**Pair Sums**

Attempted by: **3016**

/

Accuracy: **81%**

/

Maximum Score: **20**

/

34 Votes

Tag(s):

Easy

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

You have been given an integer array *A* and a number *K*. Now, you need to find out whether any **two different** elements of the array *A* sum to the number *K*. Two elements are considered to be different if they lie at different positions in the array. If there exists such a pair of numbers, print "**YES**" (without quotes), else print "**NO**" without quotes.

**Input Format**:

The first line consists of two integers *N*, denoting the size of array *A* and *K*. The next line consists of *N* space separated integers denoting the elements of the array *A*.

**Output Format**:

Print the required answer on a single line.

**Constraints**:

1≤N≤106

1≤K≤2∗106

1≤A[i]≤106

**SAMPLE INPUT**

5 9

1 2 3 4 5

**SAMPLE OUTPUT**

YES

**Explanation**

Here, A[4]+A[5]=4+5=9. So, the answer is **YES**.

**Time Limit:**1.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded when all the testcases pass.

**Allowed Languages:**Bash, C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), TypeScript, Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Swift, Swift-4.1, Visual Basic

<https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/pair-sums/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static string SumanK(int[] arr, int k)

{

//return "NO";

Dictionary<int, int> hash =

new Dictionary<int, int>();

for (int i = 0; i < arr.Length; i++)

{

if (hash.ContainsKey(arr[i])) hash[arr[i]]++;

else hash[arr[i]] = 1;

}

for (int i = 0; i < arr.Length; i++)

{

if (hash.ContainsKey(k - arr[i]))

{

if (k - arr[i] == arr[i])

{

if (hash[arr[i]] > 1)

{

return "YES";

}

else

{

return "NO";

}

}

else

{

return "YES";

}

}

}

return "NO";

}

static void Main(string[] args)

{

string[] input = Console.ReadLine().Trim().Split(' ');

int N = int.Parse(input[0]);

int K = int.Parse(input[1]);

int[] arr = Array.ConvertAll(Console.ReadLine().Trim().Split(' '), e => int.Parse(e));

Console.WriteLine(SumanK(arr, K));

Console.ReadLine();

}

}

}