C-2024

Attempts allowed: 4

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

Time limit: 1 hour 30 mins

Grading method: Highest grade

Your attempts

Attempt 1

Status	Finished
Started	Thursday, 16 January 2025, 3:33 PM
Completed	Thursday, 16 January 2025, 3:57 PM
Duration	23 mins 31 secs

Review

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

Launch Safe Exam Browser

Download configuration

GE23131-Programming Using C-2024

Status	Finished
Started	Thursday, 16 January 2025, 3:33 PM
Completed	Thursday, 16 January 2025, 3:57 PM
Duration	23 mins 31 secs

Question **1**

Incorrect

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

- 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

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

Input Format for Custom Testing

always exists.

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 arr.

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

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

```
-----    ------------------
3      → arr[] size n = 3
1      → arr = [1, 2, 1]
2
1
```

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'
3   *
4   * The function is expected t
5   * The function accepts INTEG
6   */
7
8  int balancedSum(int arrc, int
9  {
10      int totalsum=0;
11      for(int i=0;arrc;i++)
```

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

```
omplete the 'balancedSum' func

he function is expected to ret
he function accepts INTEGER_AR


balancedSum(int arrc, int* arr

int totalsum=0;
for(int i=0;arrc;i++)
{
    totalsum+=arr[i];
}
int leftsum=0;
for(int i=0;i<arrc;i++)
{
    int rightsum=totalsum-left
    if(leftsum==rightsum)
    {
        return i;
    }
    leftsum+=arr[i];
}
return 1;
```

```
 9
10
11  i++)
12
13  [i];
14
15
16  ;i++)
17
18  totalsum-leftsum-arr[i];
19  ightsum)
20
21
22
23  i];
24
25
26
27
```

Got	
Run error* egmentation fault (core dumped)	✗

Testing was aborted due to error.
Your code must pass all tests to earn any marks. Try again.

Show differences

Question **2**

Correct

Flag question

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

$1 \leqslant n \leqslant 10^4$

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

Sample Case 0

Sample Input 0

```
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]
```

Sample Output 1

24

Explanation 1

12 + 12 = 24.

Answer: (penalty regime: 0 %)

Reset answer

```
/*
 * Complete the 'arraySum' fu
 *
 * The function is expected t
 * The function accepts INTEG
 */

int arraySum(int numbers_coun
{
    int sum=0;
    for(int i=0;i<numbers_cou
        sum=sum+numbers[i];
    }
    return sum;
}

```

Reset answer

```
1
2  ction below.
3
4   return an INTEGER.
5  R_ARRAY numbers as parameter.
6
7
8  , int *numbers)
9
10
11 t;i++){
12
13
14
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

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' fun
3  *
4  * The function is expected t
5  * The function accepts INTEG
6  */
7 #include <stdio.h>
```

because there are only two elements. The final answer is |3 - 2| = 1.

Answer: (penalty regime: 0 %)

Reset answer

```
/*
 * Complete the 'minDiff' fun
 *
 * The function is expected t
 * The function accepts INTEG
 */
#include <stdio.h>
int compare(const void*a,cons
    return(*(int*)a-*(int*)b)
}
int minDiff(int arr_count, in
{
    qsort(arr,arr_count,sizeo
    int totaldiff=0;
    for(int i=1;i<arr_count;i
        totaldiff+=abs(arr[i]
    }
    return totaldiff;
}
```

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

Passed all tests! ✓

Finish review

```
function below.

d to return an INTEGER.
TEGER_ARRAY arr as parameter.


onst void*b){
)b);

 int* arr)

zeof(int),compare);

t;i++){
[i]-arr[i-1]);
```

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

Passed all tests! ✓

Finish review

Quiz navigation

1 2 3