Problem M. Rikka with Illuminations

Input file: standard input
Output file: standard output

Rikka loves convex polygons, so she decides to install some illuminants to embellish polygons.

Now she has a large convex polygon with n sides. She also has m different points strictly outside the polygon which are all legal positions to install illuminants.

An illuminant can light up some exterior boundaries of the polygon.

Rikka wants to install some illuminants to light up all exterior boundaries of the polygon. She asks you to calculate the least number of illuminants which she needs and provide a feasible scheme.

Input

The input contains several test cases, and the first line contains a single integer T ($1 \leq T \leq 100$), the number of test cases.

For each test case, the first line contains two integers n ($3 \leq n \leq 1000$) and m ($1 \leq m \leq 1000$).

Each of the following n lines describes a vertex on the convex polygon with two integers x and y ($|x|, |y| \leq 10^9$), the Cartesian coordinates of the vertex. All these vertices are given in counter-clockwise order and any three of them are not collinear.

Then the following m lines contain m different points outside the polygon describing all legal positions to install illuminants. Each of them contains two integers x and y ($|x|, |y| \leq 10^9$), the Cartesian coordinates of a legal position. They are numbered from 1 to m. All these positions would not lie in some extension lines for the sides of the polygon.

Output

For each test case, if it's impossible to light up all exterior boundaries of the polygon, output a single line with a single integer -1. Otherwise, output two lines. Firstly, output a line with a single integer k, representing the least number of illuminants Rikka needs to light up all the boundaries. Then, output a line with k space-separated distinct integers, describing a feasible scheme, each of which is the index of a selected position.

All feasible schemes are allowed, so you can output any of them.

Example

standard input	standard output
1 3 3 0 0 1 0 0 1 -1 -1 3 -1 -1 3	2 2 1

This page is intentionally left blank