Problem 156: Playfair Cipher

Difficulty: Hard

Author: Brett Reynolds, Annapolis Junction, Maryland, United States

Originally Published: Code Quest 2021

Problem Background

Substitution ciphers are methods of encrypting text that involve replacing each letter in the "plaintext" with another letter to generate a "ciphertext." While quick and easy to implement and remember, they are inherently weak ciphers. Certain letters appear more frequently than others, allowing a codebreaker to make educated guesses about what certain letters represent. By the mid-19th century, such methods were well known to government forces, and intercepted messages could be broken with relative ease.

In 1854, English inventor Sir Charles Wheatstone created a cipher intended to solve this problem. His cipher - named the Playfair cipher, after its supporter, Lord Playfair - replaced letters in pairs rather than each letter individually. Since the English language contains 26 letters, encrypting letters in pairs meant there were potentially over 600 possible pairings, making frequency analysis impractical. The Playfair cipher was also easy to use, and so it remained in use by military forces through World War II and the advent of the computer.

Problem Description

The Playfair cipher uses a word or short phrase as a keyword, which is used to build an encryption table. The encryption table is a five-by-five square filled with the letters of the English alphabet (for the purposes of this problem, the letter 'J' will be omitted; we'll cover how to handle it later).

For example, if the keyword is "PLAYFAIR DEMO", we would start by removing spaces and repeat letters from the keyword. In this case, the letter A appears twice, so its second occurrence is removed: "PLAYFIRDEMO." We then fill these letters into the five-by-five grid, starting in the top left corner and moving from left to right, top to bottom:

P	L	A	Y	F
I	R	D	E	M
O				

The remaining letters of the alphabet (again, except J) are then filled into the grid in order. Any letter already present in the table (in the keyword) is skipped.

P	L	A	Y	F
I	R	D	E	M
O	B	C	G	H
K	N	Q	S	T
U	V	W	X	Z

This completes the encryption table, and it can now be used to encrypt (or decrypt) a message. To begin with, the letters in the original message are broken into pairs. Since we've omitted the letter 'J' from the encryption table, we'll also replace any instance of the letter 'j' with the letter 'x' during encryption. If we encrypt the phrase "code quest," this results in:

co de qu es tx

Notice that since "code quest" contains an odd number of letters, we've added an 'x' at the end to complete the final pair. Each pair of letters is then encrypted according to the following rules:

1. If both letters are the same (e.g. "ss"), replace the second letter with an 'x' ("sx") and continue encryption with the following rules.
2. If the letters appear in the same row of the encryption table, replace each letter with the one to its immediate right in the table (wrapping around to the left side as needed).
3. If the letters appear in the same column of the encryption table, replace each letter with the one immediately below it in the table (wrapping around to the top side as needed).
4. If the letters share neither a row nor column, replace each letter with the one on the same row as itself, but in the same column as its partner.

To demonstrate with our earlier example (keyword: "PLAYFAIR DEMO", plaintext "code quest"), these rules would be followed as shown:

<table>
<tr><th>Original Pair</th><th>Rule Applied</th><th colspan="5">Encryption Table</th><th>Result Pair</th></tr>
<tr><td rowspan="5">co</td><td rowspan="5">2 - Same row;
replace each letter with
the one to its right</td><td>P</td><td>L</td><td>A</td><td>Y</td><td>F</td><td rowspan="5">GB</td></tr>
<tr><td>I</td><td>R</td><td>D</td><td>E</td><td>M</td></tr>
<tr><td>O</td><td>B</td><td>C</td><td>G</td><td>H</td></tr>
<tr><td>K</td><td>N</td><td>Q</td><td>S</td><td>T</td></tr>
<tr><td>U</td><td>V</td><td>W</td><td>X</td><td>Z</td></tr>
<tr><td rowspan="4">de</td><td rowspan="4">2 - Same row;
replace each letter with
the one to its right</td><td>P</td><td>L</td><td>A</td><td>Y</td><td>F</td><td rowspan="4">EM</td></tr>
<tr><td>I</td><td>R</td><td>D</td><td>E</td><td>M</td></tr>
<tr><td>O</td><td>B</td><td>C</td><td>G</td><td>H</td></tr>
<tr><td>K</td><td>N</td><td>Q</td><td>S</td><td>T</td></tr>
</table>

Original Pair	Rule Applied	Encryption Table					Result Pair
		U	V	W	X	Z	
qu	4 - Different row/column; replace each letter with the one in its row and its partner's column	P	L	A	Y	F	KW
		I	R	D	E	M	
		O	B	C	G	H	
		K	N	Q	S	T	
		U	V	W	X	Z	
es	3 - Same column; replace each letter with the one below it	P	L	A	Y	F	GX
		I	R	D	E	M	
		O	B	C	G	H	
		K	N	Q	S	T	
		U	V	W	X	Z	
tx	4 - Different row/column; replace each letter with the one in its row and its partner's column	P	L	A	Y	F	SZ
		I	R	D	E	M	
		O	B	C	G	H	
		K	N	Q	S	T	
		U	V	W	X	Z	

The resulting ciphertext, then, is "GB EM KW GX SZ." Note that the letter appears twice in our ciphertext, in place of the letters 'c' and 'e' - this is part of what foils attempts at frequency analysis. Depending on how the letters get paired up, a single letter in the ciphertext could represent any letter in the plaintext.

For this problem, you must write a program that decrypts the Playfair cipher, given the encrypted message and the keyword used to encrypt it. Decryption works in a similar manner to encryption; just make sure to move to the left or up when letters appear in the same row or column, respectively. Do not worry about replacing any 'x' letters that appear in the plaintext with 'j's or spaces; leave them as they appear.

Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include the following lines:

- A line containing the following information, separated by spaces:
  - A positive integer, X, representing the number of lines in the ciphertext
  - The keyword used to encrypt a message using the Playfair cipher, which will contain only uppercase letters.
- X lines containing the ciphertext to decrypt, consisting entirely of uppercase letters.

```
2
1 PLAYFAIRDEMO
GBEMKWGXSZ
2 LOCKHEEDMARTIN
KRLTUZBIDIBK
PLDIHGKH
```

Sample Output

For each test case, your program must print the plaintext obtained after decrypting the message, in lowercase letters.

```
codequestx
havefuntoday
goodluck
```