

Universe in the Cupboard

Input: stars.in

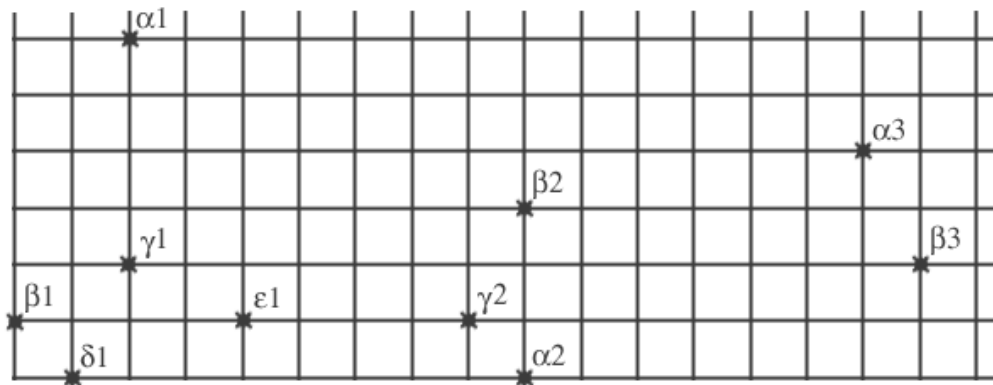
Little Omri has been playing with his magic cupboard again and has accidentally created a parallel universe, complete with stars, planets, people and everything. The people of the planet Parallearth gazed up at the sky on their first night of existence, and noticed that the stars appear to be in groups, which they chose to call *constellations*. Before the big vote to choose names for the constellations, they want to know how many there are, and that's where you come in.

The Parallearthlings have taken a photograph of the night sky, which renders all the bright stars in a rectangular frame. They then put a regular grid on the photograph so they could determine the coordinates of all of the stars. These coordinates have been written down in a list. Your program should process that list of two-dimensional coordinates and print out how many constellations can be made from the visible stars.

There are very simple rules for constructing constellations:

1. Every star is in the same constellation as its closest neighbor.
2. If a star doesn't have a unique closest neighbor—*i.e.* two or more neighbors are equally close and no others are closer—then it is in the same constellation as all of those closest neighbors.
3. If A is in the same constellation as B , then B is in the same constellation as A .
4. If A is in the same constellation as B , and B is in the same constellation as C , then A is in the same constellation as C .

For example, if the following were the picture of the night sky:



The first constellation consists of α_1 , β_1 , γ_1 , δ_1 , and ϵ_1 . Stars α_1 and γ_1 are in the same constellation because γ_1 is the closest star to α_1 (*i.e.*, by rule 1). Since γ_1 has three closest neighbors (β_1 , δ_1 , and ϵ_1), all of these are in the same constellation as γ_1 (by rule 2). Similarly, α_2 is the closest neighbor of γ_2 , and γ_2 is the closest neighbor to β_2 , so they are all in the same constellation (by rule 4). Finally, α_3 and β_3 are in a third constellation.

Input

The input to the program will consist of a sequence of universe specifications. Each begins with a single line containing a single integer n ($0 < n \leq 1000$), the number of stars in that universe.

There then follow n lines, each containing two integers x and y ($0 \leq x, y \leq 9999$), which are the coordinates of a star on the official photograph.

The last universe specification has $n = 0$, and should not be processed.

Output

For each universe specification, your program should print a single line that looks like:

Universe n contains c constellations

where n is replaced by the universe number (the first is 1), and c is replaced by the number of constellations.

Sample Input

```
10
0 1
16 2
1 0
2 6
9 0
4 1
2 2
8 1
9 3
15 4
5
10 10
10 11
20 10
20 11
15 5
0
```

Output Corresponding to Sample Input

```
Universe 1 contains 3 constellations
Universe 2 contains 1 constellations
```