

Registering Points scored by Basketball Players

A basketball agency is looking for the better basketball players around, so they would like to know the number of basketball points that each player made during a match.

In order to know this, they have typed the names of basketball players and the number of points that they scored in each event of the match.

Your company has asked you to provide an efficient implementation **using search trees** for solving this problem

You are asked to provide the names of basketball players with their points ignoring the ones that they did not score any point. The list of names should be ordered alphabetically.

Input

The first line will indicate the number of cases. Each case will represent a match with the following information with just one line:

n name1 points1 name2 points2 nameN pointsN

where

- n is the number of events
- name-ith is the name of basketball player of i-th event
- points-ith is the number of points scored in the i-th event

An example of input can be the following one:

10 Li 2 Juan 3 Lu 2 Maria 3 Lu 2 Li 1 Li 1 Lu 3 Lu 3 Li 2

Output

The output of each case should be printed in one line. The output of each case will have the following notation without any spaces:

(name2:sumPoints1)(name2:sumPoints2) . . . (nameM:sumPointsM)

The example of this output line for the aforementioned input example follows:

(Juan:3)(Li:6)(Lu:10)(Maria:3)

Implementation Details

In the virtual campus, there are the implementation of search trees and some supporting material with basic examples of using this implementation.

Example of input

```
3
10 Li 2 Juan 3 Lu 2 Maria 3 Lu 2 Li 1 Li 1 Lu 3 Lu 3 Li 2
0
14 Xi 2 Lu 3 Xi 2 Lu 3 Xi 2 Lu 3 Su 3 Su 3 Lu 2 Lu 3 Su 3 Su 3 Su 2 Li 1
```

Example of output

(Juan:3)(Li:6)(Lu:10)(Maria:3)
(Li:1)(Lu:14)(Su:14)(Xi:6)