

The Lunch Line

Filename: lunchline

The CEO of Elderly Food Catering Service has decided that when old people go to get food at his restaurants that they should be served in order of age. As everyone knows older people get sick of waiting faster than younger people, so older people should get to eat first at Elderly Food Catering Service. It is also well known that old people are stubborn and once they enter a line they will refuse to leave it until they are served. The problem with this system is that it is hard to keep the line organized! This is difficult because people can arrive at different times. The CEO has tasked you to design a system to organize the line for the company. This system needs to be able to support two kinds of queries, one where someone new enters the line and getting the first person in line to get food.

The Problem

Given a number of events where people either enter the line or the first person in line is called to get food, print out the names of people who are served in the order they are served food.

The Input

The first line of the input file will contain a number n indicating the number of restaurants that the Elderly Food Catering Service runs. For each restaurant a number q will be given indicating the number of events that will occur. Each event will start with a number id , which will indicate which kind of event it is. If id is 1 than someone new has arrived in line and the next string on the line will be their name (which is always unique) followed by their birthdate (in $mm/dd/yyyy$ format), their birthdates will also be unique. If id is 2 simply print the name of the person who is served next. All birthdates will correspond to valid days with A.D. years.

The Output

The first line of the output should be “Restaurant # r ” where r indicates which restaurant is currently being considered. For each query of type 2 print the name of the person who is served. It is guaranteed that if there is a query of type 2 that there is still someone in line. Print a blank line in between each restaurant.

Sample Input

```
2
6
1 Danny 05/22/1993
1 Sawyer 04/23/1994
2
1 Arup 09/14/1975
2
2
4
1 Sawyer 04/23/1994
2
1 Danny 05/22/1993
2
```

Sample Output

Restaurant #1

Danny

Arup

Sawyer

Restaurant #2

Sawyer

Danny