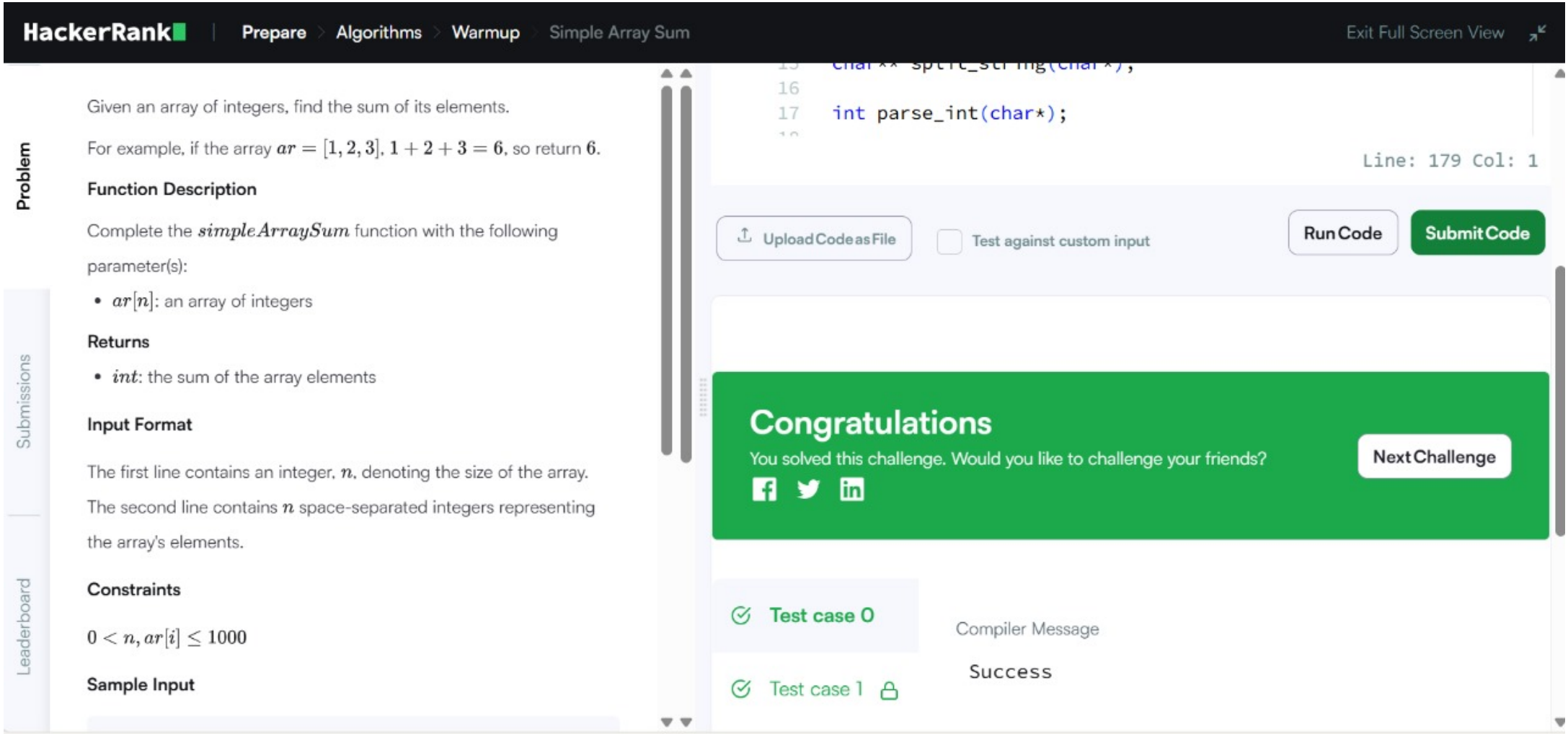
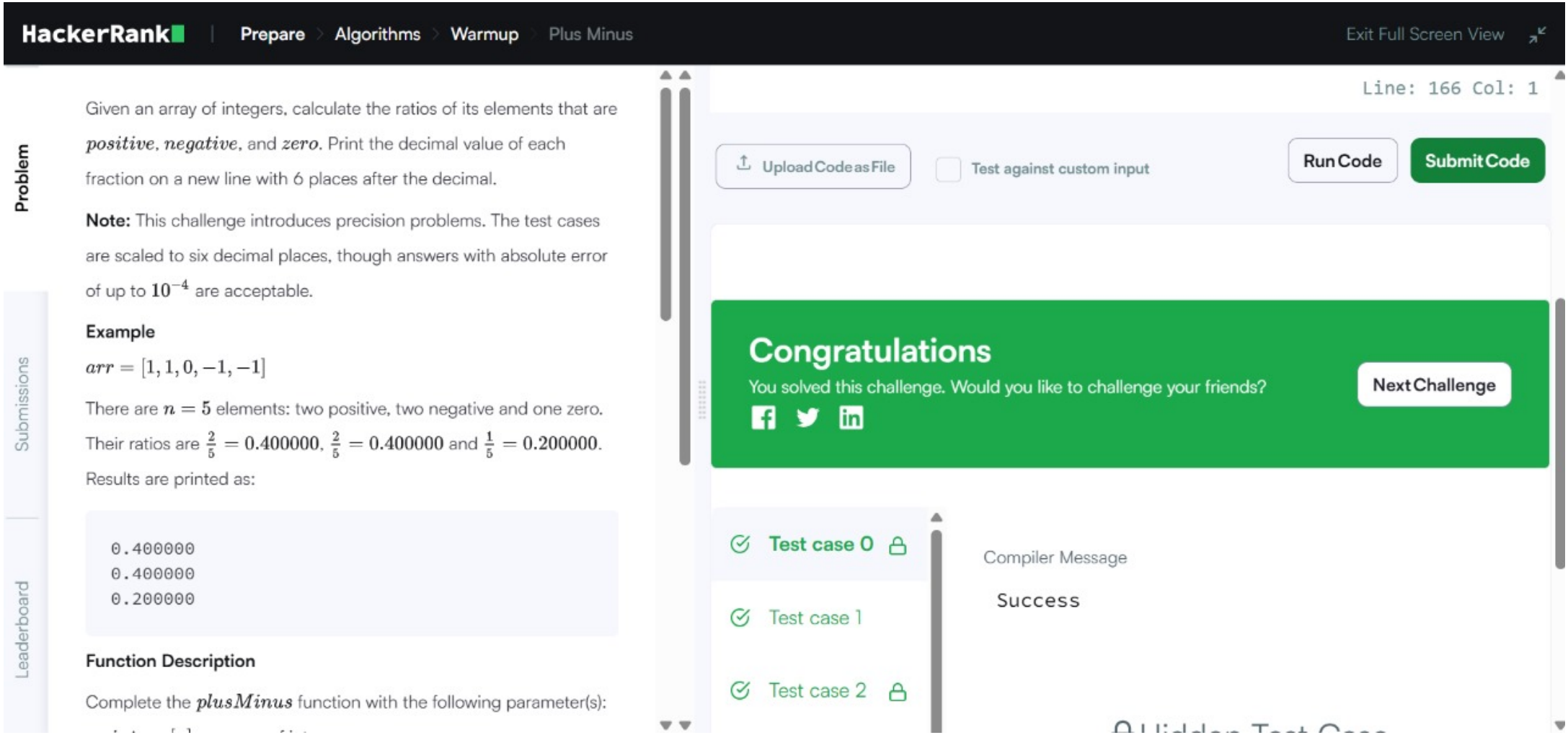
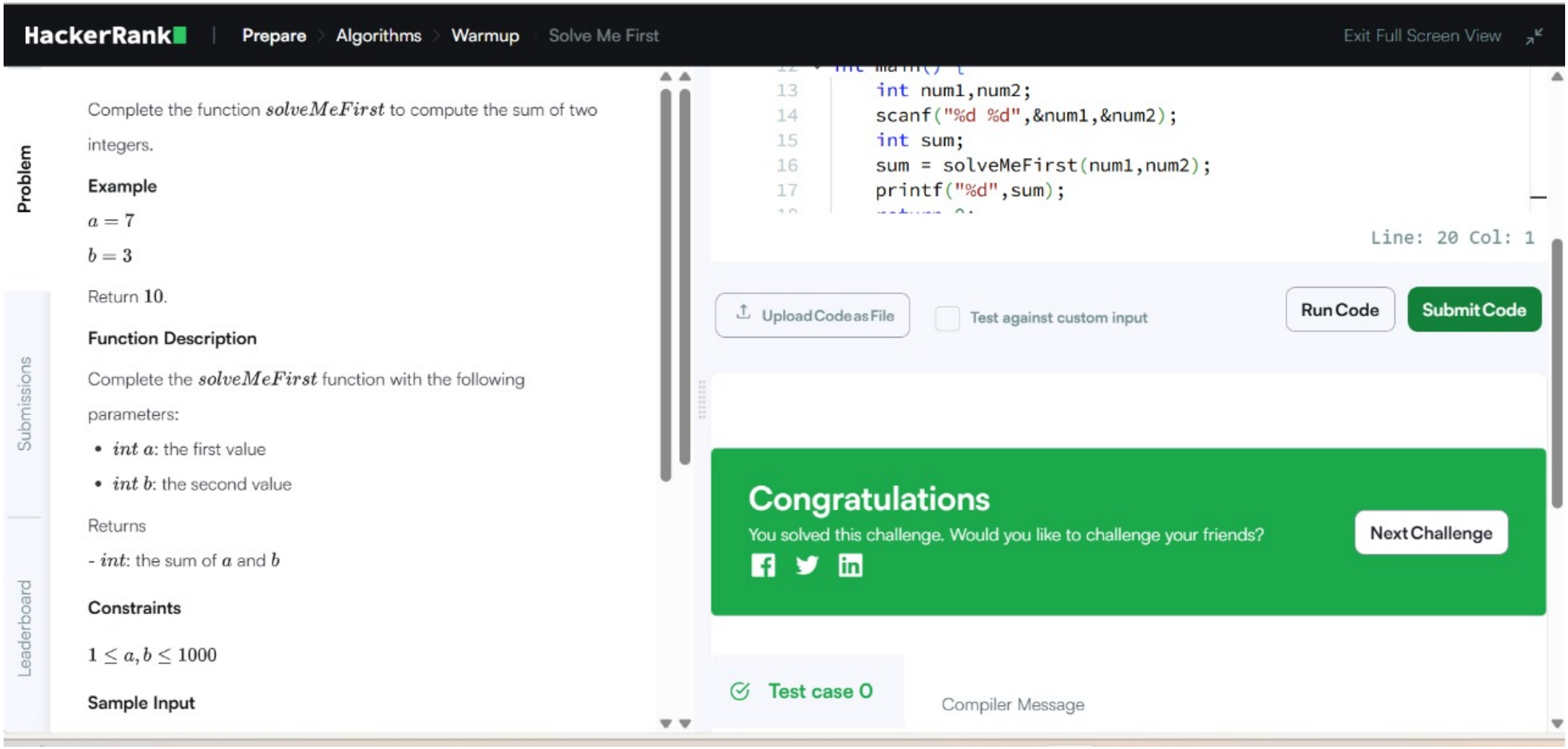
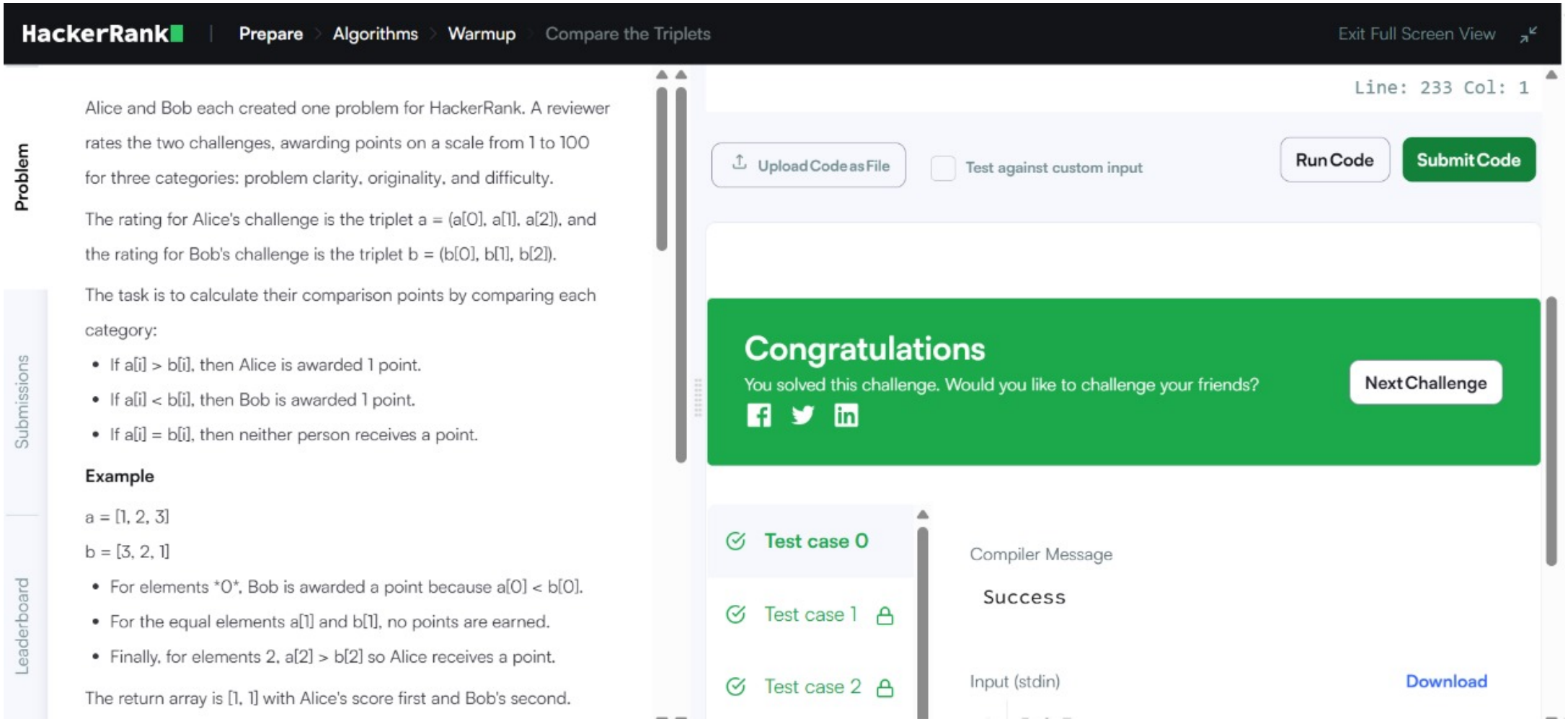
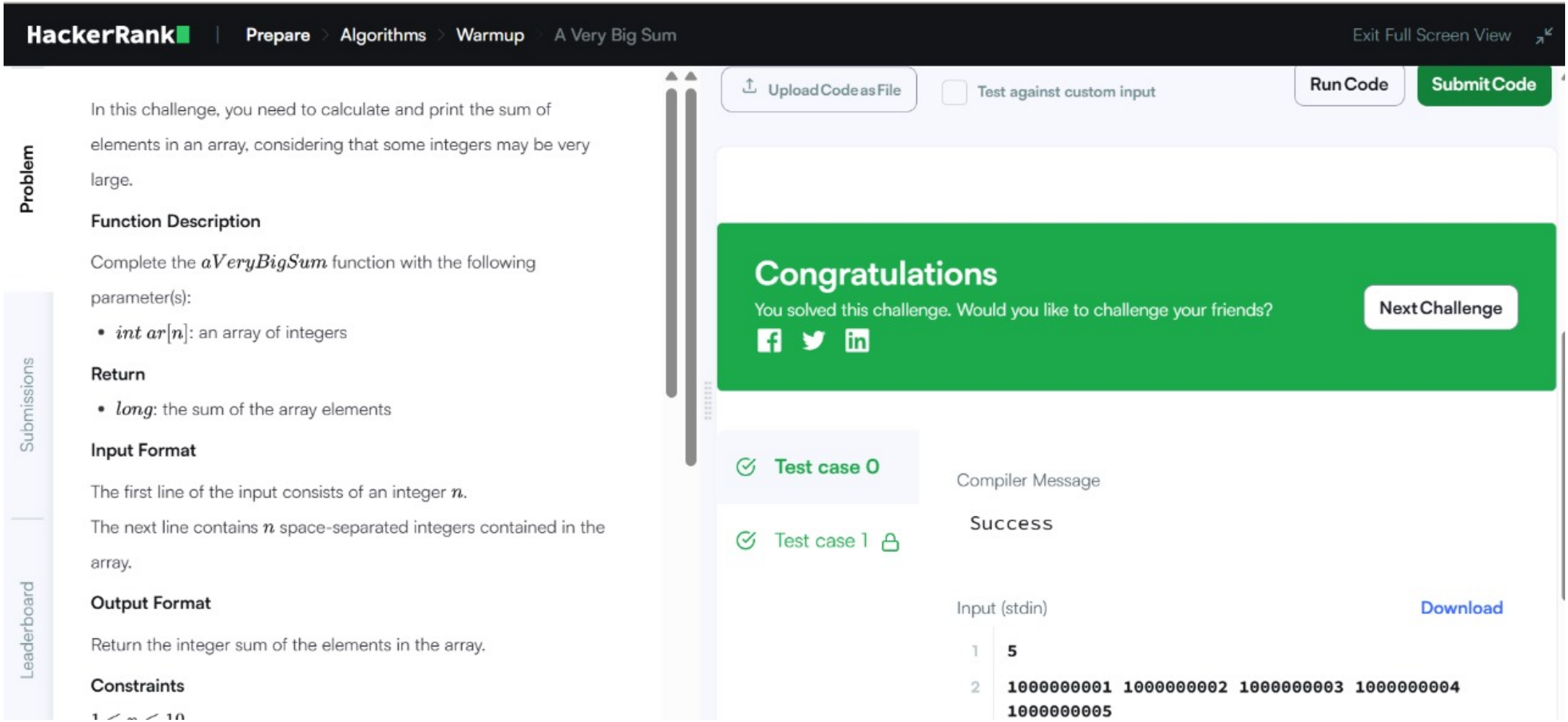

HackerRank
Prepare
Algorithms
Warmup
Simple Array Sum
Exit Full Screen View
Problem
Submissions
Leaderboard
Given an array of integers, find the sum of its elements.
For example, if the array $ar = [1, 2, 3]$, $1 + 2 + 3 = 6$, so return 6.
Function Description
Complete the $simpleArraySum$ function with the following parameter(s):
$ar[n]$: an array of integers
Returns
int: the sum of the array elements
Input Format
The first line contains an integer, n, denoting the size of the array.
The second line contains n space-separated integers representing the array's elements.
Constraints
$0 < n, ar[i] \leq 1000$
Sample Input
int parse_int(char*);
Line: 179 Col: 1
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Compiler Message
Success


HackerRank
Prepare
Algorithms
Warmup
Plus Minus
Exit Full Screen View
Problem
Submissions
Leaderboard
Given an array of integers, calculate the ratios of its elements that are positive, negative, and zero. Print the decimal value of each fraction on a new line with 6 places after the decimal.
Note: This challenge introduces precision problems. The test cases are scaled to six decimal places, though answers with absolute error of up to 10^{-4} are acceptable.
Example
$arr = [1, 1, 0, -1, -1]$
There are $n = 5$ elements: two positive, two negative and one zero. Their ratios are $\frac{2}{5} = 0.400000$, $\frac{2}{5} = 0.400000$ and $\frac{1}{5} = 0.200000$. Results are printed as:
0.400000
0.400000
0.200000
Function Description
Complete the plusMinus function with the following parameter(s):
Line: 166 Col: 1
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Test case 2
Compiler Message
Success


HackerRank
Prepare
Algorithms
Warmup
Solve Me First
Exit Full Screen View
Problem
Submissions
Leaderboard
Complete the function $solveMeFirst$ to compute the sum of two integers.
Example
$a = 7$
$b = 3$
Return 10.
Function Description
Complete the $solveMeFirst$ function with the following parameters:
$int\ a$: the first value
$int\ b$: the second value
Returns
- int: the sum of a and b
Constraints
$1 \leq a, b \leq 1000$
Sample Input
int num1,num2;
scanf("%d %d",&num1,&num2);
int sum;
sum = solveMeFirst(num1,num2);
printf("%d",sum);
Line: 20 Col: 1
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Compiler Message


HackerRank | Prepare > Algorithms > Warmup > Compare the Triplets
Exit Full Screen View
Problem
Submissions
Leaderboard
Alice and Bob each created one problem for HackerRank. A reviewer rates the two challenges, awarding points on a scale from 1 to 100 for three categories: problem clarity, originality, and difficulty.
The rating for Alice's challenge is the triplet a = (a[0], a[1], a[2]), and the rating for Bob's challenge is the triplet b = (b[0], b[1], b[2]).
The task is to calculate their comparison points by comparing each category:
If a[i] > b[i], then Alice is awarded 1 point.
If a[i] < b[i], then Bob is awarded 1 point.
If a[i] = b[i], then neither person receives a point.
Example
a = [1, 2, 3]
b = [3, 2, 1]
For elements *0*, Bob is awarded a point because a[0] < b[0].
For the equal elements a[1] and b[1], no points are earned.
Finally, for elements 2, a[2] > b[2] so Alice receives a point.
The return array is [1, 1] with Alice's score first and Bob's second.
Line: 233 Col: 1
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Test case 2
Compiler Message
Success
Input (stdin)
Download


HackerRank | Prepare > Algorithms > Warmup > A Very Big Sum
Exit Full Screen View
Problem
Submissions
Leaderboard
In this challenge, you need to calculate and print the sum of elements in an array, considering that some integers may be very large.
Function Description
Complete the $aVeryBigSum$ function with the following parameter(s):
$int\ ar[n]$: an array of integers
Return
$long$: the sum of the array elements
Input Format
The first line of the input consists of an integer n.
The next line contains n space-separated integers contained in the array.
Output Format
Return the integer sum of the elements in the array.
Constraints
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Compiler Message
Success
Input (stdin)
Download
1 5
2 1000000001 1000000002 1000000003 1000000004 1000000005

Problem

Submissions

Leaderboard

Given a square matrix, calculate the absolute difference between the sums of its diagonals.

For example, the square matrix arr is shown below:

```
1 2 3
4 5 6
9 8 9
```

- The left-to-right diagonal = $1 + 5 + 9 = 15$.
- The right-to-left diagonal = $3 + 5 + 9 = 17$.

Their absolute difference is $|15 - 17| = 2$.

Function description

Complete the *diagonalDifference* function with the following parameter:

- $int\ arr[n][m]$: a 2-D array of integers

Return

- int: the absolute difference in sums along the diagonals

Input Format

Line: 189 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Test case 1

Test case 2

Compiler Message

Success

Input (stdin) Download

1 3

Comparison Sorting

Quicksort usually has a running time of $n \times log(n)$, but is there an algorithm that can sort even faster? In general, this is not possible. Most sorting algorithms are comparison sorts, i.e. they sort a list just by comparing the elements to one another. A comparison sort algorithm cannot beat $n \times log(n)$ (worst-case) running time, since $n \times log(n)$ represents the minimum number of comparisons needed to know where to place each element. For more details, you can see these notes (PDF).

Alternative Sorting

Another sorting method, the counting sort, does not require comparison. Instead, you create an integer array whose index range covers the entire range of values in your array to sort. Each time a value occurs in the original array, you increment the counter at that index. At the end, run through your counting array, printing the value of each non-zero valued index that number of times.

Example

$arr = [1, 1, 3, 2, 1]$

All of the values are in the range $[0 \ldots 3]$, so create an array of zeros, $result = [0, 0, 0, 0]$. The results of each iteration follow:

i	arr[i]	result
0	1	[0, 1, 0, 0]
1	1	[0, 2, 0, 0]
2	3	[0, 2, 0, 1]
3	2	[0, 2, 1, 1]
4	1	[0, 3, 1, 1]

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Test case 1

Test case 2

Hidden test case

Test case 3

Test case 4

Test case 5

Compiler Message

Success

Download

```
100
63 25 73 1 98 73 56 84 86 57 16 83 8 25 81 56 9 53 98 67
99 12 83 89 80 91 39 86 76 85 74 39 25 90 59 10 94 32 44
3 89 30 27 79 46 96 27 32 18 21 92 69 81 40 40 34 68 78
24 87 42 69 23 41 78 22 6 90 99 89 50 30 20 1 43 3 70 95
33 46 44 9 69 48 33 60 65 16 82 67 61 32 21 79 75 75 13
87 70 33
```

Expected Output

Download

```
0 2 0 2 0 0 1 0 1 2 1 0 1 1 0 0 2 0 1 0 1 2 1 1 1 3 0 2 0
0 2 0 3 3 1 0 0 0 0 2 2 1 1 1 2 0 2 0 1 0 1 0 0 1 0 0 2 1
```

Given an array of integers, find the sum of its elements.

For example, if the array $ar = [1, 2, 3]$, $1 + 2 + 3 = 6$, so return 6.

Function Description

Complete the $simpleArraySum$ function with the following parameter(s):

- $ar[n]$: an array of integers

Returns

- int: the sum of the array elements

Input Format

The first line contains an integer, n, denoting the size of the array.

The second line contains n space-separated integers representing the array's elements.

Constraints

$0 < n, ar[i] \leq 1000$

Sample Input

```
STDIN        Function
-----        --------
6            ar[] size n = 6
1 2 3 4 10 11   ar = [1, 2, 3, 4, 10, 11]
```

Sample Output

```
31
```

```
15
16 int simpleArraySum(vector<int> ar) {
17     int sum = 0;
18     for(int num : ar) {
19         sum += num;
20     }
21     return sum;
22 }
23
24 int main()
25 {
```

Line: 93 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Test case 1

Compiler Message

Success

There is a collection of input strings and a collection of query strings. For each query string, determine how many times it occurs in the list of input strings. Return an array of the results.

Example

$stringList = ['ab', 'ab', 'abc']$

$queries = ['ab', 'abc', 'bc']$

There are 2 instances of 'ab', 1 of 'abc', and 0 of 'bc'. For each query, add an element to the return array: $results = [2, 1, 0]$.

Function Description

Complete the function $matchingStrings$ with the following parameters:

- $string\ stringList[n]$: an array of strings to search
- $string\ queries[q]$: an array of query strings

Returns

- $int[q]$: the results of each query

Input Format

The first line contains and integer n, the size of $stringList[]$.

Each of the next n lines contains a string $stringList[i]$.

The next line contains q, the size of $queries[]$.

Each of the next q lines contains a string $queries[i]$.

Constraints

$1 \leq n \leq 1000$

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

- Sample Test case 0
- Sample Test case 1
- Sample Test case 2

Input (stdin) Download

```
4
aba
baba
aba
xzxb
3
aba
xzxb
ab
```

Your Output (stdout)

```
2
```

Use the counting sort to order a list of strings associated with integers. If two strings are associated with the same integer, they must be printed in their original order, i.e. your sorting algorithm should be stable. There is one other twist: strings in the first half of the array are to be replaced with the character - (dash, ascii 45 decimal).

Insertion Sort and the simple version of Quicksort are stable, but the faster in-place version of Quicksort is not since it scrambles around elements while sorting.

Design your counting sort to be stable.

Example

$arr = [[0,'a'],[1,'b'],[0,'c'],[1,'d']]$

The first two strings are replaced with '-'. Since the maximum associated integer is 1, set up a helper array with at least two empty arrays as elements. The following shows the insertions into an array of three empty arrays.

i	string	converted list	
0			[[],[],[]]
1	a	-	[[-],[],[]]
2	b	-	[[-],[-],[]]
3	c		[[-,c],[-],[]]
4	d		[[-,c],[-,d],[]]

The result is then printed: - c - d .

Function Description

Complete the countSort function in the editor below. It should construct and print the sorted strings.

```
18
19    // Process each element
20    for(int i = 0; i < n; i++) {
21        int key = stoi(arr[i][0]);
22        string value = (i < n/2) ? "-" : arr[i][1];
23        buckets[key].push_back(value);
24    }
25
```

Line: 102 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

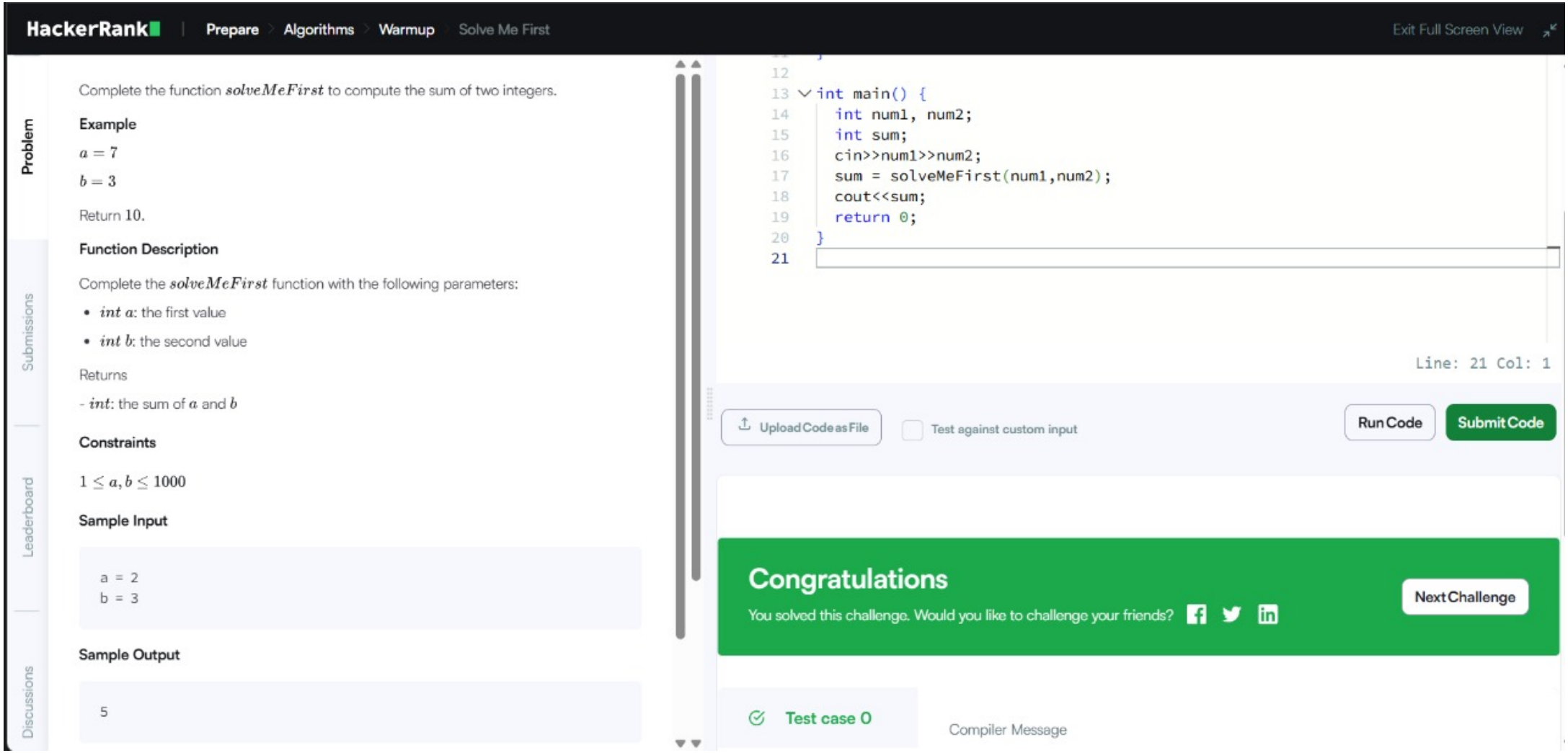
Test case 1

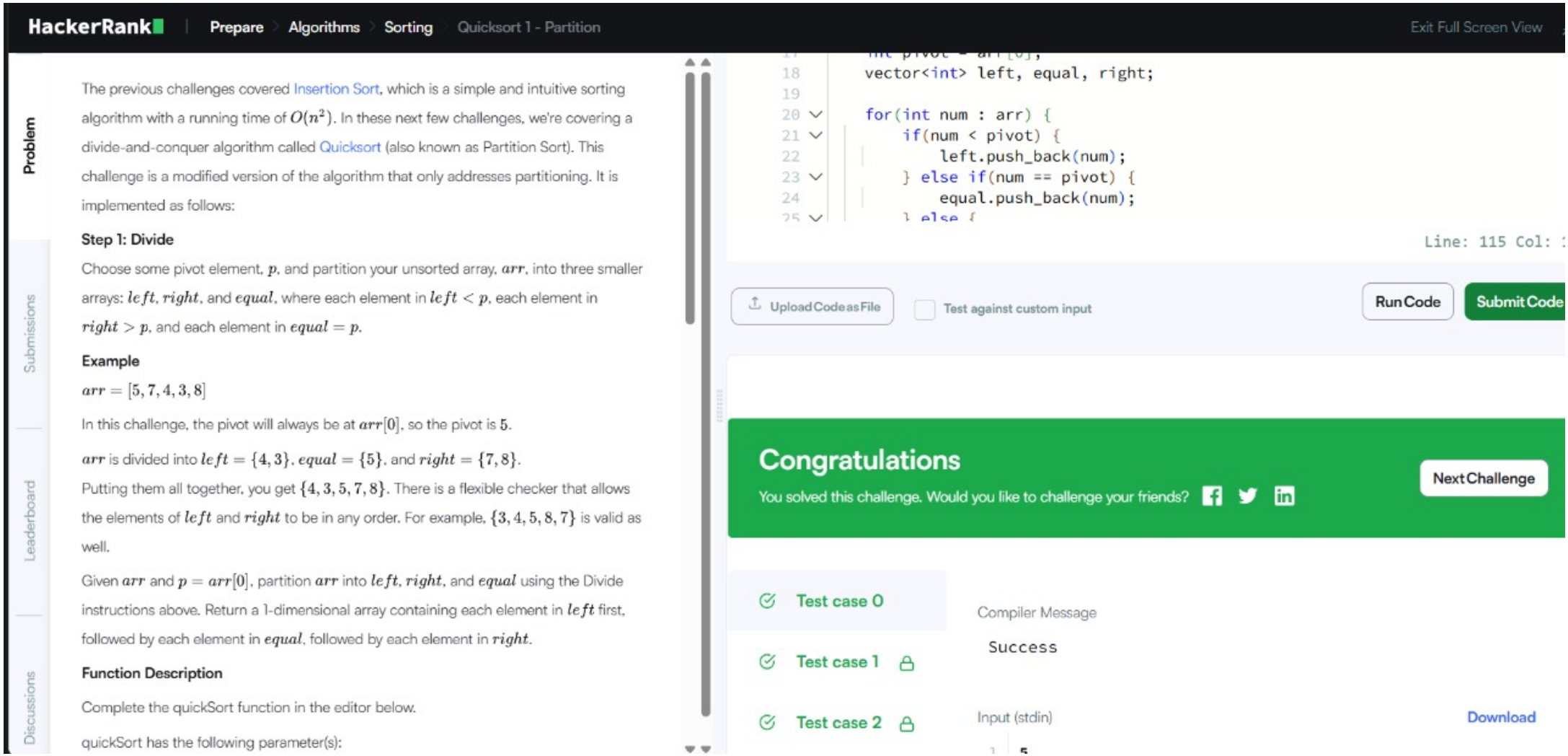
Test case 2

Compiler Message

Success

Input (stdin) Download


HackerRank
Prepare
Algorithms
Warmup
Solve Me First
Exit Full Screen View
Problem
Submissions
Leaderboard
Discussions
Complete the function $solveMeFirst$ to compute the sum of two integers.
Example
$a = 7$
$b = 3$
Return 10.
Function Description
Complete the $solveMeFirst$ function with the following parameters:
$int\ a$: the first value
$int\ b$: the second value
Returns
- int: the sum of a and b
Constraints
$1 \leq a, b \leq 1000$
Sample Input
a = 2
b = 3
Sample Output
5
int main() {
int num1, num2;
int sum;
cin>>num1>>num2;
sum = solveMeFirst(num1,num2);
cout<<sum;
return 0;
}
Line: 21 Col: 1
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Compiler Message


HackerRank
Prepare
Algorithms
Sorting
Quicksort 1 - Partition
Exit Full Screen View
Problem
Submissions
Leaderboard
Discussions
The previous challenges covered Insertion Sort, which is a simple and intuitive sorting algorithm with a running time of $O(n^2)$. In these next few challenges, we're covering a divide-and-conquer algorithm called Quicksort (also known as Partition Sort). This challenge is a modified version of the algorithm that only addresses partitioning. It is implemented as follows:
Step 1: Divide
Choose some pivot element, p, and partition your unsorted array, arr, into three smaller arrays: $left$, $right$, and $equal$, where each element in $left < p$, each element in $right > p$, and each element in $equal = p$.
Example
$arr = [5, 7, 4, 3, 8]$
In this challenge, the pivot will always be at $arr[0]$, so the pivot is 5.
arr is divided into $left = \{4, 3\}$, $equal = \{5\}$, and $right = \{7, 8\}$.
Putting them all together, you get $\{4, 3, 5, 7, 8\}$. There is a flexible checker that allows the elements of $left$ and $right$ to be in any order. For example, $\{3, 4, 5, 8, 7\}$ is valid as well.
Given arr and $p = arr[0]$, partition arr into $left$, $right$, and $equal$ using the Divide instructions above. Return a 1-dimensional array containing each element in $left$ first, followed by each element in $equal$, followed by each element in $right$.
Function Description
Complete the quickSort function in the editor below.
quickSort has the following parameter(s):
18 vector<int> left, equal, right;
20 for(int num : arr) {
21 if(num < pivot) {
22 left.push_back(num);
23 } else if(num == pivot) {
24 equal.push_back(num);
Line: 115 Col:
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Test case 2
Compiler Message
Success
Input (stdin)
Download