## Bombs (Multidimensional Arrays)

You will be given a square matrix of integers, each integer separated by a **single space**, and each row on a new line. Then on the last line of input you will receive indexes - coordinates to several cells separated by a **single space**, in the following format: **row1,column1 row2,column2 row3,column3…**

On those cells there are bombs. You have to proceed **every** **bomb**, one by one in the order they were given. When a bomb explodes deals damage **equal** to its **own** **integer** **value**, to **all** the cells **around** it (in every direction and in all diagonals). One bomb can't explode more than once and after it does, its value becomes **0**. When a cell’s value reaches **0 or below**, **it dies**. Dead cells **can't explode**.

You must **print the count of all alive cells** and **their sum**. Afterwards, print the matrix with all of its cells (including the dead ones).

### Input

* On the first line, you are given the integer N – the size of the square matrix.
* The next N lines holds the values for every row – N numbers separated by a space.
* On the last line you will receive the coordinates of the cells with the bombs in the format described above.

### Output

* On the first line you need to print the count of all alive cells in the format:

“**Alive cells: {aliveCells}**”

* On the second line you need to print the sum of all alive cell in the format:

“**Sum: {sumOfCells}**”

* In the end print the matrix. The cells must be **separated by a single space**.

### Constraints

* The size of the matrix will be between **[0…1000].**
* The bomb coordinates will **always** be in the matrix.
* The bomb’s values will always be **greater** than **0**.
* The integers of the matrix will be in range **[1…10000].**

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 4  8 3 2 5  6 4 7 9  9 9 3 6  6 8 1 2  1,2 2,1 2,0 | Alive cells: 3  Sum: 12  8 -4 -5 -2  -3 -3 0 2  0 0 -4 -1  -3 -1 -1 2 | First the bomb with value **7** will explode and reduce the values of the cells around it. Next the bomb with coordinates **2,1** and value **9** will explode and reduce its neighbour cells. In the end the bomb with coordinates **2,0** and value **9** will explode. After that you have to print the count of the alive cells, which is 3, and their sum is 12. Print the matrix after the explosions. |
| 3  7 8 4  3 1 5  6 4 9  0,2 1,0 2,2 | Alive cells: 3  Sum: 8  4 1 0  0 -3 -8  3 -8 0 |  |