

## **Programming using C**

week07 practice session and coding

**Name:S.Jananejaipratha**

**Department:AIML-'B'**

**Roll No.:241501079**

Sam and Johnny like to spend their money and go to the store to create public identity. Johnny never buys the same thing that Sam does. The only other rule they have is that they spend all of their money.

Given a list of prices for the flowers at a store, return the list that will cost all of their money but none.

For example, they have `10` to spend and there are flowers costing `[1, 2, 3, 4, 5, 6]`. The best flowers costing `3` and `7` meet the criteria. Using `3` and `7` including they are an odd sum. `3` and `7` are the only two that work.

### Function Descriptions

Complete the cards in the table below. It should reflect an array containing the values of the given `ArrayList` array.

© 1996 Blackwell Publishers Ltd.

as an integer denoting the amount of money they have to spend  
(such as integer array denoting the price of each item of an array)

### Input: Normal

The first line contains an integer,  $k$ , denoting the number of legs in the insect genus. The next  $k$  lines of the input describe a suit. Each leg is described as follows:

1. The integers  $a_i$  are amounts of money they have paid.
2. The integers  $b_i$  are numbers of buses, offered at the time.
3.  $a_i$  are extended integers denoting the cost which bus  $i$  cost  $ans[ans[i]], ans[2], \dots, ans[n]$ .

**Note:** The index values formed along represent the Name of the investment portfolio.

### Conclusions

$T \in \Gamma(\mathbb{R})$   
 $T \in \Gamma(\mathbb{R}^2)$   
 $T \in \Gamma(\mathbb{R}^3)$   
 $T \in \text{mod}(\Gamma(\mathbb{R}^2))$   $\rightarrow \Gamma(\mathbb{R})$   
 These all always fit a unique solution

### Output Format

For each level, we print two square separated integers denoting the indices of the two factors mentioned in ascending order.

#### Example Input

3  
4  
5  
146.2  
6  
6  
1.249

Sample Output:

14  
13

**Supplements**

Survey and industry studies show following best practices for quality

1. The first team they paid together  $2n + d$  dollars. If the four teams available that day, Teams 1 and 2 have a total cost of  $3 + 4 = d$ .
2. The second team they paid together  $2n + d$  dollars. If the four teams available that day, Teams 1 and 2 have a total cost of  $2 + 2 = d$ .

Answers (usually require 2 %).

```

1 // MergeSort.java
2
3 import java.util.*;
4
5 public class MergeSort {
6     // MergeSort algorithm
7     public static void mergeSort(int[] arr) {
8         // Base case: array of size 1 or less is already sorted
9         if (arr.length <= 1)
10             return;
11
12         // Divide the array into two halves
13         int mid = arr.length / 2;
14         int[] left = new int[mid];
15         int[] right = new int[arr.length - mid];
16
17         // Copy the first half into left array
18         for (int i = 0; i < mid; i++)
19             left[i] = arr[i];
20
21         // Copy the second half into right array
22         for (int i = mid; i < arr.length; i++)
23             right[i - mid] = arr[i];
24
25         // Recursively sort the left and right arrays
26         mergeSort(left);
27         mergeSort(right);
28
29         // Merge the sorted left and right arrays back into the original array
30         merge(arr, left, right);
31     }
32
33     // Merge function
34     public static void merge(int[] arr, int[] left, int[] right) {
35         // Create a temporary array to hold the merged result
36         int[] temp = new int[arr.length];
37
38         // Merge the two sorted arrays
39         int i = 0, j = 0, k = 0;
40         while (i < left.length && j < right.length) {
41             if (left[i] <= right[j])
42                 temp[k++] = left[i++];
43             else
44                 temp[k++] = right[j++];
45         }
46
47         // Copy the remaining elements of left array
48         while (i < left.length)
49             temp[k++] = left[i++];
50
51         // Copy the remaining elements of right array
52         while (j < right.length)
53             temp[k++] = right[j++];
54
55         // Copy the merged array back into the original array
56         for (int i = 0; i < temp.length; i++)
57             arr[i] = temp[i];
58     }
59 }

```

	Input	Expected	Got	
10	10	10.00	10.00	10
20	20	20.00	20.00	20
30	30	30.00	30.00	30
40	40	40.00	40.00	40
50	50	50.00	50.00	50

## Activate Windows

Go to Settings to activate Windows.

264 is present in both words. Its frequency in *avr* is 2, while its frequency in *ber* is 1. Similarly, 265 and 266 occur twice in *avr*, but three times in *ber*. The rest of the numbers have the same frequencies in both lists.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,s,c,cl=0,co;
5     scanf("%d",&n);
6     int arr[n];
7     for(int a=0;a<n;a++){
8         scanf("%d",&arr[a]);
9     }
10    scanf("%d",&m);
11    int brr[m],ans[n];
12    for(int b=0;b<m;b++){
13        scanf("%d",&brr[b]);
14    }
15    for(int j=0;j<m;j++){
16        {
17            c=0;
18            for(int i=0;i<n;i++){
19                if(arr[i]==brr[j]){
20                    c++;
21                    arr[i]=-1;
22                    break;
23                }
24            }
25            if(c==0){
26                ans[cl]=brr[j];
27                cl++;
28            }
29        }
30        for(int a=0;a<cl;a++){
31            co=0;
32            for(int b=0;b<n;b++){
33                if(ans[a]==arr[b]){
34                    co++;
35                }
36            }
37            int temp=ans[a];
38            ans[a]=ans[co];
39            ans[co]=temp;
40        }
41        for (int i=0;i<cl;i++)
42            printf("%d ",ans[i]);
43    }
44 }
45 }
```

	Input	Expected	Got	
✓	18 203 204 205 206 207 208 203 204 205 206 13 203 204 204 205 206 207 205 208 203 206 205 206 204	204 205 206	204 205 206	✓

Passed all tests! ✓

Activate Windows  
Go to Settings to activate Windows.

Problem 3

Points

Standard cost of 1.00

0 / 1000 points

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, 4, 2]`, `4` 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  $0 \leq i < n$ .

Constraints

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

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
```

Explanation 0

For the first test case, no such index exists.

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

Sample Input 1

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

Sample Output 1

```
YES
YES
YES
```

Activate Windows  
Go to Settings to activate Windows.

```

1 #include <stdio.h>
2 int main()
3 {
4     int T,n,i,left_sum,right_sum;
5     scanf("%d",&T);
6     while(T--)
7     {
8         scanf("%d",&n);
9         int arr[n];
10        for(i=0;i<n;i++)
11        {
12            scanf("%d",&arr[i]);
13        }
14        left_sum=0;
15        right_sum=0;
16        for(i=0;i<n;i++)
17        {
18            right_sum+=arr[i];
19        }
20        for(i=0;i<n;i++)
21        {
22            right_sum-=arr[i];
23            if(left_sum==right_sum)
24            {
25                printf("YES\n");
26                break;
27            }
28            left_sum+=arr[i];
29        }
30        if(left_sum!=right_sum)
31        {
32            printf("NO\n");
33        }
34    }
35    return 0;
36 }
37

```

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

Passed all tests! ✓

Activate Windows