

- Flood lights
- Chess board
- Movie hall
- TV

01	01	01
01	01	01
01	01	01


```
int A[][] = new int [ rows ][ cols ];
int B[] = new int [ N ];
```

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

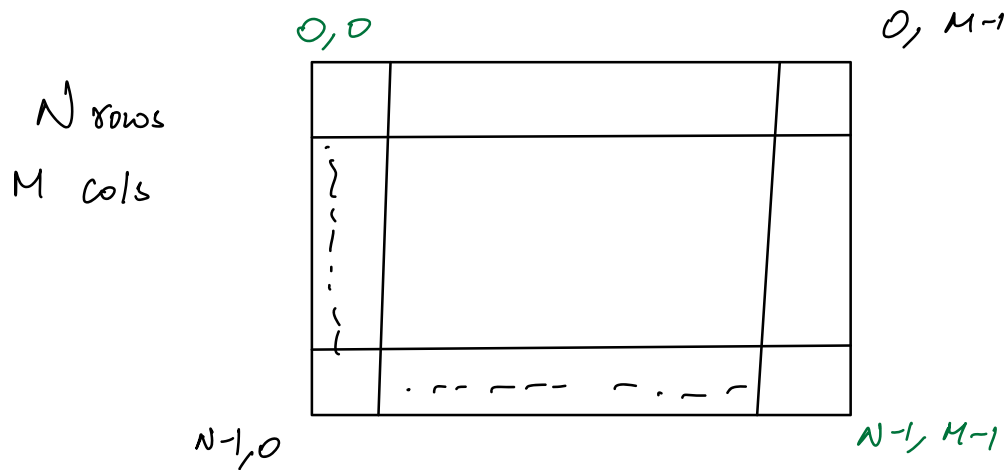
		Cols 0 1 2 3			
Rows 0	0				
	1				
	2				

mat[1][2]

mat[2][1]

Total cells $\Rightarrow 12$

$\Rightarrow \text{Rows} \times \text{Cols}$



Q: Print 1st row of a matrix (N, M)

$[0][0] \quad [0][1] \quad [0][2] \quad \dots \quad [0][M-1]$

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

Q: Print 1st col of a matrix (N, M)

```
[0][0]    for (int row = 0; row < N; row++) {  
[1][0]        SOP(mat[row][0]);  
[2][0]    }  
  ;  
[N-1][0]
```

Q: Print a matrix row-col

$A(N, M)$

1st row \leftarrow
2nd row \leftarrow
3rd row \leftarrow
 \vdots

last row \leftarrow

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

SOP(nC);

}

Q: Print a matrix col-col

1st col \leftarrow
2nd col \leftarrow
 \vdots

last col

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

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

```
}
```

```
sopln();
```

```
y
```

Break 10pm

Resuming in 2 mins