UCP-Clustering

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

Clustering algorithms are algorithms that partition data into subsets named *clusters*, intending to place similar data into the same cluster and different data into different clusters.

Professor Jihoon Ko discovered the revolutionary clustering algorithm *UCP-Clustering*. Given N distinct points in a 2-dimensional space, the algorithm partitions the points into K clusters with the following algorithm.

- Each cluster is assigned a *representative coordinate* (RC) which will be used later. Initially, K points from the given N points are chosen, and each cluster's RC corresponds to one of the chosen points. Even though the coordinates are chosen from the points, all clusters are empty at this point.
- Repeat the following algorithm `recompute`:
  - For each point, insert it into the cluster which minimizes the distance to its RC. If there is more than one such cluster, pick the one with the smallest RC per their lexicographic order.
  - For each cluster, recompute its RC. The x-coordinate of the new RC is the median of the x-coordinates of points in the cluster. Similarly, the y-coordinate is the median of the y-coordinates.
  - If none of the clusters had their RC change, the algorithm terminates. Otherwise, empty all the clusters and run the `recompute` algorithm again (emptying does not reinitialize the RC).

Here, the following definitions are used:

- Point (x_1, y_1) is lexicographically smaller than (x_2, y_2) if and only if $x_1 < x_2$ or $x_1 = x_2$ and $y_1 < y_2$.
- For a sequence of length $n > 0$, the median is defined as the $(n+1)/2$-th largest value if n is odd, and as the **average** of the $n/2$-th and $n/2+1$-th largest values if n is even.

For example, consider the case where $N = 3$ points $\{(1,2),(3,4),(5,6)\}$ are clustered into $K = 2$ sets. Suppose that the points $(1,2)$ and $(3,4)$ are selected as RCs. The first iteration of the `recompute` algorithm clusters the points into $\{(1,2)\}$ and $\{(3,4),(5,6)\}$ with their RCs being $(1,2)$ and $(4,5)$. The second iteration does not change the RCs, thus the `recompute` algorithm is executed twice. Suppose that the points $(1,2)$ and $(5,6)$ are selected as RCs. Then the first iteration of the `recompute` algorithm clusters the point into $\{(1,2),(3,4)\}$ and $\{(5,6)\}$ with their RCs being $(2,3)$ and $(5,6)$. The second iteration does not change the RCs, thus the `recompute` algorithm is executed twice. From the above example, we can see that the initial choice of the RC may change the result of the clustering.

In this problem, let's focus on the case $K = 2$. Suppose that all $N(N-1)/2$ possible initial representative coordinates are chosen for each cluster equiprobably. Please find all the possible sets of representative coordinates at the end of the algorithm, and the expected value of execution count for *recompute* if the algorithm reaches this set of points.

Input

The first line contains a single integer N ($2 \leq N \leq 512$).

The next N lines contain two integers x_i and y_i denoting point (x_i, y_i). All points are distinct ($-10^6 \leq x_i, y_i \leq 10^6$).

Output

Output all the possible sets of representative coordinates, one per line, in the following format.

Let $(x_1, y_1), (x_2, y_2)$ be the final representative coordinates. Output five real numbers x_1, y_1, x_2, y_2, E where (x_1, y_1) is lexicographically smaller than (x_2, y_2), and E is the expected value of execution count for *compute* if the algorithm reaches this set of points.

The sets must be printed in lexicographic order by (x_1, y_1), with sets that have the same representative coordinate for (x_1, y_1) printed in lexicographic order by (x_2, y_2).

All values must be within an absolute or relative error of 10^{-6}.

The input data are designed such that any choice of initial RC will not make the algorithm reach some state where any of the following conditions is true:

- Some cluster becomes empty.
- The *recompute* algorithm is invoked infinitely many times.
- Two or more clusters have the same RC.

Examples

standard input	standard output
4 0 0 0 3 3 0 3 3	0.000000 0.000000 3.000000 3.000000 1.000000000000 0.000000 1.500000 3.000000 1.500000 2.000000000000 0.000000 3.000000 3.000000 0.000000 1.000000000000 1.500000 0.000000 1.500000 3.000000 2.000000000000
3 1 2 3 4 5 6	1.000000 2.000000 4.000000 5.000000 2.000000000000 2.000000 3.000000 5.000000 6.000000 2.000000000000