Closed Fences

A closed fence in the plane is a set of non-crossing, connected line segments with N corners (3 < N < 200). The corners or vertices are each distinct and are listed in counter-clockwise order in an array $\{x_i, y_i\}$, i in (1..N).

Every pair of adjacent vertices defines a side of the fence. Thus $\{x_i\ y_i\ x_{i+1}\ y_{i+1}\}$ is a side of the fence for all i in (1..N). For our purposes, N+1 = 1, so that the first and last vertices making the fence closed.

Here is a typical closed fence and a point x,y:

```
                         * x3,y3
                 x5,y5  / \
   x,y *              *   /   \
                  / \ /     \
                 /   *       \
         x6,y6*   x4,y4     \
              |             \
              |              \
         x1,y1*----------------* x2,y2
```

Write a program which will do the following:

- Test an ordered list of vertices $\{x_i,y_i\}$, i in (1..N) to see if the array is a valid fence.
- Find the set of fence sides that a person (with no height) who is standing in the plane at position (x,y) can "see" when looking at the fence. The location x,y may fall anywhere not on the fence.

A fence side can be seen if there exists a ray that connects (x,y) and any point on the side, and the ray does not intersect any other side of the fence. A side that is parallel to the line of sight is not considered visible. In the figure, above the segments x3,y3-x4,y4; x5,y5-x6,y6; and x6-y6-x1,y1 are visible or partially visible from x,y.

PROGRAM NAME: fence4

INPUT FORMAT

Line 1:	N, the number of corners in the fence
Line 2:	Two space-separated integers, x and y, that are the location of the observer. Both integers will fit into 16 bits.
Line 3-N+2:	A pair of space-separated integers denoting the X,Y location of the corner. The pairs are given in counterclockwise order. Both integers are no larger than 1000 in magnitude.

NOTE: I have added anNew test case #12 for this task. Let me know if you think it's wrong. Rob Be sure to include USACO in your mail subject!

SAMPLE INPUT (file fence4.in)

```
13
5 5
0 0
7 0
5 2
7 5
5 7
3 5
4 9
1 8
2 5
0 9
-2 7
0 3
-3 1
```

OUTPUT FORMAT

If the sequence is not a valid fence, the output is a single line containing the word "NOFENCE".

Otherwise, the output is a listing of visible fence segments, one per line, shown as four space-separated integers that represent the two corners. Express the points in the segment by showing first the point that is earlier in the input, then the point that is later. Sort the segments for output by examining the last point and showing first those points that are earlier in the input. Use the same rule on the first of the two points in case of ties.

SAMPLE OUTPUT (file fence4.out)

```
7
0 0 7 0
5 2 7 5
7 5 5 7
5 7 3 5
-2 7 0 3
0 0 -3 1
0 3 -3 1
```