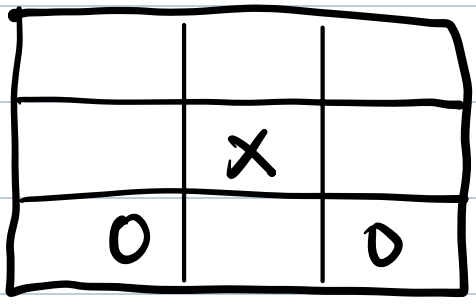
2D Arrays

Hello Everyone

Today's Content

1) Introduction to 2D Arrays
2) Print the top row of a matrix
3) Print the leftmost column of a matrix
4) Print matrix row by row
5) Print matrix column by column
6) Sum of Matrix
7) Waveform printing
8) Row wise sum
9) Column wise max

2D Arrays:



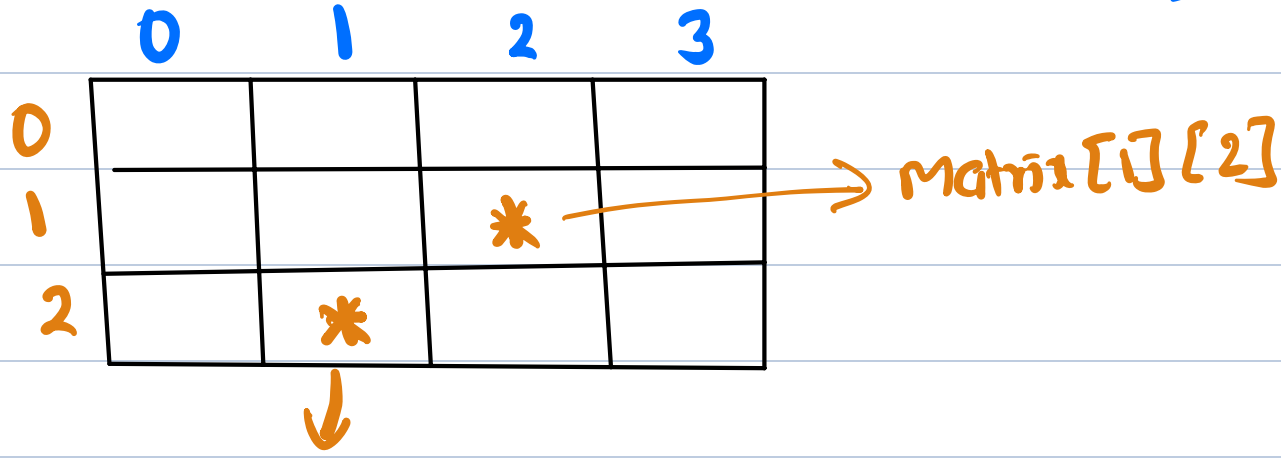
Eg. Chess, Theater seats, bus seats etc.

Syntax:

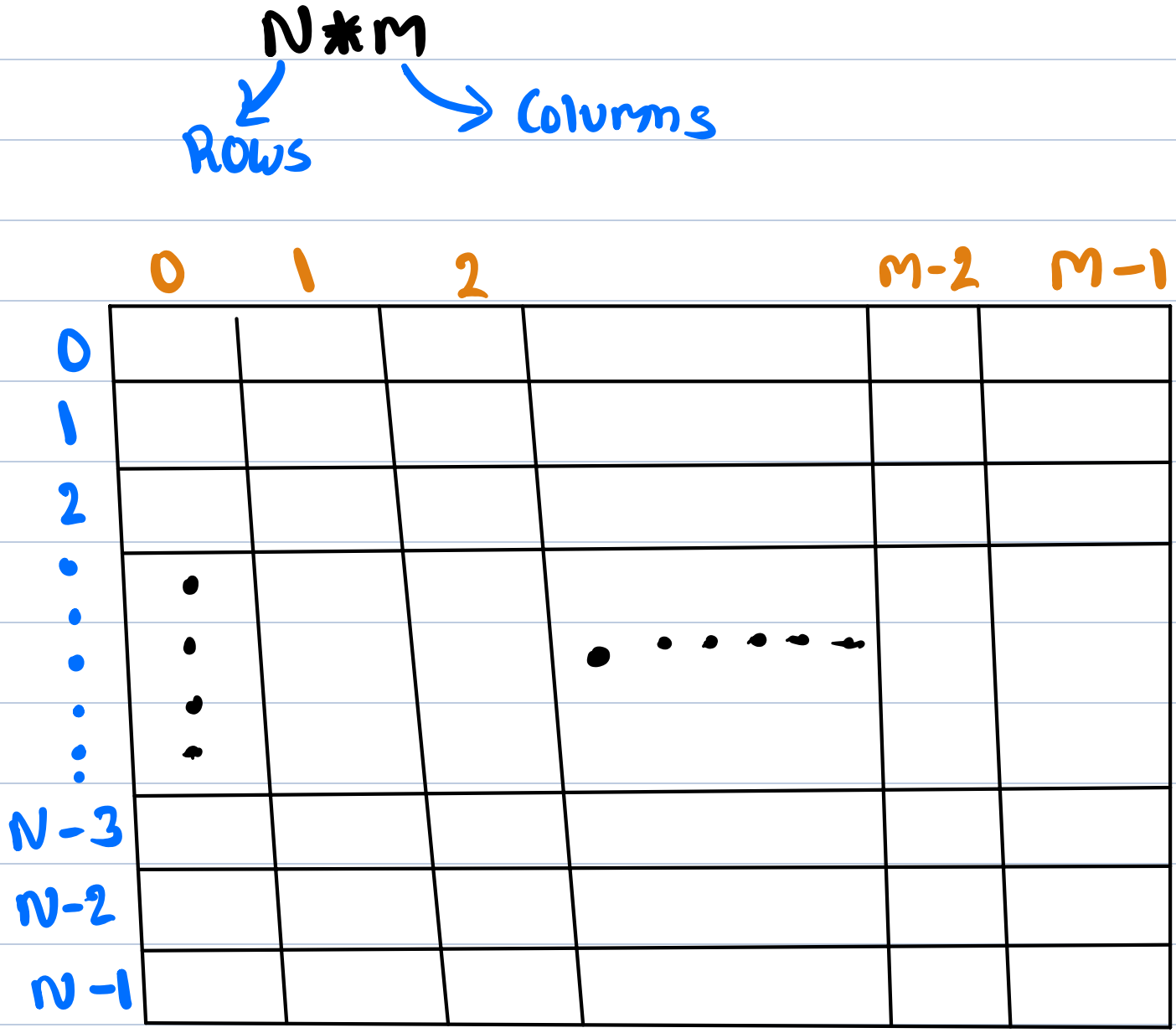
```
datatype[][] arrayName = new datatype
                         [row][col];
```

rows = 3 col = 4

```
int[][] matrix = new int[3][4];
```




matrix[2][1]
row index
col index


N*M
Rows
Columns
0
1
2
M-2
M-1
0
1
2
N-3
N-2
N-1

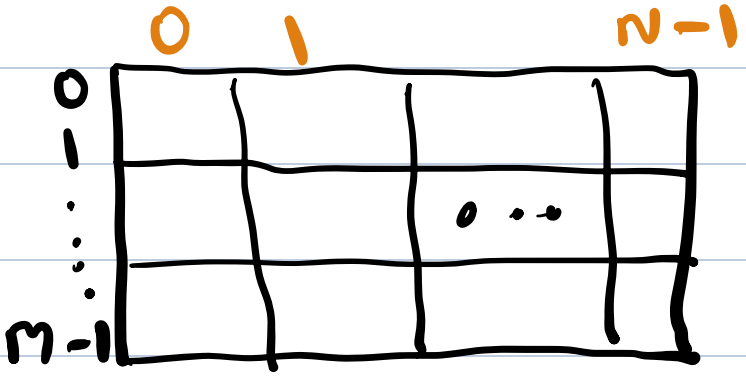
Quiz 1: Matrix with 5 columns and 7 rows.

```
int[][] matrix = new int[7][5];
```

Quiz 2: Top left corner.

[0][0]

Quiz 3: Bottom right corner.



[M-1] [N-1]

Q.1) Given a Matrix of size n*m, print it's top view.

	0	1	2
0	3	5	7
1	9	8	11
2	15	1	0
3	-2	8	9

Rows = 4
Cols = 3

Output: 3 5 7 {00 01 02}

Matrix ← 2D Array name.

```
//row    N = matrix.length;      4
// col   M = matrix[0].length;   3
```

```
for(int col=0; col<m; col++)
{
    S.O.P(matrix[0][col] + " ");
}
```

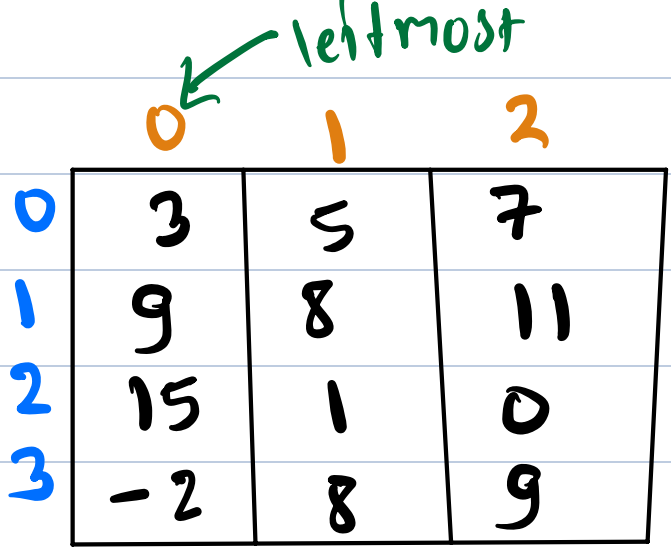
Col

0 X X 3

Output:

3 5 7

Q.2) Given a matrix of size N*M print it's leftmost col of matrix.



Output: 3 9 15 -2

00 10 20 30

```
for(int row=0; row<n; row++)
{
    S.O.P(matrix[row][0]+" ");
}
```

row

0 1 2 3 4

output:

3 9 15 -2

Q.3) Given a matrix N*M, print matrix row by row.

	0	1	2
0	3	5	7
1	9	8	11
2	15	1	0
3	-2	8	9

ROW = 4

COL = 3

Output:

3 (0,0)	5 (0,1)	7 (0,2)
9 (1,0)	8 (1,1)	11 (1,2)
15 (2,0)	1 (2,1)	0 (2,2)
-2 (3,0)	8 (3,1)	9 (3,2)

N=4 M=3

```
for(int row=0; row<N; row++)
{
    for(int col=0; col<M; col++)
    {
        S.O.P(matrix[row][col]+" ");
    }
    S.O.Println();
}
```

row	col
0	0 1 2 3
1	0 1 2 3
2	0 1 2 3
3	0 1 2 3

Output

```
3  5  7
9  8  11
15 1  0
-2 8  9
```

4	0	1	2
0	3	5	7
1	9	8	11
2	15	1	0
3	-2	8	9

Q.4) Given a matrix n*m, print column by column.

	0	1	2
0	3	5	7
1	9	8	11
2	15	1	0
3	-2	8	9

ROW = 4
COL = 3

Output:

3 (0,0) 9 (1,0) 15 (2,0) -2 (3,0)
5 (0,1) 8 (1,1) 1 (2,1) 8 (3,1)
7 (0,2) 11 (1,2) 0 (2,2) 9 (3,2)

```
for(int col=0; col<m; col++)        // m = 3
{
    for(int row=0; row<n; row++)    // n = 4
    {
        S.O.P.(matrix[row][col] + " ");
    }
    S.O.Println();
}
```

Col=0 row=0 1 2 3 4

Col=1 row=0 1 2 3 4

Col=2 row=0 1 2 3 4

Col=3

	0	1	2
0	3	5	7
1	9	8	11
2	15	1	0
3	-2	8	9

Output:

3 9 15 -2

5 8 1 8

7 11 0 9

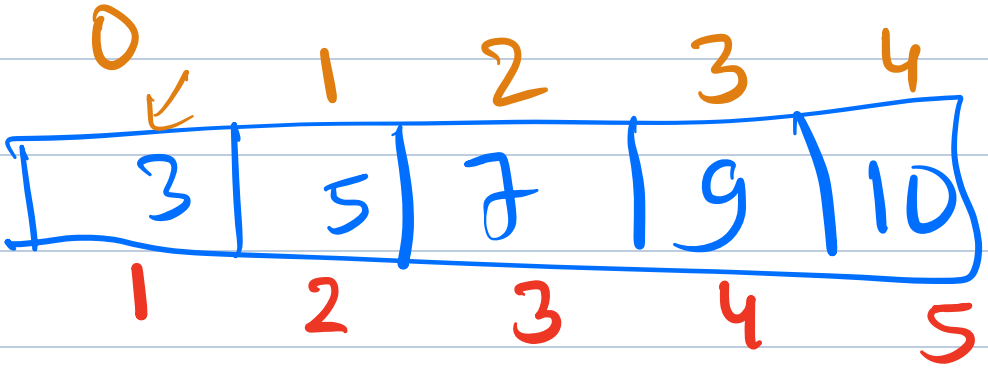
Break: 10.15 Pm

Doubt session

10	15	20
22	33	45
13	21	40

N - matrix.length → 3

M - matrix[3].length




```
for(int i=1; i<=N; i++)
{
    S.O.P(arr[i-1]);
```

}