Problem I. Vieleck

Input file:	`vieleck.in`
Output file:	`vieleck.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

In order to generate random testcases for problem "Entfernung", we need a way to randomly generate polygons which are not necessarily convex (but of course without self-intersections and self-touchings).

At first, we've employed the following strategy to generate a polygon with n vertices:

1. Generate n independent points by picking each of $2n$ coordinates independently and uniformly at random from integers between -1000 and 1000.
2. In case any two points coincide or any three points are on the same line, go back to step 1.
3. Generate a random permutation of n points uniformly (each permutation is picked with the same probability). Connect those n points into a polygon in the order of the permutation, also connecting the last element of the permutation with the first element of the permutation.
4. In case the resulting polygon has self-intersections, go back to step 3, otherwise we're done!

After trying out this strategy for a while, we've noticed that it can take a really long time even for moderate values of n, because self-intersections are actually very likely when we connect the vertices in the order of a random permutation.

We've decided to improve the strategy as follows:

1. As before.
2. As before.
3. As before.
4. In case the resulting polygon does't have any self-intersections, we're done!
5. Otherwise, pick any pair of intersecting edges, uniformly at random from all such pairs. Remove those two edges. Connect the four loose ends with two edges in such a way that we still have one polygon, and it's different from the one we just had (there's exactly one such way), then go back to step 4.

It's possible to prove that this strategy always terminates, and in fact it terminates much faster than the first strategy.

However, it turns out that it tends to generate somewhat different polygons. Given a polygon, determine which strategy was used to generate it.

Input

The first line of the input file contains one integer t, the number of testcases. Each testcase starts with the number of vertices n on a line by itself, followed by n lines, each containing the coordinates of a vertex.

In non-sample inputs one of the two strategies was picked uniformly at random for each testcase, and the polygon was generated using the picked strategy.

In non-sample inputs t is always 1000 and n is always 12.

There are 10 non-sample inputs in this problem.

Output

Output t lines, each containing either the word "FIRST" (without quotes), or the word "SECOND" (without quotes) denoting the strategy picked to generate the corresponding polygon. Your output for non-sample inputs will be accepted if at least 666 out of 1000 strategies are guessed correctly. Any correctly formatted output will be accepted for the sample input.

Examples

vieleck.in	vieleck.out
2 4 0 0 0 10 10 10 10 0 4 0 0 10 0 1 1 0 10	SECOND FIRST