Kirby starts at the point (0,0) on the Euclidean plane. He wants to get to the point (a,b), where a and b are nonnegative integers. The only kinds of steps he can take are to step one unit in either the positive x direction, or the positive y direction. In other words, he can go from (x,y) to either (x+1,y) or (x,y+1).

Kirby is wondering how many ways there are for him to get to (a,b). But, all he wants your help for is to find whether that number is even or odd.

For example, if (a,b) = (2,1), then there are three paths Kirby can take: (0,0) -> (0,1) -> (1,1) -> (2,1) (0,0) -> (1,0) -> (1,1) -> (2,1) (0,0) -> (1,0) -> (2,0) -> (2,1) Since three is odd, you should return True.

* **[time limit] 3000ms (cs)**
* **[input] integer a**

0 <= a <= 100

* **[input] integer b**

0 <= b <= 100

* **[output] boolean**

True if the number of paths is odd, False if it's even.

<https://codefights.com/challenge/WcnbBd7e4iZvWdvmG/main>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static bool IntegerWalk(int a, int b)

{

if (a == 0 && b == 0)

{

return true;

}

if (a == 1 && b == 1)

{

return false;

}

int[,] tablero = new int[a + 1, b + 1];

for (int i = 0; i <= a; i++)

{

tablero[i, 0] = 1;

// tablero[i, i] = 1;

}

for (int i = 0; i <= b; i++)

{

tablero[0, i] = 1;

}

for (int i = 1; i <= a; i++)

{

for (int j = 1; j <= b; j++)

{

tablero[i, j] = tablero[i - 1, j] + tablero[i, j - 1];

}

}

//for (int i = 0; i <= a; i++)

//{

// for (int j = 0; j <= b; j++)

// {

// Console.Write(tablero[i, j] + " ");

// }

// Console.WriteLine();

//}

if (tablero[a, b] % 2 != 0)

{

return true;

}

return false;

}

static void Main(string[] args)

{

Console.WriteLine( IntegerWalk(12,3));

Console.ReadLine();

}

}

}