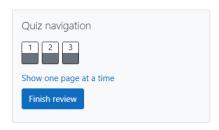
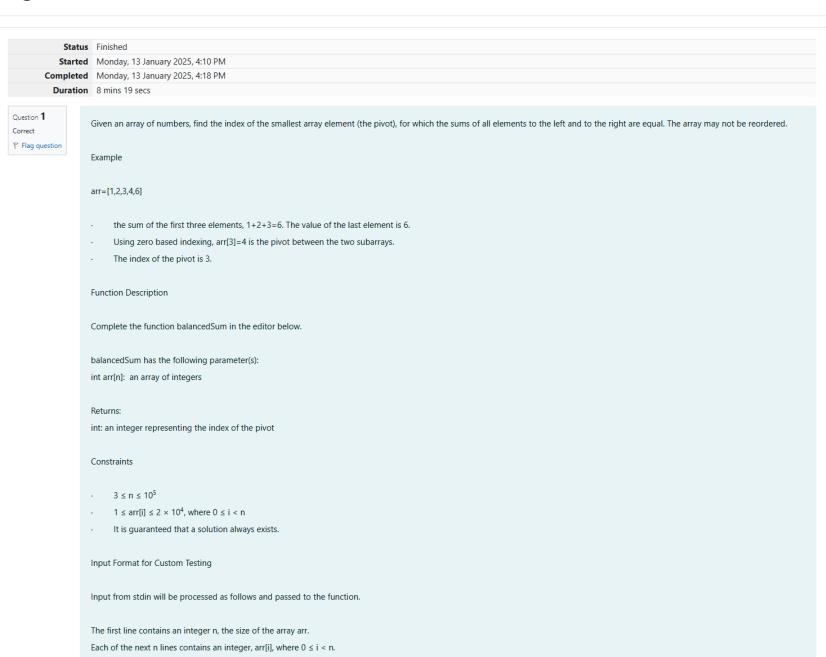
## GE23131-Programming Using C-2024





```
Sample Case 0
Sample Input 0
STDIN Function Parameters
4 \rightarrow arr[] size n = 4
1 \rightarrow arr = [1, 2, 3, 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 \rightarrow arr[] size n = 3
1 \rightarrow arr = [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 %)
```

```
* Complete the 'balancedSum' function below.
3
4
     \ensuremath{^{*}} The function is expected to return an INTEGER.
     * The function accepts INTEGER_ARRAY arr as parameter.
 6
 7
 8
    int balancedSum(int arr_count, int* arr)
 9 ,
10
    int totalsum=0;
    for(int i=0;i<arr_count;i++)
11
12 totalsum+=arr[i];
13 int lefts=0;
14 * for(int i=0;i<arr_count;i++){
int rights=totalsum-lefts-arr[i];
     if(lefts==rights)
16
17
       return i;
18
       lefts+=arr[i];
19
20
21
    return 1;
22
23
```

## Question 2 Correct

♥ Flag question

Example

numbers = [3, 13, 4, 11, 9]

The sum is 3 + 13 + 4 + 11 + 9 = 40.

Calculate the sum of an array of integers.

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

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

## Answer: (penalty regime: 0 %)

```
Reset answer
```

```
2
     * Complete the 'arraySum' function below.
3
4
     * The function is expected to return an INTEGER.
5
     * The function accepts INTEGER_ARRAY numbers as parameter.
 6
 8
    int arraySum(int numbers_count, int *numbers)
9 ,
10
     int i,sum=0;
    for(i=0;i<numbers count;i++)</pre>
    sum=sum+numbers[i];
12
13
    return sum;
14
15
```

	Test	Expected	Got	
<b>~</b>	<pre>int arr[] = {1,2,3,4,5}; printf("%d", arraySum(5, arr))</pre>	15	15	<b>~</b>

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  $2 \le n \le 105$   $0 \le arr[i] \le 109$ , where  $0 \le i < n$  Input Format For Custom Testing The first line of input contains an integer, n, the size of arr. Each of the following  $n \ge 100$  in lines contains an integer that describes  $n \ge 100$  in Sample Case  $n \ge 100$  Sample Input For Custom Testing STDIN Function  $n \ge 100$  arr  $n \ge 100$  ar

Answer: (penalty regime: 0 %)

## Reset answer

```
1 * /*
2     * Complete the 'minDiff' function below.
3     *
4     * The function is expected to return an INTEGER.
5     * The function accepts INTEGER_ARRAY arr as parameter.
6     */
7     int minDiff(int arr_count, int* arr)
9     *
10 * for(int i=0;i<arr_count-1;i++){</pre>
```

```
if(arr[j]>arr[j+1]){
    int temp=arr[j];
    arr[j]=arr[j+1];
    arr[j+1]=temp;
}

int sum=0;
    for(int i=0;i<arr_count-1;i++){
    sum+=abs(arr[i+1]-arr[i]);
}
return sum;
}
</pre>
```

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

Passed all tests! <

Finish review