

## Codeforces Beta Round 2

### A. Winner

1 second, 64 megabytes

The winner of the card game popular in Berland "Berlogging" is determined according to the following rules. If at the end of the game there is only one player with the maximum number of points, he is the winner. The situation becomes more difficult if the number of such players is more than one. During each round a player gains or loses a particular number of points. In the course of the game the number of points is registered in the line "name score", where name is a player's name, and score is the number of points gained in this round, which is an integer number. If score is negative, this means that the player has lost in the round. So, if two or more players have the maximum number of points (say, it equals to  $m$ ) at the end of the game, then wins the one of them who scored at least  $m$  points first. Initially each player has 0 points. It's guaranteed that at the end of the game at least one player has a positive number of points.

#### Input

The first line contains an integer number  $n$  ( $1 \leq n \leq 1000$ ),  $n$  is the number of rounds played. Then follow  $n$  lines, containing the information about the rounds in "name score" format in chronological order, where name is a string of lower-case Latin letters with the length from 1 to 32, and score is an integer number between -1000 and 1000, inclusive.

#### Output

Print the name of the winner.

input
3 mike 3 andrew 5 mike 2
output
andrew

input
3 andrew 3 andrew 2 mike 5
output
andrew

### B. The least round way

2 seconds, 64 megabytes

There is a square matrix  $n \times n$ , consisting of non-negative integer numbers. You should find such a way on it that

- starts in the upper left cell of the matrix;
- each following cell is to the right or down from the current cell;
- the way ends in the bottom right cell.

Moreover, if we multiply together all the numbers along the way, the result should be the least "round". In other words, it should end in the least possible number of zeros.

#### Input

The first line contains an integer number  $n$  ( $2 \leq n \leq 1000$ ),  $n$  is the size of the matrix. Then follow  $n$  lines containing the matrix elements (non-negative integer numbers not exceeding  $10^9$ ).

#### Output

In the first line print the least number of trailing zeros. In the second line print the correspondent way itself.

input
3 1 2 3 4 5 6 7 8 9
output
0 DDRR

### C. Commentator problem

1 second, 64 megabytes

The Olympic Games in Bercouver are in full swing now. Here everyone has their own objectives: sportsmen compete for medals, and sport commentators compete for more convenient positions to give a running commentary. Today the main sport events take place at three round stadiums, and the commentator's objective is to choose the best point of observation, that is to say the point from where all the three stadiums can be observed. As all the sport competitions are of the same importance, the stadiums should be observed at the same angle. If the number of points meeting the conditions is more than one, the point with the maximum angle of observation is preferred.

Would you, please, help the famous Berland commentator G. Berniev to find the best point of observation. It should be noted, that the stadiums do not hide each other, the commentator can easily see one stadium through the other.

#### Input

The input data consists of three lines, each of them describes the position of one stadium. The lines have the format  $x, y, r$ , where  $(x, y)$  are the coordinates of the stadium's center ( $-10^3 \leq x, y \leq 10^3$ ), and  $r$  ( $1 \leq r \leq 10^3$ ) is its radius. All the numbers in the input data are integer, stadiums do not have common points, and their centers are not on the same line.

#### Output

Print the coordinates of the required point with five digits after the decimal point. If there is no answer meeting the conditions, the program shouldn't print anything. The output data should be left blank.

input
0 0 10 60 0 10 30 30 10
output
30.00000 0.00000