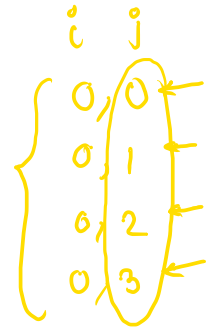
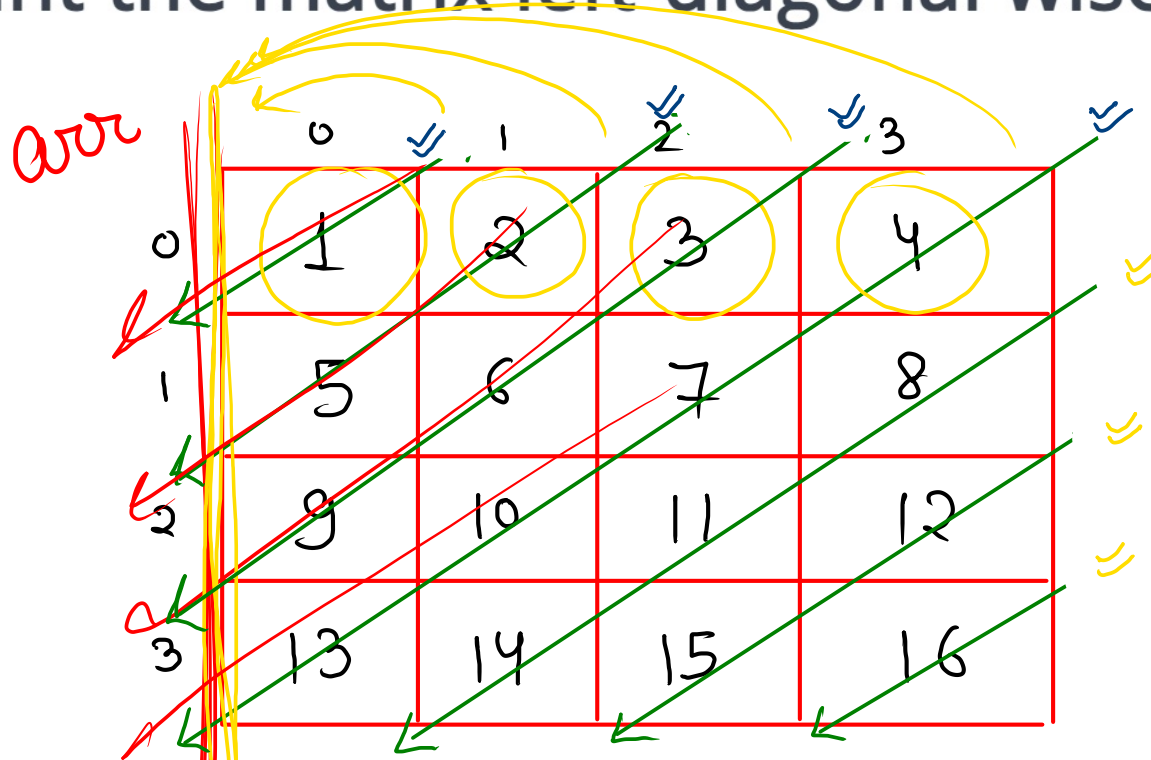
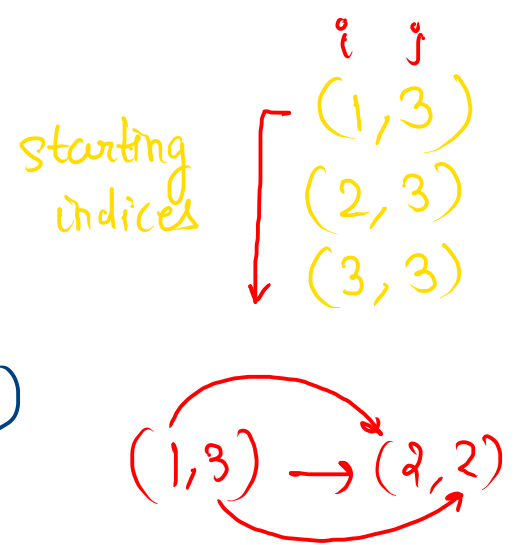
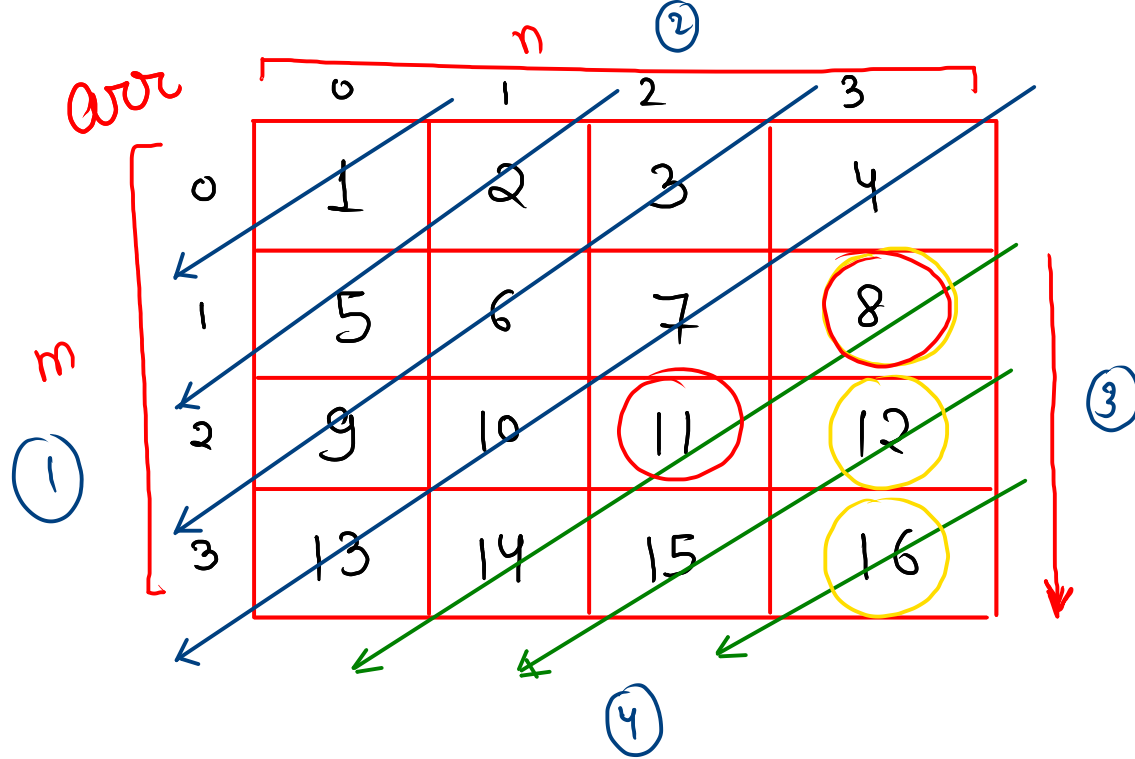


Print the matrix left-diagonal wise



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



```

for ( int gap = 1 ; gap < m ; gap++ ) {
    for ( int i = gap, j = n-1 ; i < m ; i++, j-- ) {
        syso( arr[i][j] + " ");
    }
}

```

complete code

```
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] + " ");  
        }  
    }  
  
    for (int gap = 1; gap < n; gap++) {  
        for (int i = gap, j = n - 1; i < n; i++, j--) {  
            System.out.print(arr[i][j] + " ");  
        }  
    }  
}
```

$$\text{gap}=0, \quad i=0, j=0 \\ i=1, j=1$$

$$\text{gap}=1, \quad i=0, j=1 \quad (2) \\ \underline{i=1, j=0} \quad (5) \\ i=2, j=-1$$

$$\text{gap}=2, \quad i=0, j=2 \quad (3) \\ i=1, j=1 \quad (6) \\ \underline{i=2, j=0} \quad (9) \\ i=3, j=-1$$

$$\text{gap}=3, \quad i=0, j=3 \quad (4) \\ i=1, j=2 \quad (7) \\ i=2, j=1 \quad (10) \\ i=3, j=0 \quad (13) \\ i=4, j=-1$$

$$\text{gap}=4$$

$$\text{gap}=1, \quad i=1, j=3 \quad (8)$$

$$i=2, j=2 \quad (11)$$

$$i=3, j=1 \quad (14)$$

$$\text{gap}=2, \quad i=2, j=3 \quad (12)$$

$$i=3, j=2 \quad (15)$$

$$\underline{i=4, j=1} \quad \propto$$

$$\text{gap}=3, \quad i=3, j=3 \quad (16)$$

$$\underline{i=4, j=2}$$

$$\text{gap}=4$$

Ques

Right diagonal

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

Transpose of Matrix of $N \times N$

all rows will become cols
and all cols will become rows

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

transpose of arr

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

{ 1st row will become 1st col
2nd row will become 2nd col
3rd row will become 3rd col
4th row will become 4th col

```

public static void transpose(int[][] arr, int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if ( j >= i ) {
                int temp = arr[i][j];
                arr[i][j] = arr[j][i];
                arr[j][i] = temp;
            }
        }
    }
    printing
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            System.out.print(arr[i][j] + " ");
        }
        System.out.println();
    }
}

```

it makes us travel
in upper triangle
only

Condⁿ

$j \geq i$

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

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

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

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

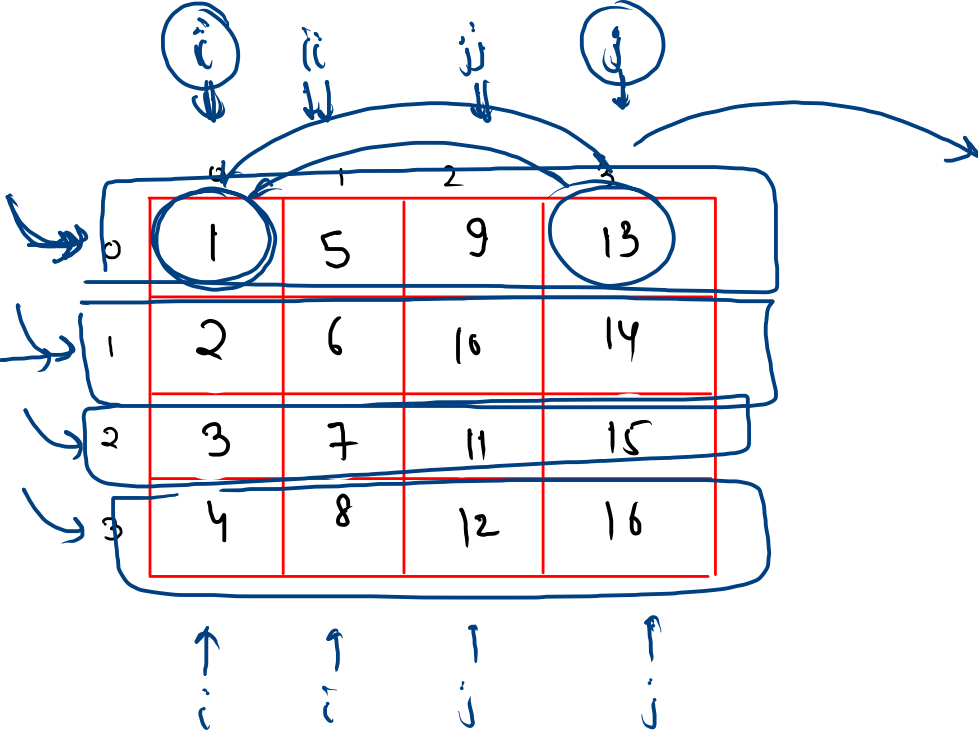
rotate 90°

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

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

swap
all cols



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

Sq matrix

arr.le = n
arr[0].le = n

arr[row][i] ↗
arr[row][j] ↘

```

public static void swapCols(int[][] arr, int n) {
    for (int row = 0; row < n; row++) {
        int i = 0;
        int j = n - 1;
        while (i < j) {
            int temp = arr[row][i];
            arr[row][i] = arr[row][j];
            arr[row][j] = temp;
            i++;
            j--;
        }
    }
}

```

row = 0, i = 0, j = 3
 i = 1, j = 2
 i = 2, j = 1 ✗

row = 1, i = 0, j = 3
 i = 1, j = 2
 i = 2, j = 1 ✗

row = 2, i = 0, j = 3
 i = 1, j = 2
 i = 2, j = 1 ✗

row = 3, i = 0, j = 3
 i = 1, j = 2
 i = 2, j = 1 ✗

row = 4 ✗

	i	i	j	j
	↓	↓	↓	↓
	0	1	2	3
0	13	9	5	1
1	14	10	6	2
2	15	11	7	3
row → 3	16	12	8	4

```

① public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt(); // row size & col size
    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);
}

```

```

② public static void rotate90(int[][] arr, int n) {
    // step1
    transpose(arr, n);

    // swap cols
    swapCols(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();
    }
}

```

```

③ public static void swapCols(int[][] arr, int n) {
    for (int row = 0; row < n; row++) {
        int i = 0;
        int j = n - 1;
        while (i < j) {
            int temp = arr[row][i];
            arr[row][i] = arr[row][j];
            arr[row][j] = temp;
            i++;
            j--;
        }
    }
}

```

```

④ public static void transpose(int[][] arr, int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (j >= i) {
                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();    // row size & col size
    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();
    }
}
```