REC-CIS

GE23131-Programming Using C-2024

Maanisha E 240901055

|  |  |
| --- | --- |
| **Status** | Finished |
| **Started** | Sunday, 12 January 2025, 7:36 PM |
| **Completed** | Sunday, 12 January 2025, 7:49 PM |
| **Duration** | 13 mins 34 secs |

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.



Question **1**

Correct

Flag question

Quiz navigation

1

2

3

[Show one page at a time](http://www.rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=151875&cmid=406&showall=0)

Finish review

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.

REC-CIS

balancedSum has the following parameter(s):

int arr[n]: an array of integers

4 → arr[] size n = 4

1 → arr = [1, 2, 3, 3]

STDIN Function Parameters

Sample Case 0

Sample Input 0

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 ≤ i < n.

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

Input Format for Custom Testing

3 ≤ n ≤ 105

1 ≤ arr[i] ≤ 2 × 104, where 0 ≤ i < n

It is guaranteed that a solution always exists.

·

·

·

Constraints

Returns:

int: an integer representing the index of the pivot

REC-CIS

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

REC-CIS

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 totalsum = 0; | |
| 11 | for(int i=0;i<arr\_count;i++) | |
| 12 ▼ | { | |
| 13 | totalsum += arr[i]; | |
| 14 | } | |
| 15 | int leftsum =0; | |
| 16 | for(int i=0;i<arr\_count;i++) | |
| 17 ▼ | { | |
| 18 | int rightsum = totalsum - leftsum -arr[i]; | |
| 19 | if(leftsum==rightsum) | |
| 20 ▼ | { | |
| 21 | return i; | |
| 22 | } | |
| 23 | leftsum +=arr[i]; | |
| 24 | } | |
| 25 | return 1; | |
| 26 | } | |

REC-CIS

27



Passed all tests!

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

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



Question **2**

Correct

Flag question

REC-CIS

int: integer sum of the numbers array

numbers[] size n = 5

numbers = [1, 2, 3, 4, 5]

→

→

5

1

2

3

4

5

STDIN Function

Sample Case 0

Sample Input 0

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 ≤ i < n.

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

Input Format for Custom Testing

1 ≤ n ≤ 104

1 ≤ numbers[i] ≤ 104

Constraints

REC-CIS

Sample Output 0

**Answer:** (penalty regime: 0 %)

12 + 12 = 24.

Explanation 1

24

Sample Output 1

numbers[] size n = 2

numbers = [12, 12]

→

→

2

12

12

STDIN Function

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

Sample Case 1

Sample Input 1

Explanation 0

15

REC-CIS

1 ▼

Reset answer

2

3

4

5

6

7

8

9 ▼

10

11

12 ▼

13

14

15

16

17

/\*

* Complete the 'arraySum' function below.

\*

* The function is expected to return an INTEGER.
* The function accepts INTEGER\_ARRAY numbers as parameter.

\*/

int arraySum(int numbers\_count, int \*numbers)

{

int sum =0;

for(int i =0;i<numbers\_count;i++)

{

sum = sum+numbers[i];

}

return sum;

}



Passed all tests!

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.

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

REC-CIS

Flag question

integer array Returns: int: the sum of the absolute differences of adjacent elements Constraints 2 ≤ n ≤105 0 ≤ arr[i] ≤ 109, where 0 ≤ 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 ≤ i < n) . Sample Case 0 Sample Input For Custom Testing STDIN Function 5 → arr[] size n =

5 5 → 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 3 → arr[] = [3, 2] 2 Sample

Question **3** The sum of those differences is 1 + 1 + 0 + 1 = 3. Function Description Complete the

Correct function minDiff in the editor below. minDiff has the following parameter: arr: an

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  3  4  5  6  7  8  9 ▼  10  11  12  13 ▼  14  15  16  17 ▼  18  19  20  21  22 | * Complete the 'minDiff' function below.   \*   * The function is expected to return an INTEGER. * The function accepts INTEGER\_ARRAY arr as parameter.   \*/  #include<stdio.h>  int compare(const void\*a,const void\*b)  {  return(\*(int\*)a-\*(int\*)b);  }  int minDiff(int arr\_count,int\*arr)  {  qsort(arr,arr\_count,sizeof(int),compare); int totaldiff=0;  for(int i =1;i<arr\_count;i++)  {  totaldiff += abs(arr[i]-arr[i-1]);  }  return totaldiff;  } | |
|  | | |





REC-CIS



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

Passed all tests!

Finish review