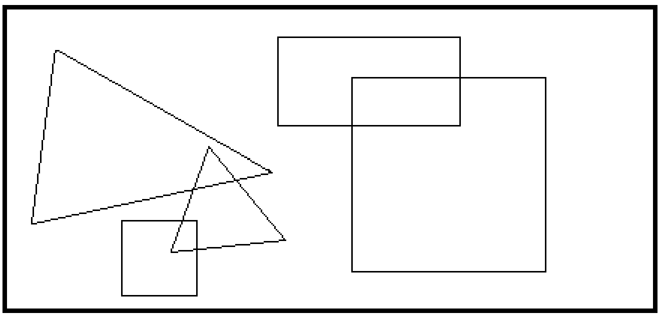
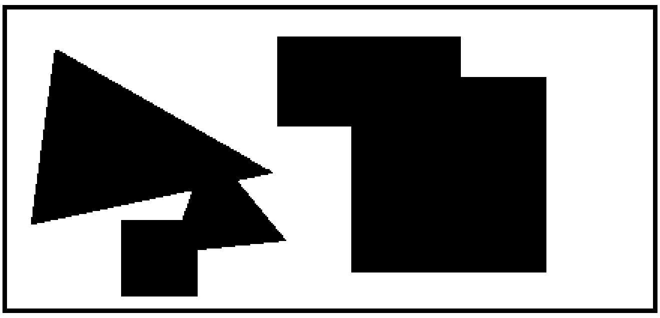
Lucca, my four-years-old daughter, loves drawing polygons in bond paper. For example, yesterday she drew two squares, a rectangle and two triangles:



Today, she wanted to fill her figures with black chinese ink. I helped her, obtaining the next result:



She asked me: how many black zones do you see?. I said: two. I'm bored answering the same question everyday. Can you help us writing a program that, given a collection of black filled polygons, determines the number of black zones on the drawing?

For a precise understanding: a *black zone* is a region of black coloured points on the sheet, where every pair of them may be connected by a continuous line within the region.

Input

The input consists of several test cases. Each test case is represented as follows:

- A line with an integer N ($1 \le N \le 40$) which indicates the number of polygons in the drawing.
- N lines, one per polygon, each one containing a list of $2 \cdot t$ integer numbers $x_1\ y_1\ x_2\ y_2\ \ldots\ x_t\ y_t$ specifying the points in the boundary of the polygon ($-10^4 \le x_i, y_i \le 10^4$, $3 \le t \le 10$). The drawn polygon is bounded by the closed path composed of the straight line segments $\overline{(x_1,y_1)(x_2,y_2)}$, $\overline{(x_2,y_2)(x_3,y_3)}$, ..., $\overline{(x_{t-1},y_{t-1})(x_t,y_t)}$, and $\overline{(x_t,y_t)(x_1,y_1)}$. You can suppose that the drawn polygon is a simple polygon (a polygon whose boundary is a non-self-intersecting closed path).

The end of the input is indicated when $N = 0$.

Output

For each case in the input, print one line with the number of black zones in the drawing after filling each one of the polygons with black chinese ink.

Sample Input

```
5
35 29 179 111 19 145
183 22 305 22 305 80 183 80
232 49 361 49 361 178 232 178
137 94 188 156 112 164
79 144 129 143 129 193 79 193
2
20 20 30 20 30 30 20 30
40 40 40 50 50 50 50 40
2
20 20 30 20 30 30 20 30
30 30 40 30 40 40 30 40
3
20 20 40 20 40 40 20 40
50 30 60 20 70 50
60 40 50 30 30 50
0
```

Sample Output

```
2
2
1
1
```