

X Alternative Encryption

Time limit: 1s

In the fight against theft of intellectual property by rivalling universities, TU Delft has decided to implement a system of secure communication, to be called New Well-Encrypted Remote Communication. All internal communication will be encrypted before being sent over the network, and then decrypted upon arrival. They have already set up the infrastructure, they have come to you for the encryption.

You are tasked to design an algorithm for both encryption and decryption of text consisting of only English lowercase letters. You do not need to tell them how you do it, in fact, they encourage you to keep it a secret, to improve security. Your algorithm only needs to meet the following criteria:



The enigma, an encryption machine that solves this problem. CC BY 2.0 by William Warby on Wikimedia Commons

- No matter what the text is, encrypting it and then decrypting the result should obviously yield the original text.
- To make sure there are no issues with the transmission, the encrypted text should also consist of only English lowercase letters and should have the same number of letters as the original text.
- To make sure the code cannot be broken easily, for all i , the i th letter of the encrypted text should differ from the i th letter of the original text.

The word “nwer`c`jury” for example may not be encrypted as “irritating”, because the number of letters does not match. Nor may you encrypt it as “imbecilic”, since the fifth letter is a ‘c’ in both. An example of an acceptable encryption is “fantastic” (both have an ‘n’ and a ‘c’, but in different positions).

Your program will be run twice for each test case. In the first pass, your program will be given a number of strings to encrypt. In the second pass, your program will be given the strings as encrypted by the first pass, which it should then decrypt to retrieve the original input. Your submission may take up to 1 second for each pass.

A testing tool is provided to help you develop your solution.

Input

The input consists of:

- One line with either “encrypt” or “decrypt”, indicating the action your program has to perform.
- One line with an integer n ($1 \leq n \leq 1000$), the number of strings.
- n lines, each with a string s ($1 \leq |s| \leq 100$), the text to encrypt or decrypt.
All input strings consist of only English lowercase letters (a–z).

Output

For each string, output its encryption or decryption, as required.

Sample Case 1

Input	Pass 1	Output
encrypt 3 plaintext nwer correct	encrypted delft balloon	
Input	Pass 2	Output
decrypt 3 encrypted delft balloon	plaintext nwer correct	