Coupling Passions

Mariana and Pedro are a young couple that loves to travel. Pedro is passionate about surfing, while Mariana is a wine lover. Like many couples, Mariana and Pedro always plan their trips together, choosing only those destinations that best satisfy their passions!

Since Booking.com launched the *Destination Finder*, Mariana and Pedro started using it to look for new places all over the world where they can travel and enjoy as many of their collective passions as possible. Can you help groups of guests like Mariana and Pedro find some hidden gems?

For this challenge, we'll provide you with the following:

- A list of guests and, for each guest, a list of their respective passions.
- A list of destinations and, for each destination, a list of passions that the destination is known to be good for.

Given the data above, find the two destinations satisfying the maximum number of the group's collective passions and print them as two space-separated strings on a single line. If more than one destination satisfies a maximum number of passions, then choose the pair of such destinations having the shortest distance between them. Use the *Law of Cosines for Spherical Trigonometry* to calculate distances, as demonstrated by the following pseudocode:

```
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;
}
```

Input Format

The first line contains a single integer, n, denoting the number of guests traveling together.
Each line i of the n subsequent lines lists guest i's passions as a sequence of space-separated values:

- The first value is an integer, m, denoting the number of passions that the guest has.
- Each of the m subsequent space-separated strings describes one of the guest's passions.

The next line contains a single integer, y, denoting the number of potential destinations.
Each line j of the y subsequent lines describes destination j as a single line of space-separated values:

- The first value is a string denoting the destination name.
- The second value is a floating-point number denoting the destination's latitude.
- The third value is a floating-point number denoting the destination's longitude.
- The fourth value is an integer, z, denoting the number of passions available at the destination.
- Each of the z subsequent space-separated strings describes a passion available at the destination.

Constraints

- $0 < n, m, y, z \leq 250$

- Assume the value of pi is $\pi = 3.14159265359$

Output Format

Print a single line with 2 space-separated destination names that cover the largest number of passions held by the group. These destinations must be ordered alphabetically; if two or more pairs of destinations cover the same number of passions, choose the pair having the shortest distance between cities.

Sample Input

```
2
3 surfing yoga walking
3 wine relaxation beach
3
amsterdam 52.374030 4.889690 4 museums canals nightlife walking
sagres 37.129665 -8.669586 3 beach surfing relaxation
biarritz 43.480120 -1.555580 6 surfing nightlife beach food wine walking
```

Sample Output

```
biarritz sagres
```

Explanation

There are two guests having the following passions:

1. Guest 1: surfing, yoga, and walking.
2. Guest 2: wine, relaxation, and beach.

There are three possible vacation destinations (passions appealing to the above guests are highlighted for emphasis):

1. *amsterdam*: museums, canals, nightlife, and `walking`.
2. *sagres*: `beach`, `surfing`, and `relaxation`.
3. *biarritz*: `surfing`, nightlife, `beach`, food, `wine`, and `walking`.

There are three possible vacations:

1. *amsterdam* and *sagres*: This trip satisfies 4 distinct guest passions (i.e., surfing, beach, relaxation, and walking).
2. *amsterdam* and *biarritz*: This trip satisfies 4 distinct guest passions (i.e., walking, surfing, beach, wine).
3. *sagres* and *biarritz*: This trip satisfies 5 distinct guest passions (i.e., beach, surfing, relaxation, wine, walking).

Because the *sagres-biarritz* vacation satisfies the maximum number of passions, we print *biarritz* and *sagres* as two alphabetically-ordered space-separated values on a new line.