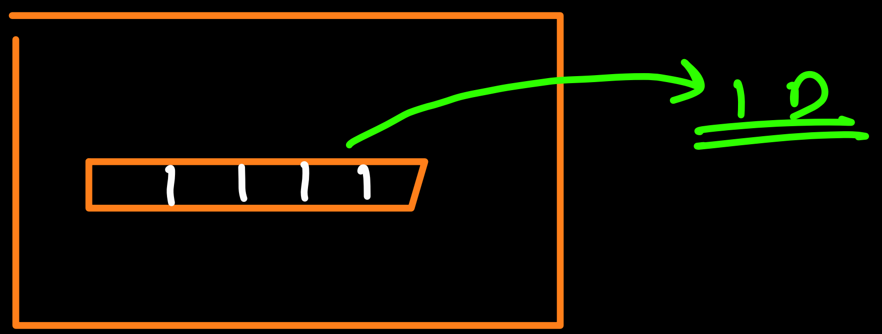
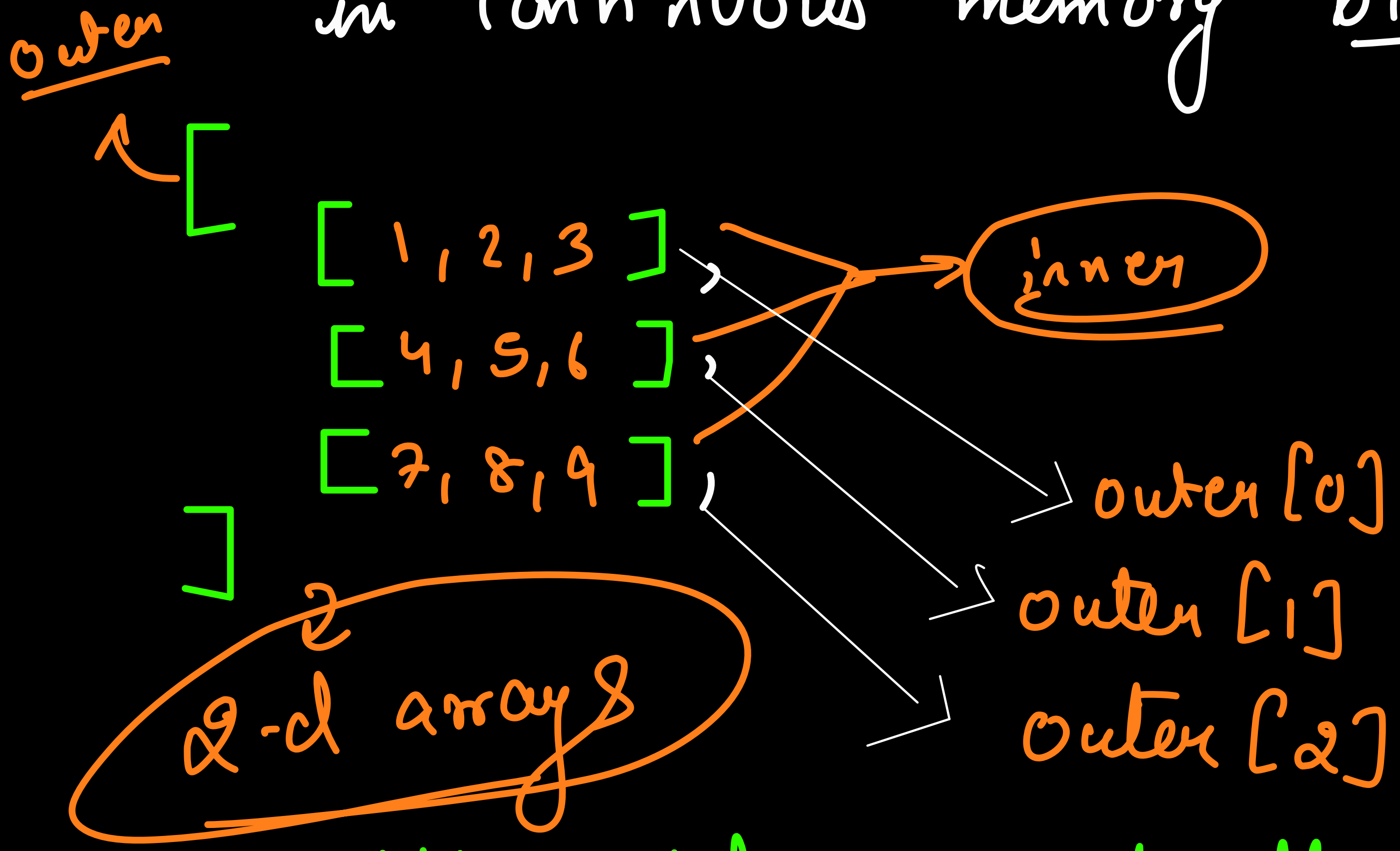
2D arrays



[2, 3, 5, -1, 6] →

array is a data structure, that stores data in continuous memory blocks



1	2	3
4	5	6
7	8	9

3x3

Bind multiple 1d arrays together inside another array to form 2d arrays a.length

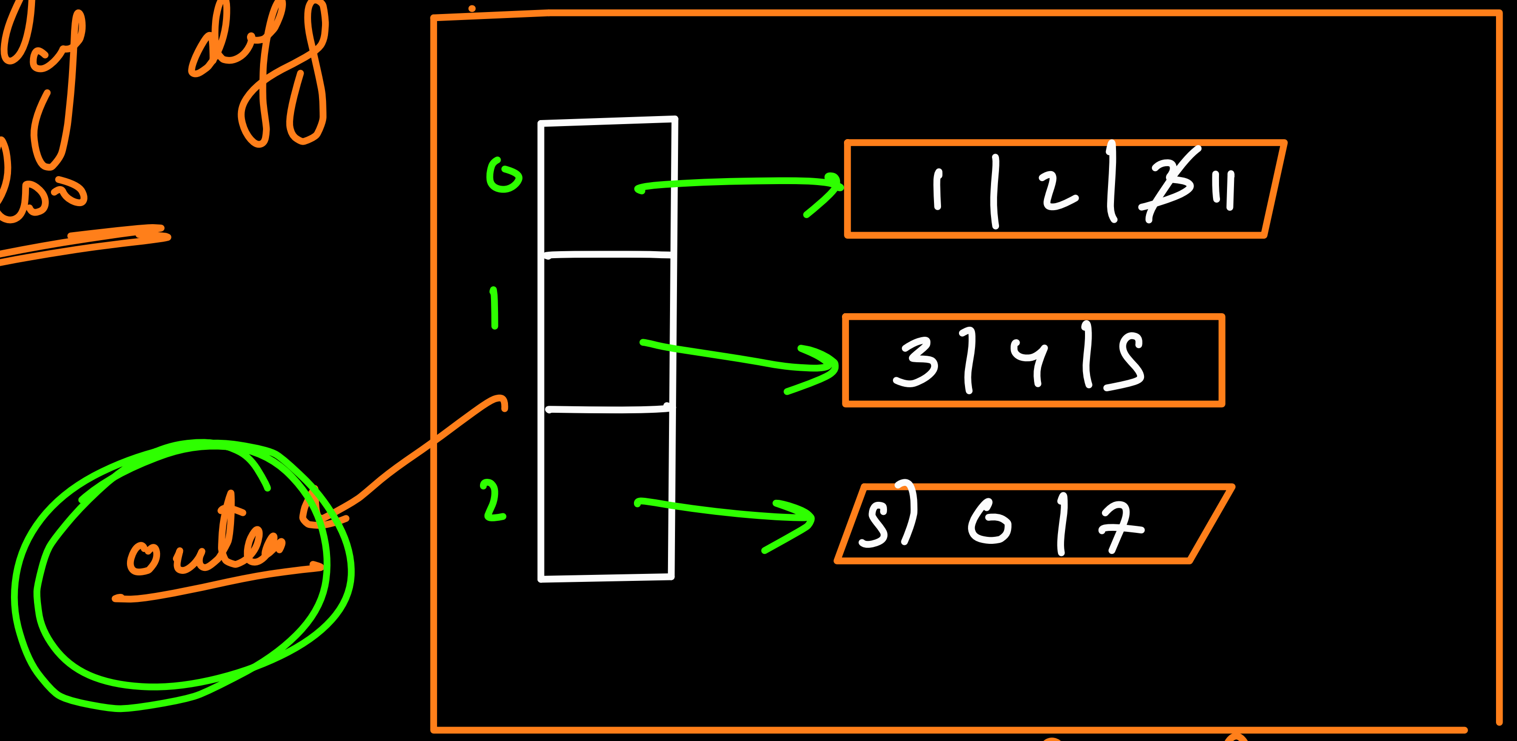
let arr = [[1,2],[3,4],[5,6]];

0 1 2

let arr1 = [[1], [1,2], [3,2 3], [4]]; → outer

inner arrays can be of diff length also

2



outer[0][2] = 11

outer[0] → [1,2,3]

outer[0][1] → 2

this is a 1d array

on the 1d array we try to access index 1

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

4x4

→ Print all elements of this 2d grid row by row

1 2 3 4 5 6 7 8 9 10 11 12

13 14 15 16

we know, grid [i] [j] → gives you element at ith row and jth col.

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

4x4

Repeatedly do some task

for every row.

row

```
str = "";
for (let i=0; i<4; i++) {
    // on each row we have a 1d array
    for (let j=0; j<4; j++) {
        str += grid[i][j] + " ";
    }
}
console.log(str);
```

Q → Given a 2d array, print it in a column wave **Form.** (no. of rows & cols can be diff)

Ex

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

grid

grid[i][j]

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

	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

grid

m x n
5 4

what if I just wanted to print the grid col by col.

(forget about wave form).

```
                    3
for ( let col = 0 ; col < n ;  col++ ) {
  ↳ for ( let row = 0 ; row < m ; row++ ) {
          str += grid[row][col] + " ";
    }
}
```

col = 0   row = ~~0~~ ~~1~~ 2

1 5 9 . . . .

	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

grid

m x n

5 b 4

3

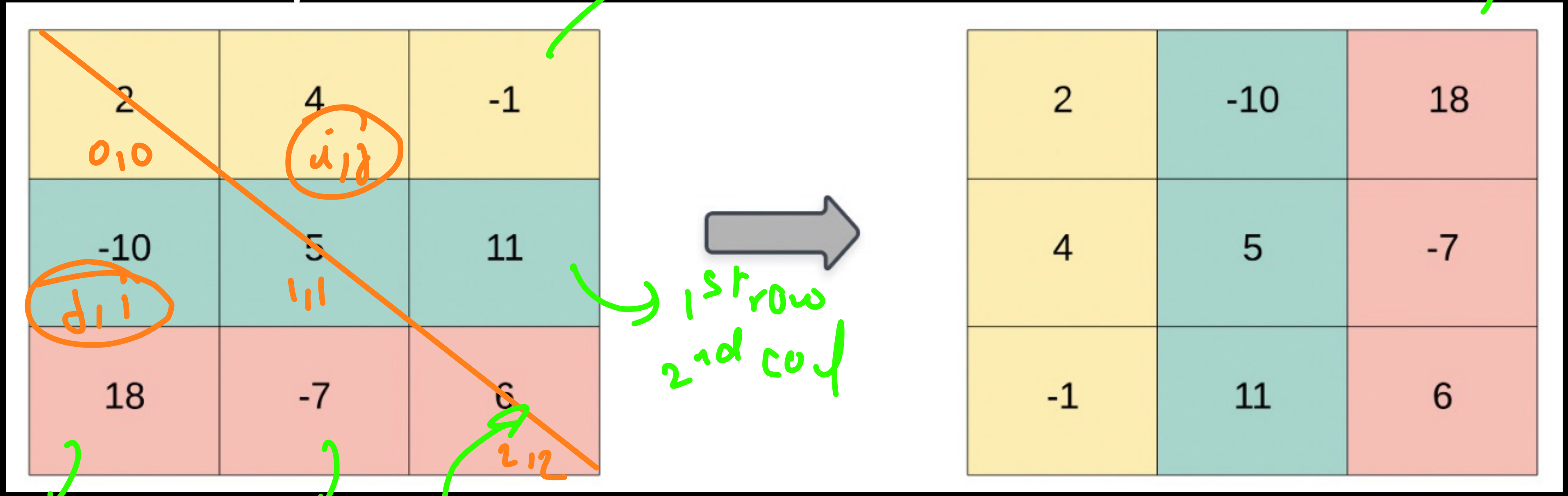
So if we want to print Some cols from down to up we need to reverse the inner loop.

Even cols → up to down

odd cols → down to up.

to solve it using another matrix

for (let row =0;row<n;row++)
for(let col =row; col<n; col++)
swap



	0	1	2
0	2 0,0	4 i,j	-1
1	-10 j,i	5 1,1	11
2	18	-7	6 2,2

2	-10	18
4	5	-7
-1	11	6

0th row 2nd col

1st row 2nd col

2nd row 0th col

2nd row 1'st col

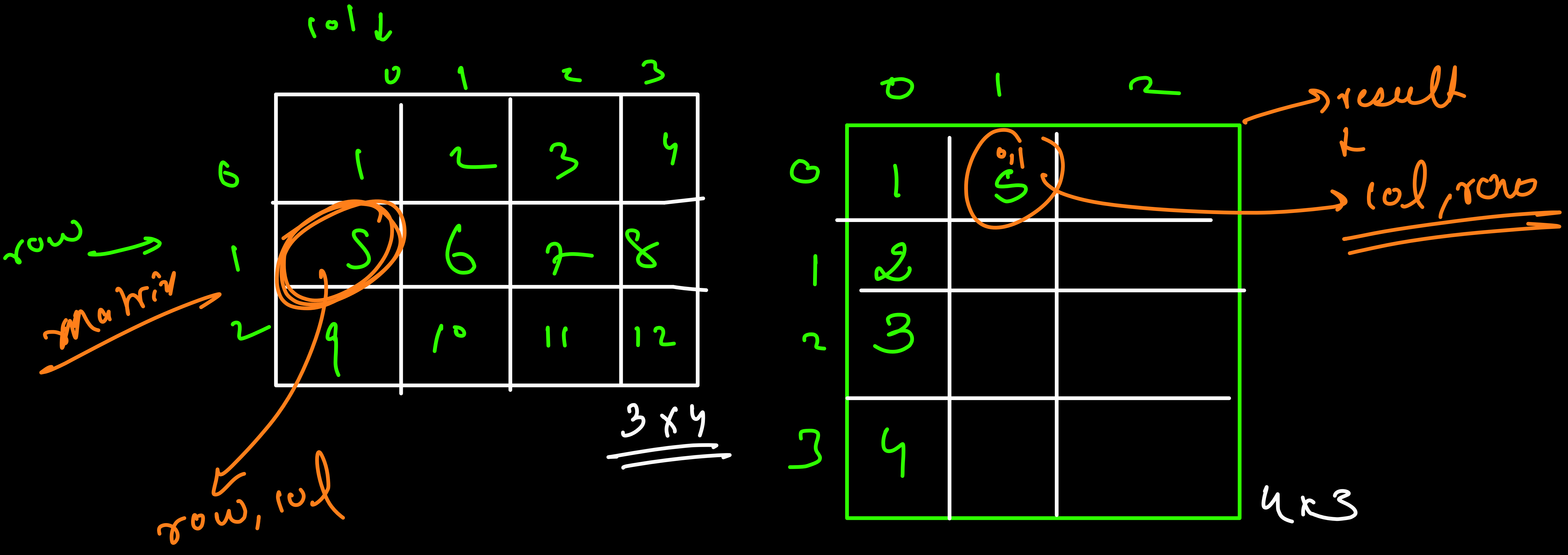
main diagonal

2	-10	18
4	5	-7
-1	11	6

arr[i][j] ←→ arr[j][i]

main diagonal → row no == col no

In JS → arrays are 0-based index



$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 2 & 3 & 4 \\ 1 & 5 & 6 & 7 & 8 \\ 2 & 9 & 10 & 11 & 12 \end{array}$$

$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 1 & 5 & \\ 1 & 2 & & \\ 2 & 3 & & \\ 3 & 4 & & \end{array}$$

Q => Given 2, 2-d arrays where the first 2d array has a dimension (m, n) and the second 2D array has a dimension (n, k). Multiply both the 2d arrays

$$a \rightarrow \underset{m \times n}{\begin{bmatrix} 1, & 1 \\ 2, & 2 \\ 5, & 3 \end{bmatrix}}_{3 \times 2} \qquad b \; \underset{n \times k}{\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix}}_{2 \times 3}$$

$$\Downarrow$$

$$\begin{bmatrix} 3, & 3, & 3 \\ 6, & 6, & 6 \\ 9, & 9, & 9 \end{bmatrix}_{3 \times 3} \quad ✓$$

$$a \rightarrow \underset{m \times n}{\begin{bmatrix} 1 , 1 \\ 2 , 2 \\ 5 , 3 \end{bmatrix}} 3 \times 2 \qquad b \underset{n \times k}{\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix}} 2 \times 3$$

we can oly multiply if no. of col of first matrix equal no. of ro no of 2nd

$$\underset{\Downarrow}{(m \times n)} \times (n \times k)$$

$$\Downarrow$$

$$\underline{\underline{(m \times k)}}$$

C →

3	3	3
6	6	.
-	.	-

$\underline{\underline{3 \times 3}}$

$C[0][0] = 1\times1 + 1\times2 \rightarrow 3$

$C[0][1] = 1\times1 + 1\times2 \rightarrow 3$

$C[0][2] = 1\times1 + 1\times2 = 3$

$C[1][0] = 2\times1 + 2\times2 = 6$

$C[1][1] = 2\times1 + 2\times2 = 6$

In any row of a we have n elements $\boxed{i=1}$

$$a \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 2 & 2 \\ 2 & 3 & 3 \end{array} \quad 3 \times 2$$

m×n

~~j=1~~ In any col of b we have n elements

$$b \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \end{array} \quad 2 \times 3$$

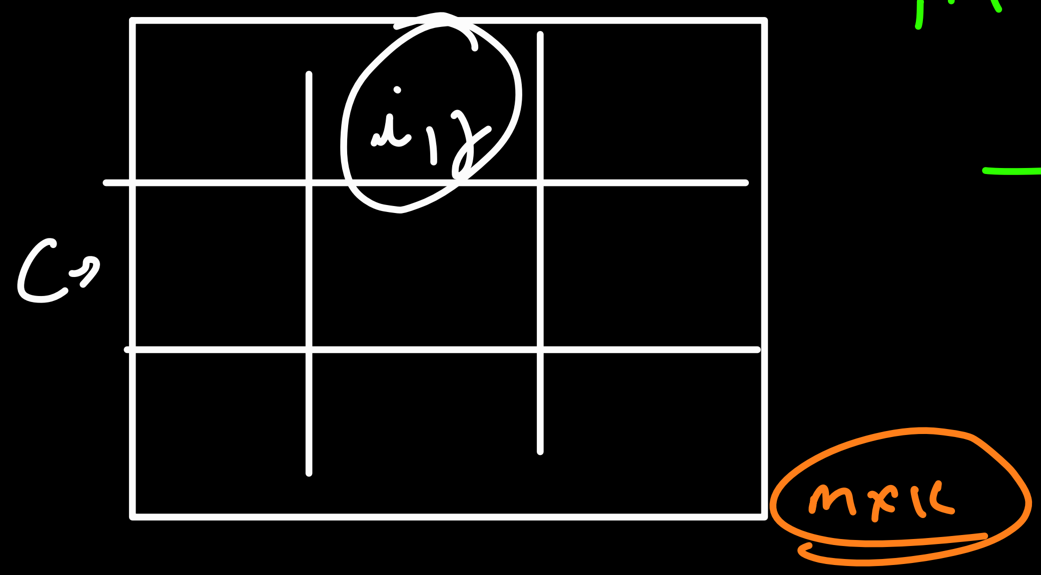
n×k

```
for(x=0; x<2; x++)
    b[x][1]
```

we know that final result matrix C will be of m×k dimensions.

→ to get C[i][j] we need to multiply ith row of a with jth col of b.

→ go to every element of ith row of a and multiply every element with elements from jth col of b.



```
// create 2d array of m x k dimension
   ↓

for (let i=0; i<m; i++) {
    for (let j=0; j<k; j++) {
        // we are at some cell i,j
        for (let x = 0; x<n; x++) {
            c[i][j] += a[i][x] * b[x][j];
        }
    }
}
```

To iterate on all cells of C

```
for (let i=0; i<m; i++) {
    for (let j=0 ;j<k; j++){
        // we are at some cell i,j
        for(let x = 0; x<n ; x++){
        c[i][j] += a[i][x] * b[x][j];
                     2  1      1  1
        }
    }
}
```

i = 2, j = 1

x = 1

3 + 6

$$a \; \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 2 & 2 \\ 2 & 3 & 3 \end{array} \quad 3 \times 2$$

m x n

$$b \; \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \end{array} \quad 2 \times 3$$

n x k

	0	1	2
0	0	0	3
1	0	0	0
2	0	9	0

0,2

2,1

C