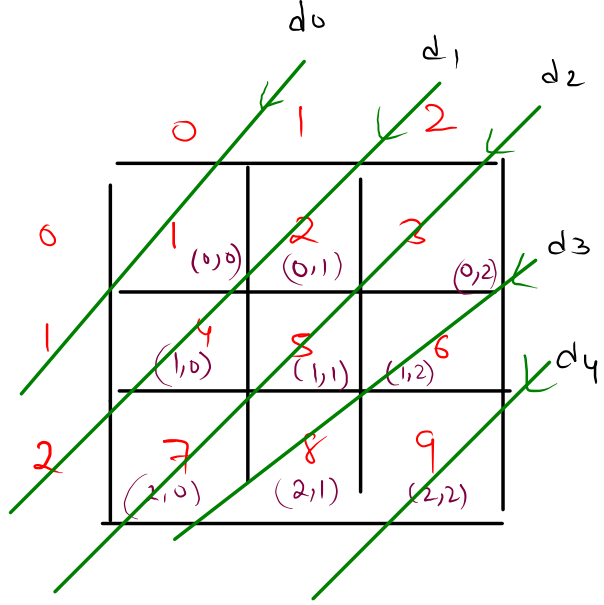
left Diagonal.



i+j = sum

Sample Input 0

```
3
1 2 3
4 5 6
7 8 9
```

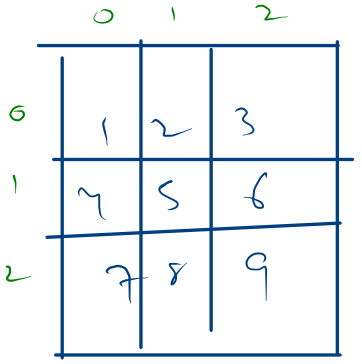
Sample Output 0

```
1 2 4 3 5 7 6 8 9
```

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

* total diagonal = 2n - 1 = 5

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int [][] A = new int[n][n];
        for(int i = 0; i < n; i++){
            for(int j = 0; j < n; j++){
                A[i][j] = scn.nextInt();
            }
        }
        //logic
        int d = 2*n-1;
        for(int s = 0; s < d; s++){
           for(int i = 0; i < n; i++){
               for(int j = 0; j < n; j++){
                   if(i + j == s){
                       System.out.print(A[i][j] + " ");
                   }
               }
           }
        }
    }
}
```



d = 5

s=0 · 0<3 ✓

i=0 0<3 ✓

j=0 1

0+0 == 0

1<3 ✓

Upper Triangle.

Sample Input 0

```
3
3
3 7 1
6 2 4
7 1 0
```

Sample Output 0

```
3 7 1
0 2 4
0 0 0
```

3 7 1

6 2 4

7 1 8

i j

0,0 0,1 0,2

1,0 1,1 1,2

2,0 2,1 2,2

3 7 1

0 2 4

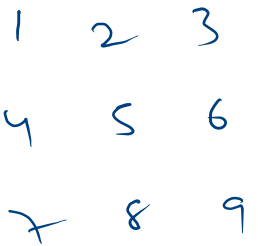
0 0 8

logic.

$i \leq j$ → val

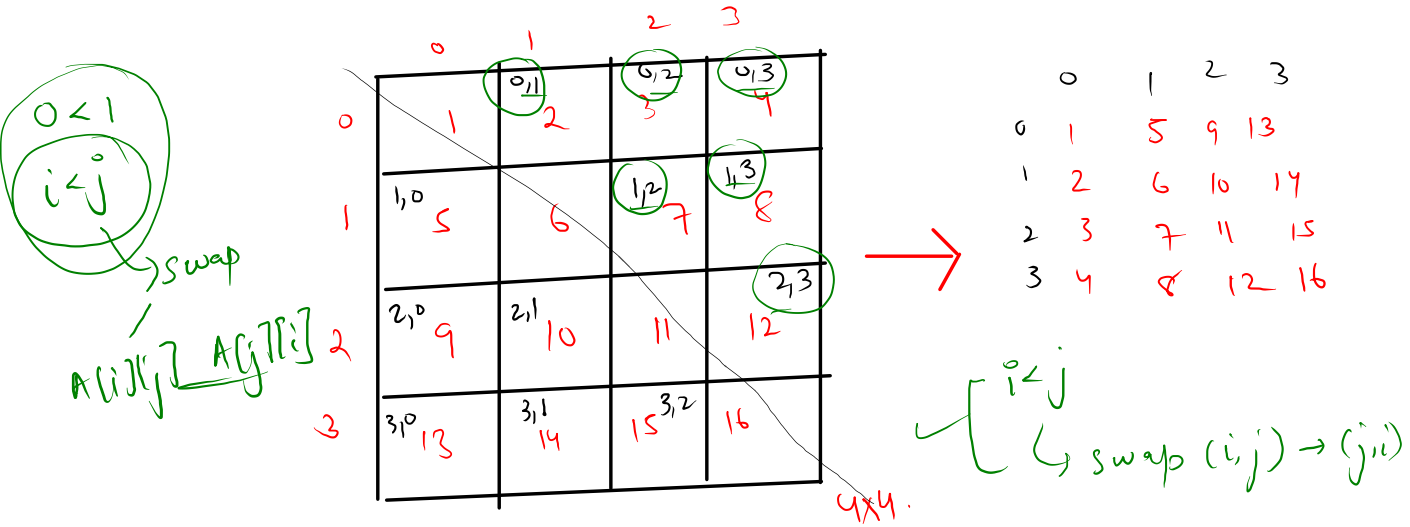
else → 0

```java
public class Solution {

    public static void swap(int [][] A, int i, int j){
        int tmp = A[i][j];
        A[i][j] = A[j][i];
        A[j][i] = tmp;
    }

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



Transpose of Matrix of N*N

n x n. Transpose. → interchange rows with columns.

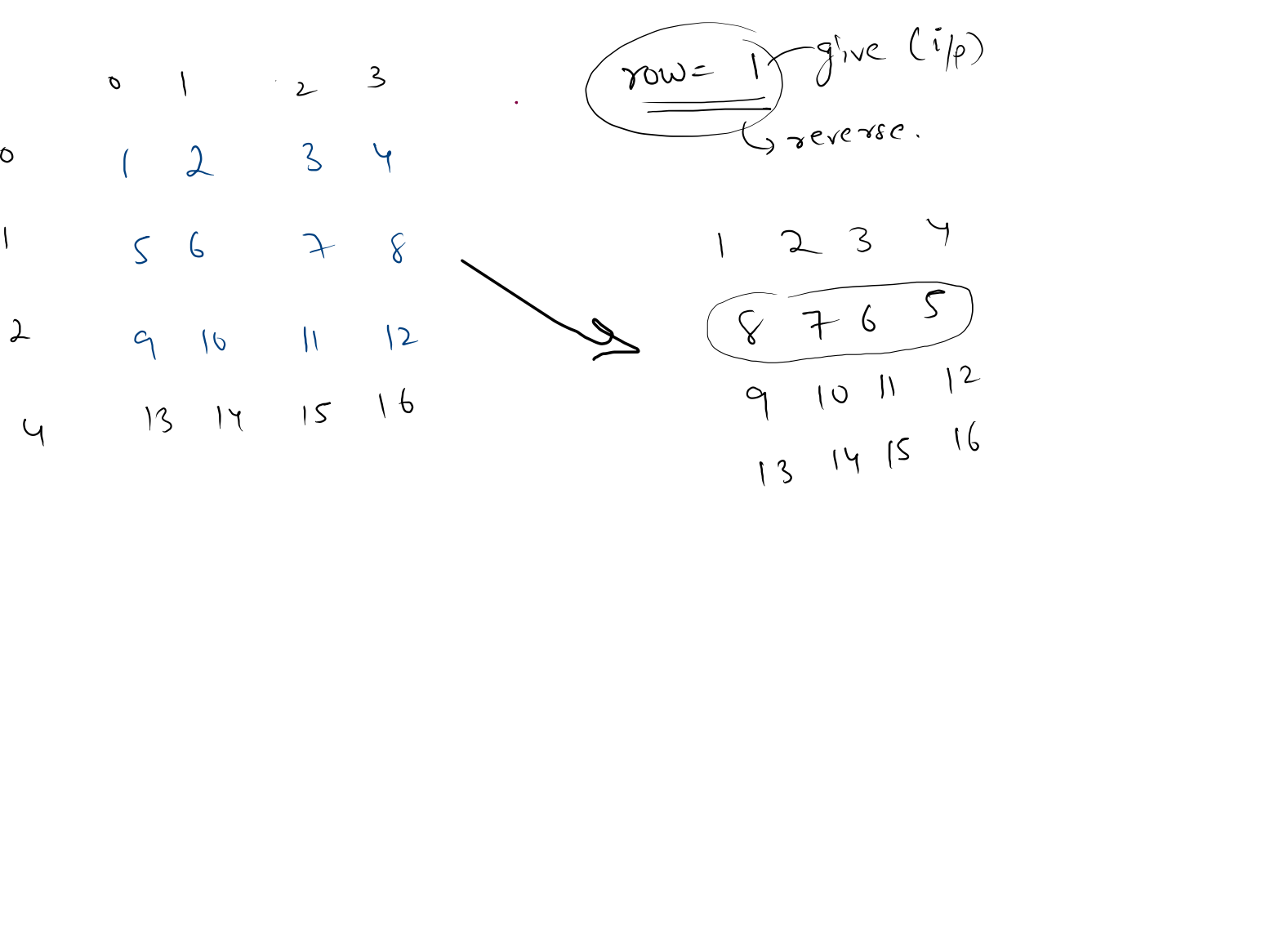


	0	1	2	3
0	1	5	9	13
1	2	6	10	14
2	3	7	11	15
3	4	8	12	16

0 < 1
i < j
↳ swap
A[i][j] A[j][i]

i < j
↳ swap (i, j) → (j, i)

```java
public class Solution {

    public static void swap(int [][] A, int i, int j){
        int tmp = A[i][j];
        A[i][j] = A[j][i];
        A[j][i] = tmp;
    }

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

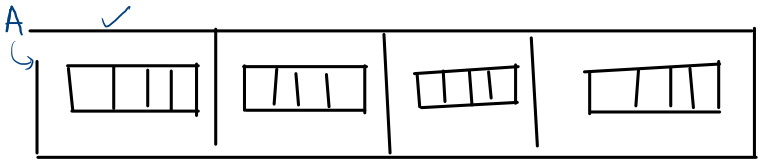

0 1 2 3
0 1 2 3 4
1 5 6 7 8
2 9 10 11 12
4 13 14 15 16
row = 1
give (i/p)
reverse.
1 2 3 4
8 7 6 5
9 10 11 12
13 14 15 16

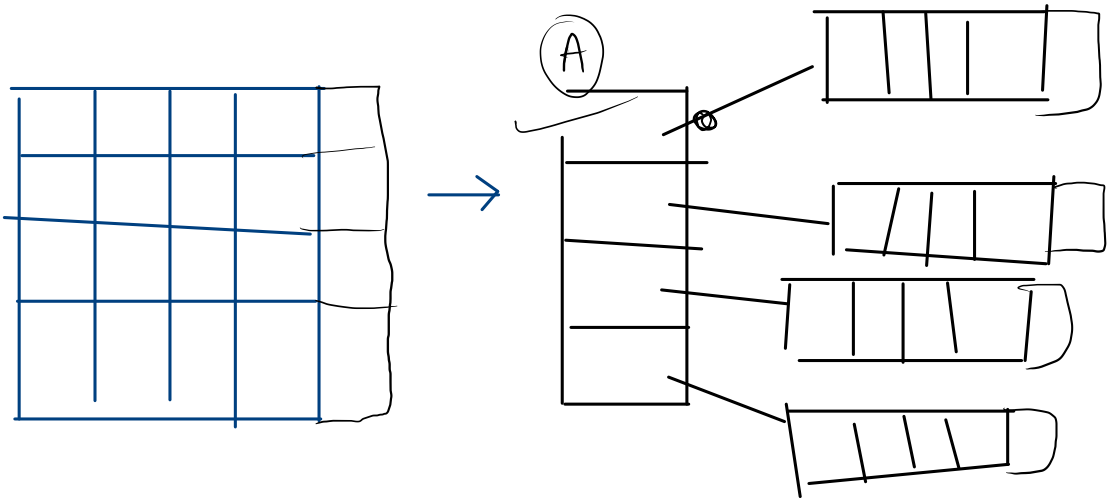
```java
public static void main(String[] args) {
    int [][] A = {{11,12,13,14},
                  {15,16,17,18},
                  {19,20,21,22},
                  {23,24,25,26}};
    int row = 1;
   // reverseRow(A,row);
   int totalRows = A.length;
   int totalCols = A[0].length;
```

2D → Array → concept.

rows → A.length

cols → A[0].length





```
{{11,12,13,14},
 {15,16,17,18},
 {19,20,21,22},
 {23,24,25,26}};
```

row=1

tmp = 16

18 17 16 15

15 16 17 18

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

j-- i++

tmp = A[1][1]

A[1][1] = A[1][2]

A[1][2] = tmp

2 < 1

```java
public static void reverseRow(int [][] A, int row){
    int i = 0;
    int j = A[0].length-1;
    while(i < j){
        int tmp = A[row][i];
        A[row][i] = A[row][j];
        A[row][j] = tmp;
        i++;
        j--;
    }
}
```

Rotate by 90°

n = 3

1	2	3
4	5	6
7	8	9

→ 1. Transpose

1	4	7
2	5	8
3	6	9

2. reverse All row

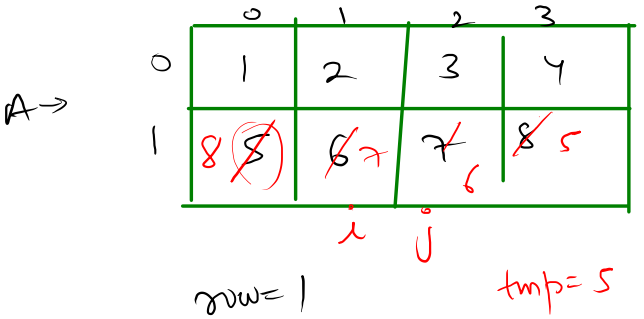
7	4	1
8	5	2
9	6	3

Sample Input 0

```
3
1 2 3
4 5 6
7 8 9
```

Sample Output 0

```
7 4 1
8 5 2
9 6 3
```



i = 0

j = 3

0 < 3 ✓

tmp = A[1][0]

A[1][0] = A[1][3]

A[1][3] = 5

```java
public static void reverseRow(int [][] A, int row){
    int i = 0;
    int j = A[0].length-1;
    while(i < j){
        int tmp = A[row][i];
        A[row][i] = A[row][j];
        A[row][j] = tmp;
        i++;
        j--;
    }
}
```

1 2 3

4 5 6

7 8 9

180°

9 8 7

6 5 4

3 2 1

90°

90°

$180^\circ = 90^\circ + 90^\circ$