

Problem C: Playing Fair with Cryptography

Encryption is the process of taking *plaintext* and producing the corresponding *ciphertext*. One method to do this is the Playfair cipher. This cipher was invented by (who else) Charles Wheatstone in the 1850's (but got its name from one of its most ardent promoters, the Scottish scientist Lyon Playfair). Unlike many substitution ciphers, the Playfair cipher does not encrypt single letters at a time, but groups of two letters called *digraphs*. The encryption process uses a 5×5 grid generated by a secret key known only to the two parties using the cipher (hopefully). The grid is generated as follows: You write the letters of the key, one letter per square row-wise in the grid starting at the top. Any repeated letters in the key are skipped. After this is done, the remaining unused letters in the alphabet are placed in order in the remaining squares, with I and J sharing the same square. For example, if the key was "ECNA PROGRAMMING CONTEST", the generated square would look like the following:

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

You encrypt a digraph using the following rules:

1. If both letters are in the same row, replace each with the letter to its immediate right (wrapping around at the end). For example, using the above grid the plaintext digraph DS is encrypted as FB, and AP is encrypted as PE.
2. If both letters are in the same column, replace each with the letter immediately below it (again wrapping around at the bottom). For example, PF is encrypted as IU (or JU), and WO is encrypted as CS.
3. Otherwise, "slide" the first character across its row until it is in the same column as the second character, and do the same for the second character. The two characters you end up on are the encrypted characters of the digraph. If you view the original digraph as corners of a rectangle in the grid, you are replacing them with the characters in the other two corners. For example, TU is encrypted as FH, UT is encrypted as HF, and ZC is encrypted as WP.

What happens when the digraph contains the same letter twice? In the original Playfair cipher you insert the letter 'X' between the two letters and continue encrypting. You also use an extra 'X' at the end of the plaintext if you have an odd number of letters. Thus the plaintext "OOPS" would first be changed to the digraphs "OX", "OP" and "SX" and then would be encrypted as "GWICBW" (using the grid above). Note that the plaintext "POOS" would not need any added letters since the two 'O's do not appear in the same digraph.

We'll modify the Playfair cipher in one simple way: Instead of always using 'X' as the inserted letter, use 'A' the first time an insertion is needed, then 'B' the next time, and so on (though you should never use 'J' as an inserted letter; just go from 'I' to 'K'). Once you hit 'Z', you go back to using 'A'. If the inserted letter would result in a digraph with the same two letters, you just skip to the next inserted letter. For example, "OOPS" would now become the digraph stream "OA", "OP", "SB" before being encrypted, and the plaintext "AABCC" would become the digraph stream "AB", "AB", "CD", "CE".

Given a key and plaintext, you are to generate the corresponding ciphertext.

Input

The input file will start with an integer n indicating the number of problem instances. Each instance consists of two lines, the first containing the key and the second the plaintext to encrypt. Both of these may contain upper- and lower-case letters, as well as spaces and non-alphabetic characters (both of which should be ignored). For simplicity, the letters 'j' and 'J' will never appear in any key or plaintext.

Output

For each problem instance, output the case number followed by the corresponding ciphertext using upper-case letters with no space. Always use the letter 'I' instead of 'J' in the ciphertext.

Sample Input

```
1
ECNA Programming Contest 2013
This is the easy problem!
```

Sample Output

```
Case 1: HVOFOFHVCPDWEIGSHNGD
```