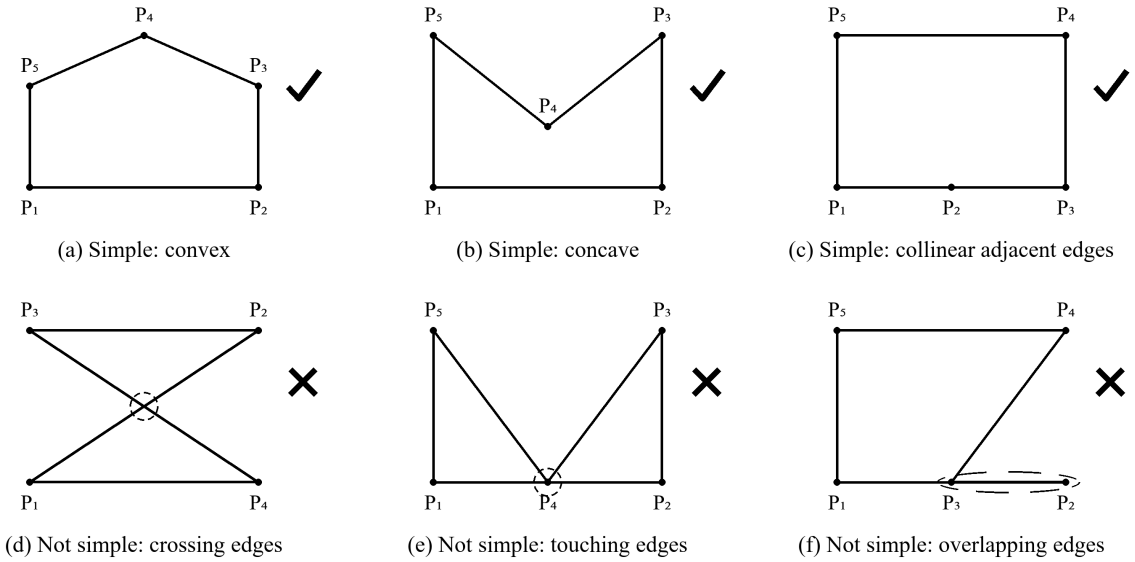
Enclose the Point

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

There are n points $P_1, P_2, \ldots, P_n$ in the plane. VegiCat has q queries for you. Each query gives a point P.

For each query, construct a simple polygon using some of the given points as its vertices. The polygon must contain P **strictly** inside it and have the **minimum number of vertices**. If no such polygon exists, output -1.

A simple polygon is a closed polygonal chain with at least three distinct vertices. Adjacent edges intersect only at their common endpoint, and nonadjacent edges do not intersect.



Examples of simple and non-simple polygons

Input

The first line contains an integer T ($1 \le T \le 2 \cdot 10^4$), the number of test cases. The description of the test cases follows.

The first line of each test case contains two positive integers n and q ($3 \le n \le 10^5$, $1 \le q \le 10^5$), representing the number of given points and the number of queries.

The i-th of the next n lines contains two integers x_i and y_i ($-10^9 \le x_i, y_i \le 10^9$), the coordinates of P_i. It is guaranteed that these n points are pairwise distinct.

Each of the next q lines contains two integers x and y ($-10^9 \le x, y \le 10^9$), the coordinates of the queried point.

It is guaranteed that the sum of n over all test cases does not exceed 10^5.

It is guaranteed that the sum of q over all test cases does not exceed 10^5.

Output

For each query, if no polygon satisfying the requirements exists, output one integer -1 on a line.

Otherwise, output a polygon with the minimum number of vertices. On the first line, output one integer m, the number of vertices.

On the second line, output m integers $a_1, a_2, \ldots, a_m$ ($1 \le a_i \le n$), the indices of the vertices in order. The polygon is $P_{a_1} P_{a_2} \ldots P_{a_m}$. The vertices may be output in either clockwise or counterclockwise order.

The indices must be pairwise distinct, and the output polygon must be simple and contain the queried point strictly inside it. If there are multiple solutions, output any of them.

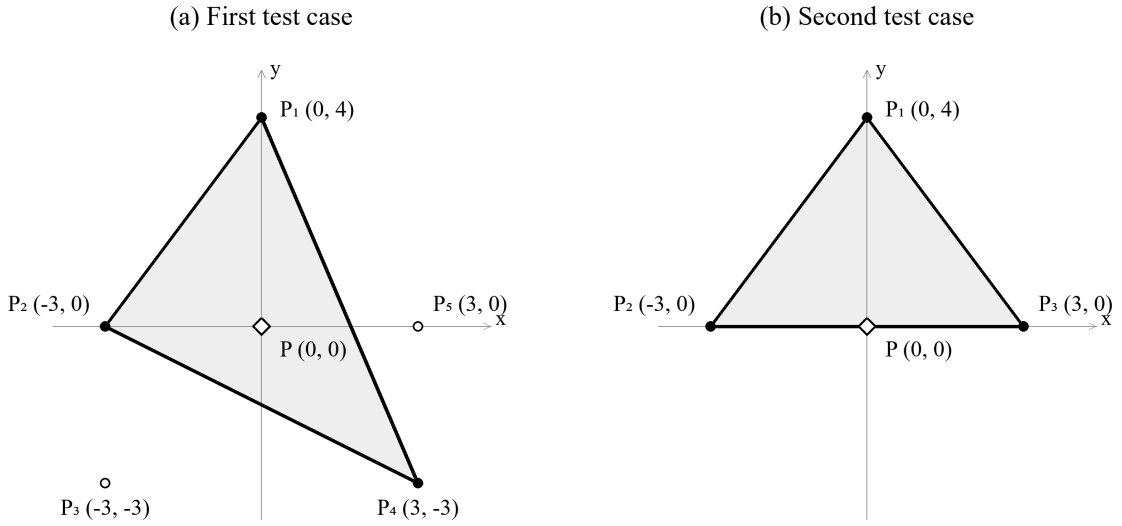
Example

standard input	standard output
2 5 1 0 4 -3 0 -3 -3 3 -3 3 0 0 0 3 1 0 4 -3 0 3 0 0 0	3 4 1 2 -1

Note

In the first test case, the triangle $P_4P_1P_2$ contains the queried point $P = (0, 0)$ strictly inside it. Since a polygon has at least three vertices, this triangle has the minimum number of vertices.

In the second test case, the only possible polygon is the triangle $P_1P_2P_3$. The queried point $P = (0, 0)$ lies on the edge P_2P_3, so the answer is -1.



Illustrations of the sample test cases