int [] arr → 0th index address

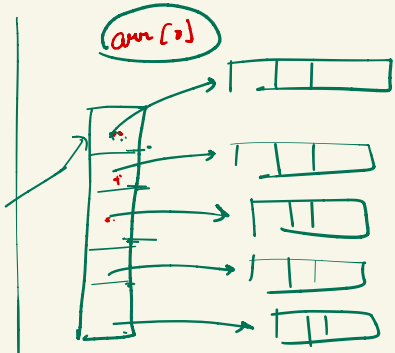
arr = new int [5]

2d array → array of arrays

array of integers

array of String

arr[0]



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

number of rows

number of columns

arr[2][0], arr[2][1], arr[2][2]

arr[1]

arr[2][0]

arr[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

```java
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.

```java
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) row sum equal

4) col sum equal.

```java
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;
}
```

n=3, m=4

	0	1	2	3
0	1	2	3	4
1	5	6	4	9
2	9	9	9	9

tar = 10

1) 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

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 ... 2

Ques Check if sum of every column is equal.

Ques Check if sum of every row is equal

n = 4

m = 3

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

first_row_sum = 10

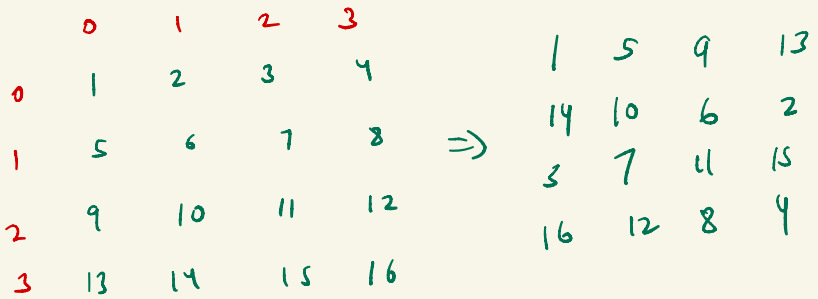
```java
public static boolean is_row_sum_equal(int[][] arr){
    int n=arr.length;
    int m=arr[0].length;

    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;
}
```

else

0,0 1,0 2,0

0,1 1,1 2,1

0,2 1,2 2,2



curr(row)(col)

```java
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 to 3
row = 0 to 4

Identity matrix

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]

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

(i, j)

(j, i)

$i < j$

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

$i < j$

$(0,1) \longleftrightarrow (1,0)$

$1 \rightarrow (0,0) \rightarrow (0,0)$

$2 \rightarrow (0,1) \rightarrow (1,0)$

$3 \rightarrow (0,2) \rightarrow (2,0)$

$4 \rightarrow (1,0) \rightarrow (0,1)$

$5 \rightarrow (1,1) \rightarrow (1,1)$

$6 \rightarrow (1,2) \rightarrow (2,1)$

$7 \rightarrow (2,0) \rightarrow (1,2)$

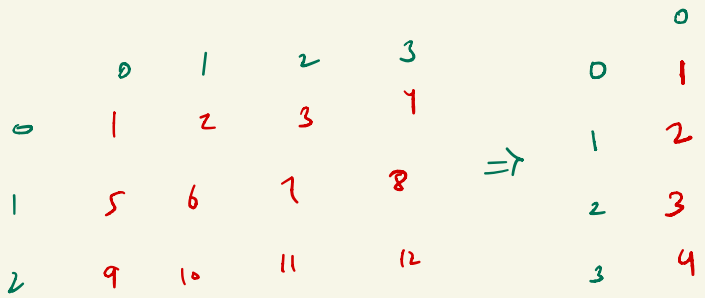
$8 \rightarrow (2,1) \rightarrow (1,4)$

$9 \rightarrow (2,2) \rightarrow (2,2)$

$(0,1) \longleftrightarrow (1,0)$

$(0,2) \longleftrightarrow (2,0)$

$(1,2) \longleftrightarrow (2,1)$



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;
}
```

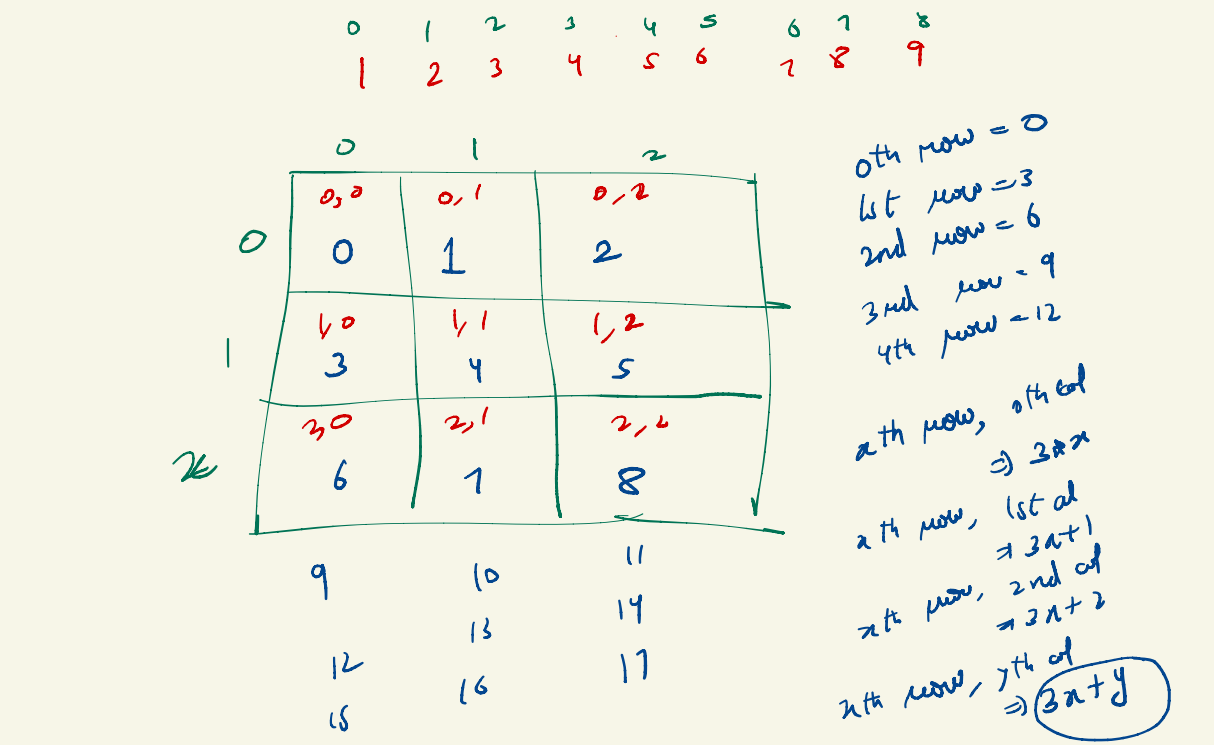
3,0 0,3

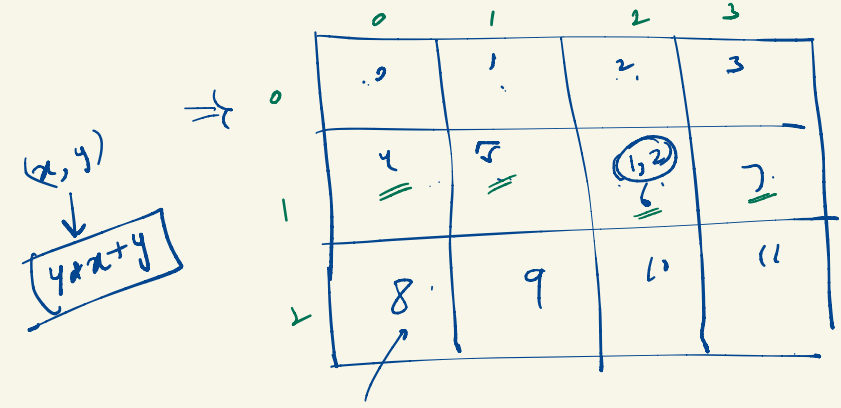
4X3

i = 0
j = ~~0 1 2 3~~

sum = ~~0~~ 1 ~~4~~ 5 ~~9~~ 7 ~~16~~ 9 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;
}
```






```java
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 * 1 old number of cols

over [x][y] = element at $x * cols + y$

$x * m + y$; m = number of columns

0 1 2 3 4 5 6 7 8

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

idx = x * m + y

idx = 7

x = 7 / 3 ⇒ 2

y = 7 % 3 ⇒ 1

x = idx / m

y = idx % m

in terms of idx and m

Multiplication of two matrices

m1 == n2

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}_{2*3} \quad n1 \times c1$$

$$B = \begin{bmatrix} 4 & 7 & 4 \\ 5 & 8 & 5 \\ 6 & 9 & 3 \end{bmatrix}_{3\times 3} \quad n2 \times c2$$

i = 0 1

j = 0

x * y y * z = x * z

1*4 + 2*5 + 3*6

$$C = \begin{bmatrix} 22 & 50 & 23 \\ \cdot & \cdot & \cdot \end{bmatrix}_{2} \quad n1 * m2$$

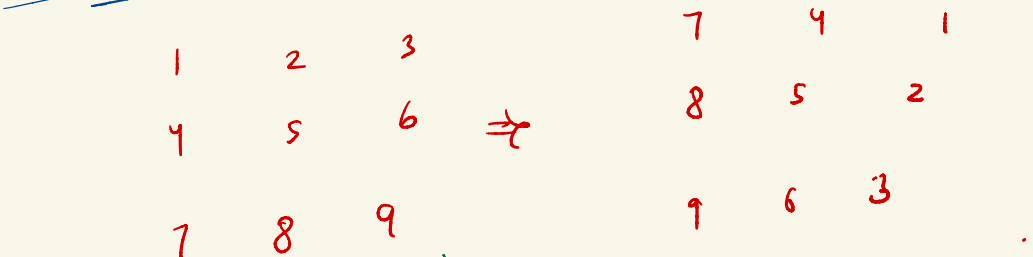
```
public static int[][] multiplyMatrix(int[][] A, int n1, int m1, int[][]B, int n2,int m
    int[][] C=new int[n1][m2];

    for(int i=0; i<n1; i++){
        for(int j=0; j<m2; j++){
            int ans=0;

            for(int k=0; k<m1; k++){
                ans=ans+(A[i][k]*B[k][j]);
            }

            C[i][j]=ans;
        }
    }

    return C;
}
```

n1, m1

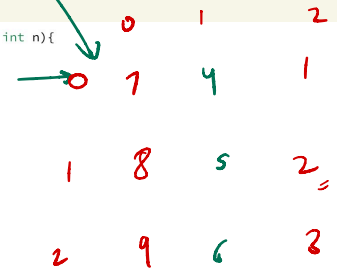
n2, m2 (n2 == m1)

n1 * m2 * m1

Rotate 2D array by 90°




```java
public static void rotateBy90Degrees(int[][] arr, int n){
    transpose(arr,n);

    for(int row=0; row<n; row++){
        // reverse the whole row

        int i=0;
        int j=n-1;

        while(i<j){
            // swap arr[row][i],arr[row][j]

            int temp=arr[row][i];

            arr[row][i]=arr[row][j];
            arr[row][j]=temp;

            i++;
            j--;
        }
    }
}
```



arr(0) i=0 j=n-1

arr(0)(i), arr(0)(j)

arr(0)(i), arr(0)(j)

row

arr(row)(i), arr(row)(j)