

# Week 13

## Question 1:

Given an array of numbers, find the index of the smallest array element (the pivot), for which the sums of all elements to the left and to the right are equal. The array may not be reordered.

### Input Format:

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  $0 \leq i < n$ .

### Program:

## Attempt 1

<b>Status</b>	Finished
<b>Started</b>	Monday, 13 January 2025, 8:54 PM
<b>Completed</b>	Monday, 13 January 2025, 9:16 PM
<b>Duration</b>	22 mins 27 secs

## Week-13-Passing Arrays and Strings to Functions: Attempt review |

Secure [rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=123456789](http://rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=123456789)

Reset answer

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

Output:

	Test	Expected	Got	
✓	int arr[] = {1,2,3,3}; printf("%d", balancedSum(4, arr))	2	2	✓

Passed all tests! ✓

Question 2:

Calculate the sum of an array of integers.

Input Format:

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  $0 \leq i < n$ .

## Attempt 1

<b>Status</b>	Finished
<b>Started</b>	Monday, 13 January 2025, 8:54 PM
<b>Completed</b>	Monday, 13 January 2025, 9:16 PM
<b>Duration</b>	22 mins 27 secs

Program:

Answer: (penalty regime: 0 / 70)

Reset answer

```

1  /*
2  * Complete the 'arraySum' function below.
3  *
4  * The function is expected to return an INTE
5  * The function accepts INTEGER_ARRAY numbers
6  */
7
8  int arraySum(int numbers_count, int *numbers)
9  {
10     int s=0;
11     for(int i=0;i<numbers_count;i++){
12         s+=numbers[i];
13     }
14     return s;
15 }
16

```

Output:

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

Passed all tests! ✓

### Question 3:

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  $1 + 1 + 0 + 1 = 3$ .

Input Format:

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  $0 \leq i < n$ ).

Program:

Attempt 1	
<b>Status</b>	Finished
<b>Started</b>	Monday, 13 January 2025, 8:54 PM
<b>Completed</b>	Monday, 13 January 2025, 9:16 PM
<b>Duration</b>	22 mins 27 secs

```

2  * Complete the 'minDiff' function below.
3  *
4  * The function is expected to return an INTEGER
5  * The function accepts INTEGER_ARRAY arr as
6  */
7
8  int minDiff(int arr_count, int* arr)
9  {
10     for(int i=0;i<arr_count;i++){
11         for(int j=i;j<arr_count;j++){
12             if(i!=j){
13                 if(arr[i]>arr[j]){
14                     int temp=arr[j];
15                     arr[j]=arr[i];
16                     arr[i]=temp;
17                 }
18             }
19         }
20     }
21     int m=0;
22     for(int i=0;i<arr_count-1;i++){
23         m+=arr[i+1]-arr[i];
24     }
25     return m;
26 }

```

Output:

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

Passed all tests! ✓