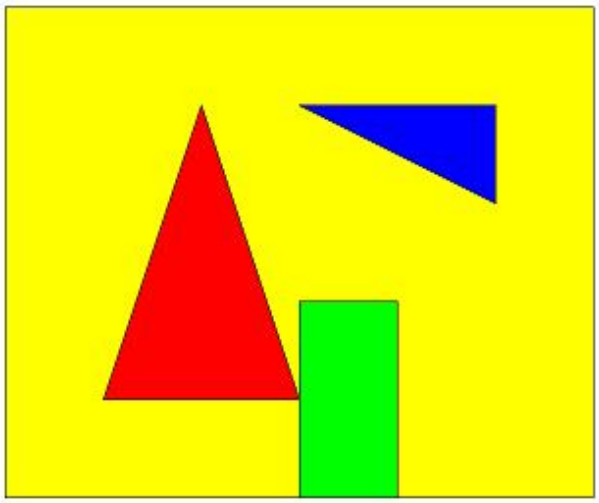
UVa Online Judge

12310 Point Location

Given a partition of the plane into disjoint regions (i.e. a planar subdivision), your task is to determine the region where each query point lies (We use labels to distinguish the regions, see below).

The planar subdivision will be given as a planar straight-line graph (PSLG) with every vertex having at least two adjacent vertices. The two sides of each segment will always belong to different regions.

Here is a sample PSG with 5 regions in the picture below (don't forget there is an exterior infinite region outside the PSGL):



Input

There will be at most 10 test cases. Each test case begins with four integers n, m, p, q ($1 \le n \le 10,000$, $1 \le m \le 30,000$, $1 \le p \le 20,000$, $1 \le q \le 100,000$), where n and m are the number of vertices and edges in PSLG, p is the number of labels, q is number of queries. The next n lines contain the coordinates of the vertices (coordinates are integers whose absolute values do not exceed 1,000,000). The next m lines contain the edges of the PSGL (vertices are numbered 1 to n). There will be no self-loops or parallel-edges. No two edges will be crossing each other at non-endpoints, and each vertex will be connected to at least two edges, and the two sides of each edge will always belong to different regions. The next p lines contain the coordinates of the labels (numbered 1 to p). There will be at most one label strictly inside each region (including the infinite region) of the PSGL. The vertices of the PSGL will be connected. The next q lines contain the coordinates of the query points (coordinates are real numbers whose absolute values do not exceed 1,000,000). The input terminates with $n = m = p = q = 0$, which should not be processed.

Output

For each query, print the label of the region in which the query point lies. If the region does not have a label, print '0'. It is guaranteed that for each label and query point, the distance to the boundary of its region will be at least 0.0001.

Sample Input

```
14 16 5 5
0 0
30 0
40 0
60 0
60 50
0 50
20 40
10 10
30 10
```

```
30 20
40 20
50 30
50 40
30 40
1 2
2 9
9 8
8 7
7 9
9 10
10 11
11 3
3 4
4 5
5 6
6 1
12 13
13 14
14 12
2 3
20 20
10 20
35 10
45 39
1 60
28 11
29 14
34 7
40 38
70 1
0 0 0 0
```

Sample Output

```
1
2
3
4
5
```