Rail fence cipher is one of the fundamental classical transposition ciphers. Due to the limitations of the key, i.e., the number of rails, it is generally considered weak and can be easily broken with a brute-force attack.

In an attempt to improve the strength of this algorithm, we do the following:

1. Perform the rail fence cipher, but instead of letters arranged in the zig-zag fashion, we put words themselves in the fence. To make this concrete, we perform it N times. The number of levels used in this process is denoted by n.
2. After this, we replace all the spaces in the enciphered text with a small string X
3. From here, we perform the classical rail fence cipher M times on this to get the final ciphered text. The number of levels used here is m.

Given the values of N, n, M, m, the string X, and a cipher text, decrypt and return the plain text.

Input Format

The input consist of 6 lines in the following format

N – integer, the number of times the *word railfence* is applied

n – integer, number of levels for *word railfence* cipher

M – integer, the number of times *character level railfence* cipher is applied

n – integer, the number of levels in *character level railfence* cipher

X – string, to replace the spaces in the text

Cipher text – string

Constraints

1. Input string will consist of only A-Z characters i.e.., english alphabets in upper case only.
2. You are not allowed to use libraries like cryptography in python.

Output Format

Plaintext - string

Sample Input 0

2

3

1

6

XXX

DXHOXXXETXETXROXMOXENXODYXXCAX

Sample Output 0

DO NOT COME HERE TODAY

Explanation 0

Please refer to the below pdf for a detailed explanation of this example: Railfence Cipher on different levels