868. Maximum Average Subarray I

* [Description](http://www.lintcode.com/en/problem/maximum-average-subarray-i/" \l "description)
* [Notes](http://www.lintcode.com/en/problem/maximum-average-subarray-i/#note)
* [Testcase](http://www.lintcode.com/en/problem/maximum-average-subarray-i/#testcase)
* [Judge](http://www.lintcode.com/en/problem/maximum-average-subarray-i/#judge)

[Discuss](http://www.lintcode.com/discuss)

Given an array consisting of nintegers, find the contiguous subarray of given length k that has the maximum average value. And you need to output the maximum average value.

 Notice

1. 1 <= k <= n <= 30,000.
2. Elements of the given array will be in the range [-10,000, 10,000].

Have you met this question in a real interview?

Yes

**Example**

Given nums = [1,12,-5,-6,50,3], k = 4, return 12.75.

Explanation:

Maximum average is (12-5-6+50)/4 = 51/4 = 12.75

[**Tags**](http://www.lintcode.com/en/problem/maximum-average-subarray-i/#tags)

[**Array**](http://www.lintcode.com/tag/array/) [**Google**](http://www.lintcode.com/tag/google/)

[**Related Problems**](http://www.lintcode.com/en/problem/maximum-average-subarray-i/#related)

[**Medium**Maximum Average Subarray II](http://www.lintcode.com/problem/maximum-average-subarray-ii)

<http://www.lintcode.com/en/problem/maximum-average-subarray-i/#>

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

**package** javaapplication23;

***/\*\****

***\****

***\* @author Usuario***

***\*/***

**public** **class** JavaApplication23 {

***/\*\****

***\* @param args the command line arguments***

***\*/***

**public** **static** **double** findMaxAverage(**int**[] nums, **int** k) {

*// Write your code here*

**double** sum =0;

**double** max\_sum =0;

**for**(**int** i =0; i<k; i++) {

             sum += nums[i];

         }

         max\_sum = sum ;

**for**(**int** i = 0; i < nums.length-k; i++) {

             sum -= nums[i];

             sum += nums[i+k];

             max\_sum = Math.max(max\_sum, sum);

         }

*//max\_sum = Math.max(max\_sum, sum);*

*//System.out.println(max\_sum);*

**return** max\_sum/k;

    }

**public** **static** **void** main(String[] args) {

*// TODO code application logic here*

**int**[] nums = {1,12,-5,-6,50,3};

**int** k = 4;

       System.out.println( findMaxAverage(nums, k));

    }

}