2002/2003 ACM International Collegiate Programming Contest   
University of Ulm Local Contest

# Problem D: Diplomatic License

Source file: diplomatic.(c|cc|hs|java|pas)   
Input file: diplomatic.in

In an effort to minimize the expenses for foreign affairs the countries of the world have argued as follows. It is not enough that each country maintains diplomatic relations with at most one other country, for then, since there are more than two countries in the world, some countries cannot communicate with each other through (a chain of) diplomats.

Now, let us assume that each country maintains diplomatic relations with at most two other countries. It is an unwritten diplomatic *"must be"* issue that every country is treated in an equal fashion. It follows that each country maintains diplomatic relations with exactly two other countries.

International topologists have proposed a structure that fits these needs. They will arrange the countries to form a circle and let each country have diplomatic relations with its left and right neighbours. In the real world, the Foreign Office is located in every country's capital. For simplicity, let us assume that its location is given as a point in a two-dimensional plane. If you connect the Foreign Offices of the diplomatically related countries by a straight line, the result is a polygon.

It is now necessary to establish locations for bilateral diplomatic meetings. Again, for diplomatic reasons, it is necessary that both diplomats will have to travel equal distances to the location. For efficiency reasons, the travel distance should be minimized. Get ready for your task!

**Input Specification**

The input contains several testcases. Each starts with the number *n* of countries involved. You may assume that *n>=3* is an odd number. Then follow *n* pairs of x- and y-coordinates denoting the locations of the Foreign Offices. The coordinates of the Foreign Offices are integer numbers whose absolute value is less than *1012*. The countries are arranged in the same order as they appear in the input. Additionally, the first country is a neighbour of the last country in the list. Input is terminated by *EOF*.

**Output Specification**

For each test case output the number of meeting locations (*=n*) followed by the x- and y-coordinates of the locations. The order of the meeting locations should be the same as specified by the input order. Start with the meeting locations for the first two countries up to the last two countries. Finally output the meeting location for the *n*-th and the first country.

**Sample Input**

5 10 2 18 2 22 6 14 18 10 18

3 -4 6 -2 4 -2 6

3 -8 12 4 8 6 12

**Sample Output**

5 14 2 20 4 18 12 12 18 10 10

3 -3 5 -2 5 -3 6

3 -2 10 5 10 -1 12

**Remark**

Note that the output can be interpreted as a polygon as well. The relationship between the sample input and output polygons is illustrated in the figure on the page of Problem E. To generate further sample input you may use your solution to that problem.