Date 31/10/2023 Page

Day - 44

2D Arrays Interview Questions

* **Column Major order:**

⇒ In Column Major order, the element stores in memory ~~in~~ as column wise.

⇒

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

~~0~~

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

⇒

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

Index =

~~m~~ ~~Famous~~

⇒ ~~We~~ ~~The~~ ~~Programming~~

⇒ C++ uses Row major order.

⇒ In real life, we use both row & column major order according to our need.

* **Vector in 2D:**

⇒ Creating a 2D vector

⇒ vector ~~<int>~~ <vector <int>> matrix;

⇒ vector <vector <int>> matrix (rows, vector <int> (col, initialize));

⇒ For finding rows —

row = matrix.size();

Date 31/10/2023 Page

=) For finding col —

col = matrix [0]. size();

* Wave Form

	0	1	2	3
0	3	6	4	2
1	7	8	11	5
2	9	3	2	1
3	17	8	5	9

=) we have to print the elements in wave form.

=) So, here, when our col no. is —

even (up → down)

odd (down → up)

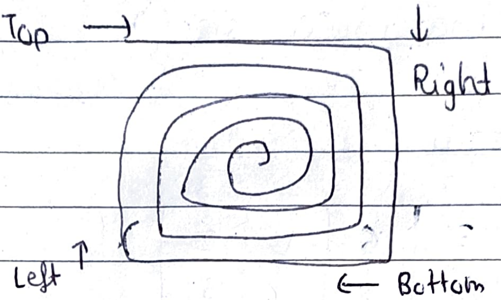
Code

```
for( int j = 0; j<col; j++){
    if( j%2 == 0)
        up → down
    else
        down → up
}
```

=) T.C → O(row * col)

Date ______________ Page ________

* Spiral Form:



Top = 0
Right = 5
Bottom = 5
left = 0

=> We will have a 2D Array & we have to print the elements in the spiral form.

=> First, we will print Top then Right then Bottom then Left.

```
while( top <= Bottom && Left <= Right){
for( int j = left ; j <= Right ; j++ ){
        cout<< matrix [top][j];
        top++;
}
for( int i = top ; i <= bottom ; i++ ){
Edge Cases      cout<< matrix [i][Right ];
        Right--;
 ↓      }
if(top <= Bottom){
 ↘ for( int j = Right ; j >= Left ; j-- ){
        cout << matrix [ bottom ] [j ];
if( left<= Right){  Bottom --; }
 ↘ for( int i = Bottom ; i >= Top ; i-- ){
        cout<< matrix [i][ left ];
        Left++; }
}
```

T.C. → O(row * col).

* Transpose Matrix:

$$\begin{array}{c|c|c|c|c} & 0 & 1 & 2 & 3 \\ \hline 0 & {}_{00}1 & {}_{01}2 & {}_{02}3 & {}_{03}4 \\ \hline 1 & {}_{10}5 & {}_{11}6 & {}_{12}7 & {}_{13}8 \\ \hline 2 & {}_{20}9 & {}_{21}10 & {}_{22}11 & {}_{23}12 \\ \hline 3 & 13_{30} & 14_{31} & 15_{32} & {}_{33}16 \end{array} \Rightarrow \begin{array}{c|c|c|c|c} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 5 & 9 & 13 \\ \hline 1 & 2 & 6 & 10 & 14 \\ \hline 2 & 3 & 7 & 11 & 15 \\ \hline 3 & 4 & 8 & 12 & 16 \end{array}$$

Code

T.C → $O(n^2)$
S.C → $O(n^2)$

```
for( i=0; i<n; i++){
    for (j=0; j<n; j++){
        arr[j][i] = matrix [i][j];
    }
}
```

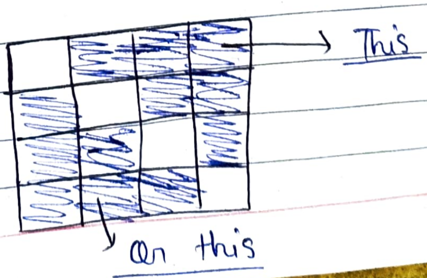
⇒ Here, we copy the i^{th} & j^{th} element at the j^{th} & i^{th} of the new array.

⇒ To solve this question in O(1) → S.C.

⇒ We have to swap matrix [i][j] with matrix [j][i].

⇒ But, if you have traverse all the array then it comes to their original array.

⇒ So we have to only traverse the array ~~in the~~ upto lower or upper triangle.



Code

```
for( int i= 0; i<row-1; i++)
      for( int j= i+1; j<col; j++){
            swap(matrix(i][j], matrix[j][i]);
      }
```