Assignment 1
Text Encoder

Text encoder is used to encrypt data. In this project, you will write a program to encode strings of text according to the algorithm outlined below. The skeleton for the program is provided. Your assignment is to complete functions code() and sum(), which are called by the main() program. Do not make any changes to the main() or to function declarations. All variables you declare must be local.

Input: A key character followed by the characters of a string to be encoded. The first character of the string is the key character.

Output: The key character, the characters of the encoded message, and a four-decimal-digit checksum that is the sum modulo 10,000 of the output encoded characters. (The key character is not included in the checksum calculation). If the input string is empty, output is an empty string.

Algorithm: (Used separately on each character)

1. Get a character.

2. If the character code is zero or greater than 126, terminate encoding, generate the checksum, and append it to the output.

3. If the character code is in the range 1 through 32, output a caret (^) followed by a second character. Initially, the code for the second character is the code of the input character plus the code of the key character. If the resulting code value is greater than 126, then 32 is added modulo 126 to the initial value of the code for the second character.

4. If the code is anything else, generate the output character(s) as follows:

    1. Subtract 33 from the character.
    2. Add the code of the key character, modulo 94.
    3. Reflect that value in the range 0 through 93.
    4. Add 33.
    5. Use the result as the output character code.
    6. If the resulting code is the code for the caret character (^), also output the key character after the generated character.

Examples:

Input: #A
Output:#;0059

Input: $3 x
Output:$H^Da0331

Input: ZThis is a test.
Output:ZO;:0^z:0^zB^z/>0/u1385

Input: qWhat grade will I get?
Output:q5$+v^3%x+('^3s#~~^3C^3%'vM1849

The following input and output are each on a single line:

```
Input: 3!"#$%&'()*+,-./0123456789:;<=>?@
       ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
       abcdefghijklmnopqrstuvwxyz{|}~
Output:3KJIHGFEDCBA@?>=<;:9876543210/.-,
        +*)('&%$#"!~}|{zyxwvutsrqponmlkj
        ihgfedcba`_^3]\[ZYXWVUTSRQPONML7524
```

Input: <Return>

Done.