ACM Intercollegiate Programming Contest
Pacific NW Region
1999

Problem F
Square Cipher

In Dorothy Sayers' novel "*Have His Carcass*", Lord Peter Wimsey describes a cryptography technique that is simple for encoding and decoding, yet relatively hard to crack. Your job is to implement this technique.

Here's how it works, in Sayers' (edited) words:

You choose a key−word of six letters or more, none of which recurs. Such as, for example, SQUANDER. Then you make a diagram of five squares each way and write the key−word in the squares like this:

S	Q	U	A	N
D	E	R		

Then you fill up the remaining spaces with the rest of the alphabet in order, leaving out the ones you've already got.

You can't put twenty−six letters in twenty−five spaces, so you pretend you're an ancient Roman or a medieval monk and treat I and J as one letter. So you get this:

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

Now let's take a message −− What shall we say? 'All is known, fly at once' −− that classic hardy perennial. We write it down all of a piece and break it into groups of two letters, reading from left to right. It won't do to have two of the same letters coming together in a pair, so where that happens, we shove in a Q, which won't confuse the reader. Now our message runs:

```
AL QL IS KN OW NF LY AT ON CE
```

If there is an odd letter at the end, we'd add on another Q to square it up. Now we take our first group, AL. We see that they come at the corners of a rectangle in which the other corners are SP. So we put down SP for the first two letters of the coded message. In the same way, QL becomes SM and IS becomes FA.

Ah, but here's KN. They both come on the same vertical line. In this case you take the letter next below each −− TC. Next comes OW, which translates to MX. Going on, SK, PV, NP, TU... If your first diagonal went from bottom to top, you must take it the same way again. ON = TU, NO would be UT.

CE come on the same horizontal line. In that case, you take the letter to the right of each. Since there isn't a letter to the right of C, you start again at the beginning of the line, producing DR.. Your coded message stands now:

```
SP SM FA TC MX SK PV NP TU DR
```

To make it look pretty and not give the method away, you can break it up into any lengths you like, or you can embellish it with punctuation as haphazard:

```
S.P. SMFA. TCMXS, KPVM, PT! UDR.
```

It's very ingenious. You can't guess it by way of the most frequent letter, because you get a different letter for each time, according as it's grouped to the next letter. And you can't guess individual words, because you don't know where the words begin and end. Is it at all possible to decode it without the key−word?

"Oh dear, yes," said Wimsey. "Any code ever coded can be decoded with pains and patience..."

Input
The input for your program will be a series of keywords and messages to encode, alternating line−by−line, until the end−of−file flag of 999. Using the above technique, you are to encode the message, using the word. For example:

```
SQUANDER
ALL IS KNOWN FLY AT ONCE
JUXTAPOSITION
THE ROOSTER CROWED AT MIDNIGHT
999
```

Input will be in upper case and will contain no punctuation. Unlike Sayers' example, letters may occur more than once in the key−words, in which case you must ignore all but the first occurrence of the letter.

Output
Print each message, encrypted, using two letter groups and no punctuation, separated by a single white space. Separate each message by one blank line.
For the sample input file, the first line of output will be:

```
SP SM FA TC MX SK PV NP TU DR
```

Print an I for IJ.

Notes:

If the last letter is both odd and repeated, treat it as repeated and put the 'Q' before it, not afterward. That is, ALL becomes ALQL for encoding purposes.

In the unlikely case of two Qs in a row, insert a Z between them. Also, augment odd−length messages ending with a Q by using a Z. Thus, FAQQAD becomes FAQZQADQ and HUQ becomes HUQZ for encoding.