Given a rectangular matrix and integers a and b, consider the union of the *ath* row and the *bth* (both *0-based*) column of the matrix. Return sum of all elements of that union.

**Example**

For

matrix = [[1, 1, 1, 1],

[2, 2, 2, 2],

[3, 3, 3, 3]]

a = 1 and b = 3, the output should be  
crossingSum(matrix, a, b) = 12.

Here (2 + 2 + 2 + 2) + (1 + 3) = 12.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] array.array.integer matrix**

2-dimensional array of integers representing a rectangular matrix.

*Constraints:*  
1 ≤ matrix.length ≤ 5,  
1 ≤ matrix[0].length ≤ 5,  
1 ≤ matrix[i][j] ≤ 100.

* **[input] integer a**

A non-negative integer less than the number of matrix rows.

*Constraints:*  
0 ≤ a < matrix.length.

* **[input] integer b**

A non-negative integer less than the number of matrix columns.

*Constraints:*  
0 ≤ b < matrix[i].length.

* **[output] integer**

<https://codefights.com/arcade/code-arcade/list-backwoods/Nh48Nqxb2zGx2NvYK>

static int crossingSum(int[][] matrix, int a, int b)

{

int sum = 0;

for (int i = 0; i < matrix[0].Length; i++)

sum += matrix[a][i];

for (int i = 0; i < matrix.Length; i++)

sum += matrix[i][b];

//Console.WriteLine(matrix[a][b] + " ");

return sum - matrix[a][b];

}