Task 10:

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;

class Result {

    /*
     * Complete the 'matrixRotation' function below.
     *
     * The function accepts following parameters:
     *  1. 2D_INTEGER_ARRAY matrix
     *  2. INTEGER r
     */

   public static void matrixRotation(List<List<Integer>> matrix, int r) {

    int m = matrix.size();
    int n = matrix.get(0).size();
    int layers = Math.min(m, n) / 2;

    for (int layer = 0; layer < layers; layer++) {

        List<Integer> elements = new ArrayList<>();

        int top = layer;
        int bottom = m - 1 - layer;
        int left = layer;
        int right = n - 1 - layer;

        // Extract elements of the current layer

        // top row
        for (int j = left; j < right; j++)
            elements.add(matrix.get(top).get(j));

        // right column
        for (int i = top; i < bottom; i++)
            elements.add(matrix.get(i).get(right));

        // bottom row
        for (int j = right; j > left; j--)
            elements.add(matrix.get(bottom).get(j));

        // left column
        for (int i = bottom; i > top; i--)
            elements.add(matrix.get(i).get(left));

        int size = elements.size();
        int rotations = r % size;

        Collections.rotate(elements, -rotations);

        int index = 0;

        // Put values back

        for (int j = left; j < right; j++)
            matrix.get(top).set(j, elements.get(index++));

        for (int i = top; i < bottom; i++)
            matrix.get(i).set(right, elements.get(index++));

        for (int j = right; j > left; j--)
            matrix.get(bottom).set(j, elements.get(index++));

        for (int i = bottom; i > top; i--)
            matrix.get(i).set(left, elements.get(index++));
    }

    // Print result
    for (List<Integer> row : matrix) {
        for (Integer val : row) {
            System.out.print(val + " ");
        }
        System.out.println();
    }
}



}

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

        String[] firstMultipleInput = bufferedReader.readLine().replaceAll("\\s+$", "").split(" ");
```

Task 10:

```
        int m = Integer.parseInt(firstMultipleInput[0]);

        int n = Integer.parseInt(firstMultipleInput[1]);

        int r = Integer.parseInt(firstMultipleInput[2]);

        List<List<Integer>> matrix = new ArrayList<>();

        IntStream.range(0, m).forEach(i -> {
            try {
                matrix.add(
                    Stream.of(bufferedReader.readLine().replaceAll("\\s+$", "").split(" "))
                        .map(Integer::parseInt)
                        .collect(toList())
                );
            } catch (IOException ex) {
                throw new RuntimeException(ex);
            }
        });

        Result.matrixRotation(matrix, r);

        bufferedReader.close();
    }
}
```