segments

Vlad really enjoys drawing points on a piece of paper. He has been doing it all day long, but now he decides that he wants to join some of these points using straight lines. There are *N* such points, numbered from *0* to *N-1*, and all of them have integer coordinates. Vlad wants to draw exactly *K* distinct line segments and for each line segment he chooses both endpoints from the set of points that he has already drawn. But that is not all - he also wants to minimize the total length of the line segments, which is the sum of all the lengths of the *K* line segments.

Task

You are going to help Vlad to determine the total minimum length as well as all the mandatory points. A point is *mandatory* if and only if it will be used as an endpoint of a line segment to at least one of the *K* line segments, no matter how we choose them with respect to the minimal length condition. So, if we would erase a mandatory point we wouldn't be able to obtain the same minimum length.

Input data

The input file `segments.in` will have on the first line two positive integers: *N* and *K*. The following *N* lines will contain the coordinates of each point. All coordinates are integers. Keep in mind that the points are numbered from 0 to N-1!

Output data

On the first line of the output file `segments.out` you should print the total minimum length, with an error of at most 10^{-4} (the first 4 decimals should be correct). The next lines will contain the indices of the mandatory points, sorted in ascending order. For details please refer to the example below.

Limits and constraints

- $2 \leq N \leq 5000$
- $1 \leq K \leq N(N+1)/2$
- $1 \leq K \leq 100$
- All the coordinates are integers in the range $[-10^9, 10^9]$.
- The coordinates of the points are not necessarily distinct.
- A point may be used as an endpoint for any number of line segments.
- Time limit: 1.1 seconds
- Memory limit: 16 MB

Example

segments.in	segments.out
6 3 1 2 4 5 2 3 1 1 8 8 7 4	4.65028 0 2 3