H Input

In a recent programming contest, one of the problems was about tiling floors with rectangular tiles. The input specification reads like this:

> The input contains several floors. The first line of the input gives the number of floors. Each floor is described in several lines. The first line contains two positive integers: the length and width of the floor, in millimeters. A floor is at most 40 000 mm long or wide. The next line contains a single number: the number t of tiles ($1 \leq t \leq 100$). The following t lines each contain the description of a tile. A tile is given as four integers:
> *xl yl xh yh*
> where (*xl*, *yl*) are the coordinates of the lower left corner of the tile, and (*xh*, *yh*) are the coordinates of the upper rightmost corner of the tile. A tile always has a positive area. The order of the coordinates of the floor and those of the tile coincide, of course.
> You may assume that the tiles are mutually disjoint, and cover the floor, the whole floor, and nothing but the floor.

The last line of this specification raised some problems. Not for the contestants, but for the judges. Some of the test cases consist of many tiles. How can we be sure that our input file meets this condition? What we need is a checking program that verifies this condition.

Problem

Given an input file in the above format, find out for each floor whether the tiles

1. are disjoint,
2. do not lie outside the floor,
3. do cover the floor.

Input

The input contains several floors. The first line of the input gives the number of floors. Each floor is described in several lines. The first line contains two positive integers: the length and width of the floor, in millimeters. A floor is at most 40 000 mm long or wide. The next line contains a single number: the number t of tiles ($1 \leq t \leq 100$). The following t lines each contain the description of a tile. A tile is given as four integers:
xl yl xh yh
where (*xl*, *yl*) are the coordinates of the lower left corner of the tile, and (*xh*, *yh*) are the coordinates of the upper rightmost corner of the tile. A tile always has a positive area. The order of the coordinates of the floor and those of the tile coincide, of course.

Output

For each floor the output contains a single line, containing one of the following words:

`NONDISJOINT`	if overlapping tiles occur;
`NONCONTAINED`	if no overlapping tiles occur, but some tiles go outside the floor;
`NONCOVERING`	if no overlapping tiles occur, and no tiles go outside the floor, but some parts of the floor are not covered;
`OK`	if none of these is true.

Example

input	output
4 4 3 2 0 0 2 2 1 1 5 5 4 3 2 0 0 2 2 −2 2 5 5 4 3 2 0 0 2 2 2 0 4 2 4 3 3 0 0 2 2 2 0 4 2 0 2 4 3	NONDISJOINT NONCONTAINED NONCOVERING OK