Closest Triplet

Closest pair is an old problem that asks to find, given a set of N points in the plane, the pair that minimizes the distance between them. This problem can easily be solved using roughly N 2 operations by testing all possible pairs of points and keeping at each step the optimal pair. With a more clever approach, the problem has been solved using ~ N log N operations.
Closest triplet is an analogous problem which also takes a set of N points as input, and asks for the triplet (group of three points) that minimizes the sum of the three distances between each pair of them. In this case there is also a trivial solution that tests all possible triplets using roughly N^3 operations. However, since you are a clever programmer, we are confident that you are able to find a better algorithm.

Input

The input contains several test cases, each one described in several lines. The first line contains an integer N indicating the number of points in the set ($3 \leq N \leq 3000$). Each of the next N lines describes a different point of the set using two integers X and Y separated by a single space ($1 \leq X, Y \leq 10^6$); these values represent the coordinates of the point in the XY plane. You may assume that within each test case no two points have the same location. The last line of the input contains a single –1 and should not be processed as a test case.

Output

For each test case output a single line with a real number representing the sum of the distances between each pair of points of any closest triplet of the set of points. Round the result to the closest rational number with three decimal places. In case of ties, round up. Always use exactly three digits after the decimal point, even if it means finishing with a zero.

Example

Input:

```
4
1 1
4 1
1 5
1000 1000
9
100000 200000
200000 200000
150000 286603
60000 140000
240000 140000
150000 340000
1 340000
300000 340000
150000 87087
-1
```

Output:
12.000
300000.796