GE23131-Programming Using C-2024

Started Thursday, 16 January

Status Finished

2025, 3:22 PM

Completed Thursday, 16 January

2025, 3:32 PM

Duration 9 mins 45 secs

Ouestion 1 Correct

P Flag question

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.

Given an array of numbers, find the index of

Example

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

1+2+3=6. The value of the last element is 6. Using zero based indexing, arr[3]=4 is

the sum of the first three elements,

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: an integer representing the index of the

int arr[n]: an array of integers

Constraints

Returns:

1 ≤ arr[i] ≤ 2 × 104, where 0 ≤ i < n

 $3 \le n \le 10^5$

exists.

Input Format for Custom Testing

Input from stdin will be processed as follows

It is guaranteed that a solution always

and passed to the function.

the array arr. Each of the next n lines contains an integer,

The first line contains an integer n, the size of

Sample Case 0 Sample Input 0

arr[i], where $0 \le i < n$.

STDIN Function Parameters

→ arr = [1, 2, 3, 3]

→ arr[] size n = 4

3

2

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.



Explanation 1

- The first and last elements are equal to1.
- 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

```
* Complete the 'balancedSum'
 3
     *
     * The function is expected t
     * The function accepts INTEG
 5
 6
     */
 7
    int balancedSum(int arr_count
 8
 9 ▼ | {
10
       int left=0,right=0;
11 •
       for(int i=0;i<arr_count;i++</pre>
           right+=arr[i];
12
13
       for(int i=0;i<arr_count;i++</pre>
14 •
           if(left==(right-arr[i])
15
           return i;
16
17
           left+=arr[i];
           right-=arr[i];
18
19
20
       return 1;
21
    }
22
```

```
Test

int arr[] = {1,2,3,3};
printf("%d", balancedSum(4, arr
```

Passed all tests! <

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

 $1 \le n \le 10^4$

 $1 \le numbers[i] \le 10^4$

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 $0 \le i < n$.

Sample Case 0

Sample Input 0

STDIN Function

→ numbers[] size n = 5 5

→ numbers = [1, 2, 3, 4, 5] 1

2

3

4

5

Sample Output 0

Explanation 1

```
12 + 12 = 24.
```

Answer: (penalty regime: 0 %)

Reset answer

```
* Complete the 'arraySum' fu
3
     *
4
     * The function is expected t
5
     * The function accepts INTEG
 6
     */
 8
    int arraySum(int numbers_coun
       int sum=0;
11 •
       for(int i=0;i<numbers_coun</pre>
12
            sum+=numbers[i];
13
14
       return sum;
15
16
```

```
Test

Int arr[] = {1,2,3,4,5};
printf("%d", arraySum(5, arr))
```

Passed all tests! <

Question 3

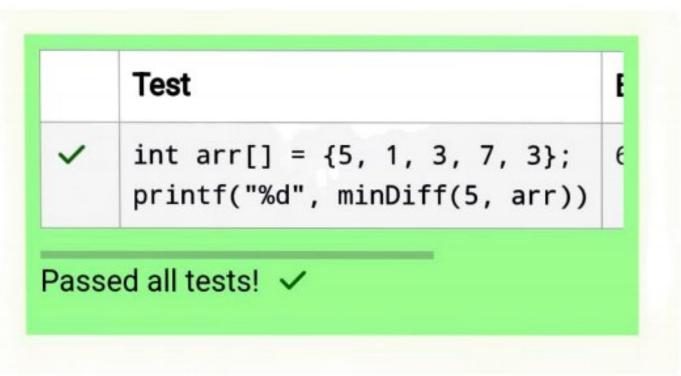
Correct

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 ≤ 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 Output 1 Explanation n = 2 arr = [3, 2] There is no need to rearrange because there are only two elements. The

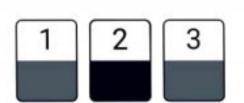
final answer is |3 - 2| = 1.

```
Answer: (penalty regime: 0 %)
 Reset answer
    1
    2
         Complete the 'minDiff' funct
    3
       * The function is expected to
    4
       * The function accepts INTEGEF
    5
    6
    7
    8
       nt minDiff(int arr_count, int
    9 •
   10 •
         for(int i=0;i<arr_count-1;i+</pre>
   11 •
              for(int j=0;j<arr_count-</pre>
   12 •
                   if(arr[j]>arr[j+1]){
   13
                       int temp=arr[j];
   14
                       arr[j]=arr[j+1];
   15
                       arr[j+1]=temp;
   16
                  }
   17
              }
   18
   19
         int sum=0;
         for(int i=0;i<arr_count-1;i+</pre>
   20 •
   21
              sum+=abs(arr[i]-arr[i+1]
   22
   23
         return sum;
   24
   25
```



Finish review

Quiz navigation



Show one page at a time

Finish review