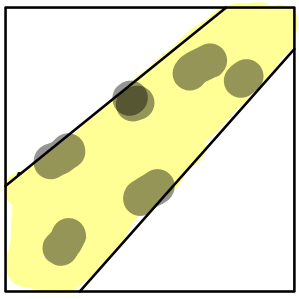
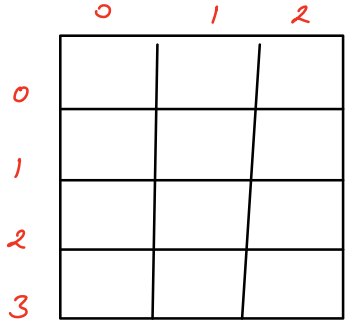


Giraffe Neck

2D Arrays

- Chess
- Images (2D arrays of pixels)
- Theater seats
- Classrooms
- Bingo



Rows => 4

Colums => 3

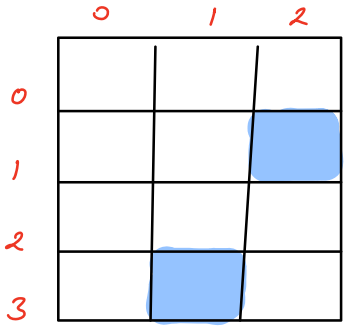
Declaration

```
int mat[][];
```

Initialise

```
mat = new int[rows][columns];
      (n)    (m)
```

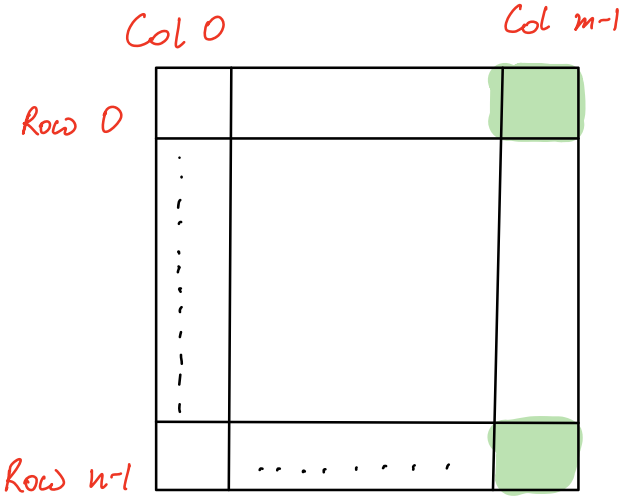
```
int mat[][] = new int [r][c];
int [][] mat = new int [r][c];
```




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

mat[3][1]
mat[1][2]

Total elements = 4 x 3 = 12 cells



Rows => N
Columns => M

mat[0][m-1]
mat[n-1][m-1]

// Q: Write the code to print elements in 1st row of matrix mat

Rows => N
Cols => M

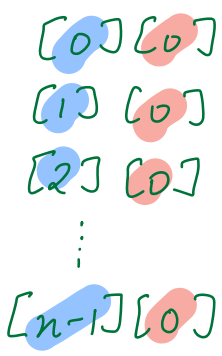
[0][0] [0][1] [0][2] .... [0][m-1]

```
for(int col=0; col<M; col++){
    SOP(mat[0][col]);
}
```

// Q: Write the code to print elements in 1st col of matrix mat

Rows => N

Cols => M




```
for(int row=0; row<N; row++){
    SOP(mat[row][0]);
}
```

// Q: Print matrix mat in row-row fashion

1	2	3
4	5	6
7	8	9

1 2 3 ↵
4 5 6 ↵
7 8 9 ↵

Row 0 => [0][0] [0][1] .... [0][m-1]
Row 1 => [1][0] [1][1] .... [1][m-1]
Row 2 => [2][0] [2][1] .... [2][m-1]
⋮
Row n-1 => [n-1][0] [n-1][1] .... [n-1][m-1]

```
for(int row=0; row<N; row++){

    for(int col=0; col<M; col++){
            SOP(mat[row][col]);
    }

    SOPln()
}
```

// Q: Print matrix mat in col-col fashion

1	2	3
4	5	6
7	8	9

1 4 7 ↵
2 5 8 ↵
3 6 9 ↵

Col 0 ⇒ [0][0] [1][0] [2][0] ... [n-1][0]
Col 1 ⇒ [0][1] [1][1] [2][1] ... [n-1][1]
⋮
Col m-1 ⇒ [0][m-1] [1][m-1] [2][m-1] ... [n-1][m-1]

```
for(int col=0; col<M; col++){

    for(int row=0; row<N; row++){
```

```
            SOP( mat[row][col]);
        }
        SOPln();
    }
```

https://www.interviewbit.com/snippet/fbbbf3a71c92f494b310/