2754 Delta Encoding and Decoding

A very simple encryption scheme which represents a big improvement over a substitution cipher is a *delta cipher*. Here's how it works: The letters 'A' through 'Z' are given a numerical values starting with 0 and ending with 25. These values aren't necessarily given in that order, but the mapping is one to one. (The trivial mapping where 'A' has value 0, 'B' has value 1, etc. is the default cipher. It's what you use when you don't have anything better.)

In order to encrypt a message, the *differences* between successive values are used.

1. All characters except letters are transferred to the encrypted message unchanged.
2. When one letter is preceeded by something other than a letter, it is treated as though it had been preceeded by an A.
3. Case both does and does not matter in this encryption scheme. If a letter is originally uppercase, the encrypted version should be uppercase. If it is lowercase, the encrypted version should be lowercase. However, 'A' and 'a' are the same letter.
4. When one letter is preceeded by another, the difference between the two values is the value of the first is subtracted from the second. The letter whose value is equivalent to this difference modulo 26 will replace the second letter in the new text.

Input

The input is merely a sequence of a commands, one per line, and is terminated by an EOF. No line contains more than 10000 characters. There are three command words: 'ENCRYPT', 'DECRYPT', and 'CIPHER'. The case of the letters in these words is not important, and each word will be followed by at least one space. Any line which begins with any word other than one of these three commands requires the response 'Command not understood.'

The 'ENCRYPT' and 'DECRYPT' commands are self-explanatory. The encrypt command is followed by plain text which needs to be encrypted and the decrypt command is followed by encrypted text which must be put into plain text. Whatever follows one of these two command words should be either encrypted or decrypted using the current cipher. If no cipher has yet been given, you should use the default cipher.

The 'CIPHER' command should be followed by the twenty-six letters of the alphabet. The order of these letters is important and that order determines the value for each character — the first letter has value 0; the second letter has value 1; etc. There may be spaces and punctuation; these should be ignored; however, this line must have exactly twenty-six letters on it with no repetitions.

Output

Any problem requires the response, 'Bad cipher. Using default.' (And, as indicated, you should actually use the default). If the cipher is understandable, you need to echo it back to the user as shown below, and then use it for all subsequent commands.

Sample Input

```
ENCRYPT The quick brown fox leaped over the lazy dog.
CIPHER qwertyuiopasdfghjklzxcvbnm
```

```
ENCRYPT The quick brown fox leaped over the lazy dog.
DECRYPT Xsf juwgv fuyzb rcd oaomza nguw xsf olpd evu.
ENCRIPT This won't work!
CIPHER qAZwSXedCRfv  tgbyhnujmikolp
ENCRYPT The quick brown fox leaped over the lazy dog.
DECRYPT Vsh puwgb gibeb rgc ocilda kbiw vsh owag ehh.
CIPHER qwertyuopasdfghjklzxcvbnm
ENCRYPT The quick brown fox leaped over the lazy dog.
DECRYPT Tox qeoui bqxir fjj ltwppz ohjn tox lpzz dls.
```

Sample Output

```
RESULT: Tox qeoui bqxir fjj ltwppz ohjn tox lpzz dls.
Good cipher. Using QWERTYUIOPASDFGHJKLZXCVBNM.
RESULT: Xsf juwgv fuyzb rcd oaomza nguw xsf olpd evu.
RESULT: The quick brown fox leaped over the lazy dog.
Command not understood.
Good cipher. Using QAZWSXEDCRFVTGBYHNUJMIKOLP.
RESULT: Vsh puwgb gibeb rgc ocilda kbiw vsh owag ehh.
RESULT: The quick brown fox leaped over the lazy dog.
Bad cipher. Using default.
RESULT: Tox qeoui bqxir fjj ltwppz ohjn tox lpzz dls.
RESULT: The quick brown fox leaped over the lazy dog.
```