**Nearbuy Attractions**

Destinations often have lots of attractions or landmarks that travelers want to visit – usually more than they can visit during their stay. What most travelers end up doing is optimizing their time by only visiting the attractions that are closest to where they’re staying.

Travelers ask you to help them to arrange holiday, so they can visit as many attractions as possible. Your task is to calculate the time it takes from the hotel to the attractions by preferred transport for the given amount of time. Transport can be only be one of the followings: foot, bike or metro.

**Notes:**

* You should consider the following average speed for the different transport options: metro, 20km/h; bike, 15km/h; and foot, 5km/h
* For distances in the problem you should use the 'Law of Cosines for Spherical Trigonometry'. Here how it is implemented as pseudo-code.

function distance\_between(point1, point2) {

var EARTH\_RADIUS = 6371;//in km

var point1\_lat\_in\_radians = degree2radians( point1.latitude );

var point2\_lat\_in\_radians = degree2radians( point2.latitude );

var point1\_long\_in\_radians = degree2radians( point1.longitude );

var point2\_long\_in\_radians = degree2radians( point2.longitude );

return acos( sin( point1\_lat\_in\_radians ) \* sin( point2\_lat\_in\_radians ) +

cos( point1\_lat\_in\_radians ) \* cos( point2\_lat\_in\_radians ) \*

cos( point2\_long\_in\_radians - point1\_long\_in\_radians) ) \* EARTH\_RADIUS;

}

* When calculating distance, always round it up or down to 2 decimal points precision. Example: 2.3467 should be rounded to 2.35 while 3.4522 should be rounded to 3.45
* All values of integer type should be considered as 4-bytes integer.
* Assume value of pi is 3.14159265359.

**Input Format**

The first line contains an integer N , which represents the number of attractions that follows. After that are N lines each containing three space-separated numbers that represent the ID (integer), the latitude (double) and the longitude (double) of the attraction.

After the attractions there will be another line containing an integer M, which represents the number of test cases that follows. After that are M lines each containing four space-separated values – the first and second are the latitude (double) and longitude (double) of the hotel the guest is staying at, the third value shows their preferred transport option and the fourth represents how long they are willing to travel in minutes starting from the hotel.

**Output Format**

The output is precisely M lines, each containing a list of space-separated attraction IDs. Each line represents the attractions that are possible to reach from the guest’s hotel in the specified amount of time and using the preferred transport option. In each line the attraction IDs should be sorted by distance starting with the closest one. When two attractions are the same distance away, they should be *sorted by ID*.

 M is no greater than 200. N is no greater than 200.

**Sample Input**

10

1 52.378281 4.900070

2 52.373634 4.890289

3 52.375737 4.896547

4 52.372995 4.893096

5 52.376237 4.902860

6 52.367066 4.893381

7 52.366537 4.911348

14 52.368832 4.892744

15 52.357895 4.892835

35 52.342497 4.855094

5

52.379141 4.880590 metro 80

52.358835 4.893867 foot 60

52.375859 4.886006 foot 30

52.371700 4.899070 metro 30

52.364055 4.898446 foot 60

**Sample Output**

2 4 3 1 14 5 6 15 7 35

15 6 14 7 4 2 3 5 1 35

2 4 3 14 1 6 5 7 15

4 3 14 5 2 6 1 7 15 35

6 14 15 7 4 2 3 5 1 35