Version Number

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

In the early days of large language models, researchers discovered an interesting phenomenon: when asked "Which is larger, 5.11 or 5.9", many models would confidently answer 5.11. It is suspected that the models confused the comparison rules for decimal numbers with those for version numbers — in version number comparison, 5.11 is indeed greater than 5.9, since the minor version 11 is greater than 9.

Each version number consists of two non-negative integers: a major version x and a minor version y, denoted as $x.y$.

The comparison rule for version numbers is as follows: first compare the major versions; the version with the larger major version is greater. If the major versions are equal, compare the minor versions; the version with the larger minor version is greater. Formally, version $x_1.y_1$ is greater than version $x_2.y_2$ if and only if $x_1 > x_2$, or $x_1 = x_2$ and $y_1 > y_2$.

Given n version numbers, you need to find the largest one among them.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 100$), indicating the number of test cases. For each test case:

The first line contains an integer n ($2 \le n \le 100$), indicating the number of version numbers.

For the following n lines, the i-th line contains two integers x_i and y_i ($0 \le x_i, y_i \le 10^3$), indicating the version number $x_i.y_i$.

Output

For each test case, output one line containing two integers x' and y' separated by a space, indicating the major version and the minor version of the largest version number.

Example

standard input	standard output
3 3 5 11 5 9 3 12 2 1 0 0 1 2 0 0 0 0	5 11 1 0 0 0