Thanks to the grant given to you as the owner of the greenest farm in the county, you can finally renovate your farm's pasture! For now, you have two separate pastures for your animals to graze on, which locations can be represented by a list of 8 integers:  [x1, x2, y1, y2, a1, a2, b1, b2] .

Here, (x1, x2) and (a1, a2) are the bottom-left corners of the two pastures, and (y1, y2) and (b1, b2) are the top-right corners of the two pastures.

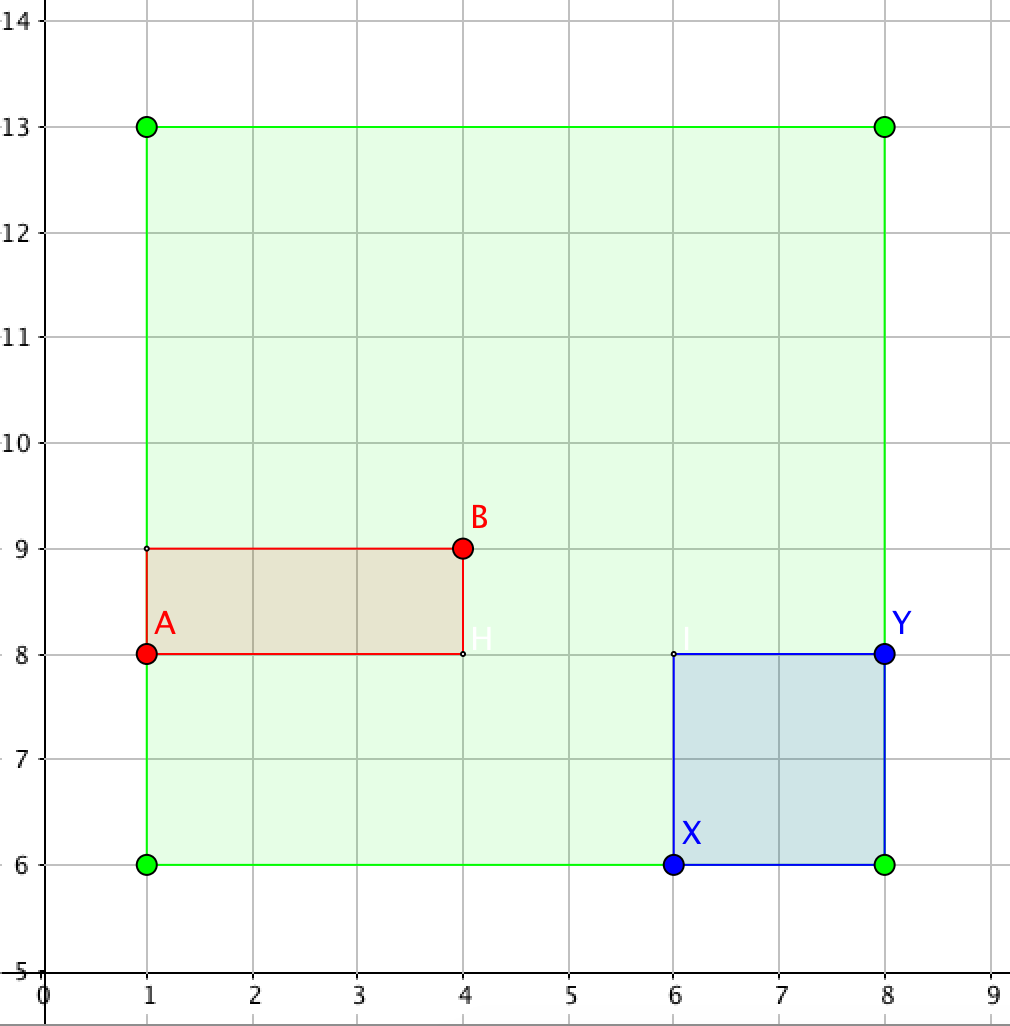
Given the vertices that represent your pastures, your task is to find the smallest **perfect square** pasture that can contain both pastures, and return its area.

**Example**

For vertices = [6, 6, 8, 8, 1, 8, 4, 9], the output should be  
combinePasture(vertices) = 49.

The first pasture is given as points X = (x1, x2) = (6, 6) and Y = (y1, y2) = (8, 8).  
The second pasture is given as points A = (a1, a2) = (1, 8) and B = (b1, b2) = (4, 9).

One of the smallest squares that contains both pastures is shown in the image below, and its area is 7 \* 7 = 49, which is the answer.



**Input/Output**

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

An array of 8 integers, where:

* + (vertices[0], vertices[1]) is the bottom left corner of the first pasture;
  + (vertices[2], vertices[3]) is the top right corner of the first pasture;
  + (vertices[4], vertices[5]) is the bottom left corner of the second pasture;
  + (vertices[6], vertices[7]) is the top right corner of the second pasture.

*Constraints:*  
vertices.length = 8,  
vertices[i] < vertices[i + 2], i ∈ {0, 1, 4, 5},  
0 ≤ vertices[i] ≤ 104.

* **[output] integer**

The area of the smallest pasture that contains two of the existing ones.

<https://codefights.com/challenge/YhSoJczcT25LtkyuY?utm_source=featuredChallenge&utm_medium=email&utm_campaign=email_notification>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int combinePasture(int[] vertices)

{

/\*

A (vertices[0], vertices[1])

B (vertices[2], vertices[3])

X (vertices[4], vertices[5])

Y (vertices[6], vertices[7])

\*/

int izq = Math.Min(vertices[0], vertices[4]);

int der = Math.Max(vertices[2], vertices[6]);

int arriba = Math.Max(vertices[3], vertices[7]);

int abajo = Math.Min(vertices[1], vertices[5]);

int lado = Math.Max(der - izq, arriba - abajo);

return lado \* lado;

}

static void Main(string[] args)

{

int[] vertices = { 6, 6, 8, 8, 1, 8, 4, 9 };

Console.WriteLine(combinePasture(vertices));

Console.ReadLine();

}

}

}