Problem E
Pizza Hawaii

You are travelling in a foreign country. Although you are also open to eat some regional food, you just cannot resist after you have found an Italian restaurant which offers pizza. Unfortunately the menu is written in the foreign language, so the list of ingredients of the pizzas are incomprehensible to you. What will you do?
One thing that you notice is that each pizza has an Italian name, which sounds quite familiar to you. You even remember for each named pizza what the ingredients of this pizza usually are. You want to use that information to figure out what the possible meaning of each word on the list of ingredients is.

Input

The first line of the input gives the number of test cases t ($0 < t \leq 20$). The first line of each input gives the number n of pizzas on the menu ($1 \leq n \leq 60$). The following $3 \cdot n$ lines describe the pizzas on the menu. Each pizza description starts with one line containing the name of the pizza. The name of the pizza consists of between 3 and 20 uppercase and lowercase letters. The next line starts with an integer m_i, giving the number of ingredients of the pizza on the menu ($1 \leq m_i \leq 20$). The rest of the line contains the m_i ingredients separated by spaces. Each ingredient is a word consisting of between 2 and 20 lowercase letters. The third line of each pizza description gives the ingredients in your native language in the same format. Note that the number of ingredients may differ, because each restaurant may use slightly different ingredients for pizzas with the same name, so the ingredients you remember for a pizza with that name may not match the actual ingredients.

Output

For each test case print all pairs of words (w_1, w_2) where w_1 is an ingredient in the foreign language that could be the same ingredient as w_2 because w_1 and w_2 appear on the same set of pizzas. Sort the pairs in increasing lexicographical order by w_1, and in case of a tie in increasing lexicographical order by w_2. Print a blank line between different test cases.

Sample Input

```
2
3
Hawaii
4 tomaten schinken ananas kaese
4 pineapple tomatoes ham cheese
QuattroStagioni
6 tomaten kaese salami thunfisch spinat champignons
6 mushrooms tomatoes cheese peppers ham salami
Capricciosa
6 champignons kaese tomaten artischocken oliven schinken
5 cheese tomatoes mushrooms ham artichoke
1
Funghi
3 tomaten kaese champignons
3 cheese tomatoes mushrooms
```

Sample Output

```
(ananas, pineapple)
(artischocken, artichoke)
(champignons, mushrooms)
(kaese, cheese)
(kaese, ham)
(kaese, tomatoes)
(oliven, artichoke)
(salami, peppers)
(salami, salami)
(spinat, peppers)
(spinat, salami)
(thunfisch, peppers)
(thunfisch, salami)
(tomaten, cheese)
(tomaten, ham)
(tomaten, tomatoes)

(champignons, cheese)
(champignons, mushrooms)
(champignons, tomatoes)
(kaese, cheese)
(kaese, mushrooms)
(kaese, tomatoes)
(tomaten, cheese)
(tomaten, mushrooms)
(tomaten, tomatoes)
```

Problem J

Track Smoothing

Bob has the task to plan a racing track of a specific length. He thought it is a great idea to just form a convex polygon that has exactly the required length. Then he was told that racing cars cannot drive such sharp corners like in his plan.

He has to ensure some minimal radius for all curves in his track. Bobs loves the shape of his track, so he don't want to change it too much. His plan is to scale the track down around the balance point of the polygon that specifies his track. Then, he wants to draw the new track with a line that has a constant distance to the scaled track. The chosen distance should be minimal distance that fulfils the given minimum radius constraint. He asks you to write a program to calculate the scale factor for a given track and minimum radius so that the resulting track has the same length as the one of his original plan.

Bob made some drawings of the first test case:

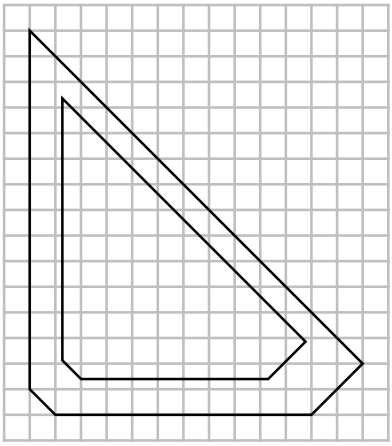

Figure 1 – Original track and scaled down track

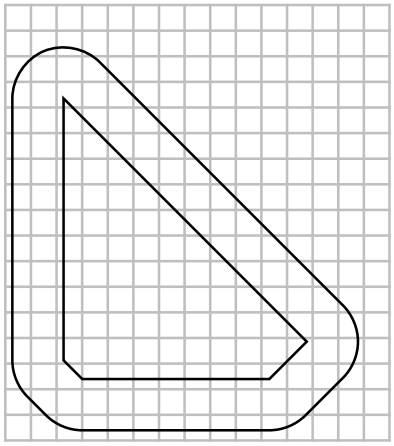

Figure 2 – Scaled down track and resulting track

Input

The input starts with the number of test cases t ($0 < t \leq 100$). Every test case starts with a line consisting of two integers: the minimal required radius r and the number of points n of the original track polygon ($0 \leq r \leq 1\,000$; $3 \leq n \leq 10\,000$). Then n lines follow, where each line specifies 2D-coordinates as two integers x_i and y_i ($-10\,000 \leq x_i, y_i \leq 10\,000$). $(0, 0)$ is the lower left corner. These are the coordinates of the original track in counterclockwise direction. You may safely assume that the area of the given polygon is non-empty.

Output

For each test case print out one line. If it is possible to construct a course according to the above description, output the scaling factor, "`Not possible`" otherwise. The factor must have a relative or absolute error smaller than 10^{-5}.

Sample Input

```
2
20 5
10 0
110 0
130 20
0 150
0 10
1 5
0 0
1 0
2 0
2 1
0 1
```

Sample Output

```
0.730494
Not possible
```