

Alice and Bob haven't met for some time. Bob isn't very happy about this, so he urges Alice to finally make time for a meeting. Let's listen to an extract from a phone call:

*Alice:* ... maybe we should meet on neutral territory.

*Bob:* I've already heard this from you — two years ago.

*Alice:* I know ; I just haven't found yet a suitable place that is roughly at the same distance from both yours and mine.

*Bob:* Well, the geometric place of the points that are equidistant from two given points on the surface of a sphere (and the earth is a sphere rather than a disc) is a great circle (namely the one which intersects the great circle through the given points orthogonally at the center of them). If you insist only on approximately equal distances though, we get a zone of some kilometers width and about 40000 km length. Not everything in this zone is water. Thus I think it is a feasible task to find a fitting place.

*Alice:* Now, if I tell you to pick any, we'll certainly land up in Honolulu.

*Bob:* Which is not a too bad idea. So, may I pick any ?

*Alice:* As long as I don't have to accept — but I'm open to suggestions.

*Bob:* Honolulu ?

*Alice:* Is it situated on aforementioned geometric place at all ???

*Bob:* Not quite ...

Nice. Now let's stop the preliminaries and come to the facts: Given two locations on the earth's surface you can find the geometric place of all equidistant points on the surface. For another given location calculate its distance on the surface to this geometric place. Assume that the earth is a sphere with a radius of 6378 km.

## Input

The input file consists of two parts: a list of locations and a list of queries.

The location list consists of up to 100 lines, one line per location. Each contains a string and two floating-point numbers, separated by whitespace, representing the name of the location, its latitude and its longitude. Names are unique and shorter than 30 characters and do not contain whitespace. Latitudes are between -90 (South Pole) and 90 (North Pole) inclusive. Longitudes are between -180 and 180 inclusive where negative numbers denote locations west of the meridian and positive numbers denote locations east of the meridian. (The meridian passes through Greenwich, London.) The location list is terminated by a line consisting of a single '#'.

Each line in the query list contains three names of locations. You can assume the first location to be Alice's home, the second location to be Bob's home and the third location to be a possible meeting point. The query list is terminated by a line consisting of a single '#'.

## Output

For each query, output a line saying '*M* is *x* km off *A/B* equidistance.' with *M*, *x*, *A*, *B* appropriately replaced by the location names and the calculated distance rounded to the nearest integer.

If one of the locations in the query didn't occur in the list of locations print '?' instead of the distance.

## Sample Input

```
Ulm          48.700 10.500
Freiburg     47.700 9.500
Philadelphia 39.883 -75.250
SanJose      37.366 -121.933
Atlanta      33      -84
Eindhoven    52      6
Orlando      28      -82
Vancouver    49      -123
Honolulu     22      -157
NorthPole    90      0
SouthPole    -90     0
#
Ulm Freiburg Philadelphia
SanJose Atlanta Eindhoven
Orlando Vancouver Honolulu
NorthPole SouthPole NorthPole
Ulm SanDiego Orlando
NorthPole SouthPole SouthPole
Ulm Honolulu SouthPole
#
```

## Sample Output

```
Philadelphia is 690 km off Ulm/Freiburg equidistance.
Eindhoven is 3117 km off SanJose/Atlanta equidistance.
Honolulu is 4251 km off Orlando/Vancouver equidistance.
NorthPole is 10019 km off NorthPole/SouthPole equidistance.
Orlando is ? km off Ulm/SanDiego equidistance.
SouthPole is 10019 km off NorthPole/SouthPole equidistance.
SouthPole is 1494 km off Ulm/Honolulu equidistance.
```