CCC '04 J4 - Simple Encryption

Time Limit: 2.0s **Memory Limit:** 64M

Canadian Computing Competition: 2004 Stage 1, Junior #4

One of the simplest ways of coding a message is to do a letter shift.

For example, if you shift the letters in the original message by 5 then `A` in your original message becomes `F` in the coded message. (`B` → `G`, `C` → `H`, …, `T` → `Y`, `U` → `Z`, `V` → `A`, …, `Z` → `E`). To decode the message, you simply need to shift back by the same number.

A slightly trickier encryption uses a keyword to determine the amount of the shift. Suppose you were using a keyword `ACT`. To encode the message, you take the original message, remove everything but the alphabetic characters, and form the message into a block that has the same width as the keyword. Here is a sample message to encrypt:

`BANANA & PEEL`

The blocked version of the message is shown below with the keyword `ACT` as a header.

A	C	T
B	A	N
A	N	A
P	E	E
L		

Now, the message is encoded using a letter shift. However, this time it is not a uniform shift, it will depend upon the keyword letter at the top of the column. If the letter at the top of the column is an `A`, then the letters in that column are not shifted. If the letter is a `B`, then the letter in that column shift by 1, and so on. In the example, the letters in the third column will shift by 19 since the `T` is the 20th letter of the alphabet.

The encoded message is:

A	C	T
B	C	G
A	P	T
P	G	X

L		

You will write a program that will accept a keyword and a string to be encoded. The keyword will never have more than 6 characters. The message will always be given in all upper case characters.

Input Specification

The first line of input consists of the keyword. The second line of the input is the message to be encoded. The keyword length will never exceed 6 characters. The total message length also will never exceed 60 characters.

Output Specification

Output the encoded message on a single line.

Sample Input 1

```
ACT
BANANA & PEEL
```

Sample Output 1

```
BCGAPTPGXL
```

Sample Input 2

```
TRICKY
I LOVE PROGRAMMING!
```

Sample Output 2

```
BCWXONKFOTKKFZVI
```

CCC problem statements in large part from the PEG OJ