E Predictive Pinball (`pinball.{c,cc,java}`)

E.1 Description

Due to recent Californian economic woes, Birch Tree University and its famed Gelded Hares athletic program are implementing cost-cutting measures. The football team, which could be doing better anyway, is slated to be replaced by a varsity pinball team.

Seeking the fame and glory accorded only to collegiate pinball champions, you've suddenly found yourself in the throes of an intense varsity pinball tryout. Everything has come down to one final game, and the hopeful player happens to be *you*. "No sweat," you think to yourself, as you fixate on the pinball machine in front of you.

You take a deep breath and carefully (but quickly, as there's not much time) examine the situation. As it so happens, the walls on both the left and right sides of the pinball machine are formed by vertical line segments, with no two touching each other.

The pinball (of radius 0) is currently at location $(0, 0)$, and due to your precision control of the pinball flippers, you can shoot it via a straight line to any point on the bottom-most wall on the right side of the pinball machine. After bouncing off this wall, the ball continues leftward until it hits a wall on the left side of the machine (or falls into the gutter, i.e., between two walls). If it hits such a wall, it bounces again, and so on, indefinitely. We number these bounces starting from 0.

There are crucial bonus points at stake, and it seems rather likely that they'll determine the outcome of the pinball tryouts. A special subset of the walls will be bonus walls, which will accumulate you valuable points if hit on specific bounces. For each i from 0 to 99, your job is to determine whether a special wall can be hit on the i-th bounce, where the first bounce is necessarily off the bottom-most right-side wall.

You may assume that the ball bounces are completely elastic, so that the internal angles between the ball and a surface (special or otherwise) are the same before and after bouncing, except if you hit the endpoint of a wall, which causes the ball to immediately fall into a gutter. Moreover, all left walls are vertically aligned with each other (that is, they have the same x-coordinate), as are all right walls. Furthermore, no two walls on the same side of the pinball machine have overlapping y-coordinates, not even at endpoints.

E.2 Input

Input cases are separated by blank lines; a line containing "0 0 0 0" follows the last input case. Each input case has the following in order:

- A line with 4 space-separated numbers n_l, n_r, s_l, s_r, denoting the number of left walls, right walls, special left walls and special right walls. $1 \leq n_l, n_r \leq 1000$, $0 \leq s_l \leq n_l$, and $0 \leq s_r \leq n_r$.
- n_l lines, each with 4 space-separated numbers (x_0, y_0, x_1, y_1), where the i-th line identifies the i-th wall on the left with bottom endpoint coordinates (x_0, y_0) and upper endpoint coordinates (x_1, y_1). $-10^6 \leq x_0 = x_1 < 0$, and $0 \leq y_0 < y_1 \leq 10^6$. The walls are listed in order from lowest to highest.
- s_l space-separated numbers i, each indicating that the i-th left-side wall is special (counting from 0).
- n_r lines, each with 4 space-separated numbers (x_0, y_0, x_1, y_1), where the i-th line identifies the i-th wall on the right with bottom endpoint coordinates (x_0, y_0) and upper endpoint coordinates (x_1, y_1). $0 < x_0 = x_1 \leq 10^6$, and $0 \leq y_0 < y_1 \leq 10^6$. The walls are listed in order from lowest to highest.
- s_r space-separated numbers i, each indicating that the i-th right-side wall is special (counting from 0).

For example:

```
2 2 1 1
-1 1 -1 5
-1 10 -1 11
```

```
1
1 1 1 2
1 10 1 11
1

5 5 1 1
-10000 5 -10000 7
-10000 13 -10000 15
-10000 21 -10000 23
-10000 29 -10000 31
-10000 37 -10000 39
4
10000 1 10000 3
10000 9 10000 11
10000 17 10000 19
10000 25 10000 27
10000 33 10000 35
4

5 5 2 2
-10000 5 -10000 7
-10000 13 -10000 15
-10000 21 -10000 23
-10000 29 -10000 31
-10000 37 -10000 39
0 3
10000 1 10000 3
10000 9 10000 11
10000 17 10000 19
10000 25 10000 27
10000 33 10000 35
0 1

0 0 0 0
```

E.3 Output

For each test case, print a line of space-separated number denoting the indices of the bounces (0-indexed) at which a special wall may be hit (not necessarily in a single shot). If no such indices exist, print -1 instead. For example:

```
-1
8 9
0 1 2 7
```