

WC '17 Contest 3 J4 - Meme Generator

Time limit: 1.0s **Memory limit:** 16M

Woburn Challenge 2017-18 Round 3 - Junior Division

As part of your internship at a popular [meme generation site](#), you've been tasked with implementing a new ASCII art feature!



You're given a grid of non-whitespace characters with R rows and C columns ($5 \leq R, C \leq 100$), representing a meme image. The user will then be able to specify two pieces of custom text to overlay onto the image, near its top and bottom edges, in order to enhance its comedic effect.

The user will first specify a non-empty string T with length no greater than $C - 2$. Each of its characters will be either an uppercase letter or an underscore, and it will neither start nor end with an underscore. This string should be laid on top of the image in the second row from the top, and horizontally centered. If it can't be perfectly centered (for example, if its length is odd while C is even), then it should instead be placed slightly to the left, as close to centered as possible. Underscores should be omitted, allowing the original image's characters to show through at those locations instead.

Finally, the user will specify a non-empty string B with the same constraints as T . It should similarly be laid on top of the image in the second row from the bottom, and horizontally centered.

Your task is to generate the resulting $R \times C$ image after both strings T and B have been laid on top of it, and give it back to the user, so that they can go post it on various social media platforms and obtain millions of well-deserved upvotes for their original, creative comedic content.

Input Specification

The first line of input consists of two space-separated integers, R and C .

R lines follow, the i -th of which consists of C characters representing the i -th row of the image grid, for $i = 1 \dots R$.

The next line consists of a single string, T .

The next line consists of a single string, B .

Output Specification

Output R lines with C characters per line, the i -th of which should be the i -th row of the updated image grid.

Sample Input

17 28

```
....._....._...
.....|+\...../++|..
.....|_ \_____.../+++|..
...../=====\_ /++++|..
.....\_ /===== \++++|..
... \ /===== \_ ++|..
.. /===/#\===== \_ |..
.. |==\##/====/##\===== \..
. |===== \##/===== \.
. |===== +++++==== |
|===#####===== +++++==== |
|=== |#####===== +++++==== |
|=== \###___/===== +++++==== |
|===== +++++==== |
. |===== |.
.. \===== |.
... \===== /..
VERY_ART
WOW
```

Sample Output

```
....._....._...
.....|+VERY.ART.../++|..
.....|_ \_____.../+++|..
...../=====\_ /++++|..
.....\_ /===== \++++|..
... \ /===== \_ ++|..
.. /===/#\===== \_ |..
.. |==\##/====/##\===== \..
. |===== \##/===== \.
. |===== +++++==== |
|===#####===== +++++==== |
|=== |#####===== +++++==== |
|=== \###___/===== +++++==== |
|===== +++++==== |
. |===== |.
.. \=====WOW===== |.
... \===== /..
```

Sample Explanation

Much ASCII.