**Hash Tables: Ice Cream Parlor**

Each time Sunny and Johnny take a trip to the Ice Cream Parlor, they pool their money to buy ice cream. On any given day, the parlor offers a line of flavors. Each flavor has a cost associated with it.

Given the value of money and the cost of each flavor for t trips to the Ice Cream Parlor, help Sunny and Johnny choose two *distinct* flavors such that they spend their entire pool of money during each visit. ID numbers are the *1- based* index number associated with a cost. For each trip to the parlor, print the ID numbers for the two types of ice cream that Sunny and Johnny purchase as two space-separated integers on a new line. You must print the smaller ID first and the larger ID second.

For example, there are n=5 flavors having const=[2,1,3,5,6]. Together they have money=5 to spend. They would purchase flavor ID's 1 and 3 for a cost of 2+3=5. Use 1 based indexing for your response.

**Note:**

* Two ice creams having unique IDs i and j *may* have the same cost (i.e., cost[i]==cost[j]).
* There will always be a unique solution.

**Function Description**

Complete the function *whatFlavors* in the editor below. It must determine the two flavors they will purchase and print them as two space-separated integers on a line.

whatFlavors has the following parameter(s):

* *cost*: an array of integers representing price for a flavor
* *money*: an integer representing the amount of money they have to spend

**Input Format**

The first line contains an integer, t, the number of trips to the ice cream parlor.

Each of the next t sets of 3 lines is as follows:

* The first line contains money.
* The second line contains an integer, n, the size of the array cost.
* The third line contains n space-separated integers denoting the cost[i].

**Constraints**

* 1<= t <=50
* 2<= money <=10^9
* 2<=n <=5\*10^4
* 1<= cost[i]<= 10^9

**Output Format**

Print two space-separated integers denoting the respective indices for the two distinct flavors they choose to purchase in ascending order. Recall that each ice cream flavor has a unique ID number in the inclusive range from 1 to |cost|.

**Sample Input**

2

4

5

1 4 5 3 2

4

4

2 2 4 3

**Sample Output**

1 4

1 2

**Explanation**

Sunny and Johnny make the following two trips to the parlor:

1. The first time, they pool together money=4 dollars. There are five flavors available that day and flavors 1 and 4 have a total cost of 1+3=4.
2. The second time, they pool together money=4 dollars. There are four flavors available that day and flavors 2 and 2 have a total cost of 2+2=4.

using System.CodeDom.Compiler;

using System.Collections.Generic;

using System.Collections;

using System.ComponentModel;

using System.Diagnostics.CodeAnalysis;

using System.Globalization;

using System.IO;

using System.Linq;

using System.Reflection;

using System.Runtime.Serialization;

using System.Text.RegularExpressions;

using System.Text;

using System;

class Solution {

// Complete the whatFlavors function below.

static void whatFlavors(int[] cost, int money) {

Hashtable lookup = new Hashtable();

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

{

if (lookup.ContainsKey(cost[i]))

{

Console.WriteLine($"{lookup[cost[i]]} {i + 1}");

return;

}

lookup.Add(money-cost[i], i+1);

}

return;

}

static void Main(string[] args) {

int t = Convert.ToInt32(Console.ReadLine());

for (int tItr = 0; tItr < t; tItr++) {

int money = Convert.ToInt32(Console.ReadLine());

int n = Convert.ToInt32(Console.ReadLine());

int[] cost = Array.ConvertAll(Console.ReadLine().Split(' '), costTemp => Convert.ToInt32(costTemp))

;

whatFlavors(cost, money);

}

}

}

**Congratulations!**

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

* **Sample Test case 0**
* **Sample Test case 1**
* **Sample Test case 2**

Input (stdin)

Download

* **2**
* **4**
* **5**
* **1 4 5 3 2**
* **4**
* **4**
* **2 2 4 3**

Your Output (stdout)

* **1 4**
* **1 2**

**Can not pass all tests**