Transpose of Matrix of N*N

Problem | Submissions | Leaderboard | Discussions

Sophie is a computer science student who loves solving programming challenges. One day, her professor gives her an interesting task - to write a program that finds the transpose of a square matrix of size N * N.

NOTE:- After answering the question, attempt the related question in the linked resource to improve your understanding of this question . Click here

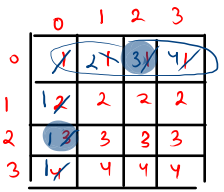
Sample Input 0

```
4
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
```

Sample Output 0

```
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
```

Transpose. n×n n=4



ans.

1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4

rows ⟷ cols

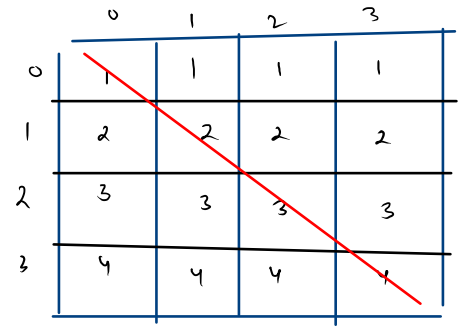
idx

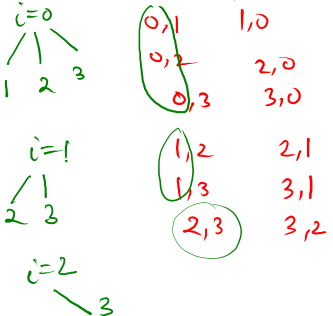
0,2

2,0

i,j → j,i

i,j → j,i





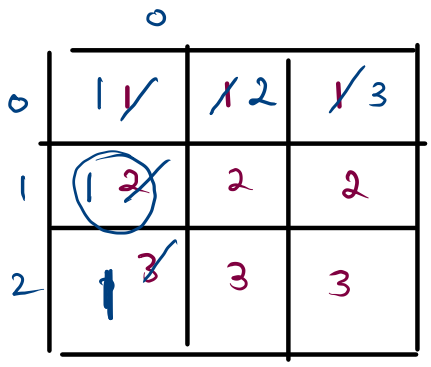
```java
import java.io.*;
import java.util.*;

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();
            }
        }

        //transpose
        for(int i = 0; i < n; i++){
            for(int j = i + 1; j < n; j++){
                int tmp = A[i][j];
                A[i][j] = A[j][i];
                A[j][i] = tmp;
            }
        }

        //print
        for(int i = 0; i < n; i++){
            for(int j = 0; j < n; j++){
                System.out.print(A[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```



n=3

i=1
j=0

i=0
j=0 1 2

Transpose → n x n

Rotate The Matrix by 90 Degree

Problem | Submissions | Leaderboard | Discussions

Take a **square** matrix of size **n*n** as input, and **rotate** the matrix by **90 degree**.

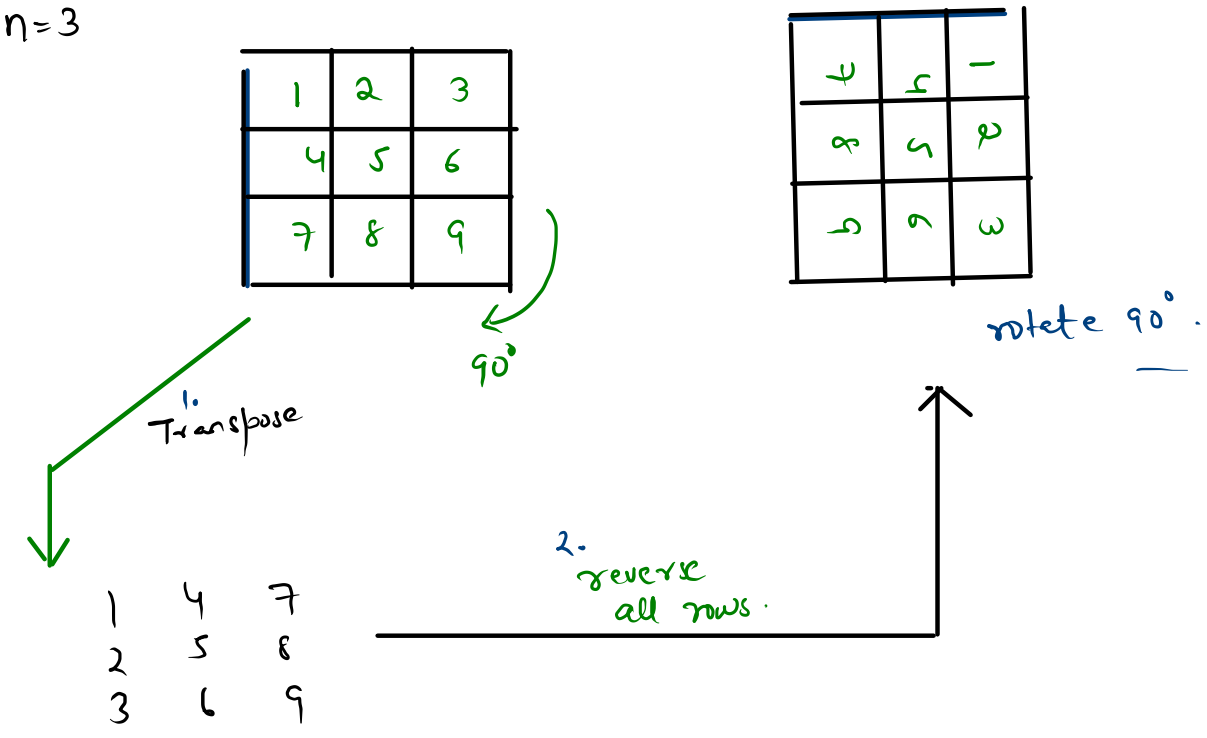
↳ same array.

Sample Input 0

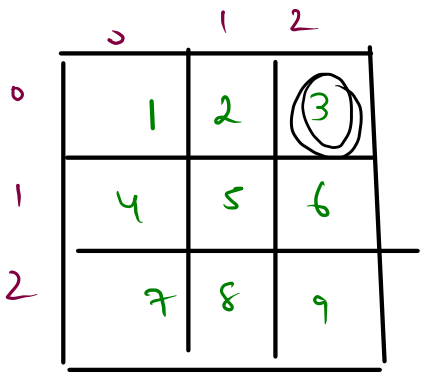
```
3
1 2 3
4 5 6
7 8 9
```

Sample Output 0

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

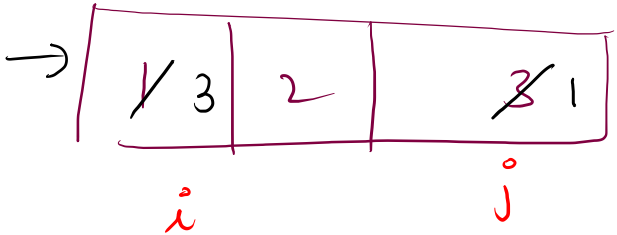


Reverse rows



row = 0

i < j



row = 0

i = 0

j = n - 1

tmp = A[row][i]

A[row][i] = A[row][j]

A[row][j] = tmp

tmp = A[0][0] = 1

A[0][0] = A[0][2]

A[0][2] = 1

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

public class Solution {
    public static void print(int [][] A){
        int n = A.length;
        for(int i = 0; i < n; i++){
            for(int j = 0; j < n; j++){
                System.out.print(A[i][j] + " ");
            }
            System.out.println();
        }
    }
    public static void transpose(int [][] A){
         int n = A.length;
        for(int i = 0; i < n; i++){
            for(int j = i+1; j < n; j++){
                int tmp = A[i][j];
                A[i][j] = A[j][i];
                A[j][i] = tmp;
            }
        }
    }
    public static void reverseRows(int [][] A){
        int n = A.length;
        for(int row = 0; row < n; row++){
            //logic to reverse single row
            int i = 0;
            int j = n-1;
            while(i < j){
                int tmp = A[row][i];
                A[row][i] = A[row][j];
                A[row][j] = tmp;
                i++;
                j--;
            }
        }
    }

    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();
            }
        }

        //1. Transpose
        transpose(A);
        //2. Reverse rows
        reverseRows(A);
        //3. print ans
        print(A);
    }
}
```

$n^2 + n^2 + n^2$

$O(n^2)$

Rotate by 180°.

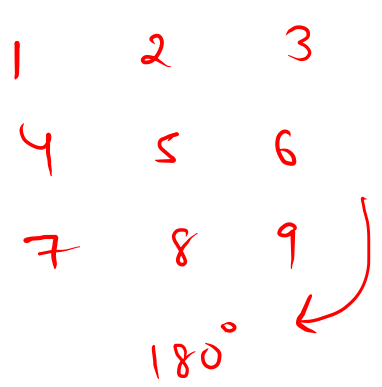
Sample Input 0

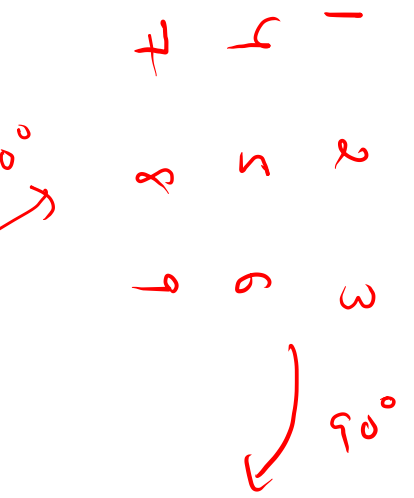
```
3
1 2 3
4 5 6
7 8 9
```

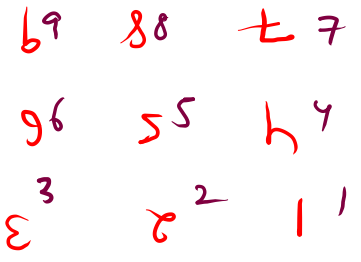
Sample Output 0

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

?







90° + 90° → 180°

Print row wise with condition

Problem | Submissions | Leaderboard | Discussions

Once upon a time, there was a programmer named Alex who was given the task of printing a **matrix row-wise**. However, there was a twist - the **even-numbered rows** had to be printed from **left to right**, and the **odd-numbered rows** had to be printed from **right to left.**

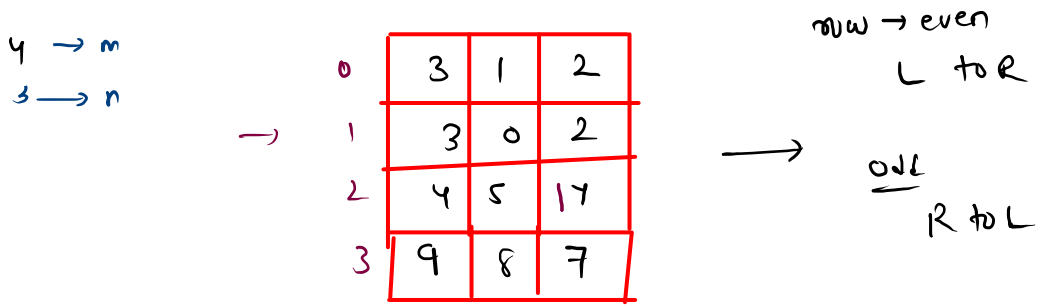
help Alex and write a program that would **iterate** through each **row** of the **matrix** and check if it was an **even or odd row.** If it was an **even row**, the program would traverse it from **left to right**, and if it was an **odd row**, the program would traverse it from **right to left.**

Sample Input 0

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

Sample Output 0

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



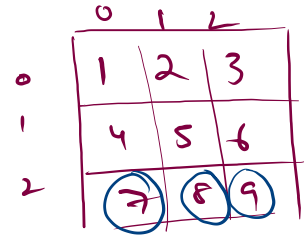
row = 0 → 3 1 2

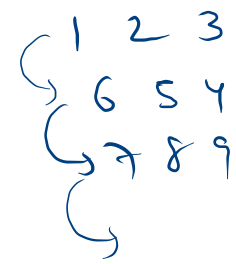
row 1 → 2 0 3

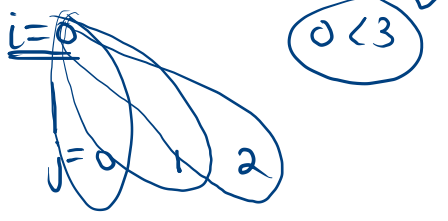
row = 2 → 4 5 14

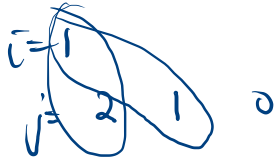
row = 3 → 7 8 9

```java
import java.io.*;
import java.util.*;
public class Solution {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int m = scn.nextInt();
        int n = scn.nextInt();
        int [][] A = new int[m][n];
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                A[i][j] = scn.nextInt();
            }
        }
        //print
        for(int i = 0; i < m; i++){
            if(i % 2 == 0){
                //L to R
                for(int j = 0; j < n; j++){
                    System.out.print(A[i][j] + " ");
                }
            }
            else{
                // R to L
                for(int j = n-1; j >= 0; j--){
                    System.out.print(A[i][j] + " ");
                }
            }
            System.out.println();
        }
    }
}
```









Convert 1-D Array to 2-D Array

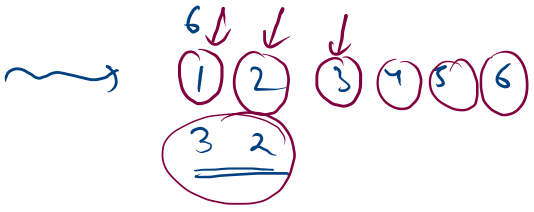
Problem | Submissions | Leaderboard | Discussions

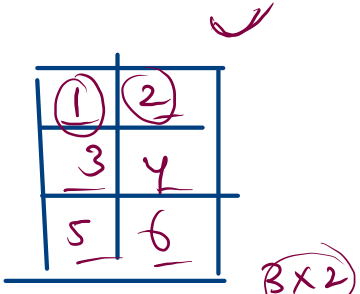
Take an **array** of size N as input, representing a 1-D array.

There are many possible factors of N, for eg:- p * q = N.

Now take p and q as input and print the 2-D array with dimensions as p*q.

Note: It is **guaranteed** that a 2-D array will be formed





1. create 2D p*q

Sample Input 0

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

Sample Output 0

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

```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 [] one = new int[n];
        for(int i = 0; i < n; i++){
            one[i] = scn.nextInt();
        }
        int p = scn.nextInt();
        int q = scn.nextInt();
        //solve
        int [][] two = new int[p][q];
        int idx = 0;
        for(int i = 0; i < p; i++){
            for(int j= 0; j < q; j++){
                two[i][j] = one[idx];
                idx++;
            }
        }
        //print
        for(int i = 0; i < p; i++){
            for(int j = 0; j < q; j++){
                System.out.print(two[i][j] + " ");
            }
             System.out.println();
        }

    }
}
```