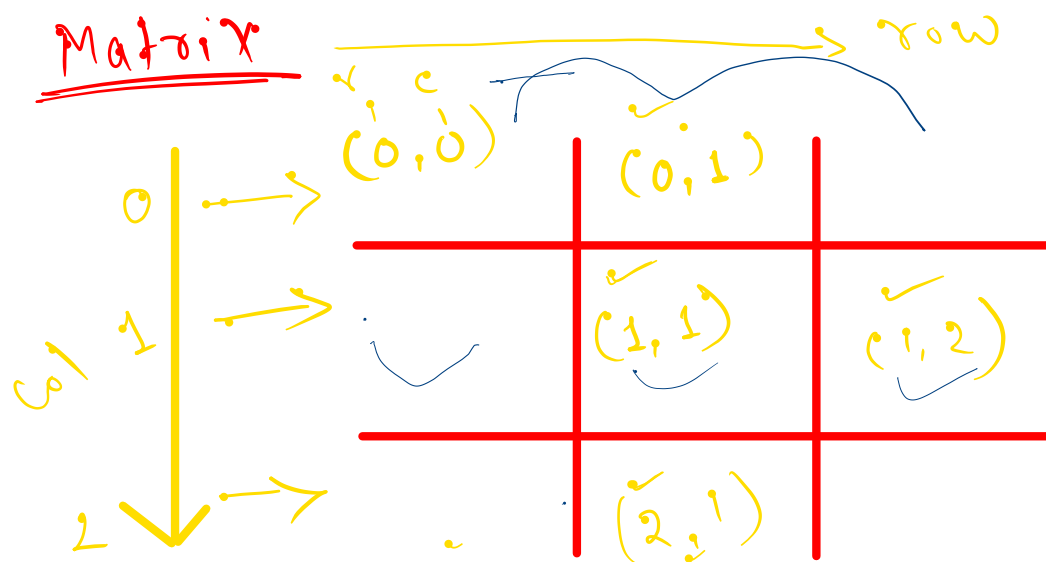
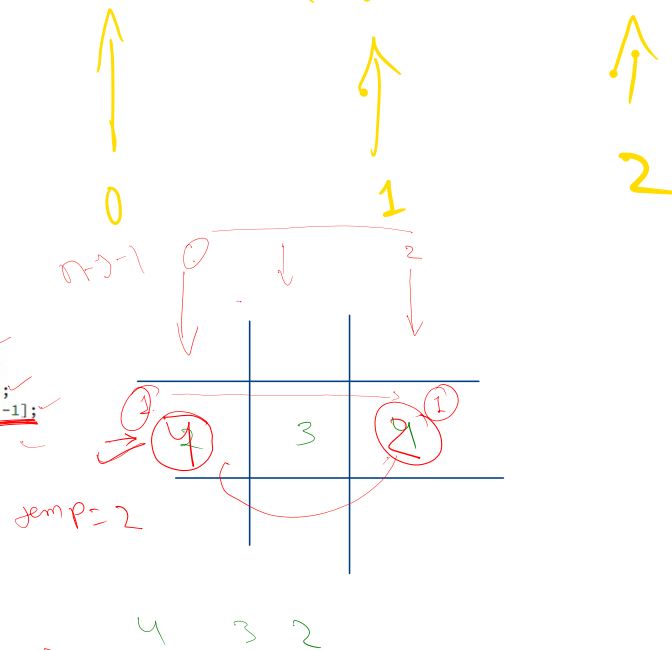


Matrix



```
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;
    }
}
```

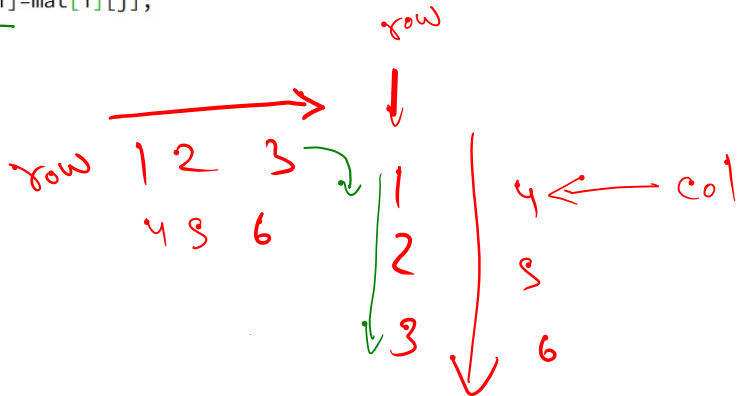


✓ $n/2$

transpose

```
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];
    }
}
```

$n \times n$
 $m \times n$
3 4



HW_Rotation Check In Matrix

Problem

Submissions

Leaderboard

Discussions

Check whether all **rows** of a **matrix** are **circular rotations** of each other.

Given a matrix of **N * N** size, the task is to find whether all **rows** are circular rotations of each other or

Input Format

First line contains integer **N** as integer input.

Second line contains **N * N** elements as Elements of matrix.

Constraints

```
1 <= N <= 1000
0 <= mat[i][j] <= 1000
```

Output Format

Return "YES" OR "NO"

Sample Input 0

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

Sample Output 0

YES

Explanation 0

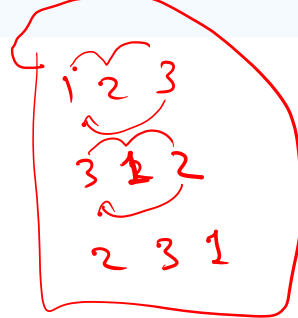
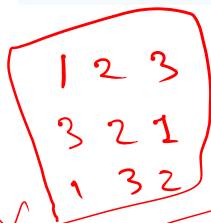
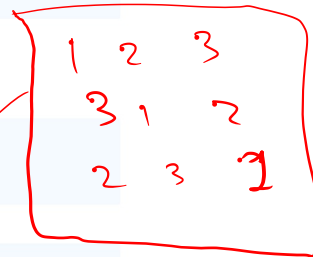
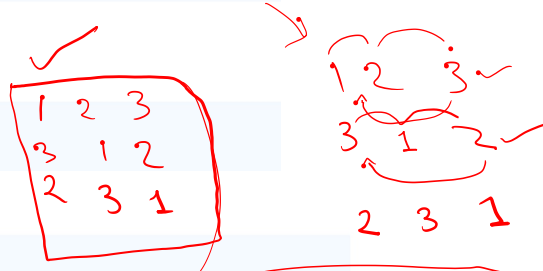
All rows are rotated permutation of each other.

Sample Input 1

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

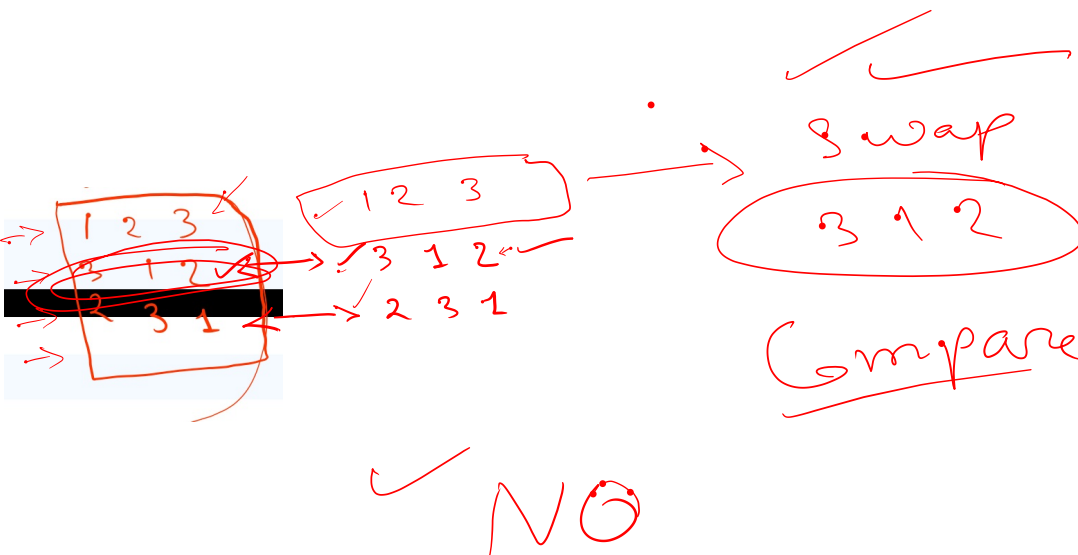
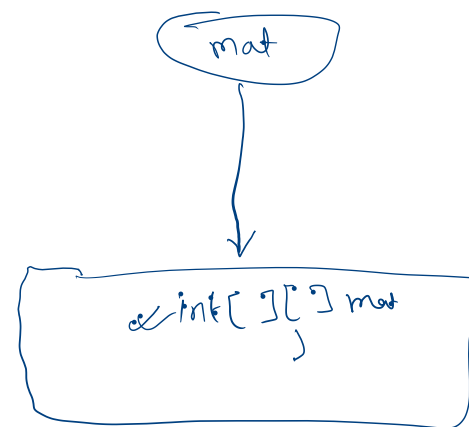
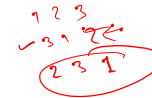
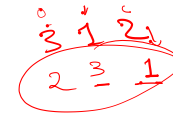
Sample Output 1

NO



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         int [][] mat = new int[n][n];
10        for(int i=0;i<n;i++){
11            for(int j=0;j<n;j++){
12                mat[i][j]=sc.nextInt();
13            }
14        }
15
16        for(int row=0;row<n;row++){
17            boolean equal = false;
18            for(int k=0;k<n;k++){
19                rotate(mat, row);
20                if(compare(mat,0,row)==true){
21                    equal = true;
22                    break;
23                }
24            }
25            if(equal == false){
26                System.out.print("NO");
27                return;
28            }
29        }
30        System.out.print("YES");
31    }
32
33    public static void rotate(int [][] mat, int row){
34        int temp = mat[row][mat[0].length-1];
35        for(int col = mat[0].length-1;col>0;col--){
36            mat[row][col] = mat[row][col-1];
37        }
38        mat[row][0] = temp;
39    }
40
41    public static boolean compare(int [][] mat, int row1, int row2){
42        for(int col = 0;col<mat[0].length;col++){
43            if(mat[row1][col]!=mat[row2][col])return false;
44        }
45        return true;
46    }
47 }
```



Compare

NO

Given a m*n matrix and you are also given an integer x. Each row and column of the matrix is sorted in increasing order. You are required to find x in the matrix and print it's location int (row, col) format as discussed in output format below. In case an element is not found, print "Not Found".

`Note :- Each row is sorted in non-decreasing order.

The first integer of each column is greater than the last integer of the previous column.`

NOTE:- After answering the question, attempt the related question in the linked resource to improve your understanding of the question . Question Link -> <https://leetcode.com/problems/search-a-2d-matrix-ii/>

Input Format

- 1. First line contains, m and n depicting the size of first matrix.
- 2. m*n Integer values, depicting all the elements of matrix.
- 3.Take a Integer input x , which is to be search.

Constraints

1 <= m and n <=1000
-1000<=mat[i][j]<=1000

Output Format

Print the row and col index in a **seperate line** if present otherwise print **"Not Found"** without quote.

Output Format

Print the row and col index in a **seperate line** if present otherwise print **"Not Found"** without quote.

Sample Input 0

2
3
4 7
5 8
8

k

matrix

123

456

789

123

456

789

Sample Output 0

1
2

123

456

789

Sample Input 1

4
5
1 4 7 11 15
2 5 8 12 19
3 6 9 16 22
10 13 14 17 24
23

k = 23

Not found

2
1

123

456

789

Sample Output 1

Not Found

123

456

789

Explanation 1

23 is not found so print "Not Found".

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 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         int k=sc.nextInt();
18
19         for(int i=0;i<row;i++){
20             for(int j=0;j<col;j++){
21                 if(mat[i][j]==k){
22                     System.out.println(i);
23                     System.out.println(j);
24                     return;
25                 }
26             }
27         }
28         System.out.print("Not Found");
29     }
30 }
```

Interchange elements of the first and last row of a matrix.

Input Format

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

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

Constraints

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

Output Format

Print the matrix after interchanging the row.

Sample Input 0

```
3
3
8 1 0
9 9 6
6 6 4
```

Sample Output 0

```
6 6 4
9 9 6
8 1 0
```

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<col;i++){
19             int first = mat[0][i];
20             mat[0][i]=mat[row-1][i];
21             mat[row-1][i]=first;
22         }
23
24         for(int i=0;i<row;i++){
25             for(int j=0;j<col;j++){
26                 System.out.print(mat[i][j]+" ");
27             }System.out.println();
28         }
29     }
30 }
31 }
```

