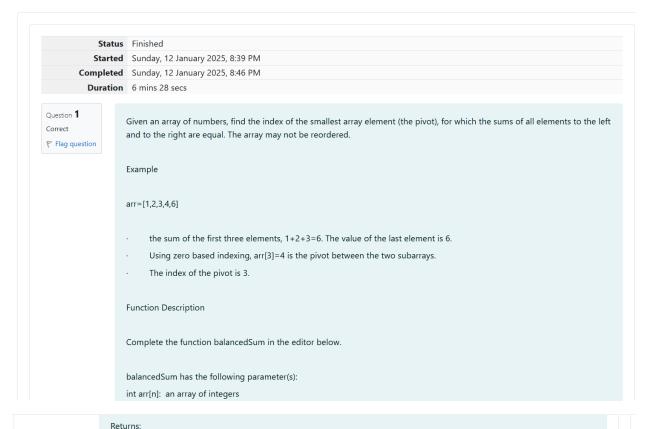
Week-13-Passing Arrays and Strings to Functions

ROLL NO: 241801294

NAME: THARUN N

Q1)



```
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
    → 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 %)

```
Reset answer
         /*

* Complete the 'balancedSum' function below.

* The function is expected to return an INTEGER.

* The function accepts INTEGER_ARRAY arr as parameter.

*/
  2
3
  7
8
9
        int balancedSum(int n, int* x)
 10
11
               int a=0, b = 0;
  12
               for (int i = 0; i < n; i++) {
 13
14
                    b+=x[i];
  15
               for (int i = 0; i < n;i++) {
   if (a==b-x[i]) {</pre>
 16
17
  18
  19
  20
                     a+=x[i];
  21
```

	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
```

```
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 ≤ n ≤ 10<sup>4</sup>
1 ≤ numbers[i] ≤ 10<sup>4</sup>

Input Format for Custom Testing
```

```
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 ≤ i < n.

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 \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
 1 \text{ v} / \text{*} * Complete the 'arraySum' function below.
         * The function is expected to return an INTEGER.   
* The function accepts INTEGER_ARRAY numbers as parameter.
    4
```

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

Passed all tests! ✓

Question **3**Correct

Flag question

Answer: (penalty regime: 0 %)

```
Reset answer
```

```
* Complete the 'minDiff' function below.
3
     * The function is expected to return an INTEGER.
     * The function accepts INTEGER_ARRAY arr as parameter.
    #include <stdlib.h>
10
    int cmp(const void* a, const void* b) {
11
       return (*(int*)a-*(int*)b);
12
13
14
    int minDiff(int n, int* a)
15 ▼ {
16
        qsort(a, n, sizeof(int), cmp);
        int s = 0;
for (int i = 0; i < (n-1); i++) {
17
18
            int c = a[i] - a[i+1];
19
20
```

```
12 }
13
    int minDiff(int n, int* a)
14
15
    {
16
        qsort(a, n, sizeof(int), cmp);
17
        int s = 0;
18
        for (int i = 0; i < (n-1); i++) {
19
            int c = a[i] - a[i+1];
21
            if (c < 0)
22
               s-=c;
23
            else
24
                s+=c;
25
26
27
        return s;
28
    }
29
```

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

Passed all tests! 🗸