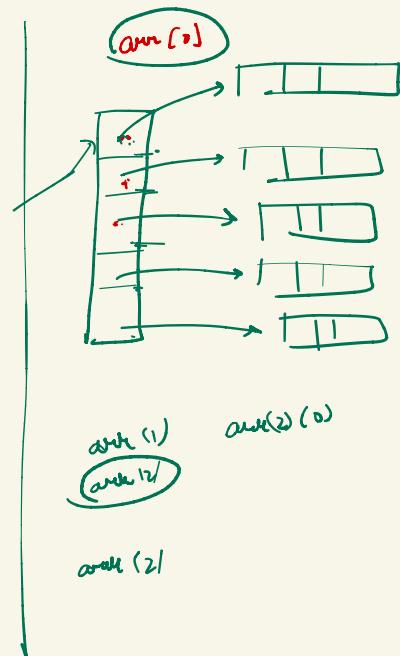



$\text{int } arr = \text{new int}[5]$ → 0th index address
 $arr[0]$
 2d array → array of arrays
array of integers
array of pointers
 $arr[0]$

$\text{int } arr = \text{new int}[5][3]$
 $\quad\quad\quad$ ↓
 $\quad\quad\quad$ number of rows
 $\quad\quad\quad$ number of columns
 $arr[2][0], arr[2][1], arr[2][2]$



$arr[i][j]$

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

row-major order
last row

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

Ques Create 2d array, take input, find sum of whole 2d array.

```

public static void main(String[] args) {
    Scanner scn=new Scanner(System.in);

    System.out.println("Enter the number of rows");
    int n=scn.nextInt();

    System.out.println("Enter the number of columns");
    int m=scn.nextInt();

    int[][] arr=new int[n][m];

    System.out.println("Enter "+n*m+" numbers for your matrix");

    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            arr[i][j]=scn.nextInt();
        }
    }

    System.out.println("Your matrix is this neo");

    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            System.out.print(arr[i][j]+" ");
        }
        System.out.println();
    }
}
    
```

$arr[2][3]$

\Downarrow

12

```

int sum=0;

for(int i=0; i<n; i++){
    for(int j=0; j<m; j++){
        int ele=arr[i][j];

        sum=sum+ele;
    }
}

System.out.println("The sum of whole array is "+sum);

```

~~1) Search~~ → create a search function.

2) Compare

3) now sum equal

4) col sum equal.

$n=3$	$m=4$		
0 1 2	1 2 3	2 3 4	3 4 5
5 6 9	6 7 9	9 9 9	9 9 9

$\text{tar} = 10$

5) Two arrays are equal if they have equal number of rows, columns and every all element is equal.

1 2 3
4 5 6
7 8 9

1 2 3
4 5 6
7 8 10

$n_1=3$
 $m_1=3$
 $n_2=3$
 $m_2=3$

```

// returns true if arr1,arr2 is same else false
public static boolean isSame(int[][] arr1, int[][] arr2){
    int n1=arr1.length;
    int m1=arr1[0].length;

    int n2=arr2.length;
    int m2=arr2[0].length;

    if(n1!=n2 || m1!=m2){
        return false;
    }

    for(int i=0; i<n1; i++){
        for(int j=0; j<m1; j++){
            if(arr1[i][j]!=arr2[i][j]){
                return false;
            }
        }
    }

    return true;
}

```

$arr1(i)(j)$
 $arr1(j)(r)$
 $arr2(o)(r)$
 $i = 0$
 $j = 2$

Ques Check if sum of every column is equal.

Ques Check if sum of every row is equal

```

public static boolean is_row_sum_equal(int[][] arr){
    int n=arr.length;
    int m=arr[0].length;

    int first_row_sum=0;

    int row=0;
    for(int col=0; col<m; col++){
        first_row_sum=first_row_sum+arr[row][col];
    }

    for(int i=1; i<n; i++){
        int row_sum=0;

        for(int j=0; j<m; j++){
            row_sum=row_sum+arr[i][j];
        }

        if(first_row_sum!=row_sum){
            return false;
        }
    }

    return true;
}

```

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

$n=4$
 $m=3$
 $first_row_sum \rightarrow 1610$

else

0, 0	1, 0	2, 0
0, 1	1, 1	2, 1
0, 2	1, 2	2, 2

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

arr[row][col]

```

public static void columnWave(int[][] arr, int n, int m){
    for(int col=0; col<m; col++){
        if(col%2==0){
            for(int row=0; row<n; row++){
                System.out.print(arr[row][col] + " ");
            }
        } else {
            for(int row=n-1; row>=0; row--){
                System.out.print(arr[row][col] + " ");
            }
        }
        System.out.println();
    }
}

```

$$\text{col} = \rho + z^3$$

$$\text{row} = x \cdot z^{10}$$

Identity matrix

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

1) square matrix

1	0	0
0	1	0
0	0	1

$i == j$
~~row == col~~

(0,0)
(1,1)
(2,2)
(3,3)

→ 1

Symmetric Matrix [transpose is equal to the matrix]

$$B = \begin{matrix} & 1 & 2 & 4 \\ 1 & & & \\ 2 & 5 & 11 & \\ & 11 & 19 & \end{matrix} ; \quad B^T = \begin{matrix} & 1 & 2 & 4 \\ 1 & & & \\ 2 & 5 & 11 & \\ 4 & 11 & 19 & \end{matrix}$$

(i < j)

$$\begin{matrix} & 0 & 1 & 2 & 3 & 0 & 1 & 4 & 7 \\ 0 & | & 2 & & & & & & \\ 1 & 4 & 5 & 6 & \Rightarrow & 1 & 2 & 5 & 8 \\ 2 & 7 & 8 & 9 & & 2 & 3 & 6 & 9 \end{matrix}$$

(i < j)

$$\begin{aligned} 1 &\rightarrow \cancel{(0,0)} \rightarrow (0,0) \\ 2 &\rightarrow \cancel{(0,1)} \rightarrow (1,1) \\ 3 &\rightarrow \cancel{(0,2)} \rightarrow (2,0) \\ 4 &\rightarrow \cancel{(1,0)} \rightarrow (0,1) \\ 5 &\rightarrow \cancel{(1,1)} \rightarrow (1,1) \\ 6 &\rightarrow \cancel{(1,2)} \rightarrow (2,1) \end{aligned}$$

$$\begin{aligned} 7 &\rightarrow \cancel{(2,0)} \rightarrow (1,2) \\ 8 &\rightarrow \cancel{(2,1)} \rightarrow (1,4) \\ 9 &\rightarrow \cancel{(2,2)} \rightarrow (2,2) \end{aligned}$$

$$\begin{array}{ccc} (0,1) & \longleftrightarrow & (1,0) \\ (0,2) & \longleftrightarrow & (2,0) \\ (1,2) & \longleftrightarrow & (2,1) \end{array}$$

0	1	2	3		0	1	2
0	1	2	3	4	1	2	
1	5	6	7	8	2	3	
2	9	10	11	12	3	4	

```

public static int[][] makeTranspose(int[][] arr, int n, int m){
    int[][] ans=new int[m][n];
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            ans[j][i]=arr[i][j];
        }
    }
    return ans;
}

```

$$\begin{array}{l} i=0 \\ j= \cancel{0} \cancel{1} \cancel{2} \end{array}$$

(4x3)

n=3							
0	1	2		0	1	2	3
0	1	2	3	0	1	2	3
1	4	5	6	1	5	6	7
2	7	8	9	2	9	10	11
	=	=		3	13	14	15
							16
							=
							25

```

public static int getDiagonalSum(int[][] arr, int n){
    int sum=0;
    for(int i=0; i<n; i++){
        for(int j=0; j<n; j++){
            if((i==j) || (i+j==n-1)){
                sum=sum+arr[i][j];
            }
        }
    }
    return sum;
}

```

$$\begin{array}{l} \text{sum=} \\ \cancel{1} \\ \cancel{4} \\ \cancel{9} \\ \cancel{16} \\ \cancel{25} \\ = \end{array}$$

0 1 2 3 4 5 6 7 8
 | 2 3 4 5 6 7 8 9

	0	1	2
0	0,0	0,1	0,2
1	1,0	1,1	1,2
2	2,0	2,1	2,2
9	6	1	8
10			11
12		13	14
15		16	17

0th row = 0
 1st row = 3
 2nd row = 6
 3rd row = 9
 4th row = 12

x th row, 0 th col
 $\Rightarrow 3x$
 x th row, 1st col
 $\Rightarrow 3x + 1$
 x th row, 2nd col
 $\Rightarrow 3x + 2$
 x th row, y th col
 $\Rightarrow 3x + y$

	0	1	2	3
0	0	1	2	3
1	4	5	1,2	7
2	8	9	10	11

(x, y)
 \downarrow
 $4x + y$

```
public static int[][] convert1Dto2D(int[] arr, int n, int rows, int cols){  

    int[][] ans=new int[rows][cols];  

    for(int i=0; i<rows; i++){  

        for(int j=0; j<cols; j++){  

            ans[i][j]=arr[i*cols+j];  

        }
    }
    return ans;
}
```

cols
 total number
 of cells

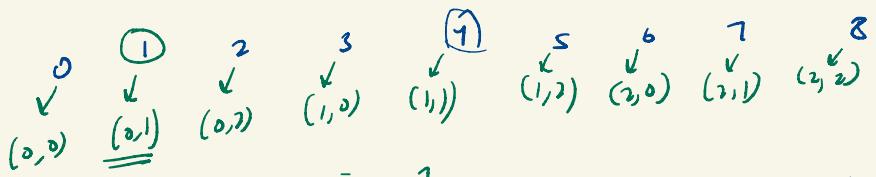
0th row
 $(0, 0) \Rightarrow 0$
 $(1, 0) = 4$
 $(2, 0) = 8$
 $(3, 0) = 12$
 x th row, 0th col
 $\Rightarrow 4x$

x th row, 1st col
 $\Rightarrow 4x + 1$
 $(x, y) \Rightarrow 4x + y$

$\text{arr}[x][y] = \text{element at } \boxed{x * \text{cols} + y}$

$$\underline{\underline{x * m + y}}$$

; $m = \text{number of columns}$



$$\text{idm} = x * m + y$$

$$\text{idm} = 7$$

$$x = 7 / 3 = 2$$

$$y = 7 \% 3 = 1$$

in terms of idm and m

$$x = \frac{\text{idm}}{\text{idm} / m}$$

$$y = \underline{\underline{\frac{\text{idm} \% m}{1}}}$$