# Problem 6
# Playfair's Cipher

You will consider a Ciphering code that was clever for its time, but is considered easy to crack with modern techniques using a computer. The method is known as Playfair's Cipher (after Baron Lyon Playfair, British scientist, 1818-1898). This method requires that the number of characters in the "alphabet" be a perfect square. You will use an 81 (9 x 9) character set for this problem; *in order,* you will use:

- the 26 letters of the English alphabet in upper case then in lower case
- the ten digits (starting from zero)
- the various keyboard characters:

| | |
|---|---|
| | space |
| ‘ | apostrophe |
| ( | left parenthesis |
| ) | right parenthesis |
| * | asterisk |
| + | plus sign |
| , | comma |
| - | hyphen |
| . | period |
| / | slash |
| : | colon |
| ; | semicolon |
| < | less than sign |
| = | equal sign |
| > | greater than sign |
| [ | left square bracket |
| ] | right square bracket |
| ^ | caret |
| ? | question mark |

Note that this ordering does not conform to the order of the characters in the ASCII character set. All other characters (tab, left curly brace {, the non-printable ASCII characters, etc.) will be regarded as errors in the input. Because of the way we process the input lines, we need not consider the carriage return in the input text stream.

The information needed (the input to your program) to encode or decode a message is:

1. whether to encode or decode,
2. a secret code word, and
3. the message to encode or decode.

The secret code word is any string of characters in the" alphabet." As an example (excluding the leading spaces, and including the internal spaces and the period):

**Cryptography Is Fun.**

The secret code word is used to make up a square table as follows: The "letters" of the code word are placed in a square table of the correct size in a row-by-row manner. When a letter is repeated, it is ignored in the construction of the table. After the secret code word is used up, all the letters in the alphabet that were not used are placed in order in the remaining spaces of the table. For the code word given, the table produced is given in the following table:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C | r | y | p | T | o | g | a | h |
| | I | s | F | U | n | . | A | B |
| D | E | G | H | J | K | L | M | N |
| O | P | Q | R | S | T | U | V | W |
| X | Y | Z | b | C | d | e | f | i |
| j | k | l | m | Q | v | w | x | z |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | ' | ( | ) | * | + | , | - | / |
| : | ; | < | = | > | [ | ] | ^ | ? |

Playfair's Cipher encodes messages by taking letters of the message two at a time (I1 and I2) and outputting two characters (01 and 02) to take their place. There are four rules governing this translation process:

1. If I1 and I2 are found at two corners of a rectangle, for example ‘f’ and‘s’, then 01 and 02 are the other two corners, Z and A.
2. If the two input characters, I1 and I2, are in the same row, the two output characters are the characters to their immediate right. If the input character is on the right edge of a row, the character on the same row on the left edge is used.
3. If the two input characters are in the same column, then the two output characters are the characters directly below the input characters, where the top row is considered to be below the bottom row if the input character is on the bottom row.
4. Two consecutive identical characters (i.e., their positions are odd/even) are not allowed.

As an example, the message (with no leading or trailing blanks):

**Computers Are Fun?**

becomes encoded, using our previous secret code word, as:

**rg3FJuYgFIIaX.unB[**

Decoding is just the reverse of encoding. One problem is that a message might have an odd number of characters. We will solve this problem by having the program insert an extra blank character at the end of a line with an odd number of characters for which the last character is not a blank (to conform with rule 4). If the line has an odd number of characters and the last character is a blank, then this last blank character should be ignored.

## Input

Three arguments will be given to your program. They will be: [encode/decode], [code word], and [message].

### Output

Output should be the encoded/decoded message.

### Sample input

```
encode “Cryptography Is Fun.” “Computers Are Fun?”
```

### Sample output

```
rg3FJuYgFIIaX.unB[
```