Title: TwoDCipher

Problem Statement:

You have to encrypt some text, but isn't encrypting text in 1 dimension a little bit boring? That's why we came up with the 2D Cipher.

Firstly, your text is displayed in an n x m grid, then you do a Caesar shift on each character, with the offset for that character being the row, column positions of character in the grid multiplied together. The grid is 1-indexed.

A Caesar shift of the character returns the character shifted rightwards by the offset amount, wrapping around if necessary. For example with an offset of 2, 'a' becomes 'c', 'b' becomes 'd' … y becomes 'a', z becomes 'b'

(Bonus follow up: Can you also implement a decrypting function?)

Input:

Integers n, m in a single line. 1 <= m, n <= 100
On the next line, a string s containing only lowercase letters. (s.length = n*m)

Output:

The 2D Cipher of size n x m.

Example:

Sample Input 1:

```
4 5
aaaaaaaaaaaaaaaaaaaa
```

Sample Output 1:

```
bcdef
cegik
dgjmp
eimqu
```

Sample Input 2:

```
5 7
thequickbrownfoxjumpsoverthelazydog
```

Sample Output 2:

```
ujhuzoj
mfxwgzt
rdsgbhn
sdqhnfg
qkoscsp
```

Explanation:

In the first sample input, the character ‘a’ at (2,2) will have an offset of 4, and thus ‘a’ is shifted by 4 letters to become ‘e’.