

As you may know, Bart has recently (painstakingly) ensured that every seat at his round poker table has the same number of poker chips. But unbelievably, when Bart’s back was turned, someone redistributed all the chips *again* such each seat does not have the same amount of chips! Bart needs to restore the balance of his table—quickly. Fortunately, with the practice he gained from the last incident, Bart no longer needs to move poker chips one at a time. He can now move all of the desired chips from one pile to an adjacent pile in one fell swoop! (He still only moves chips between adjacent seats, to be safe.) Given Bart’s impressive new skill, what is the minimum number of chip moves Bart will need to make to once again distribute the chips evenly?

Example

For `chips = [4, 5, 6, 10, 5]` , the output should be `pokerChips2(chips) = 3` .

The array represents a circular table, so we are permitted to move chips between the last and the first index in the array. Thus Bart can bring the chips back to equilibrium with the following 3 steps (1 -indexed):

- move 3 chips from seat 5 to seat 1 ;
- move 4 chips from seat 4 to seat 5 ;
- move 1 chip from seat 1 to seat 2 .

After this sequence of 3 moves, each seat will have 6 chips, and there is no sequence of fewer moves doing the same.

Input/Output

- **[execution time limit] 4 seconds (js)**
- **[input] array.integer chips**

The chip counts of each seat.

Guaranteed constraints:

$0 \leq \text{chips.length} \leq 10^6,$
 $0 \leq \text{chips}[i] \leq 10^5.$

- **[output] integer**

The minimum number of moves required to restore the chip counts.

[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;
}
```