You are given two arrays rowSum and colSum of non-negative integers where rowSum[i] is the sum of the elements in the ith row and colSum[j] is the sum of the elements of the jth column of a 2D matrix. In other words, you do not know the elements of the matrix, but you do know the sums of each row and column.

Find any matrix of **non-negative** integers of size rowSum.length x colSum.length that satisfies the rowSum and colSum requirements.

Return *a 2D array representing****any****matrix that fulfills the requirements*. It's guaranteed that **at least one**matrix that fulfills the requirements exists.

**Example 1:**

**Input:** rowSum = [3,8], colSum = [4,7]

**Output:** [[3,0],

[1,7]]

**Explanation:**

0th row: 3 + 0 = 0 == rowSum[0]

1st row: 1 + 7 = 8 == rowSum[1]

0th column: 3 + 1 = 4 == colSum[0]

1st column: 0 + 7 = 7 == colSum[1]

The row and column sums match, and all matrix elements are non-negative.

Another possible matrix is: [[1,2],

[3,5]]

**Example 2:**

**Input:** rowSum = [5,7,10], colSum = [8,6,8]

**Output:** [[0,5,0],

[6,1,0],

[2,0,8]]

**Example 3:**

**Input:** rowSum = [14,9], colSum = [6,9,8]

**Output:** [[0,9,5],

[6,0,3]]

**Example 4:**

**Input:** rowSum = [1,0], colSum = [1]

**Output:** [[1],

[0]]

**Example 5:**

**Input:** rowSum = [0], colSum = [0]

**Output:** [[0]]

**Constraints:**

* 1 <= rowSum.length, colSum.length <= 500
* 0 <= rowSum[i], colSum[i] <= 108
* sum(rows) == sum(columns)