

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 contains **M** and **N** as integers.

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

Constraints

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

Output Format

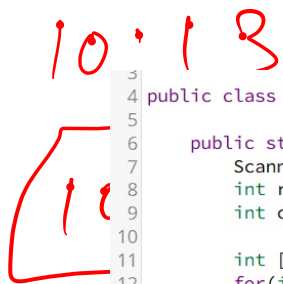
Print the **Matrix** with condition.

Sample Input 0

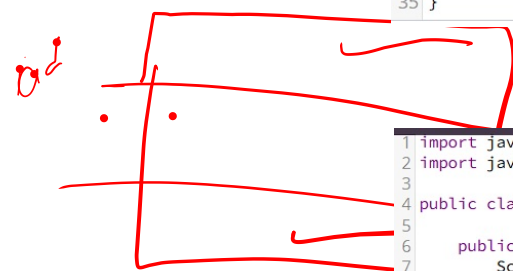
```
3 3
1 2 3
4 5 6
7 8 9
```

Sample Output 0

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



```
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int row = sc.nextInt();
9         int col = sc.nextInt();
10
11         int [][] mat = new int[row][col];
12         for(int i=0;i<row;i++){
13             for(int j=0;j<col;j++){
14                 mat[i][j]=sc.nextInt();
15             }
16         }
17
18         for(int i=0;i<row;i++){
19             if(i%2!=0){
20                 for(int j=0 ;j<col/2;j++){
21                     int temp = mat[i][j];
22                     mat[i][j]=mat[i][col-j-1];
23                     mat[i][col-j-1]=temp;
24                 }
25             }
26         }
27
28         for(int i=0;i<row;i++){
29             for(int j=0;j<col;j++){
30                 System.out.print(mat[i][j]+" ");
31             }
32             System.out.println();
33         }
34     }
35 }
```



mat [] []

return

```
Language: Java 15
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int row = sc.nextInt();
9         int col = sc.nextInt();
10
11         int [][] mat = new int[row][col];
12         for(int i=0;i<row;i++){
13             for(int j=0;j<col;j++){
14                 mat[i][j]=sc.nextInt();
15             }
16         }
17
18         for(int i=0;i<row;i++){
19             if(i%2==0){
20                 for(int j=0;j<col;j++){
21                     System.out.print(mat[i][j]+" ");
22                 }
23             }else{
24                 for(int j=col-1;j>=0;j--){
25                     System.out.print(mat[i][j]+" ");
26                 }
27             }
28             System.out.println();
29         }
30     }
31 }
```

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int row = sc.nextInt();
9         int col = sc.nextInt();
10
11         int [][] mat = new int[row][col];
12         for(int i=0;i<row;i++){
13             for(int j=0;j<col;j++){
14                 mat[i][j]=sc.nextInt();
15             }
16         }
17
18         mat= oddEven(mat,row,col);
19         for(int i=0;i<row;i++){
20             for(int j=0;j<col;j++){
21                 System.out.print(mat[i][j]+" ");
22             }
23             System.out.println();
24         }
25     }
26     public static int[][] oddEven(int [][] mat, int row, int col){
27         for(int i=0;i<row;i++){
28             if(i%2!=0){
29                 for(int j=0 ;j<col/2;j++){
30                     int temp = mat[i][j];
31                     mat[i][j]=mat[i][col-j-1];
32                     mat[i][col-j-1]=temp;
33                 }
34             }
35         }
36         return mat;
37     }
38 }
```

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 reverse 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
```

Output Format

Print the matrix after reversing each row.

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
```

Handwritten red annotations showing the reversal of each row in the sample input. The first row [1, 2, 4] is reversed to [4, 2, 1]. The second row [1, 7, 9] is reversed to [9, 7, 1]. The third row [1, 0, 4] is reversed to [4, 0, 1].

Submitted Code

Language: Java 15

```
1 import java.io.*;  
2 import java.util.*;  
3  
4 public class Solution {  
5  
6     public static void main(String[] args) {  
7         Scanner sc = new Scanner(System.in);  
8         int n = sc.nextInt();  
9  
10        int [][] mat = new int[n][n];  
11        for(int i=0;i<n;i++){  
12            for(int j=0;j<n;j++){  
13                mat[i][j]=sc.nextInt();  
14            }  
15        }  
16  
17        mat= oddEven(mat,n);  
18        for(int i=0;i<n;i++){  
19            for(int j=0;j<n;j++){  
20                System.out.print(mat[i][j]+" ");  
21            }  
22            System.out.println();  
23        }  
24    }  
25    public static int[][] oddEven(int [][] mat, int n){  
26        for(int i=0;i<n;i++){  
27  
28            for(int j=0 ;j<n/2;j++){  
29                int temp = mat[i][j];  
30                mat[i][j]=mat[i][n-j-1];  
31                mat[i][n-j-1]=temp;  
32            }  
33  
34        }  
35        return mat;  
36    }  
37 }
```

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

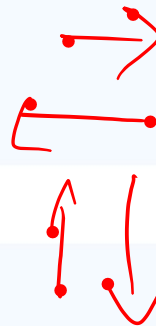
```
1 <= M , N <= 1000
-1000 <= mat[i][j] <= 1000
```

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.

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int row = sc.nextInt();
9         int col = sc.nextInt();
10
11         int [][] mat = new int[row][col];
12         for(int i=0;i<row;i++){
13             for(int j=0;j<col;j++){
14                 mat[i][j]=sc.nextInt();
15             }
16         }
17
18         mat= oddEven(mat,row,col);
19
20         int trans [][] = new int [col][row];
21         for(int i=0;i<row;i++){
22             for(int j=0;j<col;j++){
23                 trans[j][i]=mat[i][j];
24             }
25         }
26
27         for(int i=0;i<col;i++){
28             for(int j=0;j<row;j++){
29                 System.out.print(trans[i][j]+" ");
30             }System.out.println();
31         }
32     }
33
34     public static int[][] oddEven(int [][] mat, int row, int col){
35         for(int i=0;i<col;i++){
36             if(i%2!=0){
37                 for(int j=0 ;j<row/2;j++){
38                     int temp = mat[j][i];
39                     mat[j][i]=mat[row-j-1][i];
40                     mat[row-j-1][i]=temp;
41                 }
42             }
43         }
44         return mat;
45     }
46 }
```