**Pair with given sum in a sorted array**

Submissions: [8907](https://practice.geeksforgeeks.org/problem_submissions.php?pid=3674)  Accuracy:

25.96%

   Difficulty: [Easy](https://practice.geeksforgeeks.org/Easy/0/0/)   Marks: 2

Show Topic Tags   

You are given an array **A** of size **N**. You need to find **all pairs** in the array that sum to a number **K**. If no such pair exists then output will be **-1**. The elements of the array are **distinct**and are in**sorted**order.  
**Note:** (a,b) and (b,a) are considered same. Also, an element cannot pair with itself, i.e., (a,a) is invalid.

**Input:**  
The first line of input is **T** denoting the number of testcase. **T** testcases follow. Each testcase contains **three**lines of input. The first line is the size of array N. The second line contains N elements separated by spaces. The third line contains the sum K.

**Output:**  
For each testcase, print all the pairs such that there sum is equal to K.

**Constraints:**  
1 <= T <= 100  
0 <= Ai <=107  
2 <= N <= 107  
0 <= K <= 107

**Examples:**  
**Input:**  
2  
7  
1 2 3 4 5 6 7  
98  
7  
1 2 3 4 5 6 7  
8  
**Output:**  
-1  
1 7 8  
2 6 8  
3 5 8

**Explanation:  
Testcase1:**We cannot find any pair that sums to 98  
**Testcase2:**We find 3 such pairs that sum to 8 (1,7) (2,6) (3,5)

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/pair-with-given-sum-in-a-sorted-array/0#ExpectOP) option \*\*

[Author: Soul\_xhacker](https://auth.geeksforgeeks.org/user/Soul_xhacker/practice/)

<https://practice.geeksforgeeks.org/problems/pair-with-given-sum-in-a-sorted-array/0>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static void MostrarPares(int[] arr, int sum)

{

HashSet<string> hs = new HashSet<string>();

//int[] indices = new int[(int)Math.Pow(10, 7) + 1];

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

//{

// indices[i] = -1;

//}

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

//{

// indices[arr[i]] = i;

//}

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

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

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

{

int resto = sum - arr[i];

if (resto >= 0)

{

if (indices.ContainsKey(resto))

{

//Console.WriteLine(arr[i] + " " + (sum - arr[i]) + " " + sum);

if (indices.ContainsKey(sum - arr[i]))

{

int min = Math.Min(arr[i], (sum - arr[i]));

int max = Math.Max(arr[i], (sum - arr[i]));

if (min != max)

{

hs.Add(min + " " + max + " " + sum);

}

}

}

}

}

if (hs.Count == 0)

{

Console.WriteLine(-1);

}

else

{

foreach (string item in hs)

{

Console.WriteLine(item);

}

}

}

static void Main(string[] args)

{

int[] arr = { 1,2,3,4,5,6,7 };

MostrarPares(arr, 8);

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

//while(T -- > 0)

//{

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

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

// int sum = int.Parse(Console.ReadLine().Trim());

// MostrarPares(arr, sum);

//}

Console.ReadLine();

}

}

}