Pathing

Lea is programming the next new hit game, Age of Conquercraft, a real time strategy game. Unfortunately, her units do not run as they are supposed to: Instead of taking the shortest path to the their target, they sometimes take a longer path or run into a dead end and have to walk back. Can you help her and write a better pathfinding algorithm?

The terrain that Lea uses is divided into squares of 1x1 cm which are either passable or impassable. The units are considerd to be a point (they do not occupy space) and can move in any direction (in particular, they do not have to move perpendicular to the grid) but cannot move into impassable squares.

Input

The first line of the input contains an integer t, the number of test cases. t test cases follow, each of them separated by a blank line.

Each test case starts with a description of the grid. The first line contains three integers w h n, where w and h are the width and height of the grid and n is the number of impassable locations. All following coordinates are 1-bases with $(1, 1)$ being the upper left corner.

n lines follow, describing the impassable locations. The i-th line contains four integers x_i y_i w_i h_i, describing an impassable square starting at grid position (x_i, y_i) of width w_i and height h_i in the grid, extending to the right and downward. The grid description is followed by a blank line.

Two lines follow, the j-th of which contains two integers a_j b_j, the first is the start point (a_1, b_1) of the unit and the next is the target point (a_2, b_2).

Output

For each test case, print a line containing "Case #i: c" where i is its number, starting at 1, and c is a space separated list of coordinates $(X_1, Y_1)(X_2, Y_2) \ldots (X_k, Y_k)$ (with brackets and comma) such that

- $(X_1, Y_1) = (a_1, b_1)$ is the start point,
- $(X_k, Y_k) = (a_2, b_2)$ is the target point,
- for every $1 \leq i < k$ it is possible to move from (X_i, Y_i) to (X_{i+1}, Y_{i+1}) in a straight line without entering an impassable grid location, and
- the length of the path along the points in c is minimal.

Any optimal solution (up to relative or absolute errors of at most 10^{-4}) will get accepted. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $5 \leq w, h \leq 5000$
- $0 \leq n \leq 100$
- $1 \leq x_i < x_i + w_i \leq w$ for all $1 \leq i \leq n$.
- $1 \leq y_i < y_i + h_i \leq h$ for all $1 \leq i \leq n$.
- $1 \leq a_j \leq w$ for all $1 \leq j \leq 2$.
- $1 \leq b_j \leq h$ for all $1 \leq j \leq 2$.
- The start point and the target points will neither touch nor be contained in an impassable location.
- The impassable locations will not touch or overlap with each other or the border.

Sample explanation

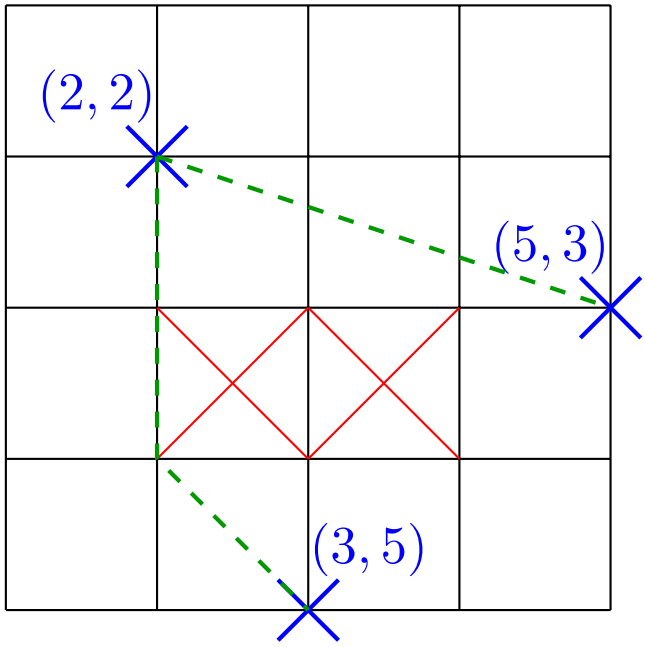

Figure 1: Illustration of the first sample input

In the first sample, depicted in Figure 1, in the first case it is possible to move from the starting point to the target in a straight line. In the second case, this is not possible because of the impassable locations, therefore we move via the point $(2, 4)$. Another accepted solution would be $(2, 2)$ $(2, 3)$ $(2, 4)$ $(3, 5)$. The solution $(2, 2)$ $(4, 3)$ $(4, 4)$ $(3, 5)$ would be wrong because the path is longer then the optimal one.

Sample Input 1	Sample Output 1
2 6 6 1 2 3 2 1 2 2 5 3 6 6 1 2 3 2 1 2 2 3 5	Case #1: (2,2) (5,3) Case #2: (2,2) (2,4) (3,5)

Sample Input 2

```
4
7 6 2
2 2 2 1
3 4 3 1
3 1
4 6

7 7 3
2 3 2 1
3 5 3 1
5 2 1 2
3 2
5 7

7 7 2
4 2 2 1
3 2 1 3
2 1
3 1

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

Sample Output 2

```
Case #1: (3,1) (2,2) (2,3) (3,5) (4,6)
Case #2: (3,2) (4,3) (5,4) (6,5) (6,6) (5,7)
Case #3: (2,1) (3,1)
Case #4: (4,8) (1,8)
```