List-2 > centered_average

prev | next | chance

Return the "centered" average of an array of ints, which we'll say is the mean average of the values, except ignoring the largest and smallest values in the array. If there are multiple copies of the smallest value, ignore just one copy, and likewise for the largest value. Use int division to produce the final average. You may assume that the array is length 3 or more.

centered_average([1, 2, 3, 4, 100]) → 3
centered_average([1, 1, 5, 5, 10, 8, 7]) → 5
centered_average([-10, -4, -2, -4, -2, 0]) → -3

Go ...Save, Compile, Run (ctrl-enter)

```
def centered_average(nums):
    diff_array = sorted(nums)
    del diff_array[0]
    del diff_array[-1]
    return sum(diff_array)/len(diff_array)
```

Go

Editor font size %: 100

Shorter output

Expected	Run		
centered_average([1, 2, 3, 4, 100]) → 3	3	OK	
centered_average([1, 1, 5, 5, 10, 8, 7]) → 5	5	OK	
centered_average([-10, -4, -2, -4, -2, 0]) → -3	-3	OK	
centered_average([5, 3, 4, 6, 2]) → 4	4	OK	
centered_average([5, 3, 4, 0, 100]) → 4	4	OK	
centered_average([100, 0, 5, 3, 4]) → 4	4	OK	
centered_average([4, 0, 100]) → 4	4	OK	
centered_average([0, 2, 3, 4, 100]) → 3	3	OK	
centered_average([1, 1, 100]) → 1	1	OK	
centered_average([7, 7, 7]) → 7	7	OK	
centered_average([1, 7, 8]) → 7	7	OK	
centered_average([1, 1, 99, 99]) → 50	50	OK	
centered_average([1000, 0, 1, 99]) → 50	50	OK	
centered_average([4, 4, 4, 4, 5]) → 4	4	OK	
centered_average([4, 4, 4, 1, 5]) → 4	4	OK	
centered_average([6, 4, 8, 12, 3]) → 6	6	OK	
other tests		OK	

All Correct

next | chance

Python > List-2

done page

Code is saved so long as this session is active. Create an account above to save code past this session.

Your progress graph for this problem

Forget It! -- delete my code for this problem

Progress graphs:
Your progress graph for this problem
Random user progress graph for this problem
Random Epic Progress Graph

Python Help

- Python Example Code
- Python Strings
- Python Lists
- Python If Boolean
- Code Badges

Difficulty: 209.0