Week 7 – 1:
ROLL NO.:240801153
Name: Kavinkumar S
Thursday, 5 December 2024

Status	Finished
Started	Sunday, 29 December 2024, 12:22 PM
Completed	Sunday, 29 December 2024, 12:59 PM
Duration	36 mins 58 secs

Q1) Sunny and Johnny like to pool their money and go to the ice cream parlor. Johnny never buys the same flavor that Sunny does. The only other rule they have is that they spend all of their money.

Given a list of prices for the flavors of ice cream, select the two that will cost all of the money they have.

For example, they have m = 6 to spend and there are flavors costing cost = [1, 2, 3, 4, 5, 6]. The two flavors costing 1 and 5 meet the criteria. Using 1-based indexing, they are at indices 1 and 4.

Complete the code in the editor below. It should return an array containing the indices of the prices of the two flavors they buy, sorted ascending.

It has the following:

- **m: an integer denoting the amount of money they have to spend**
- **cost: an integer array denoting the cost of each flavor of ice cream**

Input Format

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

The next t sets of lines each describe a visit. Each trip is described as follows:

1. **The integer m, the amount of money they have pooled.**
2. **The integer n, the number of flavors offered at the time.**

3. n space-separated integers denoting the cost of each flavor: cost[cost[1], cost[2], .
. . , cost[n]].
Note: The index within the cost array represents the flavor of the ice cream purchased.
Constraints
· 1 ≤ t ≤ 50
· 2 ≤ m ≤ 104
· 2 ≤ n ≤ 104
· 1 ≤ cost[i] ≤ 104, i [1, n]
· There will always be a unique solution.
Output Format
For each test case, print two space-separated integers denoting the indices of the two
flavors purchased, in ascending order.
Sample Input
2
4
5
1 4 5 3 2
4
4
2 2 4 3
Sample Output 1 4
1 2

Code:

```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;
}
```

OUTPUT:

	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! ✓

Q2) Numeros the Artist had two lists that were permutations of one another. He was very proud.
Unfortunately, while transporting them from one exhibition to another, some numbers
were lost out of the first list. Can you find the missing numbers?
As an example, the array with some numbers missing, arr = [7, 2, 5, 3, 5, 3]. The original
array of numbers brr = [7, 2, 5, 4, 6, 3, 5, 3]. The numbers missing are [4, 6].
Notes
· If a number occurs multiple times in the lists, you must ensure that the frequency of
that number in both lists is the same. If that is not the case, then it is also a missing
number.
· You have to print all the missing numbers in ascending order.
· Print each missing number once, even if it is missing multiple times.
· The difference between maximum and minimum number in the second list is less than
or equal to 100.
Complete the code in the editor below. It should return a sorted array of missing numbers.
It has the following:
· arr: the array with missing numbers
· brr: the original array of numbers
Input Format
There will be four lines of input:
n - the size of the first list, arr
The next line contains n space-separated integers arr[i]
m - the size of the second list, brr
The next line contains m space-separated integers brr[i]

Constraints

1 ≤ n, m ≤ 2 x 105, n ≤ m, 1 ≤ brr[i] ≤ 2 x 104, Xmax − Xmin < 101

Output Format

Output the missing numbers in ascending order.

Sample Input

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

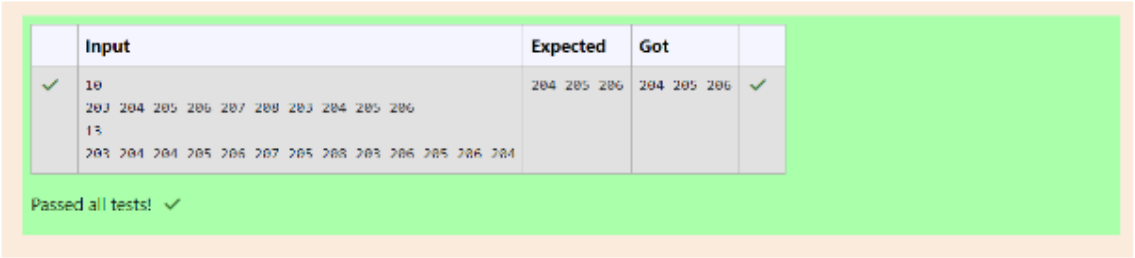
Sample Output

204 205 206

Code:

```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[n];
    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;
}
```

OUTPUT:



	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! ✓

Q3) 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

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 ≤ 105, 1 ≤ arr[i] ≤ 2 x 104, 0 ≤ i ≤ 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
```

Code:

```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;
}
```

OUTPUT:

	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! ✓