Given a rectangular matrix of integers and integers n and m, we are looking for the [submatrix](keyword://submatrix" \t "_blank)of size n × m that has the maximal sum among all submatrices of the given size.

Example

For

matrix = [[1, 12, 11, 10],

[4, 3, 2, 9],

[5, 6, 7, 8]]

n = 2, and  
m = 1, the output should be  
maxSubmatrixSum(matrix, n, m) = 19.

Input/Output

* **[execution time limit] 3 seconds (cs)**
* **[input] array.array.integer matrix**

2-dimensional array of integers representing a rectangular matrix.

*Guaranteed constraints:*  
1 ≤ matrix.length ≤ 5,  
1 ≤ matrix[0].length ≤ 5,  
-15 ≤ matrix[i][j] ≤ 15.

* **[input] integer n**

A positive integer not greater than the number of matrix rows.

*Guaranteed constraints:*  
1 ≤ n ≤ matrix.length.

* **[input] integer m**

A positive integer not greater than the number of matrix columns.

*Guaranteed constraints:*  
1 ≤ m ≤ matrix[i].length.

* **[output] integer**
  + The sum of all elements in the desired n × m submatrix.

<https://app.codesignal.com/challenge/E9FdTL6H2qa2bBjJp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp4

{

class Program

{

static int maxSubmatrixSum(int[][] matrix, int n, int m)

{

int max\_sum = int.MinValue;

for (int i = 0; i < matrix.Length - n +1 ; i++)

{

for (int j = 0; j < matrix[i].Length - m+1 ; j++)

{

int sum = 0;

for (int k = i; k < i + n ; k++)

{

for (int l = j; l < j + m ; l++)

{

sum += matrix[k][l];

}

}

max\_sum = Math.Max(max\_sum, sum);

}

}

return max\_sum;

}

static void Main(string[] args)

{

int[][] matrix = {

new int[] {1, 12, 11, 10},

new int[] {4, 3, 2, 9},

new int[] {5, 6, 7, 8}

};

int n = 2;

int m = 1;

Console.WriteLine(maxSubmatrixSum(matrix, n, m));

Console.ReadLine();

}

}

}