## \* Crossfire

You will receive two integers which represent the dimensions of a matrix. Then, you must fill the matrix with increasing integers starting from 1, and continuing on every row, like this:  
first row: 1, 2, 3… n  
second row: n+1, n+2, n+3… n + n.

You will also receive several commands in the form of 3 integers separated by a space. Those 3 integers will represent a row in the matrix, a column and a radius. You must then **destroy** the cells which correspond to those arguments **cross-like.**

**Destroying** a cell means that, **that** cell becomes completely **nonexistent** in the matrix.Destroying cells **cross-like** means that you form a **cross figure**, with center point - equal to the cell with coordinates – the **given row** and **column**, and **lines** with length equal to the **given radius**. See the examples for more info.

The input ends when you receive the command “Nuke it from orbit”. When that happens, you must print what has remained from the initial matrix.

### Input

* On the first line you will receive the dimensions of the matrix. You must then fill the matrix according to those dimensions.
* On the next several lines you will begin receiving 3 integers separated by a single **space**, which represent the row, col and radius. You must then destroy cells according to those coordinates.
* When you receive the command “Nuke it from orbit” the input ends.

### Output

* The output is simple. You must print what is left from the matrix.
* Every row must be printed on a new line and every column of a row - separated by a space.

### Constraints

* The dimensions of the matrix will be integers in the range [2, 100].
* The given rows and columns will be valid integers in the range [-231 + 1, 231 - 1].
* The radius will be in range [0, 231 - 1].
* Allowed time/memory: 250ms/16MB.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| 5 5  3 3 2  4 3 2  Nuke it from orbit | 1 2 3 4 5  6 7 8 10  11 12 13  16  21 | Initial matrix:  1 2 3 4 5  6 7 8 9 10  11 12 13 14 15  16 17 18 19 20  21 22 23 24 25  Result from first destruction:  1 2 3 4 5  6 7 8 10  11 12 13 15  16  21 22 23 25  Result from second destruction:  1 2 3 4 5  6 7 8 10  11 12 13  16  21 |

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 5  4 4 4  Nuke it from orbit | 1 2 3 4  6 7 8 9  11 12 13 14  16 17 18 19 |