Planting trees
1 second, 256 MB

A field of width **R** and length **C** is divided into unit **R** x **C** cells of size 1 x 1. Each cell is referred to by its row **i** (1<=**i**<=**R**) and its column **j** (1<=**j**<=**C**). **N** trees will be planted in N of these cells. For 1<=**k**<=**N**, tree **k** will be planted in the cell at row $\mathbf{A_k}$ and column $\mathbf{B_k}$. Trees on the same row and trees on the same column look very beautiful together. You want to find a tree that *maximizes* the number of trees on the same row and the same column.

Consider the following example where R = 5, C = 7, and N = 6. The trees will be planted in to the following cells

k	1	2	3	4	5	6
A_k	1	3	3	5	5	3
B_k	3	3	1	6	4	4

The locations in the field can be shown below. The cells contain the tree number. For example, tree 1 will be planted at the cell at row A_1=1 and column B_1=3.

		1			7	
3		2	6			
			5		4	

You want to find the index **k** such that the number of trees in the same row or same column as tree **k** (including tree **k** itself) is maximized. If there are many trees, you should answer the smallest index. In the example above, there are 4 trees in the same row or same column with tree 2. There are also 4 trees in the same row or colum with tree 6. But you should answer the smallest index; therefore, the correct answer for this case is 2.

Input
The first line contains three integers **R C** and **N**. (1 <= **R** <= 100,000; 1 <= **C** <= 100,000; 1 <= **N** <= 100,000) There are test cases worth 30% points that **R** <= 100, **C** <= 100, and **N** <= 1,000.

The next N lines describe the cell locations. More specifically, for 1<=**k**<=**N**, line 1+**k** contains two integers $\mathbf{A_k}$ and $\mathbf{B_k}$ (1<=$\mathbf{A_k}$<=**R**; 1<=$\mathbf{B_k}$<=**C**). No pairs of trees will be planted on the same cell.

Output
You program should output two integers: the **index k** of the tree with the maximum number of trees on the same row or column and the **number** of trees on the same row or column of tree **k**.

Example

Input	Output
5 7 6 1 3 3 3 3 1 5 6 5 4 3 4	2 4