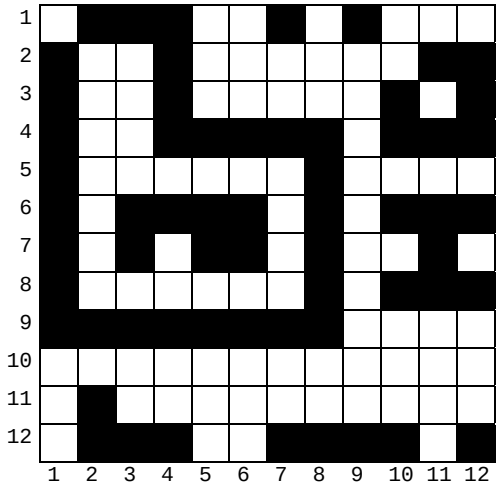
SHIPS

The board of "Ships" game consists of N*N squares. Each square may belong to some ship or be empty. If two squares belong to ships and have common edge, then both squares belong to the same ship. Squares of different ships have no common points. *Tonnage* of ship is number of squares belonging to this ship.

In the given example squares belonging to ships are marked black and on the game board there are: one 29-ton ship, three 7-ton ships, two 4-ton ships and three one-ton ships.



Task

Write program which for given description of game board calculates number of ships and tonnage of each ship.

Input

In the first line of text file `ships.in` one positive integer N (N<30000) is given.

In each of next N input file lines there is given information about one board row, consecutively describing groups of squares from left to right, which belongs to ships in one of two formats:

- `<number_of_square_column>`
  , if square in given column belongs to ship, but squares to the left and to the right are free;
- `<number_of_first_square_column>-<number_of_last_square_column>`
  , if all consecutive squares from first to last (including) belongs to ship and squares to the left and to the right from this group are free.

Square groups are separated by commas, each line ends with semicolon. There are no spaces in lines. If in some board row there are no ship's squares, then corresponding file line contains only one semicolon. It is known that total number of ships does not exceed 1000 and tonnage of any ship does not exceed 1000 tons.

Output

In the text file `ships.out` your program must output information about ships. In each file line there must be exactly two integers separated by space symbol. First number must be tonnage and second must be number of ships having this tonnage. Tonnages must be given in decreasing order and tonnage must be outputed only if there is at least one ship having this tonnage.

Example (corresponds to given example)

ships.in	ships.out
12 2-4,7,9; 1,4,11-12; 1,4,10,12; 1,4-8,10-12; 1,8; 1,3-6,8,10-12; 1,3,5-6,8,11; 1,8,10-12; 1-8; ; 2; 2-4,7-10,12;	29 1 7 3 4 2 1 3