

Print row wise with condition

arr = 0

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

modify

arr = 0

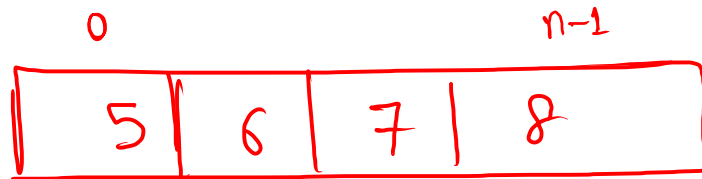
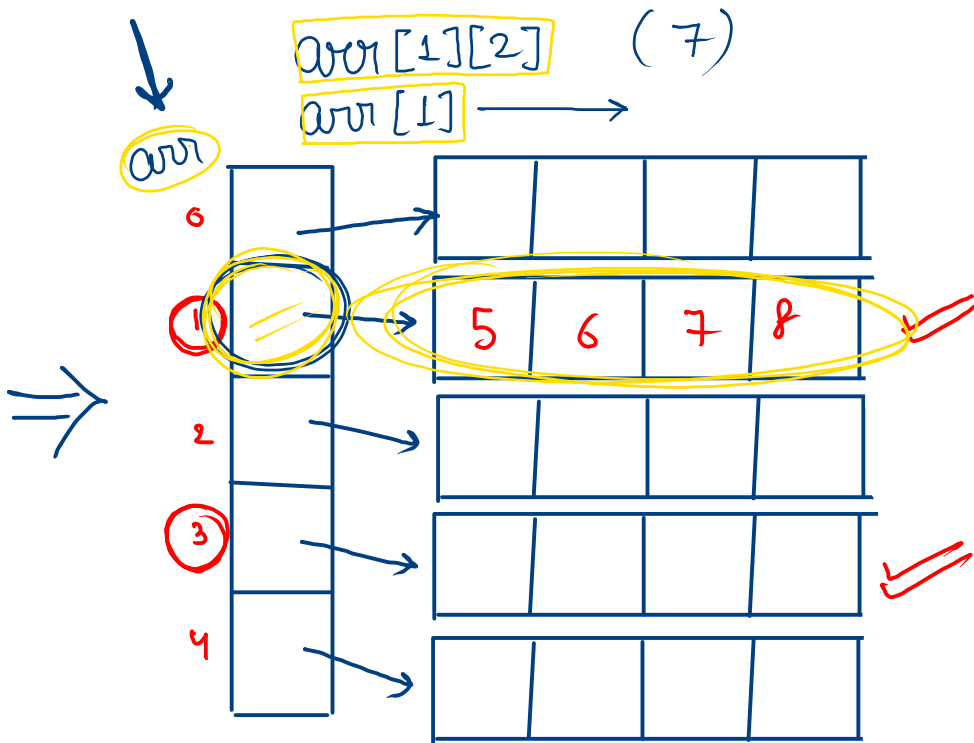
	0	1	2	3
0	1	2	3	4
1	8	7	6	5
2	9	10	11	12
3	16	15	14	13
4	17	18	19	20

(5 * 4)

arr

$m \times n$

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



n

reverse



code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int m = scn.nextInt(); // no of rows
    int n = scn.nextInt(); // no of cols
    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();
        }
    }
    printRowwise(arr, m, n);
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            System.out.print(arr[i][j] + " ");
        }
        System.out.println();
    }
}

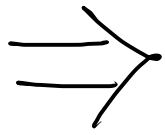
public static void printRowwise(int[][] arr, int m, int n) {
    for (int i = 0; i < m; i++) {
        if (i % 2 != 0) {
            int si = 0;
            int ei = n - 1;
            while (si < ei) {
                swap(arr[i], si, ei);
                si++;
                ei--;
            }
        }
    }
}

public static void swap(int[] ar, int si, int ei) {
    int temp = ar[si];
    ar[si] = ar[ei];
    ar[ei] = temp;
}
```

$$T.C = O(m \times n)$$

linear T.C

$$\underline{\underline{S.C = O(1)}}$$



Transpose
($n \times n$)

(convert all rows into cols)
and vice versa

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

modify
→

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

S.C = $O(1)$

Transpose of Matrix of N*N

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

modify →

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

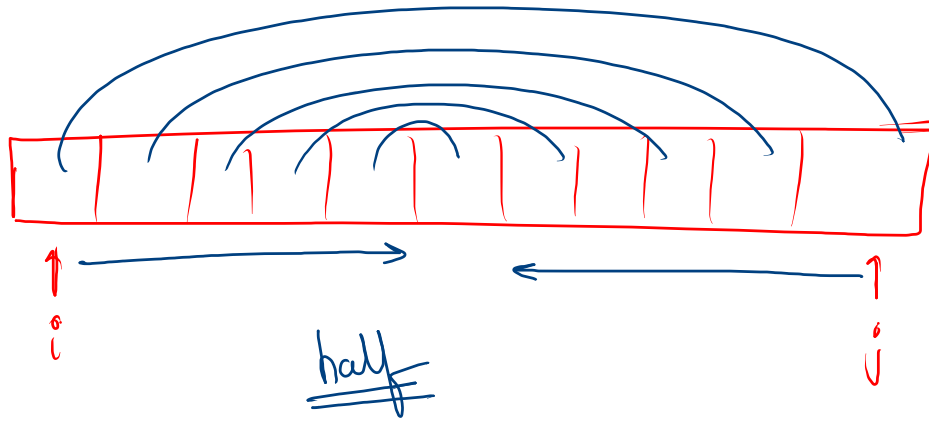
$(0,1)$
 $(0,2)$
 $(0,3)$
 $(1,2)$
 $(1,3)$
 $(2,3)$
 $(i < j)$

$$(i, j) \rightarrow (j, i)$$

observation

original → swap

$(0,0) \rightarrow (0,0)$
 $(0,1) \rightarrow (1,0) \checkmark$
 $(0,2) \rightarrow (2,0)$
 $(0,3) \rightarrow (3,0)$
 $(1,0) \rightarrow (0,1) \checkmark$
 $(1,1) \rightarrow (1,1)$
 $(1,2) \rightarrow (2,1)$
 $(1,3) \rightarrow (3,1)$
 $(2,0) \rightarrow (0,2)$
 $(2,1) \rightarrow (1,2)$
 $(2,2) \rightarrow (2,2)$
 $(2,3) \rightarrow (3,2)$
 $(3,0) \rightarrow (0,3)$
 $(3,1) \rightarrow (1,3)$
 $(3,2) \rightarrow (2,3)$
 $(3,3) \rightarrow (3,3)$



reverse

size = n ,

operation = $\frac{n}{2}$

code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[][] arr = new int[n][n];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }
    transpose(arr, n);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            System.out.print(arr[i][j] + " ");
        }
        System.out.println();
    }
}

public static void transpose(int[][] arr, int n) {

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if ( i > j ) {
                int temp = arr[i][j];
                arr[i][j] = arr[j][i];
                arr[j][i] = temp;
            }
        }
    }
}
```

$$\text{operations} = \frac{n * n}{2}$$

$$\underline{\underline{T.C = O(n^2)}}$$

Rotate The Matrix by 90 Degree

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

90°

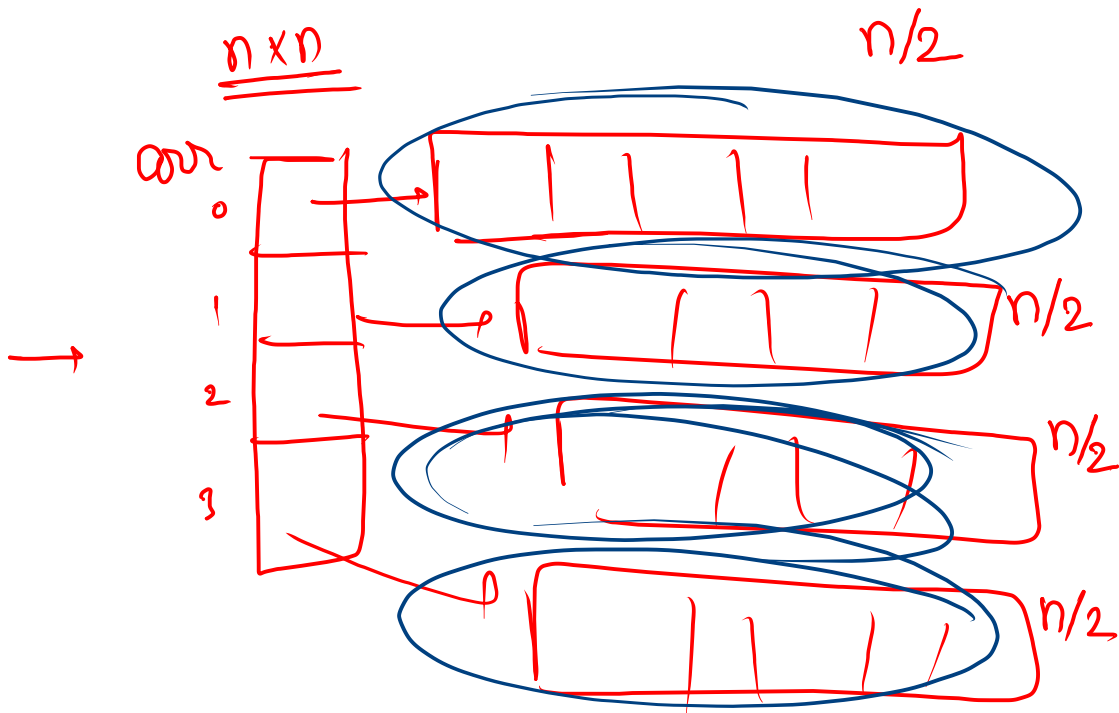
transpose

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

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

reverse
every row

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



code

```
① public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
    int[][] arr = new int[n][n];  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < n; j++) {  
            arr[i][j] = scn.nextInt();  
        }  
    }  
    rotate90(arr, n);  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < n; j++) {  
            System.out.print(arr[i][j] + " ");  
        }  
        System.out.println();  
    }  
}
```

```
② public static void rotate90(int[][] arr, int n) {  
    // step1 = transpose →  $O(n \times n)$   
    transpose(arr, n);  
    // step2 = reverse every row →  $O(n \times n)$   
    reverseEveryRow(arr, n);  
}
```

$$T.C = O(2 \times n^2)$$

$$T.C \approx O(n^2)$$

```
③ public static void reverseEveryRow(int[][] arr, int n) {  
    for (int i = 0; i < n; i++) {  
        int si = 0;  
        int ei = n - 1;  
        while (si < ei) {  
            swap(arr[i], si, ei);  
            si++;  
            ei--;  
        }  
    }  
}
```

```
④ public static void swap(int[] ar, int si, int ei) {  
    int temp = ar[si];  
    ar[si] = ar[ei];  
    ar[ei] = temp;  
}
```

```
⑤ public static void transpose(int[][] arr, int n) {  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < n; j++) {  
            if (i > j) {  
                int temp = arr[i][j];  
                arr[i][j] = arr[j][i];  
                arr[j][i] = temp;  
            }  
        }  
    }  
}
```

Rotate The Matrix by 180 Degree

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[][] arr = new int[n][n];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }
    rotate90(arr, n);
    rotate90(arr, n);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            System.out.print(arr[i][j] + " ");
        }
        System.out.println();
    }
}
```

$$\underline{\underline{T.C = O(n^2)}}$$

just call
this fun"
twice