

Week-13-Passing arrays and strings to functions



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

Source code

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

Result

	Test	Expected	Got	
~	<pre>int arr[] = {1,2,3,3}; printf("%d", balancedSum(4, arr))</pre>	2	2	~
Passe	d 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.

Source code

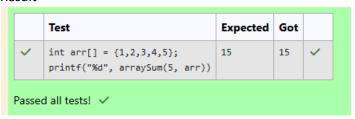
```
st Complete the 'arraySum' function below.
2
   *

* The function is expected to return an INTEGER.

* The function accepts INTEGER_ARRAY numbers as parameter.

*/
3
4
5
6
8 int arraySum(int numbers_count, int *numbers)
9 + {
10
       int sum = 0;
       for(int i=0; i<numbers_count; i++){</pre>
11 v
12
        sum=sum+numbers[i];
13
       return sum;
14
15
16 }
17
```

Result



Question **3**Correct

F Flag question

Answer: (penalty regime: 0 %)

```
Reset answer
```

```
* Complete the 'minDiff' function below.
2
3
     ^{st} The function is expected to return an INTEGER.
     \ensuremath{^{*}} The function accepts <code>INTEGER_ARRAY</code> arr as parameter.
6
 7
    #include <stdlib.h>
 8 | int compare(const void *a,const void *b){
9
       return(*(int*)a - *(int*)b);
10
11
    int minDiff(int arr_count, int* arr)
        qsort(arr, arr_count,sizeof(int), compare);
13
14
        int totaldiff=0;
15
        for(int i=1; i<arr_count; i++){</pre>
         totaldiff += abs(arr[i]-arr[i-1]);
16
17
18
        return totaldiff;
19
20
21
22
```

	Test	Expected	Got	
~	int arr[] = {5, 1, 3, 7, 3}; printf("%d", minDiff(5, arr))	6	6	~
	ed all tests! 🗸			