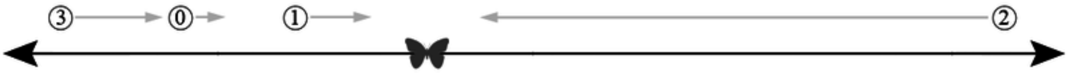
Who Gets the Catch?

locked

by kevinsogo

A wild venotoise has spawned in the middle of the street, and n *catchers* are nearby! Venotoises are quite rare and disappear quickly, so the n catchers begin racing towards it. Your task is to find who will catch the venotoise.



We represent the street as a long, straight line. The i^{th} catcher is located at position x_i along this line, and the venotoise is located at position x. The venotoise is stationary, and the i^{th} catcher is moving towards the venotoise at a speed of v_i units per second. A catcher moving at a speed of v units per second can travel a distance of d units in exactly $\frac{d}{v}$ seconds.

The first catcher that makes it to the location of the venotoise catches it. If there isn't a unique "first catcher", that is, if there are two or more catchers that initially reach the venotoise at the exact same time, then the venotoise disappears, and no one gets the catch.

Input Format

The first line contains two space-separated integers, n, the number of catchers, and x, the venotoise's location. The second line contains n space-separated integers, $x_0, x_1, \ldots, x_{n-1}$, denoting the locations of the catchers.
The third line contains n space-separated integers, $v_0, v_1, \ldots, v_{n-1}$, denoting the speeds of the catchers.

Constraints

- $1 \leq n \leq 1000$
- $1 \leq x, x_i \leq 3000$
- $1 \leq v_i \leq 60$
- Each catcher can reach the venotoise's location in an integer number of seconds.

Output Format

Print one line containing a single integer denoting the index of the catcher that catches the venotoise, or -1 if no one gets the catch.

Sample Input 0

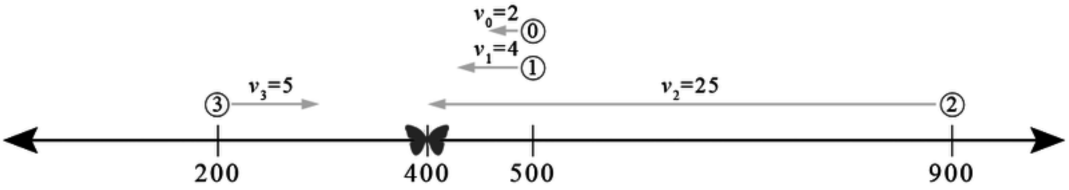
```
4 400
500 500 900 200
2 4 25 5
```

Sample Output 0

```
2
```

Explanation 0

In this example, there are $n = 4$ catchers and the venotoise is at location $x = 400$.



- Catcher 0 is at location $x_0 = 500$ and has a speed of $v_0 = 2$. It will take her $\frac{100}{2} = 50$ seconds to reach the venotoise.
- Catcher 1 is at location $x_1 = 500$ and has a speed of $v_1 = 4$. It will take him $\frac{100}{4} = 25$ seconds to reach the venotoise.
- Catcher 2 is at location $x_2 = 900$ and has a speed of $v_2 = 25$. It will take him $\frac{500}{25} = 20$ seconds to reach the venotoise.
- Catcher 3 is at location $x_3 = 200$ and has a speed of $v_3 = 5$. It will take her $\frac{200}{5} = 40$ seconds to reach the venotoise.

Thus, catcher 2 reaches the venotoise ahead of everyone else, so we print `2`.

Sample Input 1

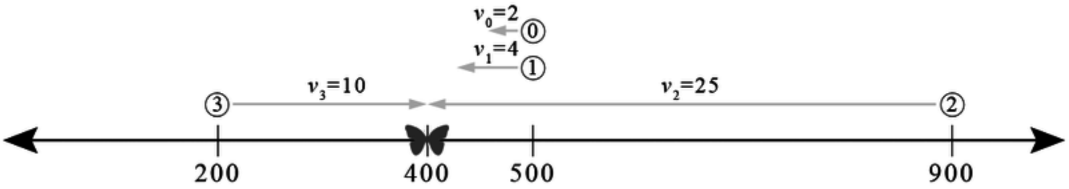
```
4 400
500 500 900 200
2 4 25 10
```

Sample Output 1

```
-1
```

Explanation 1

This is similar to the first example, except that catcher 3 now has a speed of $v_3 = 10$, so it will take her $\frac{200}{10} = 20$ seconds to reach the venotoise.



Thus, catcher 2 and catcher 3 reach the venotoise at the same time and ahead of everyone else, so the venotoise disappears and no one gets the catch. Thus, we print `-1`.

Sample Input 2

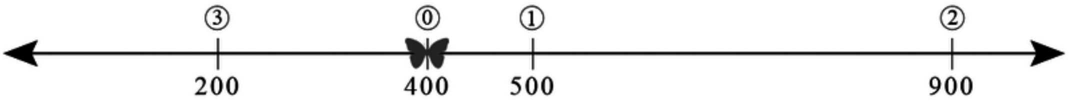
```
4 400
400 500 900 200
2 4 25 5
```

Sample Output 2

```
0
```

Explanation 2

This is similar to the first example, except that catcher 0 is now located at $x_0 = 400$, which means she's already in the same location as the venotoise! Specifically, it will take her $\frac{0}{2} = 0$ seconds to reach the venotoise.



Thus, catcher 0 reaches the venotoise ahead of everyone else, so we print `0`.

Submissions: 2563
Max Score: 20
Difficulty: Easy

Rate This Challenge:

☆☆☆☆☆

More

Current Buffer (saved locally, editable)

C#

```csharp
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
class Solution {

    static int whoGetsTheCatch(int n, int x, int[] X, int[] V){
        // Complete this function
        List<int> lista = new List<int>();
            for (int i = 0; i < n; i++)
            {
                int distancia = Math.Abs(X[i] - x);
                int segundos = distancia / V[i];

                //if (lista.Contains(segundos))
                //{
                //    return -1;
                //}
                lista.Add(segundos);
            }

            int minSeg = lista.Min();
            if (lista.IndexOf(minSeg) != lista.LastIndexOf(minSeg))
            {
                return -1;
            }

            return lista.IndexOf( minSeg);

    }

    static void Main(String[] args) {
        //  Return the index of the catcher who gets the catch, or -1 if no one gets the catch.
        string[] tokens_n = Console.ReadLine().Split(' ');
        int n = Convert.ToInt32(tokens_n[0]);
        int x = Convert.ToInt32(tokens_n[1]);
        string[] X_temp = Console.ReadLine().Split(' ');
        int[] X = Array.ConvertAll(X_temp,Int32.Parse);
        string[] V_temp = Console.ReadLine().Split(' ');
        int[] V = Array.ConvertAll(V_temp,Int32.Parse);
        int result = whoGetsTheCatch(n, x, X, V);
        Console.WriteLine(result);
    }
}

```

Line: 28 Col: 43

Upload Code as File □ Test against custom input

Run Code Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature