## 3.Air Pollution

Write a program that tracks the **pollution** **in the air** Sofia. You will receive **two arguments** – the **first** is the **map** of Sofia represented by a **matrix of numbers** and the second is an **array of strings** representing the **forces** **affecting** the **air quality**. The **map** will **always** be with **5 rows** and **5 columns** in a **total** **of** **25 elements - blocks**. Each block’s particle pollution (PM) is **affected** by **3 forces** receivedin the following formats**:**

* **"breeze {index}" –** index is **the row** where **all column’s value drops** by **15** PM
* **"gale {index}" –** index is **the column in all rows** where **value drops** by **20** PM
* **"smog {value}" – all** **blocks** in the map **increase** equally by **the given value’s** PM

The threshold in each block is **50** PM. If it is **below** **that number**, the block’s air is considered **normal** but if it **reaches or goes over it,** that block’s air is considered **polluted**. Also, note that the **polluted particles** in a block **cannot go below** **zero**.

Finally, your program needs to **find** if there are **any polluted blocks** and **print them** in the format given below.

**Input**

You will receive **two** **arguments**:

* The **first** argument is an **array with five strings** – **rows** of the matrix with **columns separated by space** that must be parsed as **numbers**, representing the **map of Sofia**.
* The **second** argument is an **array of strings** – each **string** consists of one of the **words (breeze/gale/smog)** and a **number** **separated by space**, representing the **different forces**.

**Output**

Print on the **console** a **single line**:

* If there are **polluted blocks** in the map, **use** their **coordinates** in the following format:
* **"[{rowIndex}-{columnIndex}]"**

Note that you must **start** from the **top left corner** of the map moving to the **bottom right corner** **horizontally**. Then **separate** each **formatted block’s coordinates** with **comma and space** and print them in a single line in the following format:

* **"Polluted areas: {block1}, {block2}, {block3}, …"**
* If there are **no polluted blocks** in the map print:
* **"No polluted areas"**

**Constraints**

* The **number** of **rows** and **columns** for the **matrix** will **always** be **5**
* The **number** in each block will be an **integer** in the range **[0..1000] inclusive**
* The **number** of **elements** in the **second input argument** will be in the range **[0..100] inclusive**
* Given **smog’s** **value** will be an **integer** in the range **[0..100] inclusive**
* Given **indexes** will **always** be **valid**

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| ['5 7 72 14 4',  '41 35 37 27 33',  '23 16 27 42 12',  '2 20 28 39 14',  '16 34 31 10 24'],  ['breeze 1', 'gale 2', 'smog 25'] | Polluted areas: [0-2], [1-0], [2-3], [3-3], [4-1] |

**Explanation**

Graphic diagram explaining the **first example’s** program flow:

**input**

**breeze 1 (-15)**

**gale 2 (-20)**

**smog 25 (+25)**

**polluted areas**



|  |  |
| --- | --- |
| **Input** | **Output** |
| ['5 7 3 28 32',  '41 12 49 30 33',  '3 16 20 42 12',  '2 20 10 39 14',  '7 34 4 27 24'],  ['smog 11', 'gale 3', 'breeze 1', 'smog 2'] | No polluted areas |

|  |  |
| --- | --- |
| **Input** | **Output** |
| ['5 7 2 14 4',  '21 14 2 5 3',  '3 16 7 42 12',  '2 20 8 39 14',  '7 34 1 10 24'],  ['breeze 1', 'gale 2', 'smog 35'] | Polluted areas: [2-1], [2-3], [3-1], [3-3], [4-1], [4-4] |