Problem D. Machine Learning

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	256 mebibytes

Lately, Byton has found interest in the science describing methods of teaching computers identifying patterns in data and drawing conclusions from them – *the machine learning*.

During his research in this field, he had to investigate properties of some complicated function f. He computed its value in a number of points $x_1, x_2 \ldots, x_n$, obtaining results $y_1, y_2, \ldots, y_n$.

He would like to approximate f by some **continuous** function g, composed of two linear parts; formally for some $x \in \mathbb{R}$, g should be linear for arguments less than x and linear for arguments greater than x.

Byton would like to achieve a faithful approximation of f. He would like to minimize the mean squared error:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - g(x_i))^2.$$

Input

The first line of the input contains a single integer n ($1 \le n \le 100\,000$). Each of the next n lines contain two integers x_i, y_i ($0 \le x_i \le 1\,000\,000$, $0 \le y_i \le 1000$). The numbers x_i are pairwise different.

Output

You should print a single real number – the minimum possible mean squared error he is able to achieve.

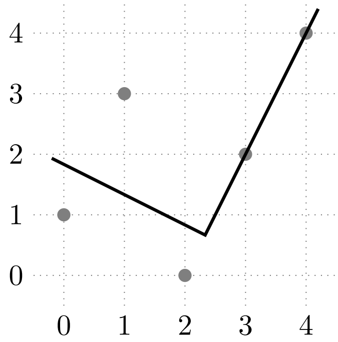
Your answer will be accepted if its absolute error does not exceed 1.

Example

standard input	standard output
5 0 1 2 0 1 3 4 4 3 2	0.8333333333333
7 0 0 1 1 2 2 3 4 4 2 5 1 6 0	0.0659340659341

Note

In the first example, the optimal mean squared error is $\frac{5}{6}$. You can get it by fixing on the left the linear function $-\frac{x}{2} + \frac{11}{6}$ and on the right, the linear function $2x - 4$.



In the second example the minimum mean squared error is $\frac{6}{91}$. The function can be constructed from lines $\frac{16}{13}x - \frac{2}{13}$ and $-\frac{16}{13}x + \frac{94}{13}$.