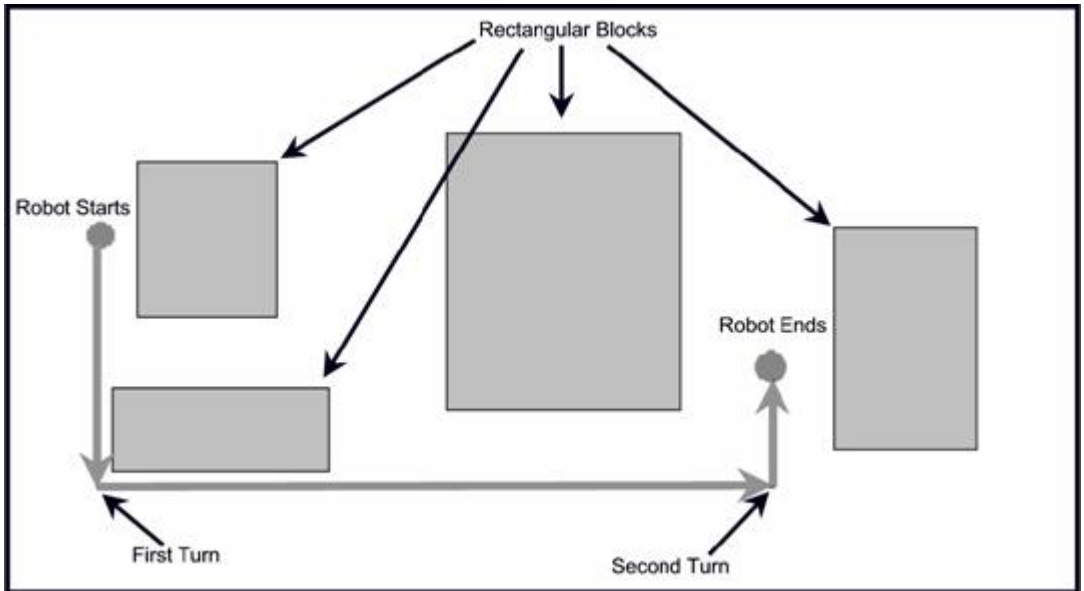


Dr. Jemison is simulating the navigation system of a robot. He is using a 2D labyrinth containing several rectangular blocks. All blocks are placed either vertically or horizontally. Spaces are available around the blocks. A robot can use these empty places for its movement. While moving, the robot is not allowed to touch or pass through any block. Also, robots movement must be either horizontal or vertical. A sample scenario is given in the following picture, where rounded blocks indicate the position of robot.



The main aspect of Jemisons experiment is to test whether a robot can turn timely in the right direction and reach its destination. Jemison has already embedded the complete map of the labyrinth and its final position inside the robots memory. As turning is costly, Jemison wants the robot to reach its destination using minimum number of turns. For example, in the above figure, it requires at least two turns to reach the destination.

In this problem you will be given a labyrinth of infinite extent. There can be zero or more rectangular blocks inside the labyrinth. Each rectangular block will be defined by its bottom-left (lx, by) and top-right (rx, ty) corners. Rectangular blocks will not overlap one another but they can share a common border line. The starting and ending positions of the robot will be denoted by Cartesian coordinates. These positions will not touch any block or stay inside it. You have to find the minimum number of turns that the robot must make to reach its destinations from starting positions.

Input

In the first line, there will be an integer, T ($1 \leq T \leq 50$) denoting the number of tests. Each input will start with an integer, N ($0 \leq N \leq 50$), where N is the number of rectangular blocks. Following N lines will contain the description of a rectangular block. Each line will contain 4 integers lx, by, rx (greater than lx), ty (greater than by). Next line will contain another integer K ($1 \leq K \leq 20$) which is the number of queries. Each query will contain starting and ending coordinates of the robot in a line (two positions will be distinct always). The coordinates will be positive integer and will not exceed 10^8 .

Two successive input cases will be separated by a blank line.

Output

For each input set, output must start with a line 'Labyrinth #D', where D is the test number starting from 1. It will be followed by minimum number of turns for each query in a separate line. If the robot somehow cannot reach to its destination, print 'Impossible.'. See sample input output for clarification.

Sample Input

```
2
0
2
10 10 20 20
10 10 10 20

1
10 10 100 100
2
9 10 101 10
1 1 1000 1000
```

Sample Output

```
Labyrinth #1
1
0
Labyrinth #2
2
1
```