Problem C: Direct Visibility

Acknowledgement: This problem is from the CTU Open Contest 2014

Building the GSM network is a very expensive and complex task. Moreover, after the *Base Transceiver Stations* (*BTS*) are built and working, we need to perform many various measurements to determine the state of the network, and propose effective improvements to be made.

The ACM technicians have a special equipment for measuring the strength of electro-magnetic fields, the transceiver power and signal quality. This equipment is packed into a huge knapsack and the technician must move with it from one BTS to another. Unfortunately, the knapsack has not enough memory for storing all of the measured values. It has a small cache only, that can store values for several seconds. Then the values must be transmitted to the BTS by an infrared connection (IRDA). The IRDA needs direct visibility between the technician and the BTS.

Your task is to find the path between two neighboring BTSes such that at least one them is always visible. For simplicity, a town is modelled as a rectangular grid of $P \times Q$ square fields. Each field is exactly 1 meter wide. For each field, a non-negative integer $Z_{i,j}$ is given, representing the height of the terrain in that place, in meters. That means the town model is made of cubes, each of them being either solid or empty.

The technician is moving in steps (*steps* stands for *Standard Technician's Elementary Positional Shift*). Each step is made between two neighboring square fields in North, South, West or East direction, it is not possible to move diagonally. The step between two fields A and B (step from A to B) is allowed only if the height of the terrain in the field B is not very different from the height in the field A. The technician can climb at most 1 meter up or descend at most 3 meters down in a single step.

At the end of each step, at least one of the two BTSes must be visible. However, there can be some point "in the middle of the step" where no BTS is visible (data are handled by the cache). The BTS is considered visible if there is a direct visibility between the unit cube just above the terrain on the BTSes coordinates and the cube just above the terrain on the square field where the technician is. Direct visibility between two cubes means that the line segment connecting the centers of the two cubes does not intersect any solid cube. However, the line can touch any number of solid cubes. In other words, consider both the BTS and the technician being points exactly half meter above the surface and in the center of the appropriate square field.

Note that the IRDA beam can also go between two cubes that touch each other by their edge, although there is no real space between them. It is because such a beam touches both of these two cubes but does not intersect them.

Input Specification

There is a single positive integer T on the first line of input. It stands for the number of test cases to follow. The first line of each test case contains two integer numbers P and Q, separated by a single space, $1 \le P, Q \le 200$. Then there are P lines, each containing Q integer numbers separated by a space. These numbers are $Z_{i,j}$, where $1 \le i \le P$, $1 \le j \le Q$ and $0 \le Z_{i,j} \le 5000$. After the terrain description, there are four numbers R_1, C_1, R_2, C_2 on the last line of each test case. These numbers represent positions of two BTSes, $1 \le R_1, R_2 \le P$, $1 \le C_1, C_2 \le Q$. The first coordinate determines the row of the town, the second coordinate determines the column.

Output Specification

You are to find the shortest possible path meeting the above criteria. That is, all steps must be done between neighboring fields, the terrain must not elevate or descend too much, and at the end of each step, at least one BTS must be visible.

For each test case, print one line containing the sentence "`The shortest path is` M `steps long.`", where M is the number of steps that must be made. If there is no such path, output the sentence "`Mission impossible!`".

Sample Input

```
4
5 5
8 7 6 5 4
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
1 1 5 1
5 8
2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2
9 9 9 9 9 9 9 2
2 2 2 2 2 2 2 2
1 2 5 1
5 8
2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2
9 9 9 9 9 9 9 2
2 2 2 2 2 2 2 2
1 5 5 1
6 12
5 5 5 5 1 5 5 5 5 5 5 5
5 5 5 5 1 5 5 5 5 5 5 5
5 5 5 5 9 5 5 5 5 5 5 5
5 9 1 5 5 5 5 5 5 5 5 5
5 5 9 5 5 5 5 5 5 5 5 5
5 5 9 5 5 5 5 5 5 5 5 5
6 1 3 12
```

Output for Sample Input

```
The shortest path is 10 steps long.
Mission impossible!
The shortest path is 14 steps long.
The shortest path is 18 steps long.
```