

Kart Racing

Filename: karts

A certain mustachioed plumber and friends are about to have their annual kart race. The winner of the race will win the prized mushroom-shaped trophy. However, a certain dragonesque king of turtle-like creatures has decided he'd be better off making a fortune in coins by betting on the races instead of trying to win for himself. In the middle of the night, the evil dragon king sneaks into the garage and takes each kart for a test drive, measuring the kart's speed (oddly, each kart travels with a constant speed, without ever slowing down or speeding up). While the evil dragon king was out, he noticed a certain giant ape (who was wearing a tie) walking around the course eating giant bananas and then carelessly dropping the banana peels on the road.

The evil dragon king found that these banana peels could not be avoided by any kart, and that every peel made a kart slip for five seconds, effectively stopping it in place. Once a kart hits a banana peel, the peel is knocked from the road and subsequent karts will not be affected by it. Note that karts can freely pass other karts at any time. It is guaranteed that every course will only have one winner (no ties), and also guaranteed that only one kart will reach a given banana (two karts won't reach the same banana at the same time). In addition, bananas are very slippery so no two bananas will ever be at the same location.

The Problem:

In order to decide who to bet on, the evil dragon king has kidnapped you to write a program to determine who will win the race.

The Input:

The first line of the input will contain a single positive integer, y , representing the number of circuits that the evil dragon king wants tested. Each circuit will begin with a single positive integer, m ($m < 9$), on a line by itself. Each of the following m lines will contain the data for one racer: the name of the racer (a string containing only upper and lowercase characters and hyphens, from 1 to 30 characters long) and the speed of the racer (an integer between 1 and 30 meters/second, inclusive), each separated by a single space from each other. Luckily, no racers have the same name within a single circuit so each name will appear in the list once.

Following the m lines, there will be a line with a single integer, n ($0 < n < 11$), representing the number of courses on the race circuit. On each of the following n lines there will be one race course description: the name of the course (a string containing only upper and lowercase characters and hyphens, 1 to 30 characters long), the length of the course in meters (an integer between 1 and 1000, inclusive), an integer b ($0 \leq b < 11$) representing the number of banana peels on the course, and finally b integers (each one between 1 and the course length, inclusive) denoting the distance from the track's starting point to a banana peel. Within a circuit, each course has a unique name. Each component of the course description will be separated by a single space.

The Output:

For each circuit, first output, on a line by itself, a header “Circuit #x:” where x represents the circuit number (beginning with 1). For each of the courses within the circuit, list the name of the course followed by a colon and one space, and then the name of the racer who first travels the entire length of the course followed by one space and the phrase “is the winner!”. Leave a blank line after the output for each circuit.

Sample Input:

```
2
3
Apple 9
Mario 10
Frog 5
2
Brother-Track 60 3 10 14 59
Rainbow-Track 600 4 1 184 397 436
2
Mario 30
Ape 4
1
Dinosaur-Track 200 2 194 14
```

Sample Output:

```
Circuit #1:
Brother-Track: Apple is the winner!
Rainbow-Track: Mario is the winner!

Circuit #2:
Dinosaur-Track: Mario is the winner!
```