

# I Conduit

Irv Kenneth Diggitt works for a company that excavates trenches, digs holes and generally tears up people's yards. Irv's job is to make sure that no underground pipe or cable is underneath where excavation is planned. He has several different maps, one for each utility company, showing where their conduits lie, and he needs to draw one large, consolidated map combining them all. One approach would be to simply draw each of the smaller maps one at a time onto the large map. However, this often wastes time, not to mention ink for the pen-plotter in the office, since in many cases portions of the conduits overlap with each other (albeit at different depths underground). What Irv wants is a way to determine the minimum number of line segments to draw given all the line segments from the separate maps.

## Input

Input will consist of multiple input sets. Each set will start with a single line containing a positive integer  $n$  indicating the total number of line segments from all the smaller maps. Each of the next  $n$  lines will contain a description of one segment in the format

$$x_1 \ y_1 \ x_2 \ y_2$$

where  $(x_1, y_1)$  are the coordinates of one endpoint and  $(x_2, y_2)$  are the coordinates of the other. Coordinate values are floating point values in the range  $0 \dots 1001$  specified to at most two decimal places. The maximum number of line segments will be 10000 and all segments will have non-zero length. Following the last input set there will be a line containing a 0 indicating end of input; it should not be processed.

## Output

For each input set, output on a single line the minimum number of line segments that need to be drawn on the larger, consolidated map.

## Example

### Input:

```
3
1.0 10.0 3.0 14.0
0.0 0.0 20.0 20.0
10.0 28.0 2.0 12.0
2
0.0 0.0 1.0 1.0
1.0 1.0 2.15 2.15
2
0.0 0.0 1.0 1.0
1.0 1.0 2.15 2.16
0
```

### Output:

```
2
1
2
```