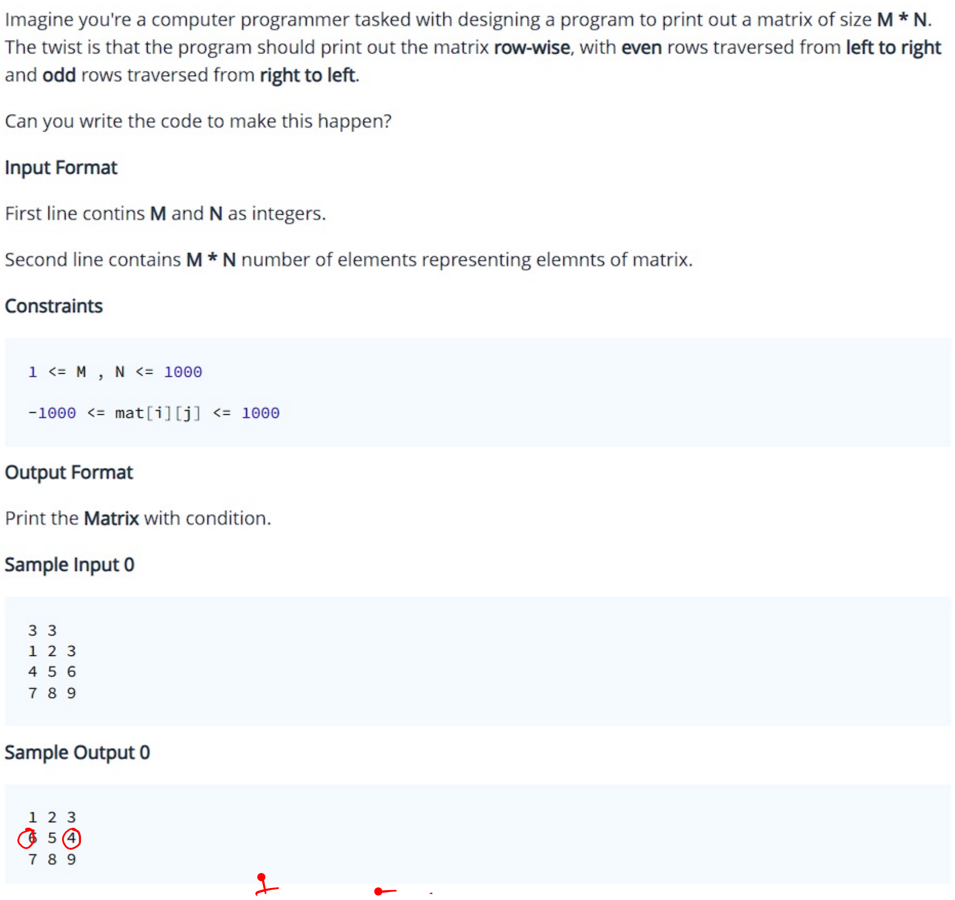


Imagine you're a computer programmer tasked with designing a program to print out a matrix of size **M * N**. The twist is that the program should print out the matrix **row-wise**, with **even** rows traversed from **left to right** and **odd** rows traversed from **right to left**.

Can you write the code to make this happen?

Input Format

First line contins **M** and **N** as integers.

Second line contains **M * N** number of elements representing elemnts of matrix.

Constraints

```
1 <= M , N <= 1000

-1000 <= mat[i][j] <= 1000
```

Output Format

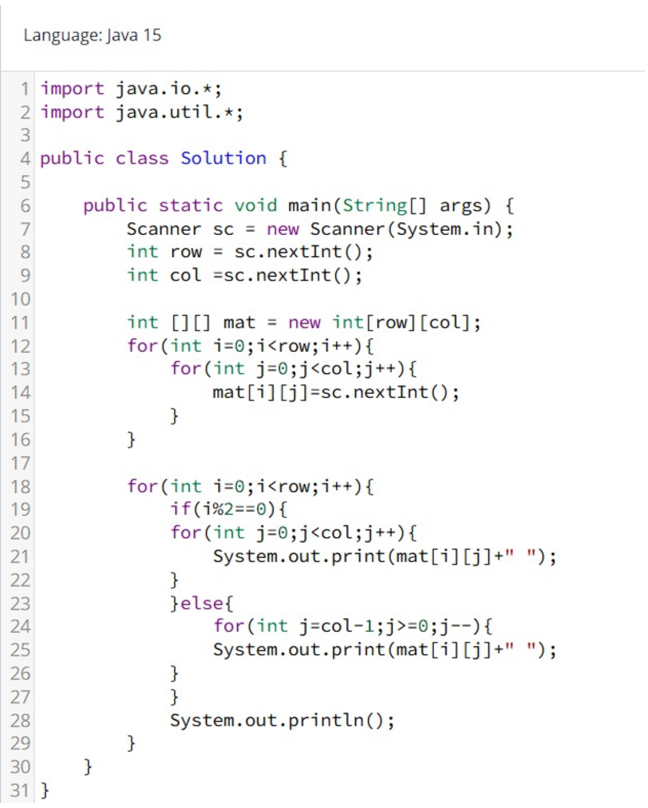
Print the **Matrix** with condition.

Sample Input 0

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

Sample Output 0

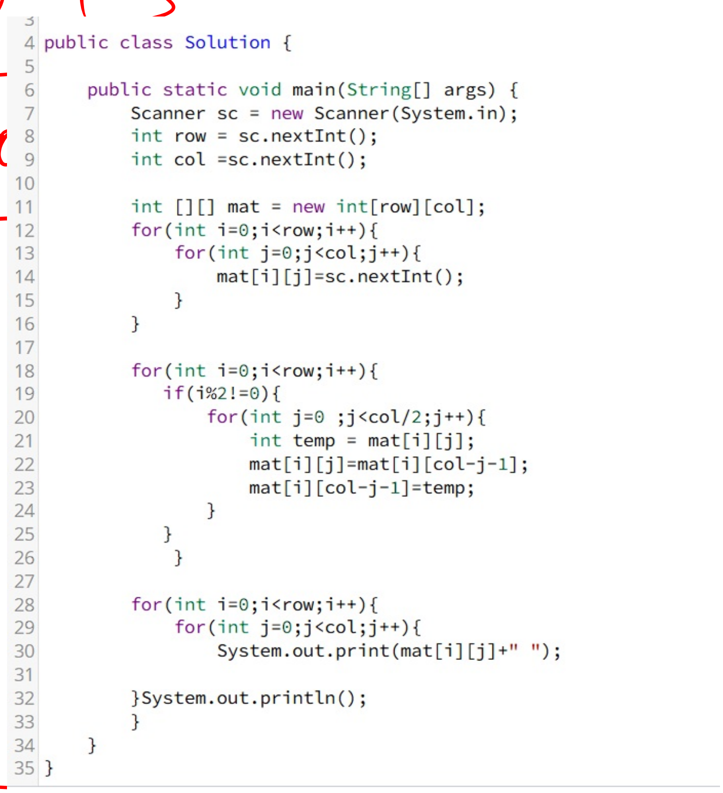
```
1 2 3
6 5 4
7 8 9
```



Language: Java 15

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

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int row = sc.nextInt();
        int col =sc.nextInt();

        int [][] mat = new int[row][col];
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
               mat[i][j]=sc.nextInt();
            }
        }

        for(int i=0;i<row;i++){
            if(i%2==0){
            for(int j=0;j<col;j++){
               System.out.print(mat[i][j]+" ");
            }
            }else{
               for(int j=col-1;j>=0;j--){
               System.out.print(mat[i][j]+" ");
            }
            }
            System.out.println();
        }
    }
}
```

10 : 18

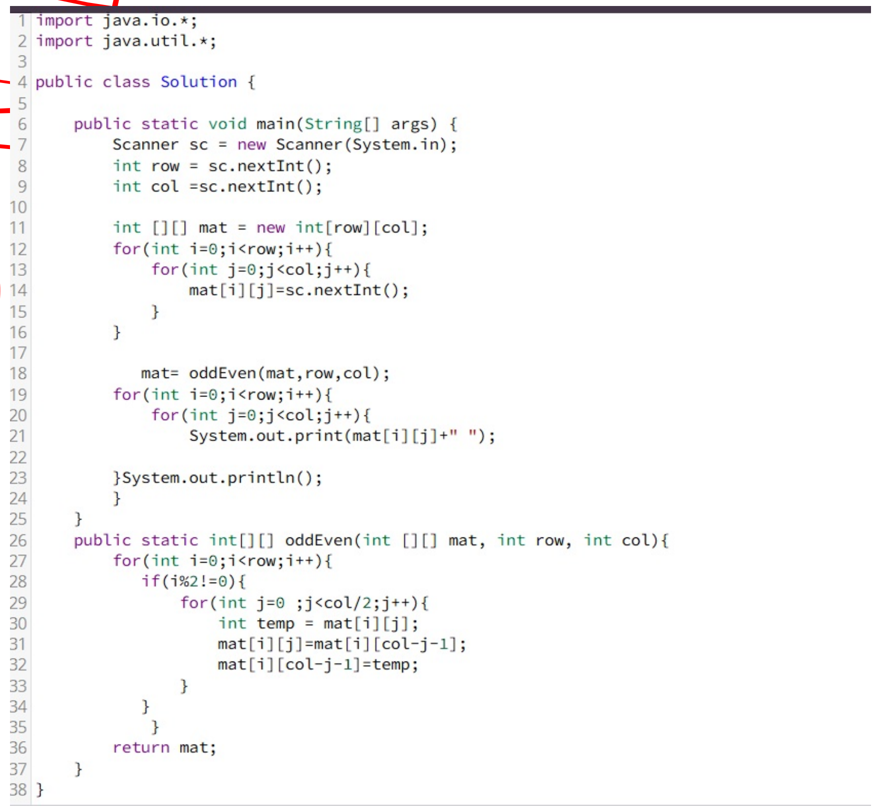


```
public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int row = sc.nextInt();
        int col =sc.nextInt();

        int [][] mat = new int[row][col];
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
               mat[i][j]=sc.nextInt();
            }
        }

        for(int i=0;i<row;i++){
           if(i%2!=0){
               for(int j=0 ;j<col/2;j++){
                  int temp = mat[i][j];
                  mat[i][j]=mat[i][col-j-1];
                  mat[i][col-j-1]=temp;
               }
           }
            }

        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
               System.out.print(mat[i][j]+" ");

        }System.out.println();
        }
    }
}
```

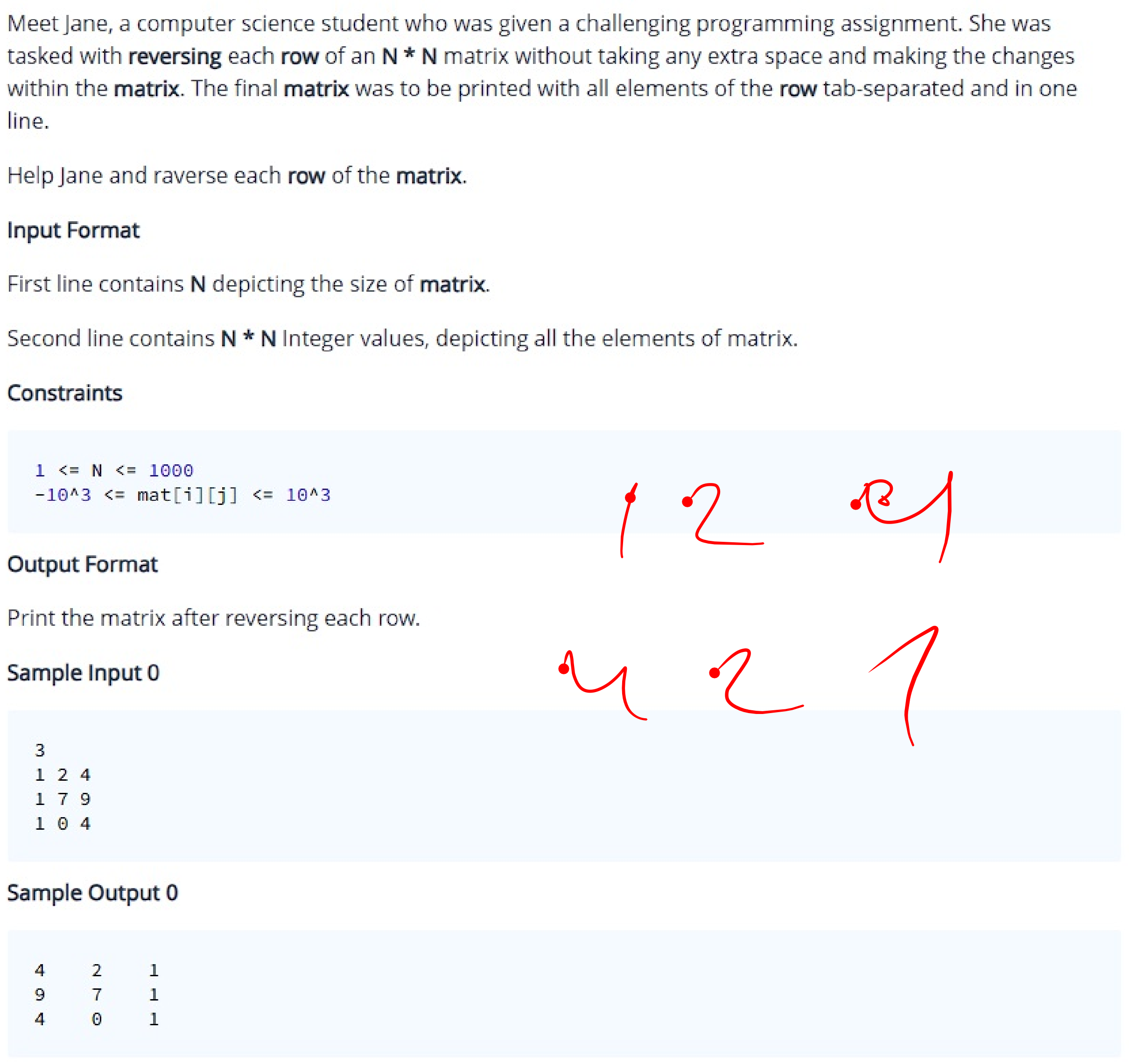
od

mat [ ] [ ]

return



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

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int row = sc.nextInt();
        int col =sc.nextInt();

        int [][] mat = new int[row][col];
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
               mat[i][j]=sc.nextInt();
            }
        }

           mat= oddEven(mat,row,col);
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
               System.out.print(mat[i][j]+" ");

        }System.out.println();
        }
    }
    public static int[][] oddEven(int [][] mat, int row, int col){
        for(int i=0;i<row;i++){
           if(i%2!=0){
               for(int j=0 ;j<col/2;j++){
                  int temp = mat[i][j];
                  mat[i][j]=mat[i][col-j-1];
                  mat[i][col-j-1]=temp;
               }
           }
            }
        return mat;
    }
}
```



Meet Jane, a computer science student who was given a challenging programming assignment. She was tasked with **reversing** each **row** of an **N * N** matrix without taking any extra space and making the changes within the **matrix**. The final **matrix** was to be printed with all elements of the **row** tab-separated and in one line.

Help Jane and raverse each **row** of the **matrix**.

Input Format

First line contains **N** depicting the size of **matrix**.

Second line contains **N * N** Integer values, depicting all the elements of matrix.

Constraints

```
1 <= N <= 1000
-10^3 <= mat[i][j] <= 10^3
```

1 2 4

Output Format

Print the matrix after reversing each row.

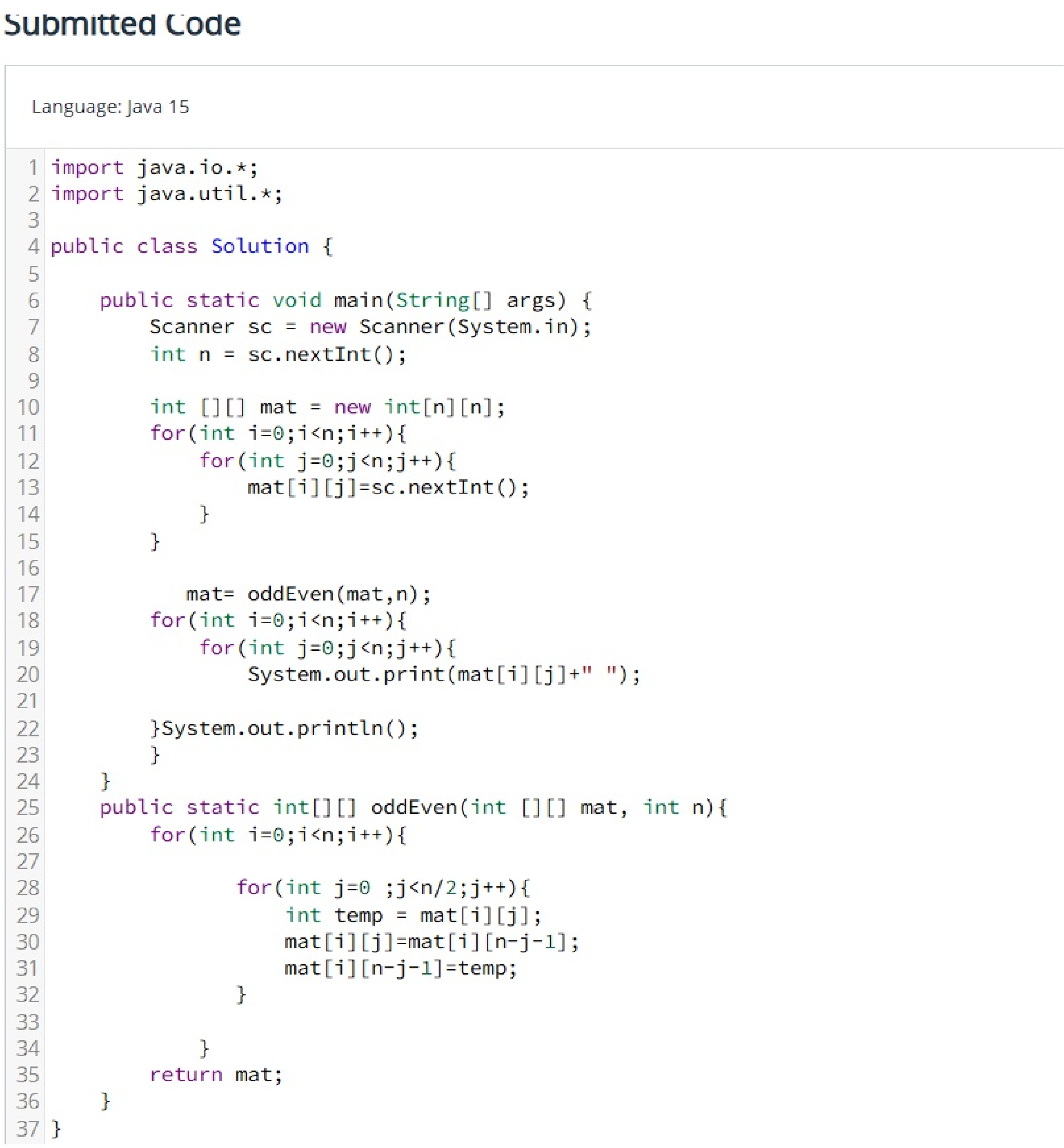
4 2 1

Sample Input 0

```
3
1 2 4
1 7 9
1 0 4
```

Sample Output 0

```
4    2    1
9    7    1
4    0    1
```



Submitted Code

Language: Java 15

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

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();

        int [][] mat = new int[n][n];
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                mat[i][j]=sc.nextInt();
            }
        }

           mat= oddEven(mat,n);
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                System.out.print(mat[i][j]+" ");

        }System.out.println();
        }
    }
    public static int[][] oddEven(int [][] mat, int n){
        for(int i=0;i<n;i++){

               for(int j=0 ;j<n/2;j++){
                   int temp = mat[i][j];
                   mat[i][j]=mat[i][n-j-1];
                   mat[i][n-j-1]=temp;
               }

            }
        return mat;
    }
}
```

Print the **matrix** column wise starting from the **0th** column such that the **even** column is traversed from **top to bottom** and **odd** column is traversed from **bottom to top**.

Input Format

First line contains **M** and **N** depicting the size of **matrix**.

Second line contains **M * N** Integer values, depicting all the elements of **matrix**.

Constraints

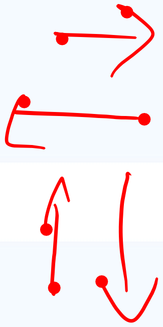
```
1 <= M , N <=1000
-1000 <= mat[i][j] <= 1000
```

Output Format

Print the **matrix.**

Sample Input 0

```
3
3
3 1 2
3 0 2
4 5 4
```

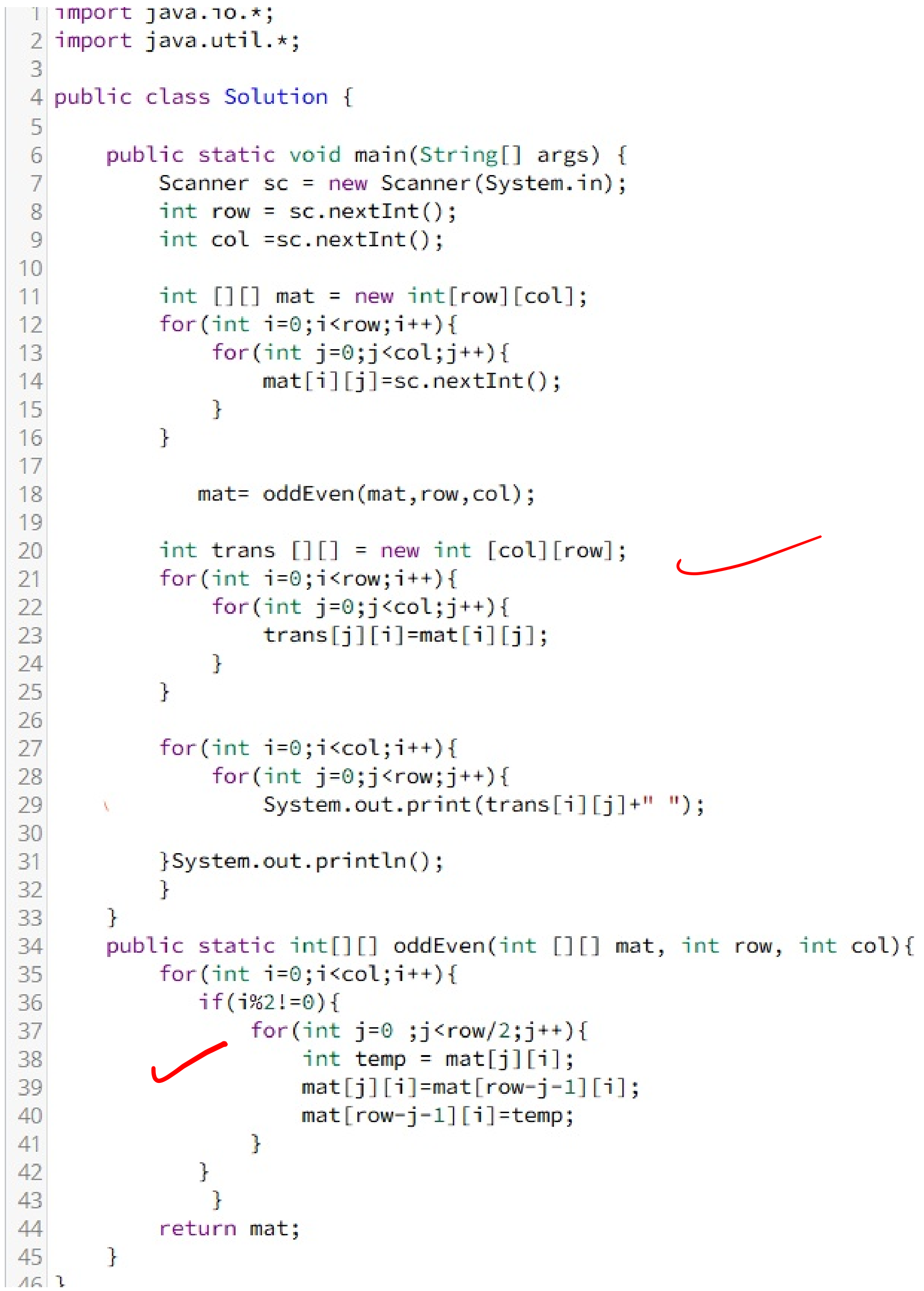


Sample Output 0

```
3 3 4
5 0 1
2 2 4
```

Explanation 0

Print the matrix according to given conditions.



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

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int row = sc.nextInt();
        int col =sc.nextInt();

        int [][] mat = new int[row][col];
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                mat[i][j]=sc.nextInt();
            }
        }

           mat= oddEven(mat,row,col);

        int trans [][] = new int [col][row];
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                trans[j][i]=mat[i][j];
            }
        }

        for(int i=0;i<col;i++){
            for(int j=0;j<row;j++){
                System.out.print(trans[i][j]+" ");

        }System.out.println();
        }
    }
    public static int[][] oddEven(int [][] mat, int row, int col){
        for(int i=0;i<col;i++){
           if(i%2!=0){
               for(int j=0 ;j<row/2;j++){
                   int temp = mat[j][i];
                   mat[j][i]=mat[row-j-1][i];
                   mat[row-j-1][i]=temp;
               }
           }
            }
        return mat;
    }
}
```