**Sum without highest and lowest number**

1592787% of 1,322600 of7,608[SteffenVogel\_79](http://www.codewars.com/users/SteffenVogel_79)

C#

* [TRAIN AGAIN](http://www.codewars.com/kata/sum-without-highest-and-lowest-number/train/csharp)
* [NEXT KATA](http://www.codewars.com/trainer/csharp)

Details

[Solutions](http://www.codewars.com/kata/sum-without-highest-and-lowest-number/solutions/csharp)

[Forks (34)](http://www.codewars.com/kata/sum-without-highest-and-lowest-number/forks/csharp)

[Discourse (232)](http://www.codewars.com/kata/sum-without-highest-and-lowest-number/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Sum all the numbers of the array (in F# and Haskell you get a list) except the highest and the lowest element (the value, not the index!).  
(The highest/lowest element is respectively only one element at each edge, even if there are more than one with the same value!)  
  
Example:

{ 6, 2, 1, 8, 10 } => 16

{ 1, 1, 11, 2, 3 } => 6

If array is empty, null or None, or if only 1 Element exists, return 0.  
**Note:**In C++ instead null an empty vector is used. In C there is no null. ;-) 

-- There's no null in Haskell, therefore Maybe [Int] is used. Nothing represents null.

Have fun coding it and please don't forget to vote and rank this kata! :-)

I have created other katas. Have a look if you like coding and challenges.

<http://www.codewars.com/kata/sum-without-highest-and-lowest-number/csharp>

public static int Sum(int[] numbers)

{

return numbers == null || numbers.Length < 2

? 0

: numbers.Sum() - numbers.Max() - numbers.Min();

}