int () arr → 0th index address

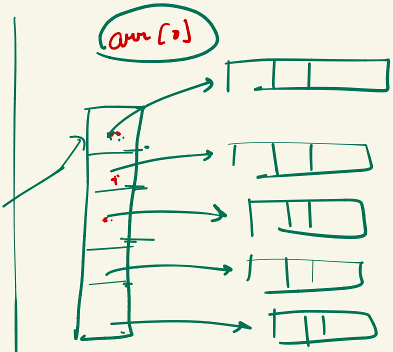
arr = new int [5]

2d array → array of arrays

array of integers

array of string

arr[0]



arr (1)

arr(2) (0)

arr (2)

arr (2)

int [][] arr = new int [5] [3]

number of rows

number of columns

arr [2][0], arr (2)(1), arr (2) (2)

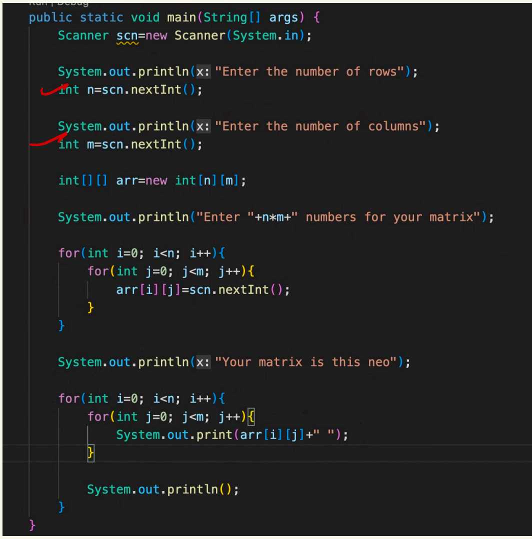
arr [i][j]

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

last row

reverse order

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



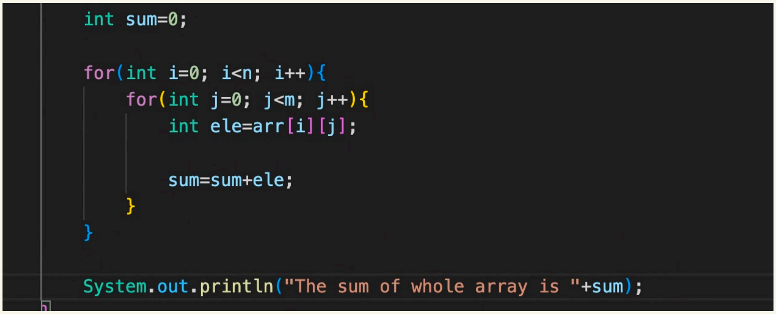
```
Run | Debug
public static void main(String[] args) {
    Scanner scn=new Scanner(System.in);

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

    System.out.println(x: "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(x: "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)

⇓

12

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



```
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);
```

~~#~~Search → create a search function.

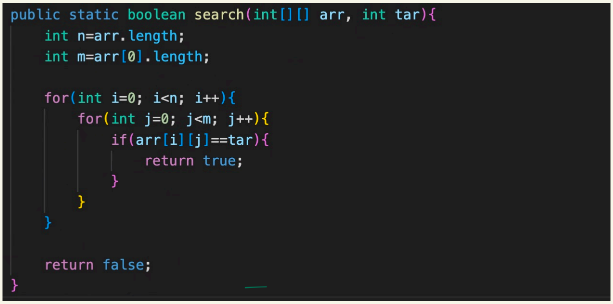
2) Compare

3) row sum equal

4) col sum equal.

$n=3, \quad m=4$

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & \underline{1} & 2 & \underline{3} & \underline{4} \\ 1 & 5 & 6 & 4 & 9 \\ 2 & 9 & 9 & 9 & 9 \end{array}$$



```
public static boolean search(int[][] arr, int tar){
    int n=arr.length;
    int m=arr[0].length;

    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            if(arr[i][j]==tar){
                return true;
            }
        }
    }

    return false;
}
```

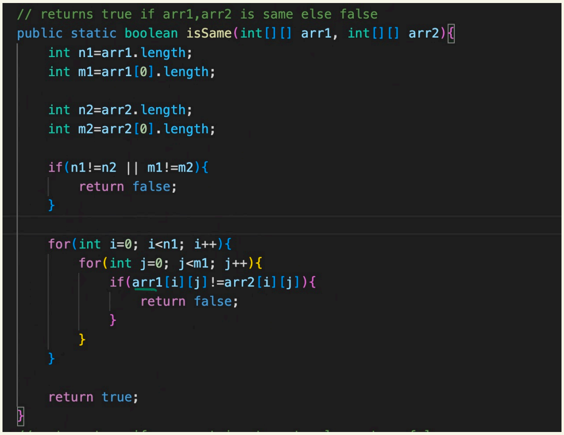
$tar=10$

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

$$\begin{array}{ccc} \underline{1} & \underline{2} & \underline{3} \\ \underline{4} & \underline{5} & \underline{6} \\ \underline{7} & \underline{8} & \boxed{9} \end{array} \qquad \begin{array}{ccc} \underline{1} & \underline{2} & \underline{3} \\ \underline{4} & \underline{5} & \underline{6} \\ \underline{7} & \underline{8} & \boxed{10} \end{array}$$

$n1=3$
$m1=3$

$n2=3$
$m2=3$



```java
// 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(0)(2)

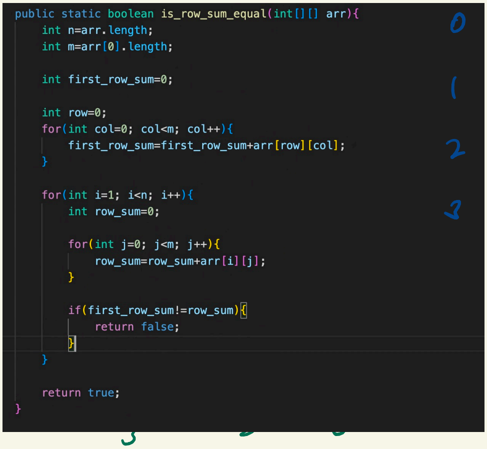
arr2(0)(2)

i=0

j=2

Ques Check if sum of every column is equal.

Ques Check if sum of every row is equal



```java
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;
}
```

$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 1 & 5 & 4 \\ 1 & 2 & 3 & 5 \\ 2 & 5 & 0 & 5 \\ 3 & 6 & 2 & 2 \end{array}$$

n=4

m=3

first_row_sum = ~~0~~ ~~1~~ 6 10

alse

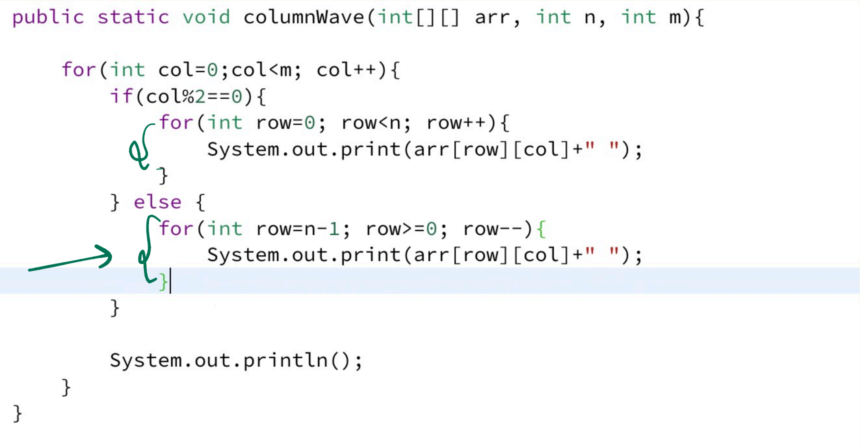
0,0
0,1
0,2

1,0
1,1
1,2

2,0
2,1
2,2

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 2 & 3 & 4 \\ 1 & 5 & 6 & 7 & 8 \\ 2 & 9 & 10 & 11 & 12 \\ 3 & 13 & 14 & 15 & 16 \end{array} \Rightarrow \begin{array}{cccc} 1 & 5 & 9 & 13 \\ 14 & 10 & 6 & 2 \\ 3 & 7 & 11 & 15 \\ 16 & 12 & 8 & 4 \end{array}$$

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();
    }
}
```

col = ~~0~~ ~~1~~ ~~2~~ 3

row = ~~3~~ ~~2~~ ~~1~~ 0

Identity matrix

1) square matrix

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

$\boxed{i==j}$

row == col

(0,0)
(1,1)
(2,2)
(3,3) → 1

Symmetric Matrix [transpose is equal to the matrix]

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

$\boxed{i < j}$

$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 1 & 2 & 3 \\ 1 & 4 & 5 & 6 \\ 2 & 7 & 8 & 9 \end{array} \Rightarrow \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 1 & 4 & 7 \\ 1 & 2 & 5 & 8 \\ 2 & 3 & 6 & 9 \end{array}$$

$\boxed{i < j}$

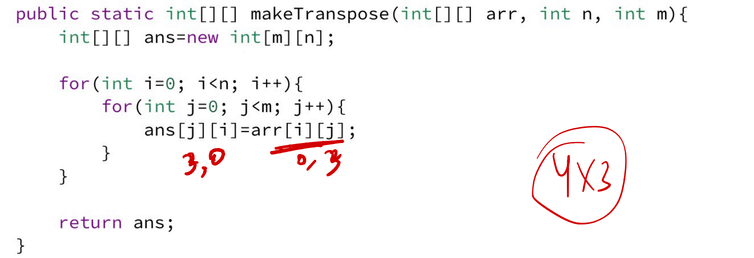
1 → ✓(0, 0) → (0, 0)
2 → ✓(0, 1) → (1, 0)
3 → ✓(0, 2) → (2, 0)
4 → ✓(1, 0) → (0, 1)
5 → ✓(1, 1) → (1, 1)
6 → ✓(1, 2) → (2, 1)

7 → ✓(2, 0) → (0, 2)
8 → ✓(2, 1) → (1, 2)
9 → ✓(2, 2) → (2, 2)

(0, 1) ⟷ (1, 0)
(0, 2) ⟷ (2, 0)
(1, 2) ⟷ (2, 1)

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

ans(2)(1)
= arr(1)(2)



```
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;
}
```

3,0 0,3

$\boxed{4X3}$

$i = 0$

$j = \cancel{0}\cancel{1}\cancel{2}3$