War

You are fighting a war and the battlefield can be mapped to an m x n matrix M such that

a. A pair (i, j) 0 <= i < m & 0 <= j < n represents a location on the battlefield
b. An element M[i][j] represents either
    i. Safe zone which is always represented by letter ‘z’
    ii. Enemy military target represented by letters [a-x]

	0	1	2	3	4	5	6
0	z	z	z	z	z	**a**	z
1	z	**b**	z	z	z	z	**c**
2	z	z	z	**d**	z	z	z
3	z	z	z	**e**	**f**	**g**	z
4	**h**	z	z	z	z	z	z
5	z	z	z	z	z	**i**	z
6	z	z	z	**j**	z	**k**	z

You have received information related to enemy military targets but it’s encoded such that you know the column index of each target but you don’t know the row index.

You do have access to an additional piece of information though, using which you should be able to recover the row index for each target and hence the whole matrix. (You can assume every element of the matrix which isn’t an enemy target is a safe zone represented by letter ‘z’)

Inputs :

a. **(m, n) -** dimensions of the matrix. m is row count and n is column count.

b. **targets -** An array of military targets picked from row 0 to row m-1 and for each row, column 0 to column n-1.
e.g. [‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, ‘h’, ‘i’, ‘j’, ‘k’] for matrix in the image above.

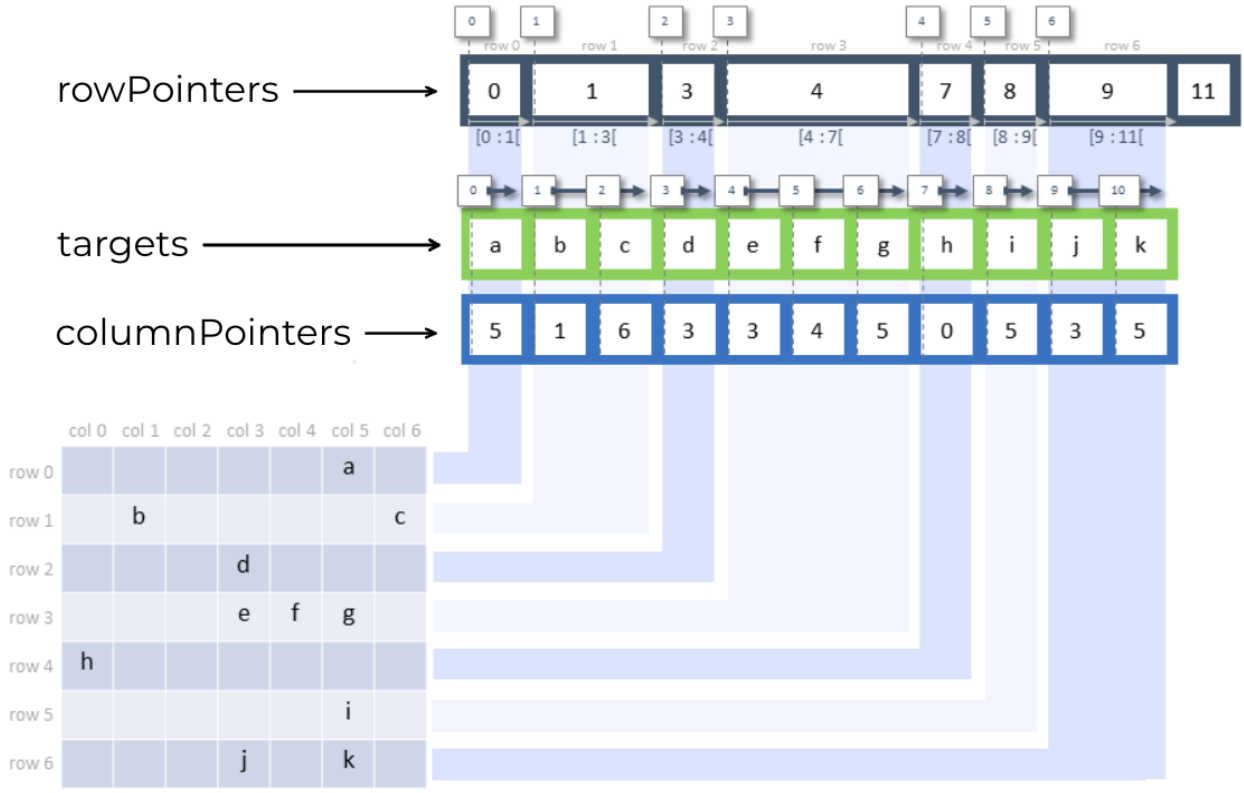
c. **columnIndices -** An array of column indices for elements in **targets**.
e.g. [5, 1, 6, 3, 3, 4, 5, 0, 5, 3, 5]

d. **rowPointers -** An array of length m + 1 such that for row **i**,
range - **[rowPointers[i] , rowPointers[i + 1])** i.e. rowPointers[**i**] inclusive to rowPointers[**i** + 1] exclusive represents indices in **targets** array that belong to row i.e.
e.g. [0, 1, 3, 4, 7, 8, 9, 11] for the matrix in the image above.

Further explanation for **rowPointers -** Matrix has m=7 rows.
Array [0, 1, 3, 4, 7, 8, 9, 11] is of length m + 1 = 8 and defines 7 ranges.

i. Range [0, 1) of size 1 for the first row containing single element ‘a’ at index 0 in **targets**.
ii. Range [1, 3) of size 2 for the second row containing 2 elements ‘b’ & ‘c’ at index 1 & 2 in **targets**.
iii. Range [3, 4) of size 1 for the third row containing single element ‘d’ at index 3 in **targets**.
iv. Range [4, 7) of size 3 for the fourth row containing 3 elements ‘e’, ‘f’ & ‘g’ at index 4, 5 & 6 in **targets**.
v. Range [7, 8) of size 1 for the fifth row containing single element ‘h’ at index 7 in **targets**.
vi. Range [8, 9) of size 1 for the sixth row containing single element ‘i’ at index 8 in **targets**.
vii. Range [9, 11) of size 2 for the last seventh row containing 2 elements ‘j’ and ‘k’ at index 9 & 10 in **targets**.

Image for visual understanding



Your task is - given **m**, **n**, **targets**, **columnIndices** and **rowPointers** , return a **m** x **n** matrix representing the enemy targets on the battlefield. All elements of the matrix which are not enemy targets should be assumed to be safe zones represented by letter ‘z’.

Input / Outputs (with example matrix) -

Example 1:

	0	1	2	3	4	5	6
0	z	z	z	z	z	**a**	z
1	z	**b**	z	z	z	z	**c**
2	z	z	z	**d**	z	z	z
3	z	z	z	**e**	**f**	**g**	z
4	**h**	z	z	z	z	z	z
5	z	z	z	z	z	**i**	z
6	z	z	z	**j**	z	**k**	z

Matrix shown for explanation purpose only

Input :

```
7
7
a b c d e f g h i j k
5 1 6 3 3 4 5 0 5 3 5
0 1 3 4 7 8 9 11
```

Output :

```
z z z z z a z
z b z z z z c
z z z d z z z
z z z e f g z
h z z z z z z
z z z z z i z
z z z j z k z
```

Example 2:

	0	1	2	3	4	5
0	z	z	z	z	z	**l**
1	z	z	z	z	z	z
2	z	z	z	**m**	**n**	z
3	**o**	z	z	z	z	z
4	z	z	**p**	z	z	z
5	z	z	z	z	**q**	z
6	**r**	**s**	z	**t**	**u**	z

Matrix shown for explanation purpose only

Input :

```
7
6
l m n o p q r s t u
5 3 4 0 2 4 0 1 3 4
0 1 1 3 4 5 6 10
```

Output :

```
z z z z z l
z z z z z z
z z z m n z
o z z z z z
z z p z z z
z z z z q z
r s z t u z
```