## Problem 2 – Plus-Remove

You are given a sequence of **text lines**, holding symbols, small and capital Latin letters. Your task is to **remove all "plus shapes"** in the text. They may consist of small and capital letters at the same time, or of any symbol. All of the **X shapes** below are **valid**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| a  aaa  a | B  BBB  B | T  TtT  T | p  PPp  P | &  &&&  & | \*  \*\*\*  \* | etc. |

Every **"plus shape"** is 3 by 3 symbols crossing each other on 3 lines. Single **"plus shape"** can be part of **multiple** **"plus shapes"**. If new **"plus shapes"** are formed after the first removal **you don't have** to remove them.

### Input

The input data comes as **array of strings**, holding the text lines.

### Output

Print at the console the input data after removing all **"plus shapes"**.

### Constraints

* The input will be passed to the first JavaScript function found in your code as **array of strings**.
* Each input line will hold 1…100 Latin letters.
* The number of input lines will be in the range [1..100].
* Allowed working time: 0.2 seconds. Allowed memory: 16 MB.

**function** *solve*(input) {  
  
 **let** matrix = input.map(row=> row.**split**(**''**));  
 **let** currentMatrix = input.map(row=> row.toLowerCase().split(**''**));  
  
 **for** (**let** i = 0; i < currentMatrix.**length** - 2; i++) {  
 **for** (**let** j = 0; j < currentMatrix[i].length - 1; j++) {  
  
 **let** current = currentMatrix[i][j + 1];  
 **let** current1 = currentMatrix[i + 1][j];  
 **let** current2 = currentMatrix[i + 1][j + 1];  
 **let** current3 = currentMatrix[i + 1][j + 2];  
 **let** current4 = currentMatrix[i + 2][j + 1];  
  
  
 **if**(current == current1 && current == current2 && current == current3 && current == current4){  
  
 matrix[i][j + 1] = **''**;  
 matrix[i + 1][j] = **''**;  
 matrix[i + 1][j + 1] = **''**;  
 matrix[i + 1][j + 2] = **''**;  
 matrix[i + 2][j + 1] = **''**;  
 }  
 }  
  
 }  
 **for** (**let** row **of** matrix) {  
 **console**.log(row.join(**''**));  
 }  
  
}  
  
*solve*([  
 **'ab\*\*l5'**,  
 **'bBb\*555'**,  
 **'absh\*5'**,  
 **'ttHHH'**,  
 **'ttth'**]);

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| a**b**\*\*l**5**  **bBb**\***555**  a**b**s**h**\***5**  tt**HHH**  ttt**h** | a\*\*l  \*  as\*  tt  ttt | 8**88**\*\***t**\*  **8888ttt**  **888ttt**<<  \***8**\*0**t**>>hi | 8\*\*\*  <<  \*\*0>>hi | @s**@**a**@**p**@**una  p**@@@@@@@**@dna  @6**@**t**@\*@\***ego  vdig**\*\*\*\*\***ne6  li??^**\***^**\*** | @sapuna  p@dna  @6tego  vdigne6  li??^^ |