

Once upon a time there lived a greedy landlord in a not far far away country. The landlord used to employ armed enforcers to ensure tax collection from his subjects. All his subjects were farmers and they depended solely on their crops to pay tax. In any single year, an individual farmer had crops in his field during a single continuous time interval. During this interval, an enforcer from the landlord was present in the farmer's field so that he could make sure to take away most of the produced crops to the landlord's barn. An enforcer could take care of at most one farmer at a time.

Now, in a glorious lazy morning, the landlord realized that a lot of his subjects were having intersecting production intervals. As he was determined to take the lion's share of crops from all the farmers, he sat down to determine the minimum number of enforcers he needed to ensure that no farmer remained unguarded while having crops in his field.

Oh your royal greediness! Don't you realize he who is greedy is always in want?

Input

There will be multiple test cases in the input file. A test case starts with an integer N ($0 \leq N \leq 1000$), denoting the number of farmers. Each of the following N lines contains 2 integers S & E ($0 \leq S \leq E \leq 31536000$), the starting & ending time respectively of the production interval of a farmer. Note that time is given in seconds from the start of the year.

The end of input will be denoted by a case with $N = -1$. This case should not be processed.

Output

For each test case, print a line in the format, 'Case X : Y ', where X is the case number & Y is the minimum number of enforcers required.

Sample Input

```
4
2 6
8 12
4 9
11 14
2
1 2
2 3
-1
```

Sample Output

```
Case 1: 2
Case 2: 2
```