**Kangaroo**

https://d3keuzeb2crhkn.cloudfront.net/s3_pub/hr-avatars/aeda8340-d6af-4f00-9aff-16617e8ba405/150x150.png**by [wanbo](https://www.hackerrank.com/wanbo)**

* [**Problem**](https://www.hackerrank.com/challenges/kangaroo)
* [**Submissions**](https://www.hackerrank.com/challenges/kangaroo/submissions)
* [**Leaderboard**](https://www.hackerrank.com/challenges/kangaroo/leaderboard)
* [**Discussions**](https://www.hackerrank.com/challenges/kangaroo/forum)
* [**Editorial**](https://www.hackerrank.com/challenges/kangaroo/editorial)

There are two kangaroos on an x-axis ready to jump in the positive direction (i.e, toward positive infinity). The first kangaroo starts at location  and moves at a rate of  meters per jump. The second kangaroo starts at location  and moves at a rate of  meters per jump. Given the starting locations and movement rates for each kangaroo, can you determine if they'll ever land *at the same location at the same time*?

**Input Format**

A single line of four space-separated integers denoting the respective values of , , , and.

**Constraints**

**Output Format**

Print YES if they can land on the same location at the same time; otherwise, print NO.

**Note:** The two kangaroos must land at the same location *after making the same number of jumps*.

**Sample Input 0**

0 3 4 2

**Sample Output 0**

YES

**Explanation 0**

The two kangaroos jump through the following sequence of locations:

Thus, the kangaroos meet after  jumps and we print *YES*.

**Sample Input 1**

0 2 5 3

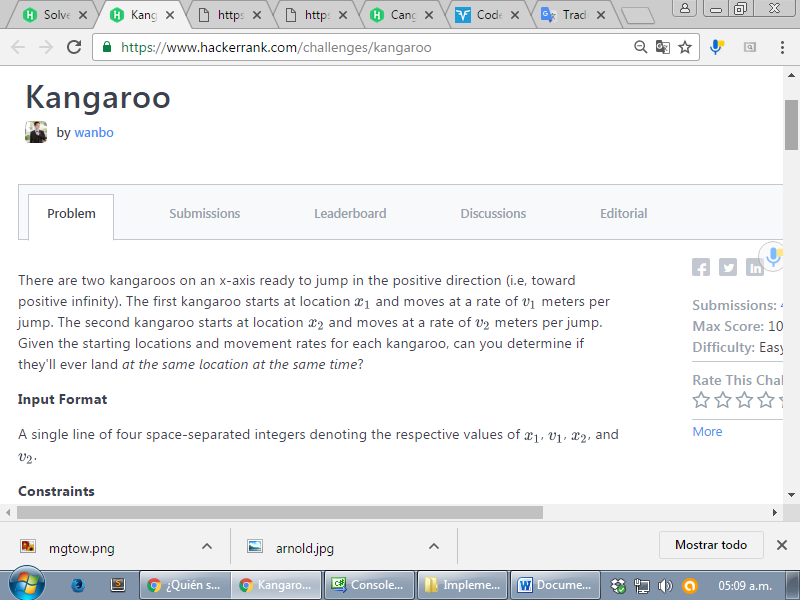
**Sample Output 1**

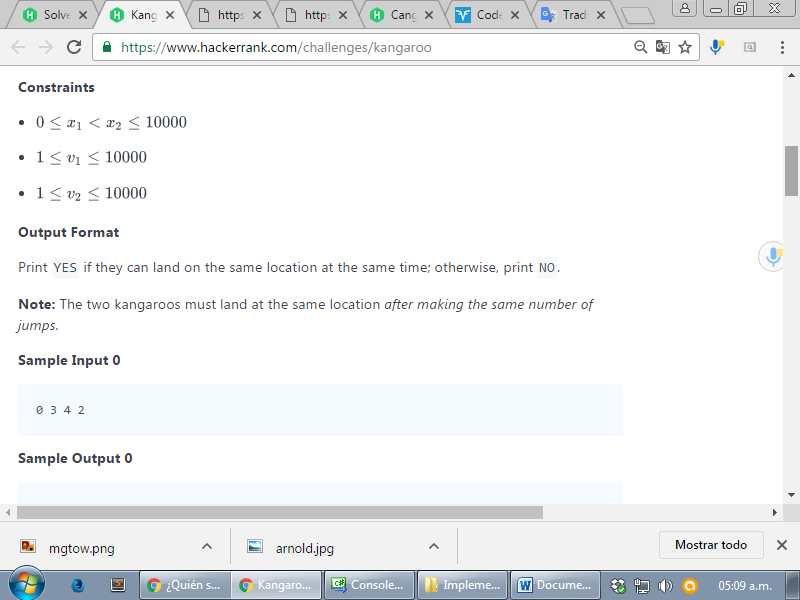
NO

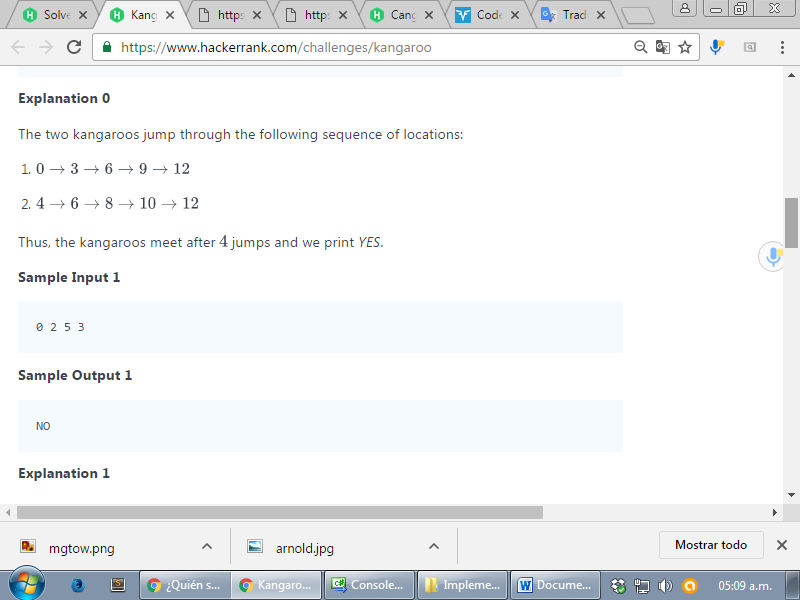
**Explanation 1**

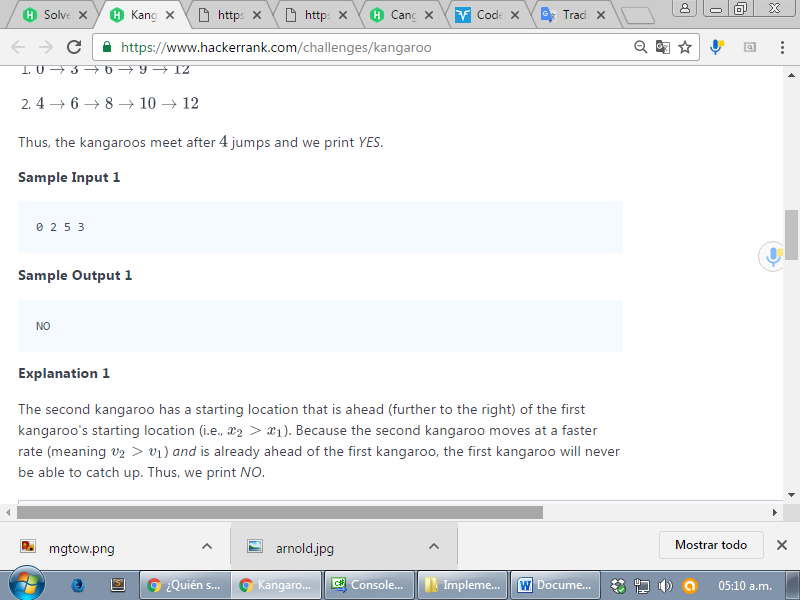
The second kangaroo has a starting location that is ahead (further to the right) of the first kangaroo's starting location (i.e., ). Because the second kangaroo moves at a faster rate (meaning ) *and* is already ahead of the first kangaroo, the first kangaroo will never be able to catch up. Thus, we print *NO*.

<https://www.hackerrank.com/challenges/kangaroo>









using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

string[] tokens\_x1 = Console.ReadLine().Split(' '); // "28 8 96 2".Split(' ');

int x1 = Convert.ToInt32(tokens\_x1[0]);

int v1 = Convert.ToInt32(tokens\_x1[1]);

int x2 = Convert.ToInt32(tokens\_x1[2]);

int v2 = Convert.ToInt32(tokens\_x1[3]);

int i = x1, j = x2; /\*les asigno a i y a j

\* las posiciones iniciales x1 y x2

\* donde empiezan los canguros\*/

int cont = 0;

int prev\_dif = 0, actual\_dif = 0;

/\*prev\_dif y actual\_dif son para saber si las diferencias

\* entre i y j aumentan, en el caso de que aumenten,

\* si con cada iteracion es mayor, entonces i y j no se

\* encuentran nunca\*/

string ans = "YES";

while (i != j )

{

if (i == j)

{

break;

}

/\* diferencia previa entre i y j antes de

\* aumentarles la velocidad v\*/

if (j > i)

{

prev\_dif = j - i;

}

else if (j < i)

{

prev\_dif = i - j;

}

//Console.WriteLine(i + " " + j);

i += v1;

j += v2;

/\* diferencia posterior entre i y j luego de asignarles la

\* velocidad v \*/

if (j > i)

{

actual\_dif = j - i;

}

else if (j < i)

{

actual\_dif = i - j;

}

/\* si la diferencia es cada vez mayor, es porque i y j

\* nunca van a ser iguales, nunca se van a encontrar,

\* si es menor se van a encontrar: en ese caso entran

\* al if i == j y sale del bucle, por lo tanto queda YES \*/

if (actual\_dif > prev\_dif)

{

ans = "NO";

break;

}

cont++;

}

Console.WriteLine(ans);

Console.ReadLine();

}

}

}