



**Problem B** (Program filename: B.cpp, B.dpr, or B.java)

## ACM: Vice City

*“Tommy, there will be a programming contest here in Vice City. One of the coaches has stolen a copy of the problem set. The chief judge wants it back. Take out the coach guy at his hotel and return the problems back. The address is taped under the phone. Do it now!”*

Not a tough job for you, Tommy Vercetti! Getting the mission at the pay phone, you must head off the coach at WK Charriot Hotel before he leaves. You have to get there fast! Get there very fast indeed! Unfortunately, the vehicle you start with may not run fast enough. But there are some fixed locations in Vice City at which you can find certain vehicles, like Diaz’s Mansion where you can find an Infernus. This way, you may change your vehicle on your way to hotel several times. For example, in the first sample input, you ride from PayPhone to CarShowRoom on a PCJ600 and drive the rest of the path in a HotRingRacer. Don’t forget that it takes one minute each time you change your vehicle.

You are given the names of these locations in the city and the distances between each pair. At each location you can find a certain vehicle anytime you get there. Knowing the top speed of each vehicle, you want to find out the minimum time in which you can reach the hotel. For the sake of simplicity, assume that you always drive at top speed of your vehicle.

### Input (filename: B.in)

The first line of the input contains a single integer  $t$  ( $1 \leq t \leq 20$ ) which is the number of test cases in the input. Each test case has three parts. The first part consists of  $m$  lines ( $1 \leq m \leq 100$ ) of the form *vehicle speed* where *vehicle* is the unique name of a vehicle and *speed* is a positive integer giving the top speed of the vehicle measured in Km/h.

The next part of the test case identifies the locations in the city and is separated from the first part by exactly one blank line. It consists of  $n$  lines ( $2 \leq n \leq 500$ ) of the form *location vehicle* where *location* is the unique name of a location in the city and *vehicle* is the name of the vehicle available in that location. The list of locations always includes the starting location PayPhone and the destination WKCharriot.

The third part of the test case identifies the roads between locations and is separated from the previous part by exactly one blank line. It consists of several lines of the form  $loc_1 loc_2 distance$  indicating there is a (two-way) road of length *distance* between the locations  $loc_1$  and  $loc_2$ . Distances are expressed in kilometers and are positive integers. The test case is terminated by a line containing a single asterisk character (\*).

All names (for vehicles and locations) are strings of at most 100 letters and digits with no space characters and are considered case sensitive. Items in an input line are separated by one or more space characters. Also, there may be arbitrary leading or trailing blanks except in empty lines used as separators.

### Output (filename: B.out)

For each test case, there is one line in the output containing the minimum time (in minutes) you need to travel from PayPhone to WKCharriot, or the word UNREACHABLE if the destination is unreachable from the starting point. Print the results as numbers with exactly three decimal digits after decimal point. That is, the possible decimal digits after the third one should be ignored, and if there are less than three digits after decimal point, zero digits should be printed for missing digits.

## Sample Input

```
2
Infernus      280
Cheetah       285
PCJ600        250
Stallion      180
HotRingRacer  300

Mansion       Infernus
CarShowRoom   HotRingRacer
VicePort      Cheetah
NorthPointMall Infernus
PayPhone      PCJ600
WKCharriot    Stallion

PayPhone      CarShowRoom   10
PayPhone      VicePort         15
VicePort      WKCharriot      20
CarShowRoom   Mansion         15
Mansion       WKCharriot      15
Mansion       NorthPointMall  5
NorthPointMall WKCharriot      5
*
Caddy         80
MrWhoopie    60
Stretch      120
CubanHermes 160
Voodoo       170

CherryPoppy  MrWhoopie
Mansion      Stretch
PayPhone     CubanHermes
LittleHaiti  Voodoo
WKCharriot   Caddy

PayPhone     CherryPoppy    10
CherryPoppy  LittleHaiti    15
Mansion      WKCharriot     20
*
```

## Sample Output

```
8.400
UNREACHABLE
```