

GE23131-Programming Using C-2024

Quiz navigation



Show all questions on one page

Finish review

Status	Finished
Started	Wednesday, 15 January 2025, 4:47 PM
Completed	Wednesday, 15 January 2025, 4:59 PM
Duration	12 mins 5 secs

Question 1
Correct
Flag question

Given an array of numbers, find the index of the smallest array element (the pivot), for which the sums of all elements to the left and to the right are equal. The array may not be reordered.

Example

```
arr=[1,2,3,4,6]
```

- the sum of the first three elements, $1+2+3=6$. The value of the last element is 6.
- Using zero based indexing, $arr[3]=4$ is the pivot between the two subarrays.
- The index of the pivot is 3.

Function Description

Complete the function `balancedSum` in the editor below.

`balancedSum` has the following parameter(s):

`int arr[n]`: an array of integers

Returns:

`int`: an integer representing the index of the pivot

Constraints

- $3 \leq n \leq 10^5$
- $1 \leq arr[i] \leq 2 \times 10^4$, where $0 \leq i < n$
- It is guaranteed that a solution always exists.

Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

Activate Windows
Go to Settings to activate Windows.

Input from user will be processed as follows and passed to the function.

The first line contains an integer n , the size of the array `arr`.

Each of the next n lines contains an integer `arr[i]`, where $0 \leq i < n$.

Sample Case 0

Sample Input 0

STDIN Function Parameters

4 → `arr[]` size $n = 4$

1 → `arr = [1, 2, 3, 3]`

2

3

3

Sample Output 0

2

Explanation 0

- The sum of the first two elements, $1+2=3$. The value of the last element is 3.
- Using zero based indexing, `arr[2]=3` is the pivot between the two subarrays.
- The index of the pivot is 2.

Sample Case 1

Sample Input 1

STDIN Function Parameters

3 → `arr[]` size $n = 3$

1 → `arr = [1, 2, 1]`

2

1

Sample Output 1

1

Activate Windows
Go to Settings to activate Windows.

Sample Output 1

1

Explanation 1

- The first and last elements are equal to 1.
- Using zero based indexing, arr[1]=2 is the pivot between the two subarrays.
- The index of the pivot is 1.

Answer: (penalty regime: 0 %)

Reset answer

```
1 /*
2  * Complete the 'balancedSum' function below.
3  *
4  * The function is expected to return an INTEGER.
5  * The function accepts INTEGER_ARRAY arr as parameter.
6  */
7
8 int balancedSum(int arr_count, int* arr)
9 {
10     int n=arr_count;
11     int p[n];
12     p[0]=arr[0];
13     for(int i=0;i<n;i++)
14     {
15         p[i]=p[i-1]+arr[i];
16     }
17     for(int i=1;i<n-1;i++)
18     {
19         if(p[i]==p[n-1]-p[i])
20         {
21             return i;
22         }
23     }
24     return 1;
25 }
26
```

	Test	Expected	Got	
✓	int arr[] = {1,2,3,3}; printf("%d", balancedSum(4, arr));	2	2	✓

Passed all tests! ✓

GE23131-Programming Using C-2024

Quiz navigation



Show all questions on one page

Finish review

Question **2**
Correct
Flag question

Calculate the sum of an array of integers.

Example

numbers = [3, 13, 4, 11, 9]

The sum is $3 + 13 + 4 + 11 + 9 = 40$.

Function Description

Complete the function arraySum in the editor below.

arraySum has the following parameter(s):

int numbers[n]: an array of integers

Returns

int: integer sum of the numbers array

Constraints

$1 \leq n \leq 10^4$

$1 \leq \text{numbers}[i] \leq 10^4$

Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer n , the size of the array numbers.

Each of the next n lines contains an integer numbers[i] where $0 \leq i < n$.

Sample Case 0

Sample Input 0

Activate Windows
Go to Settings to activate Windows.

```
STDIN  Function
-----
5  →  numbers[] size n = 5
1  →  numbers = [1, 2, 3, 4, 5]
2
3
4
5
```

Sample Output 0

15

Explanation 0

$1 + 2 + 3 + 4 + 5 = 15$.

Sample Case 1

Sample Input 1

```
STDIN  Function
-----
2  →  numbers[] size n = 2
12 →  numbers = [12, 12]
12
```

Sample Output 1

24

Explanation 1

$12 + 12 = 24$.

Answer: (penalty regime: 0 %)

Reset answer

```
1 // *
2 * Complete the 'arraySum' function below.
3 *
```

Activate Windows
Go to Settings to activate Windows.

2 → numbers[] size n = 2
12 → numbers = [12, 12]
12

Sample Output 1

24

Explanation 1

12 + 12 = 24.

Answer: (penalty regime: 0 %)

Reset answer

```
1. /*
2.  * Complete the 'arraysum' function below.
3.  *
4.  * The function is expected to return an INTEGER.
5.  * The function accepts INTEGER_ARRAY numbers as parameter.
6.  */
7.
8. int arraySum(int numbers_count, int *numbers)
9. {
10.     int sum=0;
11.     for(int i=0;i<numbers_count;i++)
12.     {
13.         sum+=numbers[i];
14.     }
15.     return sum;
16. }
17.
```

Test	Expected	Got	
✓ int arr[] = {1,2,3,4,5}; printf("%d", arraySum(5, arr))	15	15	✓

Passed all tests! ✓

Activate Windows
Go to Settings to activate Windows.

← Previous page

Next page →

Quiz navigation



Show all questions on one page

Finish review

Question 3
Correct
Flag question

Given an array of n integers, rearrange them so that the sum of the absolute differences of all adjacent elements is minimized. Then, compute the sum of those absolute differences. Example $n = 5$ $arr = [1, 3, 3, 2, 4]$ If the list is rearranged as $arr' = [1, 2, 3, 3, 4]$, the absolute differences are $|1 - 2| = 1, |2 - 3| = 1, |3 - 3| = 0, |3 - 4| = 1$. The sum of those differences is $1 + 1 + 0 + 1 = 3$. Function Description Complete the function `minDiff` in the editor below. `minDiff` has the following parameter: `arr` an integer array Returns: `int`: the sum of the absolute differences of adjacent elements Constraints $2 \leq n \leq 105$ $0 \leq arr[i] \leq 109$, where $0 \leq i < n$ Input Format For Custom Testing The first line of input contains an integer, n , the size of `arr`. Each of the following n lines contains an integer that describes `arr[i]` (where $0 \leq i < n$). Sample Case 0 Sample Input For Custom Testing STDIN Function ----- `5` `arr[]` size $n = 5 \rightarrow arr[] = [5, 1, 3, 7, 3] 1 3 7 3 Sample Output 6 Explanation $n = 5$ $arr = [5, 1, 3, 7, 3]$ If `arr` is rearranged as `arr' = [1, 3, 3, 5, 7]`, the differences are minimized. The final answer is $|1 - 3| + |3 - 3| + |3 - 5| + |5 - 7| = 6$. Sample Case 1 Sample Input For Custom Testing STDIN Function ----- `2` `arr[]` size $n = 2 \rightarrow arr[] = [3, 2] 2 Sample Output 1 Explanation $n = 2$ $arr = [3, 2]$ There is no need to rearrange because there are only two elements. The final answer is $|3 - 2| = 1$.$$

Answer: (penalty regime: 0 %)

Reset answer

```
1 //  
2 * Complete the 'minDiff' function below.  
3 *  
4 * The function is expected to return an INTEGER.  
5 * The function accepts INTEGER_ARRAY arr as parameter.  
6 */  
7  
8 int minDiff(int arr_count, int* arr)  
9 {  
10     int n=arr_count;  
11     for(int i=0;i<n;i++)  
12     {  
13         for(int j=i+1;j<n;j++)  
14         {  
15             if(arr[j]>arr[i])  
16             {  
17                 int temp=arr[j];  
18                 arr[j]=arr[i+1];  
19                 arr[i+1]=temp;  
20             }  
21         }  
22     }  
23     int diff=0;  
24     for(int i=0;i<n-1;i++)  
25     {  
26         diff+=arr[i+1]-arr[i];  
27     }  
28     return diff;  
29 }
```

Test	Expected	Got
✓ int arr[] = {5, 1, 3, 7, 3}; printf("%d", minDiff(5, arr))	6	6 ✓

Passed all tests! ✓

Previous page

Activate Windows

Go to Settings to activate Windows.

Finish review