

Simple Image Mirroring Program

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

In this problem we are given an image which is a 2D matrix of integers. The matrix has N rows and M columns where each entry in the matrix is an integer value from 0 to 9. You are supposed to perform some queries on the image which will be described below. However, note that the image is given to you as a 1D array of size $N \times M$ and at no point in your program you can store it as a 2D matrix. This constraint is strict.

Queries: There are 2 types of queries on the image - Horizontal Mirroring, and Vertical Mirroring.

Example:

Consider an image represented by the following matrix -

```
1 2
3 4
```

After **horizontal** mirroring, the original image will be -

```
2 1
4 3
```

After **vertical** mirroring, the original image will be -

```
3 4
1 2
```

Note 1: You can only use 1D arrays/vectors. Usage of 2D arrays/vectors anywhere in your code will fetch you zero marks.

Note 2: You are not expected to use any Object Oriented concepts for this problem.

Input Format

First line has 2 inputs M and N , where M is the width of the image (number of columns), and N is the height of the image (number of rows).

The next line contains $N \times M$ elements. Note: The image is given to you as flattened. i.e. The first row of the image is from 0 to $M-1$, the second row of the image is from M to $2M-1$, and so on.

The next line contains the query **string** of length k . The string will be composed of 2 characters - x representing horizontal mirroring, and y representing vertical mirroring. Example - $xyyxyxy$.

Constraints

$1 \leq M \leq 2500$ (width of image)

$1 \leq N \leq 2500$ (height of image)

$1 \leq k \leq 2500$ (size of query string)

$0 \leq$ Each image element ≤ 9

Make your algorithm as efficient as possible. If you implement an $O(k \times M \times N)$ algorithm, at most 6 out of the 10 test cases will pass.

Output Format

You are expected to perform all the mirror queries and display the final image only. You should not display the intermediate outputs.

The output should be the image displayed in the matrix form, that is, each row on a separate line.

Sample Input 0

```
2 3
1 2 3 4 5 6
xyyx
```

Sample Output 0

```
6 5
4 3
2 1
```

Explanation 0

The original image is -

```
1 2
3 4
5 6
```

The query string is - **xyyx**, i.e. 4 queries - x , then x , then y , then x .

After processing the **first** query, the resultant image will be -

```
2 1
4 3
6 5
```

After processing the **first three** queries, the resultant image will be -

```
5 6
3 4
1 2
```

After processing the **all the four** queries, the resultant image will be -

```
6 5
4 3
2 1
```