PRACTICE COMPETE JOBS LEADERBOARD Search lkutsarov

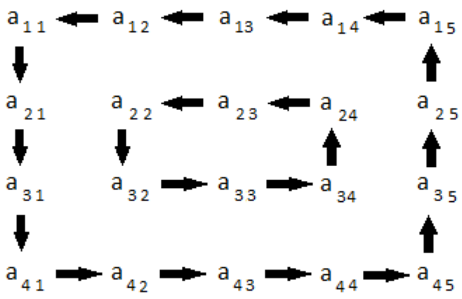
Practice > Algorithms > Implementation > Matrix Layer Rotation

Matrix Layer Rotation ☆

Problem Submissions Leaderboard Discussions Editorial

You are given a 2D matrix of dimension $m \times n$ and a positive integer r. You have to rotate the matrix r times and print the resultant matrix. Rotation should be in anti-clockwise direction.

Rotation of a $4x5$ matrix is represented by the following figure. Note that in one rotation, you have to shift elements by one step only.



Matrix Rotation

It is guaranteed that the minimum of *m* and *n* will be even.

As an example rotate the Start matrix by 2:

```
Start        First          Second
 1 2 3 4      2  3  4  5     3  4  5  6
12 1 2 5  ->  1  2  3  6 ->  2  3  4  7
11 4 3 6     12  1  4  7     1  2  1  8
10 9 8 7     11 10  9  8    12 11 10  9
```

Input Format

The first line contains three space separated integers, m, n, and r, the number of rows and columns in $matrix$, and the required rotation.
The next m lines contain n space-separated integers representing the elements of a row of $matrix$.

Constraints

$2 \leq \mathrm{m, n} \leq 300$
$1 \leq r \leq 10^9$
$min(m, n)\%2 = 0$
$1 \leq a_{ij} \leq 10^8 \text{ where } i \in [1 \ldots m] \text{ and } j \in [1 \ldots n]$

Output Format

Print each row of the rotated matrix as space-separated integers on separate lines.

Sample Input

Sample Input #01

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

Author	Heraldo
Difficulty	Hard
Max Score	80
Submitted By	20736

NEED HELP?

View discussions

View editorial

View top submissions

RATE THIS CHALLENGE

☆ ☆ ☆ ☆ ☆

MORE DETAILS

Download problem statement

Download sample test cases

Suggest Edits

```
9 10 11 12
13 14 15 16
```

Sample Output #01

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

Explanation #01

The matrix is rotated through two rotations.

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

Sample Input #02

```
5 4 7
1 2 3 4
7 8 9 10
13 14 15 16
19 20 21 22
25 26 27 28
```

Sample Output #02

```
28 27 26 25
22 9 15 19
16 8 21 13
10 14 20 7
4 3 2 1
```

Explanation 02

The various states through 7 rotations:

```
1  2  3  4     2  3  4 10    3  4 10 16    4 10 16 22
7  8  9 10     1  9 15 16    2 15 21 22    3 21 20 28
13 14 15 16 ->  7  8 21 22 -> 1  9 20 28 -> 2 15 14 27 ->
19 20 21 22    13 14 20 28    7  8 14 27    1  9  8 26
25 26 27 28    19 25 26 27    13 19 25 26   7 13 19 25

10 16 22 28    16 22 28 27    22 28 27 26    28 27 26 25
 4 20 14 27    10 14  8 26    16  8  9 25    22  9 15 19
 3 21  8 26 ->  4 20  9 25 -> 10 14 15 19 -> 16  8 21 13
 2 15  9 25     3 21 15 19     4 20 21 13    10 14 20  7
 1  7 13 19     2  1  7 13     3  2  1  7     4  3  2  1
```

Sample Input #03

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

Sample Output #03

```
1 1
1 1
```

Explanation #03

All of the elements are the same, so any rotation will repeat the same matrix.

Current Buffer (saved locally, editable)

Java 7

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    static void matrixRotation(int[][] matrix) {
        // Complete this function
    }

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int m = in.nextInt();
        int n = in.nextInt();
        int r = in.nextInt();
        int[][] matrix = new int[m][n];
        for(int matrix_i = 0; matrix_i < m; matrix_i++){
            for(int matrix_j = 0; matrix_j < n; matrix_j++){
                matrix[matrix_i][matrix_j] = in.nextInt();
            }
        }
        matrixRotation(matrix);
        in.close();
    }
}

```

Line: 1 Col: 1

Upload Code as File □ Test against custom input Run Code Submit Code