

## D. "Damaged ticket"

### Statement

At last came the day of Bob Kerley's concert! Your best agents are checking the tickets of the people standing in a queue in front of the concert hall. Strangely, the queue keeps getting longer and longer... Your agents can't scan the tickets of many people, damaged as they are, folded in eight or crammed in a pocket. They urge you to find a solution by writing a program that tries to fix the damaged codes.

Lucky you! These 2D barcodes, composed of dashes '-' and spaces represented by the '\*' character, include control data! They provide you with the **total number of dashes on each line and column** of the barcode. Thanks to this information, you can determine whether the code where some dashes are missing can be fixed or not. If it is possible to deduce the original (undamaged) code from the control data and the damaged code, print the fixed version. If it isn't, well the owner of the ticket should have been more careful.



When you realise you're not listening to reggae tonight...

### Input

- On the first line, two integers  $L$  and  $C$  separated by a space, corresponding to the **number of lines and the number of columns of the 2D barcode** ( $1 \leq L, C \leq 20$ );
- On each of the  $L$  following lines,  $C$  characters '-' or '\*' corresponding to a **dash** or a **space** in the barcode;
- On the following line,  $L$  space-separated integers  $l_i$  corresponding to the **number of dashes that should be on each line  $i$**  of the barcode ( $0 \leq l_i \leq C$ );
- On the last line,  $C$  space-separated integers  $c_j$  corresponding to the **number of dashes that should be on each column  $c$**  of the barcode ( $0 \leq c_j \leq L$ ).

## Output

If the barcode can be **reconstructed in a unique way**, print it ; otherwise, print 'NON'.

## Examples

### Example 1

Input	Output
<pre> 4 5 -***- ***-* *-*** ***** 3 1 1 2 2 1 0 2 2 </pre>	<pre> -***- ***-* *-*** -***- </pre>

In this first example, the barcode can be recreated in a single way (the one in the output). Any other combination would not respect the control data that tells us that there is a missing dash on the first line, and two missing dashes on the last one, knowing that there are 3 dashes missing on the first, 4th and 5th columns.

### Example 2

Input	Output
<pre> 4 2 ** ** ** ** 1 0 0 1 1 1 </pre>	<pre> NON </pre>

In this second example, the code cannot be fixed because there are two different ways to recreate it according to the control data : with dashes on the first line, first column and last line, last column, or with dashes on the first line, last column and last line, first column.