

Week-07-Searching Algorithms-Linear and Binary

Dashboard / My courses / GE23131-PUC-2024 / Week-07-Searching Algorithms-Linear and Binary

Navigation

✓ Dashboard

Site home

Site pages

✓ My courses

✓ GE23131-PUC-2024

Participants

Competencies

Grades

General

Skill Test-01-MCQ & Coding

Lecture Notes

Week-01-Overview of C, Constants, Variables and Da...

Assessment-01-Overview of C, Constants, Variables ...

Week-02-Operators and Expressions, Managing

◀ Week-06-01-One-Dimensional Arrays

▶ Assessment-07-Searching Algorithms-Linear and Binary

Week-07-01-Practice Session-Coding

✓ Done

◀ Week-06-01-One-Dimensional Arrays

Jump to...

▶ Assessment-07-Searching Algorithms-Linear and Binary



GE23131-Programming Using C-2024

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Thursday, 19 December 2024, 9:37 AM
Duration	4 days 7 hours

Question 1

Correct

Marked out of
5.00

Flag question

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 $\text{cost} = [1, 2, 3, 4, 5, 6]$. The two flavors costing **1** and **5** meet the criteria. Using **1-based indexing**, they are at indices **1** and **4**.

Function Description

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

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

REC-CIS

GE23131-Programming Using C-2024

Navigation

Dashboard

[Site home](#)[Site pages](#)

My courses

GE23131-PUC-2024

[Participants](#) [Competencies](#) [Grades](#)[General](#)[Skill Test-01-MCQ & Coding](#)[Lecture Notes](#)[Week-01-Overview of C, Constants, Variables and Da...](#)[Assessment-01-Overview of C, Constants, Variables ...](#)[Week-02-Operators and Expressions, Managing Input ...](#)[Assessment-02- Operators and Expressions, Managing](#)

Attempts allowed: 2

This quiz has been configured so that students may only attempt it using the Safe Exam Browser.

Time limit: 2 hours

Grading method: Highest grade

Your attempts

Attempt 1

Status Finished

Started Monday, 23 December 2024, 5:33 PM

Completed Thursday, 19 December 2024, 9:37 AM

Duration 4 days 7 hours

[Review](#)

The Safe Exam Browser keys could not be validated. Check that you're using Safe Exam Browser with the correct configuration file.

[Download Safe Exam Browser](#)[Launch Safe Exam Browser](#)[Download configuration](#)

Type here to search



USD/INR +0.32%
12-01-2025

03:54
ENG

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 $m = 4$ dollars. Of the five flavors available that day, flavors **1** and **4** have a total cost of $1 + 3 = 4$.
2. The second time, they pool together $m = 4$ dollars. Of the four flavors available that day, flavors **1** and **2** have a total cost of $2 + 2 = 4$.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int t,m,n,c=0;
5     scanf("%d",&t);
6     for(int i=0;i<t;i++)
7     {
8         c=0;
9         scanf("%d\n%d",&m,&n);
10        int arr[n];
11        for(int j=0;j<n;j++)
12        {
13            scanf("%d",&arr[j]);
14        }
15        for(int a=0;a< n-1;a++)
16        {
17            for(int b=a+1;b<n;b++)
18            {
19                if(arr[a]+arr[b]==m)
20                {
21                    printf("%d %d\n",a+1,b+1);
22                    c=1;break;
23                }
24            }
25            if(c==1)break;
26        }
27    }
28    return 0;
29 }
```

	Input	Expected	Got	
✓	2	1 4	1 4	✓
	4	1 2	1 2	

REC-CIS

```
27  
28     return 0;  
29 }
```

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

Passed all tests! ✓

Question 2

Correct

Marked out of
5.00[Flag question](#)

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.



Type here to search



USD/INR +0.32%



03:54

12-01-2025

5.00

[Flag question](#)

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 an 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]**



Type here to search



USD/INR +0.32%



03:54

12-01-2025

REC-CIS

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

- $1 \leq n, m \leq 2 \times 10^5$
- $n \leq m$
- $1 \leq brr[i] \leq 2 \times 10^4$
- $X_{max} - X_{min} < 101$

Output Format

Output the missing numbers in ascending order.

Sample Input

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
```

Explanation

204 is present in both arrays. Its frequency in **arr** is **2**, while its frequency in **brr** is **3**. Similarly, **205** and **206** occur twice in **arr**, but three times in **brr**. 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 m,n,c,c1=0,co;  
5     scanf("%d",&n);  
6     int arr[n];  
7     for(int a=0;a<n;a++)  
8     {  
9         scanf("%d",&arr[a]);
```

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int m,n,c,c1=0,co;
5     scanf("%d",&n);
6     int arr[n];
7     for(int a=0;a<n;a++)
8     {
9         scanf("%d",&arr[a]);
10    }
11    scanf("%d",&m);
12    int brr[m],ans[m];
13    for(int b=0;b<m;b++)
14    {
15        scanf("%d",&brr[b]);
16    }
17    for(int j=0;j<m;j++)
18    {
19        c=0;
20        for(int i=0;i<n;i++)
21        {
22            if(arr[i]==brr[j])
23            {
24                c=1;
25                arr[i]=-1;
26                break;
27            }
28        }
29    }
30    if(c==0)
31    {
32        ans[c1]=brr[j];
33        c1++;
34    }
35 }
36 for(int a=0;a<c1;a++)
37 {
```

```
26
27
28     }
29 }
30 if(c==0)
31 {
32     ans[c1]=brr[j];
33     c1++;
34 }
35 }
36 for(int a=0;a<c1;a++)
37 {
38     co=0;
39     for(int b=0;b<c1;b++)
40     {
41         if(ans[b]<ans[a])
42             co++;
43     }
44     int temp=ans[a];
45     ans[a]=ans[co];
46     ans[co]=temp;
47 }
48 for(int i=0;i<c1;i++)
49 printf("%d ",ans[i]);
50 return 0;
51 }
```

	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! ✅

	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! ✓

Question 3

Correct

Marked out of
5.00

Flag question

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



Type here to search



USD/INR +0.32%

ENG 03:54
12-01-2025

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

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

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
```

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, $\text{arr}[0] + \text{arr}[1] = \text{arr}[3]$, therefore index 2 satisfies the given conditions.

REC-CIS

Sample Output 0

NO

YES

Explanation 0

For the first test case, no such index exists.

For the second test case, $\text{arr}[0] + \text{arr}[1] = \text{arr}[3]$, 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

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***.

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**.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int t,n,Is,rs,m;
5     scanf("%d",&t);
6     for(int i=0;i<t;i++)
7     {
8         Is=0;
9         rs=0;
10        scanf("%d",&n);
11        int arr[n];
12        for(int j=0;j<n;j++)
13            scanf("%d",&arr[j]);
14        m=n/2;
15        if(arr[m]==0)
16        {
17            for(m=0;arr[m]==0&& m<n;m++);
18        }
19        for(int j=0;j<=m;j++)
20            Is=Is+arr[j];
21        for(int j=m;j<n;j++)
22            rs=rs+arr[j];
23        printf("%s\n", (Is==rs)?"YES":"NO");
24    }
25    return 0;
26 }
```

REC-CIS

```
19     for(int j=0;j<=m;j++)
20         Is=Is+arr[j];
21         for(int j=m;j<n;j++)
22             rs=rs+arr[j];
23             printf("%s\n", (Is==rs)? "YES": "NO");
24     }
25     return 0;
26 }
```

	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! ✓

Finish review

Week-07-Searching Algorithms-Linear and Binary

Dashboard / My courses / GE23131-PUC-2024 / Week-07-Searching Algorithms-Linear and Binary

Navigation

✓ Dashboard

Site home

Site pages

✓ My courses

✓ GE23131-PUC-2024

Participants

Competencies

Grades

General

Skill Test-01-MCQ & Coding

Lecture Notes

Week-01-Overview of C, Constants, Variables and Da...

Assessment-01-Overview of C, Constants, Variables ...

Week-02-Operators and Expressions, Managing Input ...

◀ Week-06-01-One-Dimensional Arrays

▶ Assessment-07-Searching Algorithms-Linear and Binary

Week-07-01-Practice Session-Coding

✓ Done

◀ Week-06-01-One-Dimensional Arrays

Jump to...

▶ Assessment-07-Searching Algorithms-Linear and Binary