UDEM

Actividad 1.1

Programación Avanzada

Instrucciones para envío en Blackboard

Guarda tus archivos con el nombre según la siguiente regla:

- M<matrícula>pX.cpp, donde <matrícula>corresponde a los 6 dígitos de su mátrícula UDEM y X al número de programa.
- Ejemplo: Si tu matrícula es 123456, el archivo para el problema 2, se deberá llamar M123456p2.cpp o .java
- Incluye el Código de Honor en cada programa.
- Si realizas los programas en pareja, basta que lo suba un solo estudiante. Pero deberás poner **ambas matrículas y nombres** en los comentarios al subir los programas en Blackboard.
- **Nota: Sino sigues esta regla, tus programas NO serán calificados.

Resolver los siguientes problema de programación.

Problem 1. *Store*

Standard input
Time limit: 3 seconds

Last year Mr. Goodman opened an online product store. Mr. Goodman decided to hire a novice programmer to develop the website. Recently some of the new customers complained about the electronics invoices. The programmer forgot to use plural words in case an item is purchased multiple times. Unfortunately the programmer hired by Mr. Goodman was fired and now he has hired you to fix this feature. Below is a description of how to determine the plural form of a word.

1. If the word is in an irregular word list, replace it with the indicated plural.
2. Else if the word ends in a consonant followed by the letter "y", replace "y" with "ies".
3. Else if the word ends in "o", "s", "ch", "sh" or "x", add "es" to the word.
4. Otherwise, add "s" to the word.

Input

The input begins with two integers L and N ($0 \leq L \leq 20$, $1 \leq N \leq 100$). The following L lines list the description of irregular words and their plural form. Each line consists of two words separated by a single space, where the first word is the singular form, and the second word is the plural form of the word irregular. After the list of irregular words, the next N lines contain one singular word, from which its plural must be obtained. You can assume that each word consists of a maximum of 40 lowercase letters of the English alphabet ("a" - "z").

Output

Tu programa deberá imprimir N líneas , donde la i-ésima línea es la forma plural de la i-ésima palabra de entrada.

Sample Input

```
3 7
goose geese
cactus cacti
spaghetti spaghetti
spaghetti
toy
cactus
cheese
fish
goose
strawberry
```

Sample Output

```
spaghetti
toys
cacti
cheeses
fishes
geese
strawberries
```

Problema 2. *Formula 1*

Standard Input
Time limit: 3.000 seconds

Formula 1 race is here and the board of directors decided that is 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.

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!

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 min Y sec Z 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 $(0 \leq X \leq 59)$, Y $(0 \leq Y \leq 59)$ and Z $(0 \leq Z \leq 999)$ 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
```