```
public class Rotator {
    public static int[][] Solution(int[][] matrix, int flag) {
        if (matrix == null || matrix.length == 0 || matrix[0].length == 0)  return matrix;
        int[][] rvalue;
        rvalue = transpose(matrix);
        flip(rvalue, flag);
        return rvalue;
    }

    private static int[][] transpose(int[][] matrix) {
        int m = matrix.length, n = matrix[0].length;
        int[][] rvalue = new int[n][m];
        for (int i = 0; i < n; i++)
            for (int j = 0; j < m; j++)
                rvalue[i][j] = matrix[j][i];
        return rvalue;
    }

    private static void flip(int[][] matrix, int flag) {
        int m = matrix.length, n = matrix[0].length;
        if (flag == 1) {    //clock wise
            for (int i = 0; i < m; i++)
                for (int j = 0; j < n / 2; j++) {
                    matrix[i][j] ^= matrix[i][n-j-1];
                    matrix[i][n-j-1] ^= matrix[i][j];
                    matrix[i][j] ^= matrix[i][n-j-1];
                }
        }
        else {
            for (int i = 0; i < m / 2; i++)
                for (int j = 0; j < n; j++) {
                    matrix[i][j] ^= matrix[m-i-1][j];
                    matrix[m-i-1][j] ^= matrix[i][j];
                    matrix[i][j] ^= matrix[m-i-1][j];
                }
        }
    }
}
```