The Internet now offers a variety of interactive map facilities, so that users can see either an overview map of a large geographic region or can "zoom in" to a specific street, sometimes even a specific building, on a much more detailed map. For instance, downtown San Jose might appear in a map of California, a map of Santa Clara county, and a detailed street map.

Suppose you have a large collection of rectangular maps and you wish to design a browsing facility that will process a sequence of map requests for locations at various levels of map detail. Locations are expressed using *location names*. Each location name has a unique pair of real coordinates (x, y). Maps are unique, labeled with identifying map names, and defined by two pairs of real coordinates—$(x_1, y_1)(x_2, y_2)$—representing opposite corners of the map. All map edges are parallel to the standard Cartesian x and y axes. A map and a location can have the same name. The *aspect ratio* of a map is the ratio of its height to its width (where width is measured in the x direction and height is measured in the y direction).

The level of detail of a map can be approximated by using the rectangular area represented by the map; i.e., assume that a map covering a smaller area contains more detailed information. Maps can overlap one another. If a location (x, y) lies within two or more maps having equal areas, the preferred map (at that level of detail) is the one in which the location is nearest the center of the map. If the location is equidistant from the centers of two overlapping maps of the same area, then the preferred map (at that level of detail) is the one whose aspect ratio is nearest to the aspect ratio of the browser window, which is 0.75. If this still does not break the tie, then the preferred map is the one in which the location is furthest from the lower right corner of the map (this heuristic is intended to minimize the need for scrolling in the user's browser window). Finally, if there is still a tie, then the preferred map is the one containing the smallest x-coordinate.

The *maximum detail level* available for a given location is the maximal number of maps of different areas that contain the location. Clearly, different locations can have different maximum detail levels. The map at detail i for the location is the map with the ith largest area among a maximal set of maps of the distinct area containing the location. Thus, the map at detail level 1 for the location will be the least detailed (largest area) map containing the location and the map at the maximum detail level will be the most detailed (smallest area) map containing the location.

Input

The input file consists of a set of maps, locations, and requests; it is organized as follows:

- The word '`MAPS`', in all uppercase letters and on a line by itself, introduces a set of one or more maps. Following the set heading, each map is described by a single line consisting of a map name (an alphabetic string with no leading, trailing, or embedded blanks) and two real coordinate pairs—x_1 y_1 x_2 y_2—representing opposite corners of the map.
- The word '`LOCATIONS`', in all uppercase letters and on a line by itself, introduces a set of one or more locations. Following this heading, each location is described by a line consisting of a location name (an alphabetic string with no leading, trailing, or embedded blanks) and a real coordinate pair—x y—representing the center of the location.
- The word '`REQUESTS`', in all uppercase letters and on a line by itself, introduces a set of zero or more requests. Following this heading, each request is described by a line consisting of a location name (an alphabetic string with no leading, trailing, or embedded blanks) followed by a positive integer representing the desired detail level for that location.
- The word '`END`', in all uppercase and on a line by itself, terminates the file.

All map and location data preceding the requests are valid. There will be no duplicate maps. The result of processing a valid request is the name of the map containing the given location at the given detail level (using the tie-breaking rules described above). Invalid requests can result from requesting unknown location names, locations that do not appear in any map, or detail levels that exceed the number of maps of different areas containing the location.

The following example should illustrate all these definitions:

Output

Each request must be echoed to the output. If the request is valid, display the name of the map satisfying the request. If the location is not on a map, display a message to that effect. If the location is on the map but the detail level is too large, display the name of the map of the smallest available area (largest possible detail level).

Sample Input

```
MAPS
BayArea -6.0 12.0 -11.0 5.0
SantaClara 4.0 9.0 -3.5 2.5
SanJoseRegion -3.0 10.0 11.0 3.0
CenterCoast -5.0 11.0 1.0 -8.0
SanMateo -5.5 4.0 -12.5 9.0
NCalif -13.0 -7.0 13.0 15.0
LOCATIONS
Monterey -4.0 2.0
SanJose -1.0 7.5
Fresno 7.0 0.1
SanFrancisco -10.0 8.6
SantaCruz -4.0 2.0
SanDiego 13.8 -19.3
REQUESTS
SanJose 3
SanFrancisco 2
Fresno 2
Stockton 1
SanDiego 2
SanJose 4
SantaCruz 3
END
```

Sample Output

```
SanJose at detail level 3 using SanJoseRegion
SanFrancisco at detail level 2 using BayArea
Fresno at detail level 2 no map at that detail level; using NCalif
Stockton at detail level 1 unknown location
SanDiego at detail level 2 no map contains that location
SanJose at detail level 4 using SantaClara
SantaCruz at detail level 3 no map at that detail level; using CenterCoast
```