

### 1314. Matrix Block Sum

Given a  $m \times n$  matrix `mat` and an integer `k`, return a matrix `answer` where each `answer[i][j]` is the sum of all elements `mat[r][c]` for:

- $i - k \leq r \leq i + k$ ,
- $j - k \leq c \leq j + k$ , and
- $(r, c)$  is a valid position in the matrix.

#### Example 1:

**Input:** `mat = [[1,2,3],[4,5,6],[7,8,9]]`, `k = 1`

**Output:** `[[12,21,16],[27,45,33],[24,39,28]]`

#### Example 2:

**Input:** `mat = [[1,2,3],[4,5,6],[7,8,9]]`, `k = 2`

**Output:** `[[45,45,45],[45,45,45],[45,45,45]]`

#### Program:

```
class Solution {
    public int[][] matrixBlockSum(int[][] mat, int k) {
        int m = mat.length;
        int n = mat[0].length;

        // Step 1: Row prefix sum
        int[][] rowPrefix = new int[m][n + 1];

        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                rowPrefix[i][j + 1] = rowPrefix[i][j] + mat[i][j];
            }
        }

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

        // Step 2: Compute block sum
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                int r1 = Math.max(0, i - k);
                int r2 = Math.min(m - 1, i + k);
                int c1 = Math.max(0, j - k);
                int c2 = Math.min(n - 1, j + k);

                int sum = 0;

                for (int r = r1; r <= r2; r++) {
                    sum += rowPrefix[r][c2 + 1] - rowPrefix[r][c1];
                }

                result[i][j] = sum;
            }
        }

        return result;
    }
}
```