

Softuniada 2019

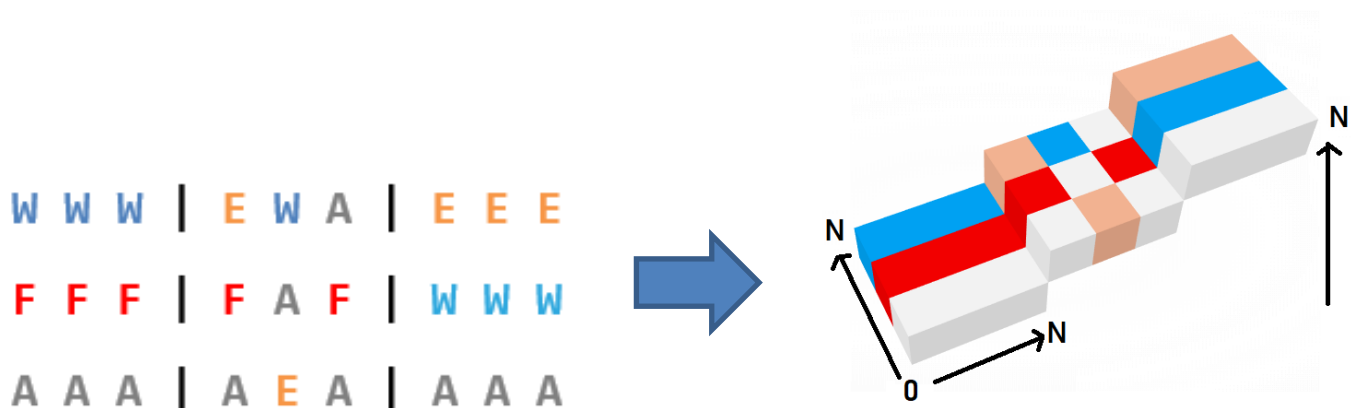
Elemelons

If there is a watermelon, then there should be earthmelon, firemelon and airmelon. Introducing, the Elemelons! Doctor Sanity has a very unusual 3-D garden with elemenlons. The elemelons like to morph into each other when they are not observed, and Doc doesn't like that. He has called for your help in implementing an algorithm which helps him observe his surroundings while reaping elemelons so that they don't change.

You will be given **N** – the **dimensions** of the **3-D matrix** – the garden of Doctor Sanity, and the matrix itself afterwards, in a series of **N rows** containing the **columns** of **N matrices** – the **layers** of the 3-D matrix. The matrix will consist of the following symbols: **W** (Watermelon), **E** (Earthmelon), **F** (Firemelon), **A** (Airmelon).

For an example, let's color **W** (blue), **E** (orange), **F** (red) **A** (gray-white)

Check the examples below, for more info:



After you've successfully initialized the 3-D matrix, you'll start receiving **coordinates** – the cell that Doctor Sanity will **harvest** a melon from. **Upon harvesting** a melon, that cell in the 3-D matrix should be **set** to '0'. **All other melons, except the ones in direct sight** of Doc (**up, down, left, right, front, back** from the **currently harvested cell**), should **morph**.

The **morphing order** is: **Watermelon (W) -> Earthmelon (E) -> Firemelon (F) -> Airmelon (A) -> Watermelon (W)**... and so on.

Upon receiving the command "**Melolemonmelon**", the input sequence should end, and you should print the current state of the 3-D matrix in the same format, that you've received it from the input. Then you should end the program.

Input

The input consists of several lines:

- On the first line you will receive an integer number **N** – the size of the 3-D matrix.
- At the next **N** lines the **layers** of the **3-D matrix** are given (from **bottom** to **top**) as a sequence of **N** matrices separated by " | ".
- Afterwards, you will start receiving **coordinates** of a **cell** in the following format: "**{layer} {row} {column}**"
- When you receive the command "**Melolemonmelon**" the input sequence should end.

Output

As output you must print the current state of the 3-D matrix, in the same format as it came from the input – rows, containing the columns of N matrices – the layers of the cube (from **bottom** to **top**).

Constraints

- The integer **N** will be in **range [0, 50]**.
- The 3-D matrix will always be in a **valid format** and will **only contain** the following characters: 'W', 'E', 'F', 'A'.
- The **coordinates** for the **cell** to be **harvested**, will **always be inside** the 3-D matrix.
- Allowed time / memory: 100ms / 16MB.

Examples

Input	Output	Comment
3 W W W E W A E E E F F F F A F W W W A A A A E A A A A 1 1 1 Melolemonmelon	E E E F W W F F F A F A F Ø F E W E W W W W E W W W W	The cell at layer 1, row 1, column 1 was changed to 'Ø', as it was harvested. The yellow marked cells were in direct sight of Doctor Sanity, which is why the melons inside them didn't morph . All other cells' melons morphed into the next melon in order .

Input	Output
4 A W F A W W W W W W W W A A A A W F W F F F F F A A A A F F F F A W E W E E E E A A A A W E E W A F W E W W W W W W W W W W W W 1 1 1 2 3 2 3 1 3 Melolemonmelon	F A E F A F A A A A A A F F F E A W A E W Ø W E F E F E E E W Ø F A W A W A W W F F E F A W W F F E A W A A F A A F Ø F A A F A