We want to create a smartphone application to help visitors of a shopping mall and you have to calculate the shortest path between pairs of locations in the mall. Given the current location of the visitor and his destination, the application will show the shortest walking path (in meters) to arrive to the destination.

The mall has N places in several floors connected by walking paths, lifts, stairs and escalators (automated stairs). Note that the shortest path in meters may involve using an escalator in the opposite direction. We only want to count the distance that the visitor has walked so each type of movement between places has a different cost in meters:

- If *walking* or taking the *stairs* the distance is the euclidean distance between the points.
- Using the *lift* has a cost of 1 meter because once we enter the lift we do not walk at all. One lift can only connect 2 points. An actual lift connects the same point of different floors, in the map all the points connected by a lift have the corresponding edge. So you do not need to worry about that. For instance, if there are three floors and one lift at position (1,2) of each floor, the input contains the edges $(0, 1, 2) \rightarrow (1, 1, 2)$, $(1, 1, 2) \rightarrow (2, 1, 2)$ and $(0, 1, 2) \rightarrow (2, 1, 2)$. In some maps it can be possible that a lift does not connect all the floors, then some of the edges will not be in the input.
- The *escalator* has two uses:
  - Moving from `A` to `B` (proper direction) the cost is 1 meter because we only walk a few steps and then the escalator moves us.
  - Moving from `B` to `A` (opposite direction) has a cost of the euclidean distance between `B` and `A` multiplied by a factor of 3.

The shortest walking path must use only these connections. All the places are connected to each other by at least one path.

INPUT

Each input file contains the map of a unique shopping mall and a list of queries.

The first line contains two integers N ($N \leq 200$) and M ($N-1 \leq M \leq 1000$), the number of places and connections respectively. The places are numbered from 0 to *N-1*. The next N lines contain floor and the coordinates x, y of the places, one place per line. The distance between floors is 5 meters. The other two coordinates x and y are expressed in meters.

The next M lines contain the direct connections between places. Each connection is defined by the identifier of both places and the type of movement (one of the following: `walking`, `stairs`, `lift`, or `escalator`). Check the cost of each type in the description above. The type for places in the same floor is *walking*.

The next line contains an integer Q ($1 \leq Q \leq 1000$) that represents the number of queries that follow. The next Q lines contain two places each `a` and `b`. We want the shortest walking path distance to go from `a` to `b`.

OUTPUT

For each query write a line with the shortest path in walked meters from the origin to the destination, with each place separated by a space.

SAMPLE INPUT

```
6 7
3 2 3
3 5 3
2 2 3
2 6 4
1 1 3
1 4 2
0 1 walking
0 2 lift
1 2 stairs
2 3 walking
3 4 escalator
5 3 escalator
4 5 walking
5
0 1
1 2
3 5
5 3
5 1
```

SAMPLE OUTPUT

```
0 1
1 0 2
3 4 5
5 3
5 3 2 0 1
```