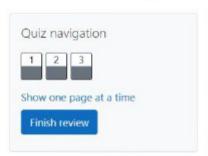
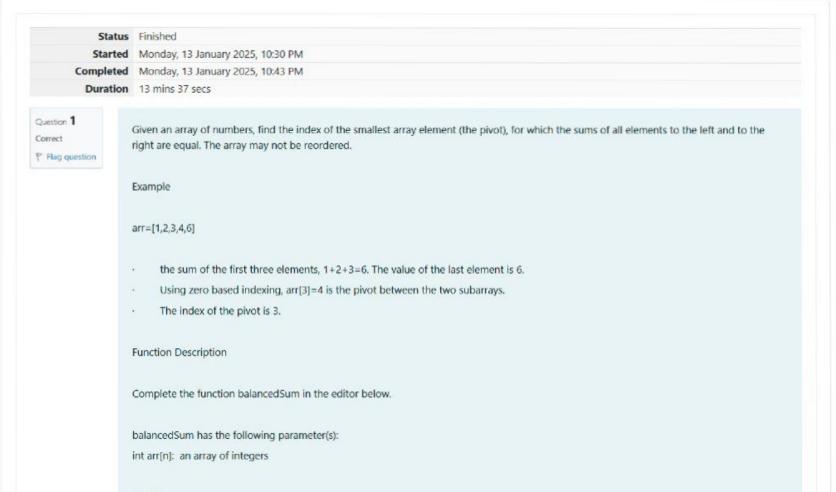
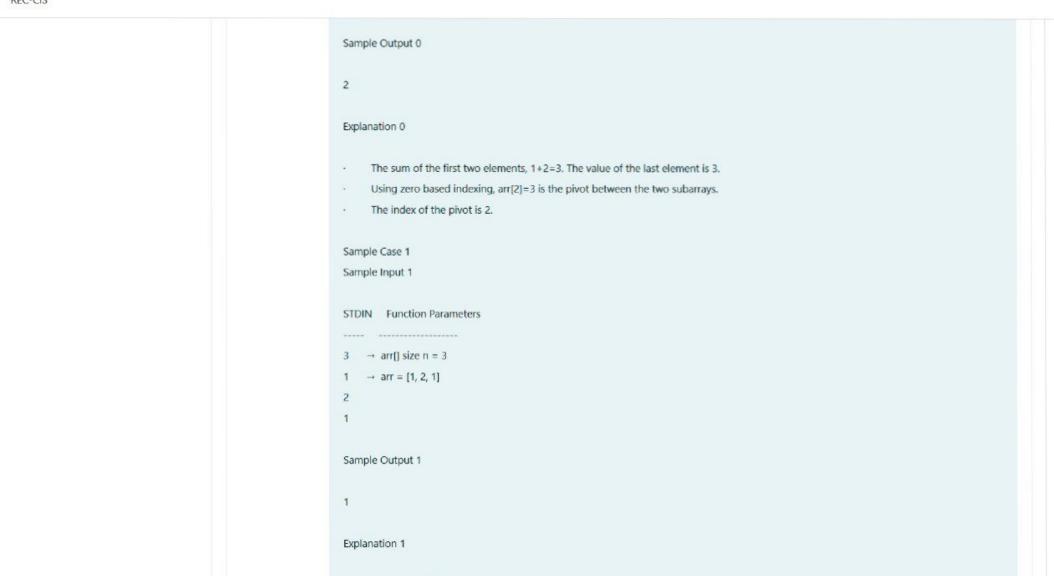
## GE23131-Programming Using C-2024





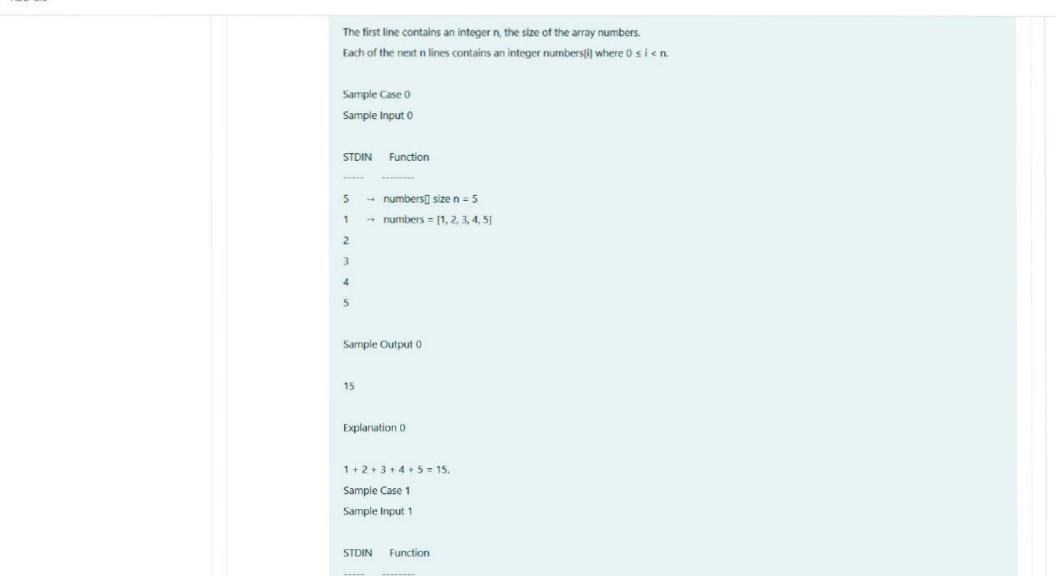
| REC-CIS |  |
|---------|--|
|         | Returns:   |
|         | int: an integer representing the index of the pivot                            |
|         | Constraints  |
|         | · 3 ≤ n ≤ 10 <sup>5</sup>  |
|         | . 1 ≤ arr[i] ≤ 2 × 10 <sup>4</sup> , where 0 ≤ i < n                           |
|         | 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 → arr[] size n = 4   |
|         | 1 $\rightarrow$ arr = [1, 2, 3, 3]   |
|         | 2  |
|         | 3  |
|         | 3  |



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. \* The function is expected to return an INTEGER. \* The function accepts INTEGER\_ARRAY arr as parameter. int balancedSum(int arr count, int\* arr) 9 + int 1 = 0, r = 0; 10 for (int i=0;i<arr\_count;i++){ 11 + r += arr[i]; 12 13 14 + for (int i=0;i<arr\_count;i++){ if (1 -- r - arr[i]){ 15 + return i; 16 17 1 += arr[i]; 18 r -= arr[i]; 19 20 21 return 1; 22 23

|   | Test  | Expected | Got |   |
|---|---|----------|-----|---|
| ~ | int arr[] = {1,2,3,3};<br>printf("%d", balanced5um(4, arr)) | 2        | 2   | ~ |

| REC-CIS               |   |
|-----------------------|---|
| Question 2<br>Correct | Calculate the sum of an array of integers.                                |
| Y Flag question       | 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 ≤ numbers[i] ≤ 10 <sup>4</sup>  |
|                       | Input Format for Custom Testing   |
|                       | Input from stdin will be processed as follows and passed to the function. |



```
2 → numbers[] size n = 2
    → numbers = [12, 12]
12
Sample Output 1
24
Explanation 1
12 + 12 = 24.
Answer: (penalty regime: 0 %)
 Reset answer
       * Complete the 'arraySum' function below.
       * The function is expected to return an INTEGER.
       * The function accepts INTEGER ARRAY numbers as parameter.
       int arraySum(int numbers_count, int *numbers)
   9 .
           int s =0;
  10
          for (int i=0;i<numbers_count;i++){
  11 +
  12
              s += numbers[i];
  13
           return s;
  14
  15
  16
```

```
12 + 12 = 24.
Answer: (penalty regime: 0 %)
 Reset answer
       * Complete the 'arraySum' function below.
       * The function is expected to return an INTEGER.
       * The function accepts INTEGER_ARRAY numbers as parameter.
       int arraySum(int numbers_count, int *numbers)
   9 +
           int s =0;
   10
          for (int i=0;i<numbers_count;i++){
  11 +
              s += numbers[i];
  12
  13
  14
           return s;
  15
  16
```

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

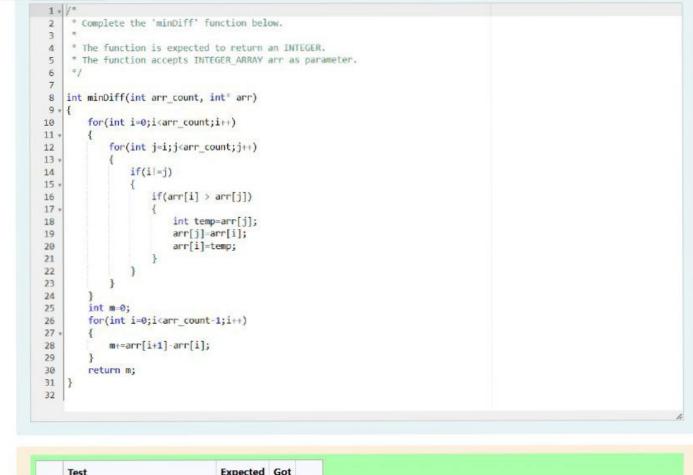
Passed all tests! 🗸

Question 3 Correct P 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

## Answer: (penalty regime: 0 %) Reset answer

```
* Complete the 'minDiff' function below.
     * The function is expected to return an INTEGER.
     * The function accepts INTEGER_ARRAY arr as parameter.
    int minDiff(int arr count, int* arr)
 8
 9
        for(int i=0;i<arr count;i++)
10
11 4
            for(int j=i;j<arr count;j++)
12
13
                if(i|=j)
14
15
                    if(arr[i] > arr[j])
16
17
                        int temp=arr[i];
18
                        arr[j]=arr[i];
19
                        arr[i]=temp;
20
21
22
23
24
25
        for(int i=0;i<arr count-1;i++)
26
```



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