**Count pairs with given sum**

[array](http://www.practice.geeksforgeeks.org/tag-page.php?tag=array&isCmp=0)

Given an array of integers, and an integer  ‘K’ , find the count of pairs of elements in the array whose sum is equal to 'K'.  
  
**Input:**  
First line of the input contains an integer T, denoting the number of test cases. Then T test cases follow. Each test case consists of two lines. First line of each test case contains 2 space separated integers N and K denoting the size of array and the sum respectively. Second line of each test case contains N space separated integers denoting the elements of the array.  
  
**Output:**  
Print the count of pairs of elements in the array whose sum is equal to the K.  
  
**Constraints:**  
1<=T<=50  
1<=N<=50  
1<=K<=50  
1<=A[i]<=100  
  
**Example:  
Input**  
2  
4 6  
1  5  7 1  
4 2  
1 1 1 1  
**Output**  
2  
6

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=676>

#include <iostream>

#include <stdio.h>

using namespace std;

int main() {

int t;

scanf("%d", &t);

while(t--) {

int n,k;

scanf("%d %d", &n, &k);

int arr[n];

for(int i =0; i<n; i++) {

scanf("%d", &arr[i]);

}

int ans =0;

for(int i =0; i<n-1; i++) {

for(int j =i+1; j<n; j++) {

if(arr[i] + arr[j] == k) {

ans++;

}

}

}

cout << ans << endl;

}

// system("pause");

return 0;

}

-----------------------C# -------------------------

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

public class GFG

{

public static int twoSum(int[] nums, int target)

{

Dictionary<int, int> map = new Dictionary<int, int>();

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

{

if (map.ContainsKey(nums[i]))

{

map[nums[i]]++;

}

else

{

map[nums[i]] = 1;

}

}

int[] keys = map.Keys.ToArray();

Dictionary<int, int> marcas = new Dictionary<int, int>();

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

{

marcas[keys[i]] = 1;

}

int cont = 0;

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

{

int complement = target - keys[i];

if (map.ContainsKey(complement))

{

if (complement == keys[i])

{

if (marcas[keys[i]] == 1)

{

cont += (map[keys[i]] \* (map[keys[i]] - 1)) / 2;

marcas[keys[i]]++;

}

}

else

{

if (marcas[keys[i]] == 1 && marcas[complement] == 1)

{

cont += map[keys[i]] \* map[complement];

marcas[keys[i]]++;

marcas[complement]++;

}

}

}

}

return cont;

}

static void Main(string[] args)

{

//int[] arr = { 2, 2, 2, 2, 2 };

//Console.WriteLine(twoSum(arr, 4));

//int[] arr = Array.ConvertAll("48 24 99 51 33 39 29 83 74 72 22 46 40 51 67 37 78 76 26 28 76 25 10 65 64 47 34 88 26 49 86 73 73 36 75 5 26 4 39 99 27 12 97 67 63 15 3 92 90".Trim().Split(' '), e => int.Parse(e));

//Console.WriteLine(twoSum(arr, 50));

//int[] arr = { 1, 5, 7, 1 };

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

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

while (t-- > 0)

{

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

int n = int.Parse(nk[0]);

int k = int.Parse(nk[1]);

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

Console.WriteLine(twoSum(arr, k));

}

Console.ReadLine();

}

}