For many centuries, most — if not all — known methods for enciphering messages considered a single cipher alphabet. For example, Julius Caesar used a substitution cipher that replaced each letter in the plaintext message by a letter three places further down in the alphabet (‘`A`’ would be ‘`D`’ and so on).

Sometime in the 16th century, a French diplomat called Blaise de Vigenère perfected a new cipher based on previous work by Leon Alberti, Johannes Trithemius and Giovanni Porta. The most significant breakthrough was the use of *multiple cipher alphabets*, which turned the cipher impregnable to frequency analysis (a technique for deducing the plaintext from a ciphertext).

How to encipher a plaintext message

The first step in encipherment is to draw a Vigenère square:

```
                  P L A I N T E X T   L E T T E R S

          | a b c d e f g h i j k l m n o p q r s t u v w x y z
----------+----------------------------------------------------
   1  (A) | B C D E F G H I J K L M N O P Q R S T U V W X Y Z A
   2  (B) | C D E F G H I J K L M N O P Q R S T U V W X Y Z A B
   3  (C) | D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
   4  (D) | E F G H I J K L M N O P Q R S T U V W X Y Z A B C D
   5  (E) | F G H I J K L M N O P Q R S T U V W X Y Z A B C D E
C  6  (F) | G H I J K L M N O P Q R S T U V W X Y Z A B C D E F
I  7  (G) | H I J K L M N O P Q R S T U V W X Y Z A B C D E F G
P  8  (H) | I J K L M N O P Q R S T U V W X Y Z A B C D E F G H
H  9  (I) | J K L M N O P Q R S T U V W X Y Z A B C D E F G H I
   -------+----------------------------------------------------
E  10 (J) | K L M N O P Q R S T U V W X Y Z A B C D E F G H I J
R  11 (K) | L M N O P Q R S T U V W X Y Z A B C D E F G H I J K
   12 (L) | M N O P Q R S T U V W X Y Z A B C D E F G H I J K L
A  13 (M) | N O P Q R S T U V W X Y Z A B C D E F G H I J K L M
   -------+----------------------------------------------------
L  14 (N) | O P Q R S T U V W X Y Z A B C D E F G H I J K L M N
P  15 (O) | P Q R S T U V W X Y Z A B C D E F G H I J K L M N O
H  16 (P) | Q R S T U V W X Y Z A B C D E F G H I J K L M N O P
   -------+----------------------------------------------------
A  17 (Q) | R S T U V W X Y Z A B C D E F G H I J K L M N O P Q
B  18 (R) | S T U V W X Y Z A B C D E F G H I J K L M N O P Q R
E  19 (S) | T U V W X Y Z A B C D E F G H I J K L M N O P Q R S
T  20 (T) | U V W X Y Z A B C D E F G H I J K L M N O P Q R S T
S  21 (U) | V W X Y Z A B C D E F G H I J K L M N O P Q R S T U
   -------+----------------------------------------------------
   22 (V) | W X Y Z A B C D E F G H I J K L M N O P Q R S T U V
   23 (W) | X Y Z A B C D E F G H I J K L M N O P Q R S T U V W
   24 (X) | Y Z A B C D E F G H I J K L M N O P Q R S T U V W X
   25 (Y) | Z A B C D E F G H I J K L M N O P Q R S T U V W X Y
   26 (Z) | A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

Then, the multiple cipher alphabets (rows in the Vigenère square) to be used in the encipherment must be chosen. A practical way of doing this is via a keyword. For example, the keyword ‘MIUP’ selects the cipher alphabets 13, 9, 21 and 16 (see underlined rows in the Vigenère square).

In the final step, each letter in the plaintext message is enciphered according to the corresponding cipher alphabet. For example, suppose the plaintext message is ‘programming’ and the keyword is ‘MIUP.’ The encoded message would be:

```
plaintext message -  p  r  o  g   r  a  m  m   i  n  g
keyword           -  M  I  U  P   M  I  U  P   M  I  U
cipher alphabets  -  13 9  21 16  13 9  21 16  13 9  21
----------------- -  ----------------------------------
encoded message   -  C  A  J  W   E  J  H  C   V  W  B
```

The letter ‘`p`’ in the plaintext message is enciphered using cipher alphabet 13 into the letter ‘`C`’. The ‘`C`’ comes from the intersection of column ‘`p`’ with row 13 in the Vigenère square. The second letter in the plaintext message, ‘`r`’, is enciphered into the letter ‘`A`’ using cipher alphabet 9 (check the intersection of column ‘`r`’ with row 9), and so on. If necessary, as in this case, the keyword can be repeated multiple times to match the length of the plaintext message.

How to decipher an encoded message

The decipherment of an encoded message requires the keyword that was used to encipher the plaintext message. For example, suppose the encoded message is ‘CAJWEJHCVWB’ and the keyword is ’MIUP’.

```
encoded message   -  C  A  J  W   E  J  H  C   V  W  B
keyword           -  M  I  U  P   M  I  U  P   M  I  U
cipher alphabets  -  13 9  21 16  13 9  21 16  13 9  21
----------------- -  ----------------------------------
plaintext message -  p  r  o  g   r  a  m  m   i  n  g
```

The multiple cipher alphabets of ‘MIUP’ are, again, 13, 9, 21 and 16 (grey rows in the Vigenère square above). Each plaintext message letter is obtained by finding the corresponding encoded letter in the row of the current cipher alphabet and then checking the letter on top of the column. For example, the letter ‘`p`’ in the plaintext message is on top of the column where ‘`C`’ (the encoded letter) appears in row (cipher alphabet) 13. The decipherment ends when the last encoded letter is deciphered.

(This background was written based on Simon Singh’s “The Code Book,” p. 45-51.)

Input

The input will consist of multiple lines.

Each line contains a message enciphered with the Vigenère cipher (in uppercase). The only knowledge about each message is that *after being deciphered* it represents a 3 digit number written as text (for example: ‘`oneoneone`’ or ‘`zerozeroseven`’).

Output

Separate the output of each input line with a single blank line.

For each line, the output is all possible decipherments of the encoded message. For example, the encoded message ‘`PQRPQRPQR`’ means ‘`oneoneone`’ when the keyword is ‘ACM’, but also means ‘`oneonetwo`’ or ‘`oneonesix`’ when the keywords are, respectively, ‘`ACMACMVTC`’ and ‘`ACMACMWHT`’, and so on.

Each decipherment is written in a single line, as shown in the sample output. The text to the left of the ‘`->`’ symbol is the keyword (in uppercase) and to the right is the corresponding plaintext 3 digit number (in lowercase). There should be *no empty line* between decipherments of the same input message.

Note: the output must be sorted in descending order by value of the plaintext number (‘nineninenine’ ¿ ‘nineninEeight’ ¿ ... ¿ ‘zerozeroone’ ¿ ‘zerozerozero’).

Sample Input

```
PQRPQRPQR
PQRPQRPQR
```

Sample Output

```
WHTWHTWHT -> sixsixsix
WHTWHTVTC -> sixsixtwo
WHTWHTACM -> sixsixone
WHTVTCWHT -> sixtwosix
WHTVTCVTC -> sixtwotwo
WHTVTCACM -> sixtwoone
WHTACMWHT -> sixonesix
WHTACMVTC -> sixonetwo
WHTACMACM -> sixoneone
VTCWHTWHT -> twosixsix
VTCWHTVTC -> twosixtwo
VTCWHTACM -> twosixone
VTCVTCWHT -> twotwosix
VTCVTCVTC -> twotwotwo
VTCVTCACM -> twotwoone
VTCACMWHT -> twoonesix
VTCACMVTC -> twoonetwo
VTCACMACM -> twooneone
ACMWHTWHT -> onesixsix
ACMWHTVTC -> onesixtwo
ACMWHTACM -> onesixone
ACMVTCWHT -> onetwosix
ACMVTCVTC -> onetwotwo
ACMVTCACM -> onetwoone
ACMACMWHT -> oneonesix
ACMACMVTC -> oneonetwo
ACMACMACM -> oneoneone

WHTWHTWHT -> sixsixsix
WHTWHTVTC -> sixsixtwo
WHTWHTACM -> sixsixone
WHTVTCWHT -> sixtwosix
WHTVTCVTC -> sixtwotwo
WHTVTCACM -> sixtwoone
WHTACMWHT -> sixonesix
WHTACMVTC -> sixonetwo
WHTACMACM -> sixoneone
VTCWHTWHT -> twosixsix
VTCWHTVTC -> twosixtwo
VTCWHTACM -> twosixone
VTCVTCWHT -> twotwosix
VTCVTCVTC -> twotwotwo
VTCVTCACM -> twotwoone
VTCACMWHT -> twoonesix
VTCACMVTC -> twoonetwo
VTCACMACM -> twooneone
ACMWHTWHT -> onesixsix
ACMWHTVTC -> onesixtwo
ACMWHTACM -> onesixone
ACMVTCWHT -> onetwosix
ACMVTCVTC -> onetwotwo
ACMVTCACM -> onetwoone
ACMACMWHT -> oneonesix
ACMACMVTC -> oneonetwo
ACMACMACM -> oneoneone
```