Desrugenstein

The city of Desrugenstein is a complete mess. Looking at the map, it seems organized, since it was created in a square grid form, but there is no directions standard. Each corner says the directions you can go from there (north, south, west, east). The mayor Daniel Snake is headstrong and lazy enough to let everything as it is and forbid any change attempt. Unable to do much, the Master Spiritual Counselor of Desrugenstein, Giordano Marfyn, asked you, Spiritual Counselor Level XVII of Desrugenstein, lead programmer of Desrugenstein, to write a program to calculate the cost of going from a corner (x, y) to another corner (z, w), considering the messy streets.

Input

Input file contains several test cases. First line of each test case contains an integer N (1 ≤ N ≤ 10) that represents height and width of the grid that maps the city (a N x N grid). The input file ends with N = 0, and it should not be processed. Each one of the next N lines represents a street of the city, starting from the further north (N – 1) until the further south. In each one of these lines there are 4*N integers, 4 for each corner: A (north) B (south) C (west) D (east). Each one is 0 if it is not possible to go on the respective direction, or 1 if it is possible.
After the city map, your program should read an integer P (1 ≤ P ≤ 100). Next P lines contain 4 integers each, x0 y0 x1 y1 representing the question: “What is the minimum cost to go from corner (x 0 , y0) to corner (x1 , y1)?”. The cost to go from a corner to the nearest corner in any direction is 1.

Output

For each question, answer “Impossible” if there is no valid (respecting corners directions) path between the corners, or the minimum cost, if there is(are) path(s). Print a blank line after each test case.

Example

Input:

```
4
0 1 0 1 0 0 1 0 0 0 0 1 0 1 0 0
0 0 0 0 1 0 0 1 0 0 0 1 0 1 0 0
1 0 0 0 1 0 1 0 1 1 0 0 0 1 1 0
0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0
5
1 3 0 3
2 3 3 0
2 3 0 0
3 1 3 2
```

```
0 3 0 0
0
```

Output:

```
1
4
7
3
Impossible
```