



3484 - Auction

Asia - Hangzhou - 2005/2006

Recently the auction house has introduced a new type of auction, the lowest price auction. In this new system, people compete for the lowest bid price, as opposed to what they did in the past. What an amazing thing! Now you could buy cool stuff with one penny. Your task is to write the software to automate this auction system.

First the auctioneer puts an upper limit on bid price for each item. Only positive price less than or equal to this price limit is a valid bid. For example, if the price limit is 100, then 1 to 100, inclusive, are all valid bid prices. Bidder can not put more than one bid for the same price on a same item. However they can put many bids on a same item, as long as the prices are different. After all bids are set, the auctioneer chooses the winner according to the following rules:

1. If any valid price comes from only one bidder, the price is a "unique bid". If there are unique bids, then the unique bid with the lowest price wins. This price is the winning price and the only bidder is the winning bidder.
2. If there are no unique bids, then the price with fewest bids is the winning bid. If there are more than one price which has the same lowest bid count, choose the lowest one. This price is the winning price. The bidder who puts this bid first is the winning bidder.

Given the price limit and all the bids that happen in order, you will determine the winning bidder and the winning price.

Input

Standard input will contain multiple test cases. The first line of the input is a single integer T ($1 \leq T \leq 10$) which is the number of test cases. T test cases follow, each preceded by a single blank line.

The first line of each test case contains two integers: U ($1 \leq U \leq 10,000$), the price upper limit and M ($1 \leq M \leq 100,000$), the total number of bids. M lines follow, each of which presents a single bid. The bid contains the bidder's name (consecutive non-whitespace characters) and the price P ($1 \leq P \leq U$), separated with a single space. All bids in the input are guaranteed to be valid ones.

Output

Results should be directed to standard output. Start each case with "Case # : " on a single line, where # is the case number starting from 1. Two consecutive cases should be separated by a single blank line. No blank line should be produced after the last test case.

For each test case, print the sentence "The winner is W ." on the first line, and "The price is P ." on the second. Replace W and P with the winning bidder's name and the winning price.

Sample Input

2

3 3

Alice 1
Bob 2
Carl 3

3 6
Alice 1
Alice 2
Alice 3
Bob 1
Bob 3
Carl 3

Sample Output

Case 1:
The winner is Alice.
The price is 1.

Case 2:
The winner is Alice.
The price is 2.

Hangzhou 2005-2006