

**2006/2007 SOUTHERN CALIFORNIA REGIONAL  
ACM INTERNATIONAL COLLEGIATE PROGRAMMING CONTEST**

**Problem 5  
Bad Birthday Present**

On my birthday my cousin Larry gave me a box inside a box inside a box inside a box inside a box, and there wasn't anything inside the innermost box! Now his birthday is coming up. Of course I'm not so crass or unimaginative as to do the same thing to him, but if I were, how many boxes could I force him to open? Your team is to write a program that will help me calculate that.

To do that, you'll be given several lists of boxes. For each list, your program's job will be to compute the greatest number of boxes in that list that can be nested. More specifically, you're to compute the size of the largest subset of the boxes in a list, such that the smallest box of the subset fits within the second smallest, the second smallest of the subset fits within the third smallest, the third smallest fits within the fourth smallest, and so forth.

Each dimension of a box is given as an integer between 1 and 999, inclusive. A box fits into a second if each dimension of the first is strictly smaller than the corresponding dimension of the second. A box can be rotated in any way necessary to allow it and a second box to nest, although it must be rotated through 90-degree angles—boxes are not to be rotated at odd angles when placed inside other boxes. Given  $n$  boxes, it's possible all  $n$  can be rotated so that all of them nest. It's also possible that no single box of the  $n$  can be rotated to fit into any other; in that case the largest number of nesting boxes is 1 (since any single standalone box forms a trivially nested singleton.)

Your program should read data from standard input. The data will contain one or more lists of boxes. The lists will immediately follow one another until the end of the file. The first line for each list is the number of boxes in the list, and each line that follows gives the dimensions of a single box in the list. No line will have any leading or trailing whitespace, and the integers of a box's dimensions will be separated from each other by a single blank.

For each list of boxes, you should print a single integer that is the maximum number of boxes that can be selected from the original set to form a fully nesting subset. The integer should be printed to standard output on a line by itself with no leading or trailing whitespace.

The sample input describes a list of five boxes followed by a list of four boxes. Since imaginary revenge is the most fun when you go completely overboard, expect that lists can contain anywhere from one to five hundred boxes.

*Sample Input*

```
5
145 472 812
827 133 549
381 371 900
271 389 128
718 217 491
4
432 123 139
942 844 783
481 487 577
677 581 701
```

*Output for the Sample Input*

```
2
4
```