⇒ 2D array

1D arr

0	1	2	3	4	5	6	7

arr[3]

size = 8

2D array

rows

	0	1	2	3	4	5
0						
1						
2						
3						
4						

col.

size = 5 x 6

arr[2][3]

row no. col. no.

declare

1D :- `int[] arr = new int[n];`

2D :- `int[][] arr = new int[5][6]`

How to access indices

1D :- `arr[i]` (single loop)

2D :- `arr[i][j]` (nested loops)

Traversal :-

arr

cols

rows

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

i = row no.
j = col. no.

i = 0, j = 0 (1)	i = 1, j = 0 (5)	i = 2, j = 0 (9)	i = 3, j = 0 (13)
j = 1 (2)	j = 1 (6)	j = 1 (10)	j = 1 (14)
j = 2 (3)	j = 2 (7)	j = 2 (11)	j = 2 (15)
j = 3 (4)	j = 3 (8)	j = 3 (12)	j = 3 (16)

length of row :- arr.length , length of col. :- arr[0].length

psudo Code

arr ; rows = m
cols = n

rows

```
for (int i=0; i< arr.length ; i++) {
    for (int j=0; j< arr[0].length ; j++) {
        print( arr[i][j] + " ");
    }
    sysoln();
}
```

cols

T.C = O(m * n) linear

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int m = scn.nextInt();    // row size
    int n = scn.nextInt();    // col size
    int[][] arr = new int[m][n];
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }

    printMatrix(arr);
}

public static void printMatrix(int[][] arr) {
    int row = arr.length;
    int col = arr[0].length;
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            System.out.print( arr[i][j] + " ");
        }
        System.out.println();
    }
}
```

rows

cols

Print Alternate Row

		0	1	2	3
row 1st	0	1	2	3	4
row 2nd	1	5	6	7	8
row 3rd	2	9	10	11	12
row 4th	3	13	14	15	16
row 5th	4	17	18	19	20

$i = 0, j = 0 \rightarrow 3$

$i = 2, j = 0 \rightarrow 3$

$i = 4, j = 0 \rightarrow 3$

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int m = scn.nextInt();   // row size
    int n = scn.nextInt();   // col size
    int[][] arr = new int[m][n];
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }

    printMatrix(arr);
}

public static void printMatrix(int[][] arr) {
    int row = arr.length;
    int col = arr[0].length;
    for (int i = 0; i < row; i += 2) { // rows
        for (int j = 0; j < col; j++) { // cols
            System.out.print( arr[i][j] + " ");
        }
        System.out.println();
    }
}
```

Print Upper triangular matrix 1

	0	1	2	3	4	5	6
0							
1	0						
2	0	0					
3	0	0	0				
4	0	0	0	0			
5	0	0	0	0	0		
6	0	0	0	0	0	0	

rows cols

$i = 0,\ j = 0 \rightarrow 6$

$i = 1,\ j = 1 \rightarrow 6$

$i = 2,\ j = 2 \rightarrow 6$

$i = 3,\ j = 3 \rightarrow 6$

$i = 4,\ j = 4 \rightarrow 6$

$i = 5,\ j = 5 \rightarrow 6$

$i = 6,\ j = 6 \rightarrow 6$

```
if (j >= i)
        arr[i][j]
else  0
```

```java
public static void printUpperTriangularMatrix(int[][] arr) {
    int row = arr.length;
    int col = arr[0].length;
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            if ( j >= i ) {
                System.out.print(arr[i][j] + " ");
            } else {
                System.out.print("0 ");
            }
        }
        System.out.println();
    }
}
```

row = 4
col = 4

arr

	0	1	2	3
0	1	2	3	4
1	0	6	7	8
2	0	0	11	12
3	0	0	0	16

i=0, j=0 true
j=1 true
j=2 true
j=3 true

i=1, j=0 ✗
j=1 ✓
j=2 ✓
j=3 ✓

i=2, j=0 ✗
j=1 ✗
j=2 ✓
j=3 ✓

i=3, j=0 ✗
j=1 ✗
j=2 ✗
j=3 ✓

Print the matrix left-diagonal wise

arr

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

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

Trick

initialise condition inc/dec

for(_____ ; _____ ; _____)

arr

j

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

i j

(0, 2)

1, 1

for(int i=0, j=2 ; j>=0 ; i++, j--)

for (i = 0, j = 0,1,2,3 ; j>=0 ; i++, j--)

arr

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

j>=0

i j
0, 1
1, 0

i j
0, 3
1, 2
2, 1
3, 0

psudo code

```
for ( int gap = 0 ; gap < arr[0].len ; gap++) {
    for ( int i=0, j= gap ; j>=0 ; i++, j--){
        Syso( arr[i][j] + " ");
    }
}
```

```java
public static void leftDiagonal(int[][] arr, int n) {
    for (int gap = 0; gap < n; gap++) {
        for (int i = 0, j = gap; j >= 0; i++, j--) {
            System.out.print(arr[i][j] + " ");
        }
    }
}
```

arr

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

1, 2, 5, 3, 6, 9, 4, 7, 10, 13

gap = 0, i = 0, j = 0
i = 1, j = -1 ✗

gap = 1, i = 0, j = 1
i = 1, j = 0
i = 2, j = -1 ✗

gap = 2, i = 0, j = 2
i = 1, j = 1
i = 2, j = 0
i = 3, j = -1 ✗

gap = 3, i = 0, j = 3
i = 1, j = 2
i = 2, j = 1
i = 3, j = 0
✗ i = 4, j = -1