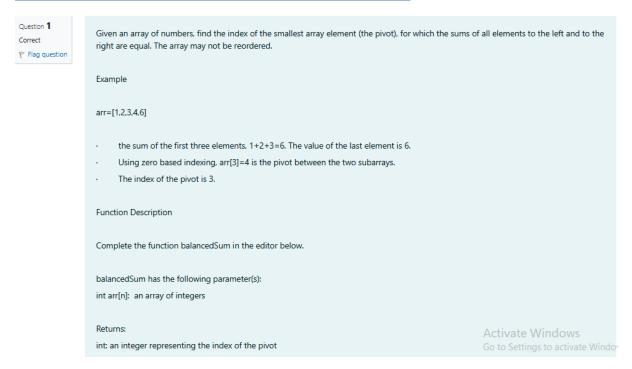
Week-13-Passing Array and Strings to Functions

Week-13-01-Practice Session-Coding



Source Code

```
2
      * Complete the 'balancedSum' function below.
 4
     * The function is expected to return an INTEGER.
     * The function accepts INTEGER_ARRAY arr as parameter.
 8
    int balancedSum(int arr count, int* arr)
 9 ,
    {
10
         int left=0,right=0;
        for(int i=0;i<arr_count;i++){</pre>
11
             right+=arr[i];
12
13
        for(int i=0;i<arr_count;i++){</pre>
14 1
            if(left==(right-arr[i]))
15
16
                return i;
            left+=arr[i];
17
18
            right-=arr[i];
19
20
         return 1;
21
```

Result



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
```

Source Code

```
/*
* Complete the 'arraySum' function below.
 3
     * The function is expected to return an INTEGER.
     \ensuremath{^*} The function accepts <code>INTEGER_ARRAY</code> numbers as parameter.
    int arraySum(int numbers_count, int *numbers)
10
         int sum=0;
11
         for(int i=0;i<numbers_count;i++){</pre>
             sum+=numbers[i];
13
14
         return sum;
15
```

Result

	Test	Expected	Got	
~	int arr[] = {1,2,3,4,5}; printf("%d", arraySum(5, arr))	15	15	~
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 lines contains an integer that describes arr[i] (where $0 \le i < i$ n) . Sample Case 0 Sample Input For Custom Testing STDIN Function ----- 5 → arr[] size n = 5 5 → arr[] = [5, 1, 3, 7, 3] 1 3 7 3 Sample Output 6 Explanation n = 5 arr = [5, 1, 3, 7, 3] If arr is rearranged as arr' = [1, 3, 3, 5, 7], the differences are minimized. The final answer is |1 - 3| + |3 - 3| + |3 - 5| + |5 - 7| = 6. Sample Case 1 Sample Input For Custom Testing STDIN Function ----- 2 → arr[] size $n=2\ 3 \rightarrow arr[]=[3,\,2]\ 2\ Sample\ Output\ 1\ Explanation\ n=2\ arr=[3,\,2]\ There\ is\ no\ need\ to\ rearrange\ because\ there\ are\ only\ two$ elements. The final answer is |3 - 2| = 1.

Source Code

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

Result

