Week7

Question1

Correct

Markedoutof 1.00

Questiontext

SunnyandJohnnyliketopooltheirmoneyandgototheicecreamparlour.Johnnyneverbuysthesame flavour that Sunny does. The only other rule they have is that they spend all of theirmoney.

Givenalistofpricesfortheflavoursoficecream,selectthetwothatwillcostallofthemoneytheyhave.

Forexample,theyhavem=6tospendandthereareflavourscostingcost=[1,2,3,4,5,6].Thetwo flavours costing 1 and 5 meet the criteria. Using 1-based indexing, they are at indices 1 and 4. 1 and 4.

FunctionDescription

Completethecodeintheeditorbelow.Itshouldreturnanarraycontainingtheindicesofthepricesofthe two flavours they buy

Ithasthefollowing:

· m: an integer denoting the amount of money they have to spend

cost:anintegerarraydenotingthecostofeachflavouroficecream

InputFormat

Thefirstlinecontainsaninteger,t,denotingthenumberoftripstotheicecreamparlor.Thenexttsetsof lines each describe a visit. Each trip is described as follows:

1. .Theintegerm,theamountofmoneytheyhavepooled.
2. Theintegern,thenumberofflavoursofferedatthetime.
3. nspace-separatedintegersdenotingthecostofeachflavour:cost[cost[1],cost[2],...,cost[n]]. The

Note: index within the cost array represents the flavour of the ice cream purchased.

Constraints

1≤ t≤ 50·

2≤ m≤ 104

2≤ n≤ 104

1≤cost[i] ≤104,"iÎ[1,n]

Therewillalwaysbeauniquesolution

.OutputFormat

Sample Input

2

4

5

14532

4

4

2243

SampleOutput

t1

4

12

Explanation

SunnyandJohnnymakethefollowingtwotripstotheparlor:

1. Thefirsttime,theypooltogetherm=4dollars.Ofthefiveflavorsavailablethatday,flavors1and4have a total cost of 1 + 3 = 4.

2. Thesecondtime,theypooltogetherm=4dollars.TOfthefourflavorsavailablethatday,flavors1and2 have a total cost of 2 + 2 = 4

```c
#include<stdio.h>
int main()
{
    int t,m,n,c=0;
    scanf("%d",&t);
    for(int i=0;i<t;i++)
    {
        c=0;
        scanf("%d\n%d",&m,&n);
        int arr[n];
        for(int j=0;j<n;j++)
        {
            scanf("%d",&arr[j]);
        }
        for(int a=0;a<n-1;a++)
        {
            for(int b=a+1;b<n;b++)
            {
                if(arr[a]+arr[b]==m)
                {
                    printf("%d %d\n",a+1,b+1);
                    c=1;break;
                }
                if(c==1)break;
            }
        }
    }
    return 0;
}
```

	Input	Expected	Got	
✓	2 4 5 1 4 5 3 2 4 4 2 2 4 3	1 4 1 2	1 4 1 2	✓

Passed all tests! ✓

Question2

CorrectMarkedoutof1.00 Question

text

NumerostheArtisthadtwolistsˆthatwerepermutationsofoneanother.Hewasveryproud. Unfortunately,whiletransportingthemfromoneexhibitiontoanother,somenumberswerelostoutofthe first list. Can you find the missing numbers?

Asanexample,thearraywithsomenumbersmissing,arr=[7,2,5,3,5,3].Theoriginalarrayofnumbers brr = [7, 2, 5, 4, 6, 3, 5, 3]. The numbers missing are [4, 6].

Notes

· Ifanumberoccursmultipletimesinthelists,youmustensurethatthefrequencyofthatnumberinboth lists is the same. If that is not the case, then it is alsoa missing number.

· Youhavetoprintallthemissingnumbersinascendingorder.

· Printeachmissingnumberonce,evenifitismissingmultipletimes.·

Thedifferencebetweenmaximumandminimumnumberinthesecondlistislessthanorequalto100.

completethecodeintheeditorbelow.Itshouldreturnanarrayofmissingnumbers. It has the following:

· arr:thearraywithmissingnumbers

· brr:theoriginalarrayofnumbers

InputFormat

Therewillbefourlinesofinput:

n-thesizeofthefirstlist,arr

Thenextlinecontainsnspace-separatedintegersarr[i

]m-thesizeofthesecondlist,brr

Thenextlinecontainsmspace-separatedintegersbrr[i] Constraints

· 1≤ n, m≤ 2x105

· n≤ m

· 1≤ brr[i]≤ 2x104

· Xmax–Xmin<101

Output Format

Outputthemissingnumbersinascending order

.SampleInput

10

203204205206207208203204205206

13

2032042042052062072052082032062052062042

SampleOutput

204 205 206

Explanation204ispresentinbotharrays.Itsfrequencyinarris2,whileitsfrequencyinbrris3.Similarly, 205and206occurtwiceinarr,butthreetimesinbrr.Therestofthenumbershavethesamefrequencies in both lists.

Answer:(penaltyregime:0%)

```c
#include<stdio.h>
int main()
{
    int n,m,c,c1=0,co;
    scanf("%d",&n);
    int arr[n];
    for(int a=0;a<n;a++)
    {
        scanf("%d",&arr[a]);
    }
    scanf("%d",&m);
    int brr[m],ans[m];
    for(int b=0;b<m;b++)
    {
        scanf("%d",&brr[b]);
    }
    for(int j=0;j<m;j++)
    {
        c=0;
        for(int i=0;i<n;i++)
        {
            if(arr[i]==brr[j])
            {
                c=1;
                arr[i]=-1;
                break;
            }
        }
        if(c==0)
        {
            ans[c1]=brr[j];
            c1++;
        }
    }
    }
    }
    for(int a=0;a<c1;a++)
    {
        co=0;
        for(int b=0;b<c1;b++)
        {
            if(ans[b]<ans[a])
            co++;
        }
        int temp=ans[a];
        ans[a]=ans[co];
        ans[co]=temp;
    }
    for(int i=0;i<c1;i++)
    printf("%d ",ans[i]);
    return 0;
}
```

	Input	Expected	Got	
✓	10 203 204 205 206 207 208 203 204 205 206 13 203 204 204 205 206 207 205 208 203 206 205 206 204	204 205 206	204 205 206	✓

Passed all tests! ✓

3,

Question text

Watson gives Sherlock an array of integers. His challenge is to find an element of the array such that the sum of all elements to the left is equal to the sum of all elements to the right. For instance, given the array ***arr = [5, 6, 8, 11]***, ***8*** is between two subarrays that sum to ***11***. If your starting array is ***[1]***, that element satisfies the rule as left and right sum to ***0***.

You will be given arrays of integers and must determine whether there is an element that meets the criterion.

Complete the code in the editor below. It should return a string, either YES if there is an element meeting the criterion or NO otherwise.

It has the following:

· arr: an array of integers 72

Input Format

The first line contains ***T***, the number of test cases.

The next ***T*** pairs of lines each represent a test case.

- The first line contains ***n***, the number of elements in the array ***arr***.

- The second line contains ***n*** space-separated integers ***arr[i]*** where ***0 ≤ i < n***.

Constraints

· ***1 ≤ T ≤ 10***

· ***1 ≤ n ≤ 10^5***

Constraints

- $1 \leq T \leq 10$
- $1 \leq n \leq 10^5$
- $1 \leq arr[i] \leq 2 \times 10^4$
- $0 \leq i \leq n$

Output Format

For each test case print YES if there exists an element in the array, such that the sum of the elements on its left is equal to the sum of the elements on its right; otherwise print NO.

Sample Input 0

```
2
3
1 2 3
4
1 2 3 3
```

Sample Output 0

```
NO
YES
```

Explanation 0

For the first test case, no such index exists.

For the second test case, ***arr[0] + arr[1] = arr[3]***, therefore index ***2*** satisfies the given conditions.

Sample Input 1


```
3
5
1 1 4 1 1
4
2 0 0 0
4
0 0 2 0
```

Sample Output 1

```
YES
YES
YES
```

Explanation 1

In the first test case, $arr[2] = 4$ is between two subarrays summing to 2.

In the second case, $arr[0] = 2$ is between two subarrays summing to 0.

In the third case, $arr[2] = 2$ is between two subarrays summing to 0.

```c
#include<stdio.h>
int main()
{
    int t,n,Is,rs,m;
    scanf("%d",&t);
    for(int i=0;i<t;i++)
    {
        Is=0;
        rs=0;
        scanf("%d",&n);
        int arr[n];
        for(int j=0;j<n;j++)
        scanf("%d",&arr[j]);
        m=n/2;
        if(arr[m]==0)
        {
            for(m=0;arr[m]==0 && m<n;m++);
        }
        for(int j=0;j<=m;j++)
        Is=Is+arr[j];
        for(int j=m;j<n;j++)
        rs=rs+arr[j];
        printf("%s\n",(Is==rs)?"YES" : "NO");
    }
    return 0;
}
```

	Input	Expected	Got	
✓	3 5 1 1 4 1 1 4 2 0 0 0 4 0 0 2 0	YES YES YES	YES YES YES	✓
✓	2 3 1 2 3 4 1 2 3 3	NO YES	NO YES	✓

Passed all tests! ✓