# natureoculusC:\Users\Ivo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\liana.pngC:\Users\Ivo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\liana.pngProblem 2 –Nature’s Prophet

Furion loves nature and that is why he has a beautiful square garden. He wants to plant it with magical flowers, so it can be even more beautiful. No one knows why, but he actually needs a software program to do that, that is why you’ll write one for him.

You will be given **N** and **M** – **integers**, indicating the **dimensions** of the **square garden**. The garden is **empty at the beginning** – it has no flowers. Furion wants every place for a flower to be presented with a **zero (0)** when it is **empty**. After you’ve finished creating the garden, you will start receiving two integers – **Row** and **Column**, **separated** by a **single space** – which represent the **position** at which Furion **currently plants a flower**. This happens until you receive the command **“Bloom Bloom Plow”**. When you receive that input, **all planted flowers** should **bloom**.

The flowers are **magical**. When a flower **blooms** it instantly **blooms flowers** to **all places** to its **left**, **right**, **up**, and **down**, **increasing** their **value** with **1**. Flowers can bloom **multiple times**, and **each time** the flower blooms – it becomes more and more beautiful, which means its **value increases**. The blooming of flowers is done from the **top-left** corner of the garden to the **bottom-right** one.

**Note**: If one flower blooms and affects several places, and then another flower blooms and affects one of the **first flower’s affected places**, it does **NOT** override their values with 1 again. Instead it blooms them one more time – **increasing** their value with **1**.

### Input

* On the first line of input you will receive two integers, separated by a single space – indicating the dimensions of the garden.
* On the next several lines you will be receiving two integers separated by a single space – indicating the position at which Furion currently plants a flower.
* When you receive the input line **“Bloom Bloom Plow”** the input sequence should end.

### Output

* The output is simple. Print the whole garden – each row of it on a new line, and each column – separated by a **single space**.

### Constraints

* The dimensions of the matrix (**N** and **M**) will be integers in the range [3, 500].
* The integers received as position of planting a flower will **always** be inside the matrix.
* The amount of input commands will be in the range [0, **N \* M**].
* Flowers will **always** be planted on **empty** places.
* Allowed time/memory: 100ms/16MB

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| **5 5**  **1 1**  **3 3**  **Bloom Bloom Plow** | **0 1 0 1 0**  **1 1 1 2 1**  **0 1 0 1 0**  **1 2 1 1 1**  **0 1 0 1 0** | **The garden has 5 rows and 5 columns.**  **They are all **0** at the beginning.**  **The planted flowers are at [1, 1] and [3, 3].**  **The affected places are:**  **0 0 0 0 0**  **0 0 0 0 0**  **0 0 0 0 0**  **0 0 0 0 0**  **0 0 0 0 0**  **We receive the blooming command and we bloom the flowers from top-left to bottom-right corner of the garden.**  **First we reach the first flower, and we bloom it, increasing all affected fields’ value with 1.**  **0 1 0 0 0**  **1 1 1 1 1**  **0 1 0 0 0**  **0 1 0 0 0**  **0 1 0 0 0**  **Then we bloom the second flower, and we increase all affected fields’ (even those from the first flower) value with 1.**  **0 1 0 1 0**  **1 1 1 2 1**  **0 1 0 1 0**  **1 2 1 1 1**  **0 1 0 1 0** |

|  |  |
| --- | --- |
| **Input** | **Output** |
| **4 4**  **0 0**  **3 3**  **1 1**  **2 2**  **Bloom Bloom Plow** | **1 2 2 2**  **2 1 2 2**  **2 2 1 2**  **2 2 2 1** |