Clustering

If you remember the task Sheep you should know that Elly often dreams how she spends her time with her sheep on the pasture. Doing this is not an easy task though. In addition to her sheep she also has hounds to guard the sheep from wolves and thieves. The danger for each sheep can be measured as the distance to the closest hound – the less, the better. The danger for the flock can be measured of the sum of these distances for each sheep.
If we consider the pasture as flat surface, we can represent the sheep as **N** points in the plane. Elly wonders how to place her guarding hounds (represented as **K** points) such that the sum of minimal distances of each sheep to the closest one is as low as possible. In other words, you are given **N** points through their **X**- and **Y**-coordinates. You should place **K** new points in such way that the sum of the minimum distances from each of the given points to the closest of the new ones is as low as possible.

Input
On the first line of the standard input are given the integers **N** and **K** – the number of sheep and the number of hounds, respectively. On each of the next **N** lines are given two integers $\mathbf{X_i}$ and $\mathbf{Y_i}$ – the coordinates of the i^{th} sheep.

Output
On the standard output print **K** pairs of real numbers – the coordinates of the hounds. It is allowed that a hound is placed on the same coordinate as a sheep.

Constraints
$1 \leq \mathbf{K} < \mathbf{N} \leq 1000$
$1 \leq \mathbf{K} \leq 100$
$0 \leq \mathbf{X_i}, \mathbf{Y_i} \leq 10000$

Scoring
For each test case your solution will be awarded:
`round(min(1, (author_score / your_score))` ^{2} `* test_score)`
points, where author_score is the result, found by the author's solution, your_score is the result of your solution, and test_score is the maximal score for the given test case.

Sample Input:	*Sample Output:*
7 2 1 2 1 4 2 5 3 2 4 4 5 6 6 5	1.750000 3.250000 5.000000 5.000000

Note that you can print sub-optimal solution and still get points.