**Cricket Balls**

Attempted by: **603**

/

Accuracy: **31%**

/

Maximum Score: **20**

/

7 Votes

Tag(s):

Easy

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

Daksh has '**N**' boxes of cricket balls and each box has certain number of cricket balls (non-zero) in it. The boxes are numbered from **1** to **N**.

Now, Daksh has to go for practicing to cricket ground to compete in the upcoming fest **URJA**. Daksh wants to carry exactly **K** number of cricket balls to ground and he can carry only **2** boxes. Can you tell him the number of ways in which he can select **2** boxes such that total number of cricket balls in them is **K**.

**Input :**

* First line of the input is the number of test cases **T**.
* Each test case has 3 lines. First line is the number of boxes **N**.
* Next line has **N** space separated integers where the **ith**integer is the number of cricket balls in the **ith** box.
* 3rd line of each test case contains value of **K**.

**Output:**

* For each test case, output a single number, the number of ways in which Daksh can select the boxes.

**Constraints:**

* **1<=T<=100**
* **1<=N<=105**
* **1<=Number of Balls in ith box<=100**

**SAMPLE INPUT**

2

5

1 2 3 4 3

6

5

1 5 3 3 3

6

**SAMPLE OUTPUT**

2

4

**Explanation**

None

**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/cricket-balls/description/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

//static int maxCuadrado(string[] s)

//{

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

// {

// for (int j = 0; j < s[i].Length; j++)

// {

// }

// }

//}

public static int 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;

}

int cont = 0;

//foreach (KeyValuePair<int, int> kvp in hash)

//{

// if (hash.ContainsKey(k - kvp.Key))

// {

// if (kvp.Key == k - kvp.Key)

// {

// cont += (kvp.Value \* (kvp.Value - 1)) / 2;

// }

// else

// {

// cont += kvp.Value \* hash[k - kvp.Key];

// }

// }

//}

//foreach (KeyValuePair<int, int> kvp in hash)

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

{

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

{

if (hash.ContainsKey(arr[i]) && hash[k - arr[i]] > 0 && arr[i] == k - arr[i])

{

cont += (hash[arr[i]] \* (hash[arr[i]] - 1)) / 2;

//hash.Remove(arr[i]);

hash[arr[i]] = 0;

}

else

{

if ( hash.ContainsKey( arr[i]) && hash.ContainsKey( k - arr[i]) && hash[k - arr[i]] > 0 && hash[arr[i]] > 0)

{

cont += hash[arr[i]] \* hash[k - arr[i]];

hash[arr[i]] = 0;

hash[k - arr[i]] = 0;

}

//hash.Remove(arr[i]);

}

//hash.Remove(arr[i]);

}

}

return cont;

}

static void Main(string[] args)

{

int T = int.Parse(Console.ReadLine());

while (T-- > 0)

{

int N = int.Parse(Console.ReadLine());

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

int K = int.Parse(Console.ReadLine());

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

}

//string input = "1 2 3 4 3";

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

//Console.WriteLine(SumanK(arr, 6));

Console.ReadLine();

}

}

}