TeamByte ⌄

IEEEXtreme 10.0 > Food Truck

Food Truck

🔒 locked

by *IEEEXtreme*

| Problem | Submissions | Leaderboard | Discussions |

This problem was created and sponsored by Nokia. If they opted in, top finishers will be contacted about possible careers at Nokia.

Madhu has a food-truck called "The Yummy Goods" that goes to a different business hotspot every day at lunch! Madhu wants to perform location-based advertising to folks in the offices near her halt. To do this she uses the GPS location as a longitude and a latitude at the stop and decides on a radius (r) value. She wants to broadcast advertisement SMSes, to customers within this radius, advertising her food-truck.

She needs your help to generate the list of phone numbers of such folks. She has access to a big file of telecom data, which among other details, contains the phone number, longitude, and latitude of active cell-phone users in the city at that moment.

In order to calculate the distance between her stops and her subscribers, she wants you to use the most recent location available for each subscriber. To calculate the distance, you should use the Haversine formula:

d = 2 × r × arcsin (sqrt ($\sin^2$(($lat1$ - $lat2$)/2) + cos($lat1$) × cos($lat2$) × $\sin^2$(($long1$ - $long2$)/2)))

where d is the distance between two points on the surface of the earth, in km's

r is the radius of the earth (6378.137 km for this problem)

$lat1$, $long1$ are the latitude and longitude, respectively, of point 1

$lat2$, $long2$ are the latitude and longitude, respectively, of point 2

Input Format

The first line contains Madhu's latitude and longitude in degrees, separated by a comma.

The second line contains the radius r in kms, within which she wants to broadcast her advertisement.

The third line is a header for the data in the subsequent lines.

The remaining lines have rows of telecom data of active cellphone users. Each line contains the following comma-separated fields:

- A time stamp in `MM/DD/YYYY hh:mm` format. `MM`, is a two-digit month, e.g. `01` for January, `DD` is a two-digit day of month (`01` through `31`), `YYYY` is a four-digit year, `hh` is the two digits of hour (`00` through `23`), and `mm` is the two digits of minute (`00` through `59`)

- The latitude of the subscriber, in degrees

- The longitude of the subscriber, in degrees

- The subscriber's phone number, as a 10-digit number

Notes:

- Some subscribers may appear multiple times. You should use the most recent entry to determine the location of a subscriber. If a subscriber appears multiple times, the date/time stamps will differ.

- None of the field values will contain commas.

Constraints

In order to eliminate rounding and approximation errors, no subscribers will be at a distance d from Madhu, such that $0.99 \times r \le d \le 1.01 \times r$

$1 \le r \le 100$

There will be at most 50,000 lines in the subscriber list.

Output Format

A comma separated list of phone numbers for subscribers within a radius r of the stop, sorted in ascending order.

Sample Input

```
18.9778972,72.8321983
1.0
Date&Time,Latitude,Longitude,PhoneNumber
10/21/2016 13:34,18.912875,72.822318,9020320100
10/21/2016 10:35,18.9582233,72.8275845,9020320024
10/21/2016 15:20,18.95169982,72.83525604,9020320047
10/21/2016 15:23,18.9513048,72.8343388,9020357980
10/21/2016 15:23,18.9513048,72.8343388,9020357962
10/21/2016 15:28,18.9548652,72.8332443,9020320027
10/21/2016 14:03,18.9179784,72.8279306,9020357972
10/21/2016 14:03,18.9179784,72.8279306,9020357959
10/21/2016 09:52,18.97523123,72.83494895,9020320007
10/21/2016 09:44,18.9715932,72.8383992,9020357607
10/21/2016 09:44,18.9715932,72.8383992,9020357593
10/21/2016 09:44,18.9715932,72.8383992,9020357584
10/21/2016 14:57,18.93438826,72.82704499,9020320011
10/21/2016 09:56,18.97596514,72.8327072,9020320045
10/21/2016 08:33,18.9811929,72.8353202,9020320084
10/21/2016 13:27,18.9159265,72.8245989,9020357896
10/21/2016 13:09,18.9077347,72.8076201,9020320094
10/21/2016 10:52,18.97523003,72.83494865,9020320007
```

Sample Output

9020320007,9020320045,9020320084,9020357584,9020357593,9020357607

Explanation

We can calculate the distance between the location "18.9778972, 72.8321983" and each of the subscribers whose details are provided. Only the 6 phone numbers, listed in the *Sample Output* set, have a distance to the location of the food-truck that is less than 1.0 km.

Max Score: 59pts   dynamic

Submissions: 1047

Max Score: 59

Difficulty: Hard

More

Current Buffer (saved locally, editable)   ⌥ ⟲   C++   ⌄   ⤢   ⚙

```cpp
1  #include <cmath>
2  #include <cstdio>
3  #include <vector>
4  #include <iostream>
5  #include <algorithm>
6  using namespace std;
7
8
9  int main() {
10     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
11     return 0;
12 }
13
```

Line: 1 Col: 1

⬆ Upload Code as File   ☐ Test against custom input   Run Code   Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Scoring | Environment | FAQ | Terms Of Service | Privacy Policy