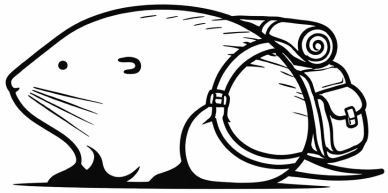
Sales-mole



Surprise! You're a mole. Not only that, you're a mole *with a job.* Every day, you must travel throughout the underground Mole Nation to sell your products at each city on your list to your fellow mole brethren. Since digging is pretty tiring and your usual route takes all day to complete, you've decided to try to speed up your trips using your Advanced Algo knowledge (yes, they teach it in Mole Nation too).

Write a program that, given the 2D coordinates of each city in Mole Nation, outputs a tour that starts and ends at the same city and visits every city *at least once.*

In addition, although you're pretty strong and can dig through dirt like a pro, there are lots of rocks scattered all throughout Mole Nation that are just a little too thick to get through. Each rock can represented by a *convex polygon* in the plane. When traveling between two cities, you must travel along the *shortest path* between them that that *avoids all rocks.* More specifically, along your tour:

1. You may not travel through the *interior* of any rock.
2. You may travel along the *boundary* (edges and corners) of a rock, as needed.

Your program must also output any intermediate coordinates that are visited between cities to account for rocks blocking your path.

A final note: you've been pretty beat down by your corporate mole life, and you aren't striving for perfection like you once did in your younger days. As such, you've decided that the tour you find doesn't quite need to be the shortest one around Mole Nation, it just needs to be at most *twice as long* as the optimal.

Input

The first line of input will contain the number of cities $3 \leq n \leq 500$ and the number of rocks $0 \leq m \leq 250$. The next n lines will contain the x y coordinates of each city.

The remaining input will contain m sections. Each section starts with an integer $3 \leq R_i \leq 150$ giving the number of points in the i'th rock, and the following R_i lines will contain the x y positions of the points that make up the corresponding convex polygon. The points of each rock will be listed in *counter-clockwise order.*

For simplicity, all coordinate values will be non-negative integers no greater than 10^6 ($\approx 2^{20}$). It's further guaranteed that no city lies on the boundary or interior of any rock, and that no two rocks *properly* intersect. That is, no point of one rock will fall within the interior of another, but it could fall on another's boundary.

Output

Your output should contain the x y coordinates of the city *and* non-city points visited in the order they appear in your calculated tour, each on a new line. The initial city should be included at both the beginning and end of your tour.

Languages

It is **highly recommended** that you use C++ for this assignment. We have working Python and C++ solutions, but our Python implementation does not meet the time constraints. We are doubtful that it would be possible to do so without extreme optimizations.

Sample Input

```
4 1
2 2
5 1
5 8
10 8
4
4 6
6 3
8 4
7 9
```

Sample Output

```
2 2
5 1
4 6
5 8
7 9
10 8
8 4
6 3
2 2
```