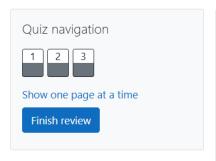
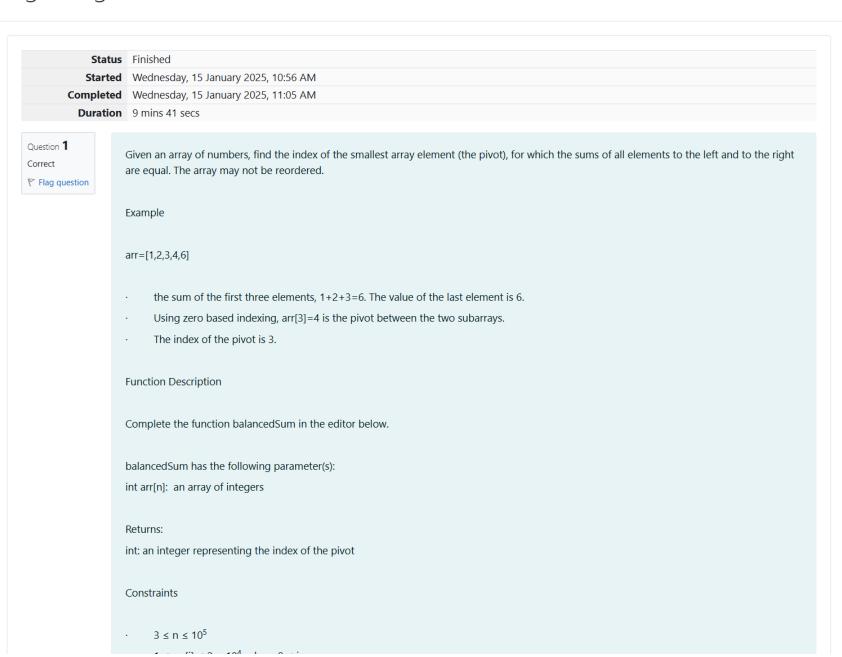
GE23131-Programming Using C-2024





T ≤ arr[i] ≤ ∠ × TUT, where U ≤ 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. 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 $0 \le i < n$. Sample Case 0 Sample Input 0 STDIN Function Parameters 4 \rightarrow arr[] size n = 4 1 \rightarrow 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
```

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

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

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

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

 $1 \le n \le 10^4$

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

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
5 \rightarrow numbers[] size n = 5
1 \rightarrow 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 \rightarrow numbers[] size n = 2
12 → numbers = [12, 12]
12
Sample Output 1
24
Explanation 1
```

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

Answer: (penalty regime: 0 %)

```
Reset answer
```

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

	Test Expecte	d Go	t
~	int arr[] = {1,2,3,4,5}; printf("%d", arraySum(5, arr))	15	~
	all tests! ✓		

Question **3**Correct

Flag question

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
```

```
2
      * Complete the 'minDiff' function below.
 3
 4
     \ensuremath{^{*}} The function is expected to return an INTEGER.
 5
     * The function accepts INTEGER_ARRAY arr as parameter.
 6
 7 ,
     int abs(int x){
        return x<0?-x:x;
 8
 9
     void bubblesort(int arr[],int n){
10
        for(int i=0;i<n-1;i++){
   for(int j=0;j<n-i-1;j++){</pre>
11
12
13 1
                 if(arr[j]>arr[j+1]){
14
                     int temp=arr[j];
                      arr[j]=arr[j+1];
15
16
                      arr[j+1]=temp;
17
18
19
20
    int minDiff(int arr_count, int* arr)
21
22
23
        bubblesort(arr,arr_count);
24
         int sum=0;
25
         for(int i=1;i<arr_count;i++){</pre>
26
             sum+=abs(arr[i]-arr[i-1]);
27
28
        return sum;
29
30
```

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

Passed all tests! <

Finish review