Problem: Kangaroo

There are two kangaroos on a number line 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:

1.
2.

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.

Solution:

```
int startOne, startTwo, speedOne, speedTwo;

    cin >>startOne >>speedOne
       >>startTwo >>speedTwo;

    int relative=speedOne-speedTwo;
    if(speedOne < speedTwo || relative==0)        //Kangaroos can never cross
       {cout<<"NO";}

    else if(speedOne > speedTwo)
       {
          int flag=0;
          for(int i=startOne; i<=startTwo; i+=relative)
             { (i+relative==startTwo ? flag=1 : relative+=0); }
          (flag==1 ? cout<<"YES" : cout<<"NO");

          /*Elegant Approach*/
          ((startTwo-startOne)%(speedTwo-speedOne)==0  && speedTwo!=speedOne ?
cout<<"YES" : cout<<"NO");

    }

       return 0;
}
```