NAME:BRINDHA.A
ROLL NO:240701083
WEEK-13

Passing Arrays and Strings to Functions

Ex. No. : 59 **Date :** 20.12.24

Balanced Array 240701083

Problem Statement:

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.

Example: arr=[1,2,3,4,6]

- the sum of the first three elements, 1+2+3=6. The value of the last element is 6.
- Using zero based indexing, arr[3]=4 is the pivot between the two subarrays.
- The index of the pivot is 3.

Function Description: Complete the function balancedSum in the editor below.

balancedSum has the following parameter(s): int arr[n]: an array of integers

Returns: int: an integer representing the index of the pivot

Constraints

- $3 \le n \le 105$
- $1 \le arr[i] \le 2 \times 104$, where $0 \le i < n$
- 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 Input

STDIN		Function Parameters
4	→	arr[] size n = 4
1	→	arr = [1, 2, 3, 3]
2		
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.

Program:

```c
/*
 * 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)
{
int total_sum=0,left_sum=0;
for(int i=0;i<arr_count;i++)
{
total_sum+=arr[i];
}
for(int i=0;i<arr_count;i++)
{
int right_sum=total_sum-left_sum-arr[i];
if(left_sum==right_sum)
{
return i;
}
left_sum+=arr[i];
}
return 1;
}
```

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

Passed all tests! ✓

Ex. No. : 60 **Date :** 20.12.24

Sum Them All 240701083

Problem Statement:

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

Constraints
$1 \le n \le 104$
$1 \le numbers[i] \le 104$

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 Input

STDIN		Function
5	→	numbers[] size n = 5
1	→	numbers = [1, 2, 3, 4, 5]
2		
3		
4		
5		

Sample Output
15

Explanation
1 + 2 + 3 + 4 + 5 = 15.

Program:

```c
/*
 * Complete the 'arraySum' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY numbers as parameter.
 */

int arraySum(int n, int *numbers)
{
    int sum=0;
    for(int i=0;i<n;i++)
    {
        sum+=numbers[i];
    }
    return sum;

}
```

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

Passed all tests! ✓

Ex. No. : 61 **Date :** 20.12.24

Minimum Difference Sum 240701083

Problem Statement:

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
$2 \leq n \leq 105$
$0 \leq arr[i] \leq 109$, where $0 \leq i < n$

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 $0 \leq i < n$) .

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.

Program:

```c
#include<stdio.h>
#include<stdlib.h>
int compare(const void*a,const void*b)
{
    return(*(int*)a - *(int *)b);
}
int minDiff(int arr_count,int*arr)
{
    qsort(arr,arr_count,sizeof(int),compare);
    int sum=0;
    for(int i=1;i<arr_count;i++)
    {
        sum+=abs(arr[i]-arr[i-1]);
    }
    return sum;
}
```

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

Passed all tests! ✓