import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

public class Solution {

    public static class Cell {

        public int row;

        public int column;

        public Cell(int row, int column) {

            super();

            this.row = row;

            this.column = column;

        }

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int m = sc.nextInt();

        int n = sc.nextInt();

        int r = sc.nextInt();

        int[][] matrix = new int[m][n];

        int[][] rotatedMatrix = new int[m][n];

        for (int i = 0; i < m; i++) {

            for (int j = 0; j < n; j++) {

                matrix[i][j] = sc.nextInt();

            }

        }

        sc.close();

        int numberOfLayers = Math.min(n, m) / 2;

        for (int i = 0; i < numberOfLayers; i++) {

            int rowLowerLimit = i;

            int rowUpperLimit = m - 1 - i;

            int columnLowerLimit = i;

            int columnUpperLimit = n - 1 - i;

            int pathLength = ((n - (i \* 2)) \* 2) + ((m - (i \* 2)) \* 2) - 4;

            ArrayList<Cell> path = new ArrayList<>();

            int currentRowIndex = i;

            int currentColumnIndex = i;

            boolean down = true;

            boolean right = false;

            boolean up = false;

            boolean left = false;

            for (int j = 0; j < pathLength; j++) {

                path.add(new Cell(currentRowIndex, currentColumnIndex));

                if (currentRowIndex == rowUpperLimit && currentColumnIndex == columnLowerLimit) {

                    down = false;

                    right = true;

                }

                if (currentRowIndex == rowUpperLimit && currentColumnIndex == columnUpperLimit) {

                    right = false;

                    up = true;

                }

                if (currentRowIndex == rowLowerLimit && currentColumnIndex == columnUpperLimit) {

                    up = false;

                    left = true;

                }

                if(down){

                    currentRowIndex = currentRowIndex + 1;

                }

                if(right){

                    currentColumnIndex = currentColumnIndex + 1;

                }

                if(up){

                    currentRowIndex = currentRowIndex - 1;

                }

                if(left){

                    currentColumnIndex = currentColumnIndex - 1;

                }

            }

            for (int j = 0; j < pathLength; j++) {

                Cell currentCell = path.get(j);

                int numberToBeMoved = matrix[currentCell.row][currentCell.column];

                Cell newCell = path.get((j + r) % pathLength);

                rotatedMatrix[newCell.row][newCell.column] = numberToBeMoved;

            }

        }

        for (int i = 0; i < m; i++) {

            for (int j = 0; j < n; j++) {

                System.out.print(rotatedMatrix[i][j] + " ");

            }

            System.out.println();

        }

    }

}