ELEVENTH INTERNATIONAL AUTUMN TOURNAMENT IN INFORMATICS "JOHN ATANASOV"
Shumen, 26.11.2011

Group A

Problem A2. CLUSTERING

Elly often 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 a 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) so that the sum of minimal distances of each sheep to the closest hound was as small as possible. In other words, you are given *N* points by their *X*- and *Y*-coordinates. You should place *K* new points in such a way that the sum of the minimum distances from each of the given points to the closest of the new ones was as small as possible. Write a program **`clustering`** to solve this problem.

Input

On the first line of the standard input are given the space separated positive integers *N* and *K* – the number of sheep and the number of hounds, respectively. On each of the next *N* lines are given two space separated integers X_i and Y_i – the coordinates of the ith sheep.

Output

On the standard output print *K* pairs of real numbers – the coordinates of the hounds, separated with a space, formatted to the sixth digit after the decimal point. It is allowed for a hound to be placed on the same coordinates as a sheep.

Constraints

$1 \le K < N \le 1000$
$1 \le K \le 100$
$0 \le X_i, Y_i \le 10000$

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.

Example

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

Clarification: Note that you are not required to print an optimal solution.