

## Problem G

# Always Late

**Input:** standard input

**Output:** standard output

**Time Limit:** 1 second

**Memory Limit:** 32 MB

Ajmat: Have you noticed one member of team 13 comes late in almost every contest?

Nejhum: You mean the team of Mahbub, Moazzem and Yeamin? What's the problem with them?

Ajmat: All of them lives far away from BUET, and they always avoid the shortest path to BUET.

Nejhum: Are you kidding? Why would they do so? A computer science student won't ever do that.

Ajmat: They *always* avoids the shortest path. They believe that they always face the worst traffic jam on their shortest way to BUET.

Nejhum: That sounds crazy!

Ajmat: They are more crazy than you think they are. They always try to use the path whose length is smaller than or equal to that of any path to BUET except the shortest one. And for that, they sometimes visit the same junction more than once.

Nejhum: But they are not using the longest path. Then why are they late almost everyday?

Ajmat: That's because they have to spend a lot of time to find out which of the paths meet their criteria. Perhaps they don't know how to find out the second-shortest path.

Nejhum: I think we should solve this *real* problem in today's contest, instead of solving imaginary ones.

Ajmat: Let's try.

### Input

The first line of each test case contains two integers:  $n$  (the number of junctions,  $1 < n < 100$ ) and  $r$  (the number of bi-directional roads connecting these junctions). The junctions are numbered with  $0, 1, \dots, n-1$ . Each of the next  $r$  lines (one for each road) contains three positive integers: two junction-numbers that the corresponding road connects and the length of the road in kilometers. The next line contains an integer  $q$ , the number of queries. Each of the next  $q$  lines contains two junction-numbers. There is a blank line after the  $q$  queries. There is at most 1 road between each junction pair. A road always connects two different junctions. Length of a road is not more than 100km.

Input is terminated by EOF.

### Output

For each set of output, print the set # (1, 2, ... ) followed by  $q$  lines, one for each query, each containing the length of the second-shortest path between the corresponding junctions. However, for the unsolvable queries, print a '?'.

### Sample Input

```
4 3
0 1 12
0 2 20
1 2 15
3
0 1
0 2
0 3
```

```
4 3
0 1 11
0 2 20
1 2 15
3
0 1
0 2
0 3
```

### Sample Output

```
Set #1
35
27
?
Set #2
33
26
?
```

---

**Mustaq Ahmed**