Narmadha S

ECE D

240801216

Question1: BalancedArray

ProblemStatement:

Givenanarrayofnumbers,findtheindexofthesmallestarrayelement(thepivot),for whichthesumsofallelementstotheleftandtotherightareequal.Thearraymaynotbe reordered.

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

- thesumofthefirstthreeelements,1+2+3=6.Thevalueofthelastelementis6.
- Usingzerobasedindexing,arr[3]=4isthepivotbetweenthetwosubarrays.
- Theindexofthepivotis3.

FunctionDescription:CompletethefunctionbalancedSumintheeditorbelow.

balancedSumhasthefollowingparameter(s):intarr[n]:anarrayofintegers

Returns:int:anintegerrepresentingtheindexofthepivot

Constraints:

- 3≤n≤105
- 1≤arr[i]≤2×104,where0≤i<n
- Itisguaranteedthatasolutionalwaysexists.

InputFormatforCustomTesting

Inputfromstdinwillbeprocessedasfollowsandpassedtothefunction.Thefirstline containsanintegern,thesizeofthearrayarr.Eachofthenextnlinescontainsaninteger, arr[i], where 0 ≤ i < n.

SampleInput:

STDINFunctionParameters

----- -----------------

4 → arr[]sizen=4

1 → arr=[1,2,3,3]

2

3

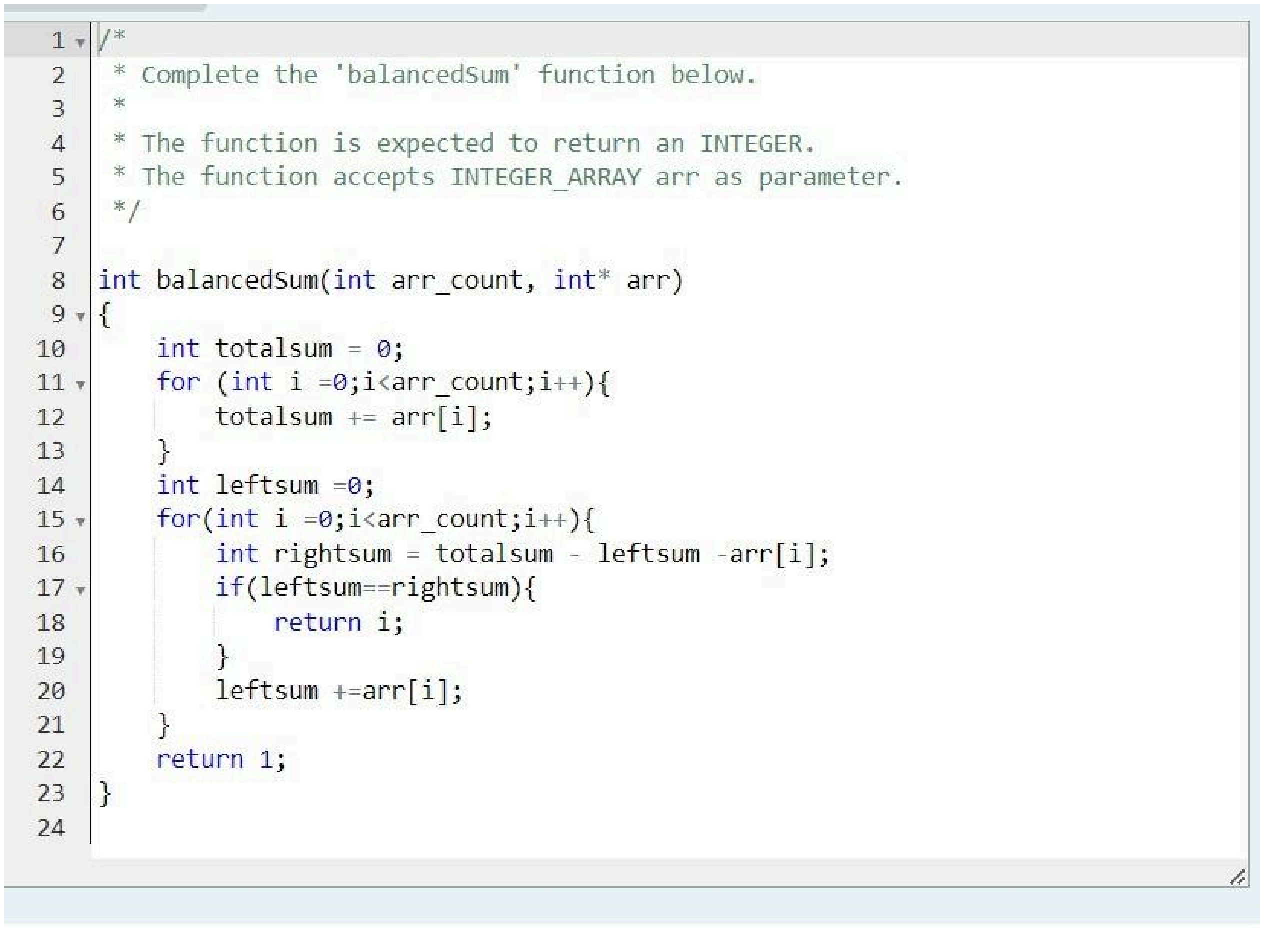
3

SampleOutput0

2

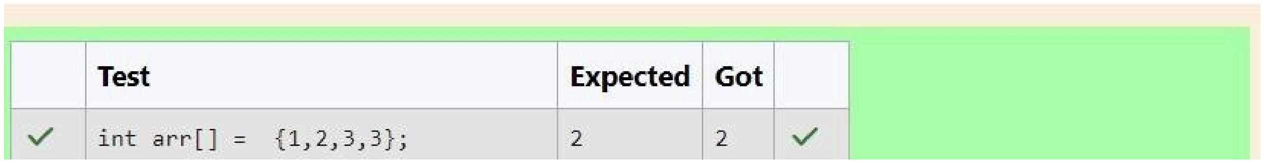
Explanation0

- Thesumofthefirsttwoelements,1+2=3.Thevalueofthelastelementis3.
- Usingzerobasedindexing,arr[2]=3isthepivotbetweenthetwosubarrays.
- Theindexofthepivotis2.



```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 totalsum = 0;
    for (int i =0;i<arr_count;i++){
        totalsum += arr[i];
    }
    int leftsum =0;
    for(int i =0;i<arr_count;i++){
        int rightsum = totalsum - leftsum -arr[i];
        if(leftsum==rightsum){
            return i;
        }
        leftsum +=arr[i];
    }
    return 1;
}
```



	Test	Expected	Got	
✓	`int arr[] =  {1,2,3,3};`	2	2	✓

Question2: SumThemAll

Calculatethesumofanarrayofintegers.

Example:

numbers=[3,13,4,11,9]

Thesumis3+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:integersumofthenumbersarray

Constraints:

- 1≤n≤104
- 1 ≤ numbers[i] ≤ 104

Input Format for Custom Testing

Inputfromstdinwillbeprocessedasfollowsandpassedtothefunction.

Thefirstlinecontainsanintegern,thesizeofthearraynumbers.

Each of the next n lines contains an integer numbers[i] where 0 ≤ i <

n.Sample Input

5 → numbers[]sizen=5

1 → numbers=[1,2,3,4,5]

2

3

4

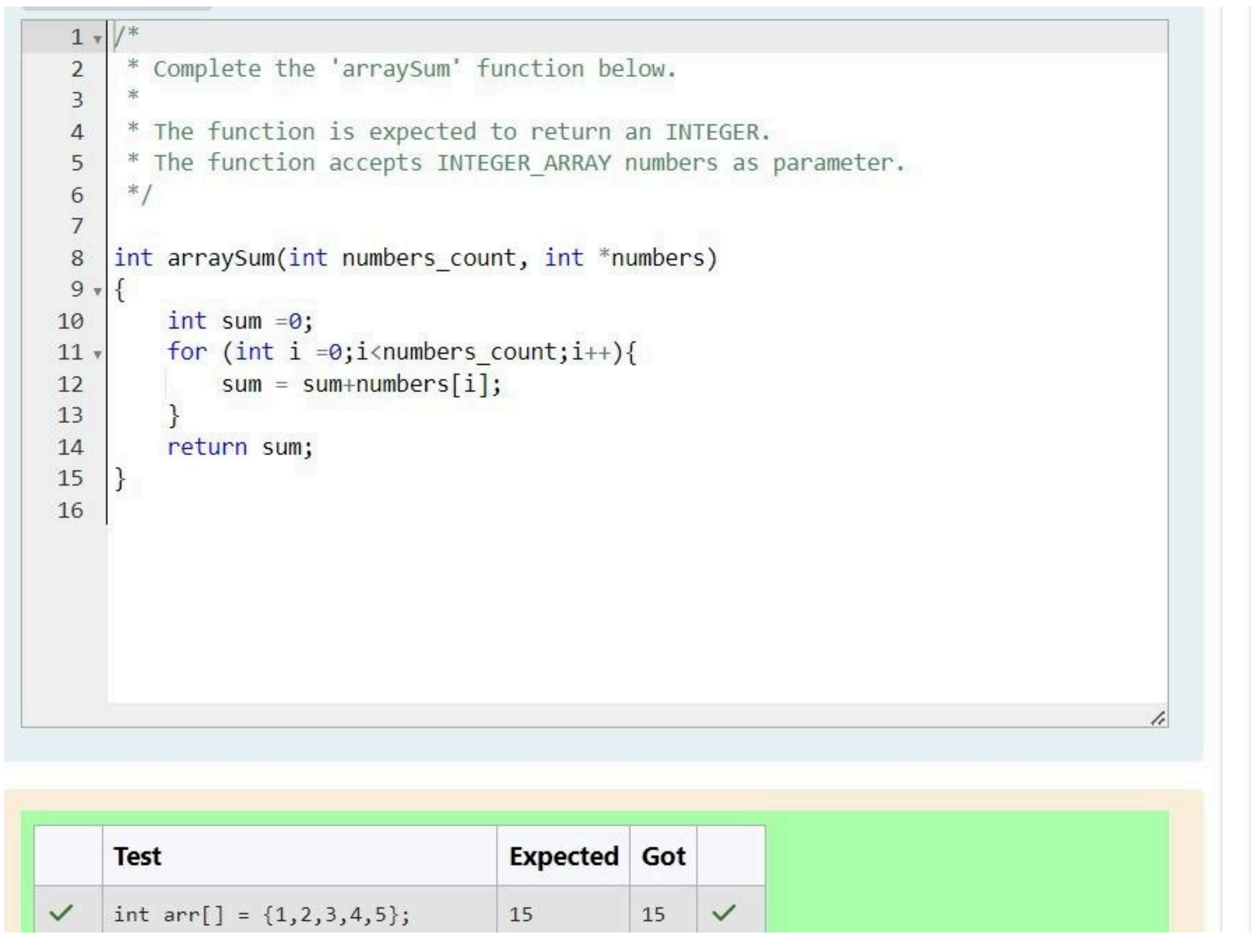
5

SampleOutput

15

Explanation

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



```
/*
 * 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)
{
    int sum =0;
    for (int i =0;i<numbers_count;i++){
        sum = sum+numbers[i];
    }
    return sum;
}

```

	Test	Expected	Got	
✓	int arr[] = {1,2,3,4,5};	15	15	✓

Question3: MinimumDifferenceSum

Givenanarrayofnintegers,rearrangethemsothatthesumoftheabsolutedifferencesof alladjacentelementsisminimized.Then,computethesumofthoseabsolutedifferences.

Example

n=5,arr=[1,3,3,2,4]

Ifthelistisrearrangedasarr'=[1,2,3,3,4],theabsolutedifferencesare|1-2|=1,|2-3| =1,|3-3|=0,|3-4|=1.Thesumofthosedifferencesis1+1+0+1=3. Function Description

CompletethefunctionminDiffintheeditorbelow. minDiff

has the following parameter:

arr: an integer array

Returns:

int: the sum of the absolute differences of adjacent

elementsConstraints

2≤n≤105

0≤arr[i]≤109,where0≤i<n Format

For Custom Testing

Thefirstlineofinputcontainsaninteger,n,thesizeofarr.

Eachofthefollowingnlinescontainsanintegerthatdescribesarr[i](where0≤i<n).

SampleInputForCustomTesting 5

→ arr[] size n = 5

5 →arr[] =[5, 1,3, 7,3]

1

3

7

3

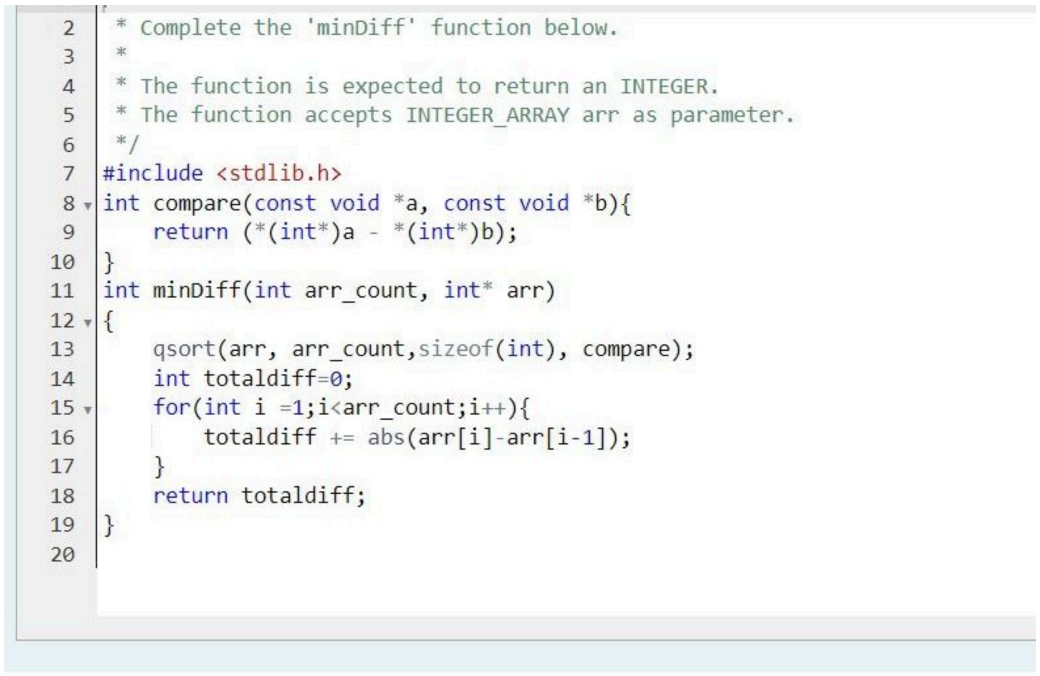
SampleOutput6

Explanation

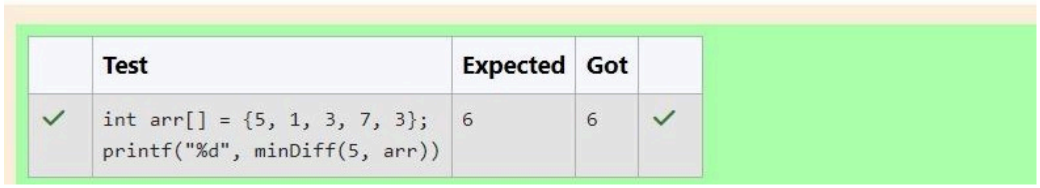
n=5,arr=[5,1,3,7,3]

If arr is rearranged as arr' = [1, 3, 3, 5, 7], the differences are

minimized.Thefinalansweris|1-3|+|3-3|+|3-5|+|5-7|=6.



```
 * Complete the 'minDiff' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY arr as parameter.
 */
#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 totaldiff=0;
    for(int i =1;i<arr_count;i++){
        totaldiff += abs(arr[i]-arr[i-1]);
    }
    return totaldiff;
}
```



	Test	Expected	Got	

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