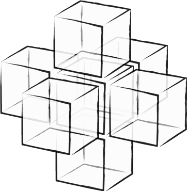
# Problem 4 – Stars in the Cube

We are given a **cube of Latin letters** of size **n** \* **n** \* **n** given as **n** layers (square matrices) of size **n** \* **n**. A **cube**, split into **layers**, is shown on the **right** (each letter is shown as different color).

Write a program to calculate how many **3D stars of 7 cells** (center, up, down, left, right, front, back) holding **equal letters** exist in the cube.

The form of the **3D star** is shown at the figure on the **left**. The same letter can be shared between several stars (stars can overlaps inside the cube).

## Input

* The input is read from the console.
* The first line holds an integer **n** – the size of the cube.
* At the next **n** lines the layers of the cube are given as sequence of **n** matrices separated by ‘**|**’.
* The cells in each matrix row are separated by space (see the examples below).

## Output

* At the **first line** at the console print the **total number of 3D stars** of equal letters in the cube.
* At the next few lines, for **each letter** in alphabetical order print the **number of its stars** found in the cube in format “letter -> count”. Skip the letters that don’t have any stars in the cube.

## Constraints

* The size of the cube **n** is integer in the range **[1…75]**.
* All cube **cells** hold lowercase **Latin letters** in the range **[‘a’…‘z’]**.
* Time limit: **100 ms**. Allowed memory: **16 MB**.

## Sample Input and Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  a a a a a | a p a a a | a a a a a | a p a a a | p p p p p  a p a a a | p p p a a | a p a a a | p p p a a | p p p p p  a a a a a | a p x x a | a p a a a | p p p z a | p p p p p  a a x x a | a x x x x | a a x x a | a p z z z | p p p z p  a a a a a | a a x x a | a a a a a | a a a z a | p p p p p | 6  a -> 1  p -> 3  x -> 2 |
| 3  x x x | x a x | x x x  x a x | a a a | x a x  x x x | x a x | x x x | 1  a -> 1 |
| 2  a a | a a  a a | a a | 0 |