

It's time for a new Formula 1 race and the board of directors decided that it's also time for a new chronometer system. They bought all the ultra-modern radar-laser-microwave sensors to install in the starting line, but just realized the manufacturer didn't provide the software to rank the cars.

That's how you come to the game. The board of directors wants you to write a program that receives the time each pilot spent on their qualification lap and prints the starting grid for the race (i.e. the order in which the pilots must start the race). And they want it fast! I bet you can do it in the time of a pit stop!

(If you don't know the rules that make the starting grid in Formula 1, consider that the least time the pilot spend on the qualification lap the best, i.e. the closest he will start of the beginning of the grid).

## Input

The input file contains several input sets. The description of each set is given below:

Each set starts with one integer  $N$  ( $1 \leq N \leq 100$ ), the number of participant pilots in the race, on the first line.

Then,  $N$  lines in the format ' $S : X \text{ min } Y \text{ sec } Z \text{ ms}$ '; will follow (quotes for clarity only). The value  $S$  represents the name of the pilot and consists of a string of 1 to 20 letters. The integers  $X$  ( $0 \leq C \leq 59$ ),  $Y$  ( $0 \leq C \leq 59$ ) and  $Z$  ( $0 \leq C \leq 999$ ) represent the number of minutes, seconds and milliseconds, respectively, that the pilot spent on his qualification lap.

There is a blank line after each input set. Input is terminated by EOF.

## Output

For each input set, your program must produce the starting grid, i.e. the order in which the pilots must start the race, in the following format. For each row in the grid (a pair of cars that start the race side by side), print a line containing "Row  $R$ ", where  $R$  represents the row number starting from 1, followed by two other lines containing the name of the pilots (as appear in the input) that start the race from that row (in order of classification). If one row contains only one pilot, just print one line after the row number. If there are two pilots with the same name on a test case, they are different pilots and the output must contain both pilots. If two or more pilots are tied in the classification time, sort them by their names (lexicographically, case-insensitive comparisons).

Print a blank line after each test case.

## Sample Input

```
3
Schumacher : 1 min 23 sec 172 ms
Barrichello : 2 min 12 sec 999 ms
Senna : 0 min 55 sec 582 ms
```

```
4
Schumacher : 1 min 23 sec 172 ms
Barrichello : 2 min 12 sec 999 ms
Senna : 0 min 55 sec 582 ms
Fangio : 1 min 03 sec 000 ms
```

```
2
BadPilot : 59 min 59 sec 999 ms
ABadPilot : 59 min 59 sec 999 ms
```

## Sample Output

```
Row 1
Senna
Schumacher
Row 2
Barrichello
```

```
Row 1
Senna
Fangio
Row 2
Schumacher
Barrichello
```

```
Row 1
ABadPilot
BadPilot
```