j < col$; $j++$)

(swap $\text{arr}[i][j] \leftrightarrow \text{arr}[j][i]$)

only

Right half



right half

If wanna do Left half

(FLW)

for ($i \rightarrow i < r$) {

for ($j = 0$; $j \leq i$; $j++$) {

swap
↓
↓

Vector \rightarrow 2-D vector

vector <vector<int>> arr



\rightarrow 2D :::

Syntax → $\text{vector} < \text{vector} < \text{int} > \gg \text{arr}[5, \text{vector} < \text{int} > (10, 0)]$

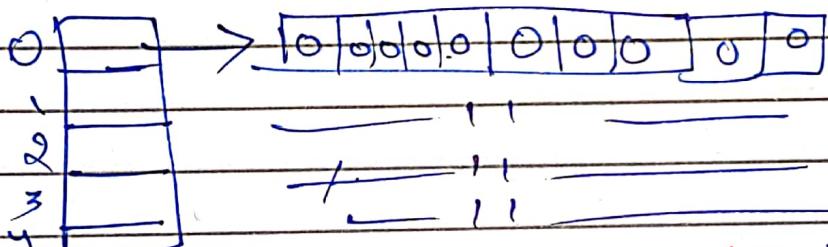
↓
 2-D vector name row
 of size hor
 vectors er
 Row
 Me

$\text{vector} < \text{int} > (10, 0)$

like address 10 cell

with 0 value

in every cell



~~arr.size()~~

Point →

~~for (i = 0; i < r)~~
~~for (j = 0; j < c)~~ * $\rightarrow \text{arr}[i].size()$
~~cout << arr[i][j]~~

Tagged array → $\text{vector} < \text{vector} < \text{int} > \gg \text{arr}$

Every column
can have different
columns

$\rightarrow \text{vector} < \text{int} > v1(5, 1);$

$\rightarrow \text{vector} < \text{int} > v2(4, 2);$

$\text{arr.push_back}(v1);$

$\text{arr.push_back}(v2);$

Yes, In vector it
can happen

0 | \rightarrow 5 5 5 5 5
 4 4 4 4