ACM (Association of Car Modernization) has recently developed a new car, "MAGIC CAR". It uses solar energy. The car has some interesting characteristics :

- It uses up constant amount of energy at the start for any initial speed.
- It also uses up constant energy when stops.
- If it changes speed to a value which is less than already achieved least speed or greater than already achieved most speed, it uses up some energy. In both cases the energy is equal to the absolute difference of the current speed and previously achieved least or most speed.
- The loss of energy does not depend on the distance the car covered (Really magic!!!).

Mr. Oberoy has such a magic car. So far he has used the car quite intelligently so that minimum energy is used up. This was easy for him as he could take any speed on any road. But recently, TCD (Transport Control Department) has decided that there will be a fixed speed for each road in the city and everybody must maintain the speed. Mr. Oberoy is in problem now. Can you help him so that he can optimally use the car?

Input

Each dataset starts with two positive integer, N $(2 \leq N \leq 200)$ denoting the number of junctions and M $(1 \leq M \leq 1000)$ denoting the number of roads in Mr. Oberoy's city. Each junction is identified by a unique integer from 1 to N. In next few lines there will be road descriptions. A road is described by three positive integers which are start, end junctions and the fixed speed of that road. There may be more than one roads between two junctions. Roads are bidirectional. In the next line there will be two positive integers which are the used up energy during start and stop of the magic car. Next line will contain an integer K $(1 \leq K \leq 5)$ indicating the number of queries. Each of following K lines will contain two integers, the source and destination junction of Mr. Oberoy. Source and destination will not be same. Input is terminated by EOF.

Output

For each dataset print the minimum possible used up energy for each query of Mr. Oberoy. It is guranteed that the destination is always reachable from source.

Sample Input

```
4 4
1 2 2
2 3 4
1 4 1
3 4 2
5 5
2
1 3
1 2
```

Sample Output

```
11
10
```