You are an aspiring robotics engineer and love working on your side projects. You are working on building a shapeshifting robot from scratch, which is currently able to navigate in a flat environment and take in inputs on how far forward to move. Unfortunately, the robot can only move straight and is unable to change directions by itself. However, you want your robot to be able to verify inputs and only follow the input if it won't run into any objects around it, as colliding with an object may cause parts to break.

Write a program that, given the shape, location, direction, and distance of the robot and the location of obstacles, determines whether or not it is possible for the robot to move that distance in the direction without colliding with anything.

The environment is viewed from a bird-eye perspective and contains several obstacles that are represented as polygons. Obstacles may either be convex or non-convex polygons; however, non-convex polygons will be split up into convex polygons in the input for ease of use. The shape of the robot is also represented by a convex polygon. As the environment is seen from a bird-eye perspective, all points are given in 2 dimensions.

If the robot even grazes the edge of an obstacle, this should be considered a collision. In the event that the robot begins inside of an obstacle, this should also be counted as a collision.

Input:

The first line contains an integer O ($1 \leq O \leq 100$) for the number of obstacles. For each obstacle, there will be an input in the following format:

- The first line will begin with an integer P ($3 \leq P \leq 20$) for the number of points in the convex polygon that will represent the obstacle.
- The next P lines will contain two real numbers x and y ($-10^3 \leq x, y \leq 10^3$) representing the coordinates of a vertex on the polygon. These points are provided in counterclockwise order

After the obstacles, input for the robot is given:

- A line containing an integer R ($3 \leq R \leq 20$) for the number of points in the convex polygon representing the robot.
- The next R lines will contain two real numbers x and y ($-10^3 \leq x, y \leq 10^3$) representing the coordinates of a vertex on the polygon. These points are provided in counterclockwise order

The next line will provide a real number Θ ($0 \leq \Theta < 360$) for the angle in degrees, specifying the direction the robot will move. A degree of 0 refers to movement in the positive x direction

The last line contains a real number D ($0 < D \leq 10^3$) for the distance the robot will move.

Output:

The program will print 0 if the robot will not collide with any obstacle during its movement. Otherwise, the program will print a line for each polygon the robot will collide with that contains the points of the polygon.

Sample Input:

```
1
4
0 2
0 4
2 4
2 2
4
0 0
0 1
1 1
1 0
90
5
```

Sample Output:

```
(0.0, 2.0) (0.0, 4.0) (2.0, 4.0) (2.0, 2.0)
```