



[BACK](#)

Shuffled Array

 177,554



[<](#) [DESCRIPTION](#) [MY SOLUTIONS](#) [LEADERBOARD](#) [COMMENTS](#) [README](#) [>](#)

[CODEWRITING](#) [SCORE: 300/300](#)

A noob programmer was given two simple tasks: sum and sort the elements of the given array $a = [a_1, a_2, \dots, a_n]$. He started with summing and did it easily, but decided to store the sum he found in some random position of the original array which was a bad idea. Now he needs to cope with the second task, sorting the original array a , and it's giving him trouble since he modified it.

Given the array `shuffled`, consisting of elements $a_1, a_2, \dots, a_n, a_1 + a_2 + \dots + a_n$ in random order, return the sorted array of original elements a_1, a_2, \dots, a_n .

Example

- For `shuffled = [1, 12, 3, 6, 2]`, the output should be `shuffledArray(shuffled) = [1, 2, 3, 6]`.
 $1 + 3 + 6 + 2 = 12$, which means that $1, 3, 6$ and 2 are original elements of the array.
- For `shuffled = [1, -3, -5, 7, 2]`, the output should be `shuffledArray(shuffled) = [-5, -3, 2, 7]`.

Input/Output

- [execution time limit] 4 seconds (js)**
- [input] array.integer shuffled**
Array of at least two integers. It is guaranteed that there is an index i such that $shuffled[i] = shuffled[0] + \dots + shuffled[i - 1] + shuffled[i + 1] + \dots + shuffled[n]$.

Guaranteed constraints:
 $2 \leq shuffled.length \leq 15$,
 $-300 \leq shuffled[i] \leq 300$.
- [output] array.integer**
A sorted array of `shuffled.length - 1` elements.

[JavaScript (ES6)] Syntax Tips

```
// Prints help message to the console
// Returns a string
function helloWorld(name) {
  console.log("This prints to the console when you Run Tests");
  return "Hello, " + name;
}
```