PROGRAMMING CHALLENGE 2: MESSAGE DECODER

A message has been encoded as a text stream that is to be read character by character. The stream contains a series of comma-delimited integers, each a positive number. However, the character represented by a particular integer depends on the current decoding mode. There are 3 modes: ***uppercase***, ***lowercase***, and ***punctuation***.

In ***uppercase*** mode, each integer represents an uppercase letter: The integer modulo 27 indicates the letter of the alphabet (where 1 = A, 2 = B, and so on). So an input value of 143 in uppercase mode would yield the letter H because 143 modulo 27 is 8, and H is the eighth letter in the alphabet.

The ***lowercase*** mode works the same but with lowercase letters; the remainder of dividing the integer by 27 represents the lower letter (1 = a, 2 = b, and so on). So an input value of 56 in lowercase mode would yield the letter b because 56 modulo 27 is 2 and b is the second letter in the alphabet.

In ***punctuation*** mode, the integer is instead considered modulo 9, with the interpretation given by Table 1 below. So 19 would yield an exclamation point because 19 modulo 9 is 1.

Number	1	2	3	4	5	6	7	8
Symbol	!	?	,	.	(space)	;	"	'

Table 1: Punctuation Decoding Mode

At the beginning of each message, the decoding mode is uppercase letters. Each time the modulo operation (by 27 or 9, depending on mode) ***results in zero (0)***, the ***decoding mode switches***. If the current mode is uppercase, the mode switches to lowercase letters. If the current mode is lowercase, the mode switches to punctuation, and if it is punctuation, it switches back to uppercase.

chr
65 → 90 97 → 122
A Z a z

Task 1: Write a function `decoder` that reads a text stream from file "`textstream.txt`", decodes the text stream, and output the decoded text on the screen.

Contents of textstream.txt

```
18,12312,171,763,98423,1208,216,11,500,18,241,0,32,20620,27,10
```

Evidence 1: Your program code for function `decoder`.

Evidence 2: Run your program and produce a screenshot.

Task 2: Write a function `encoder` that

- accepts user input of a sentence, which contains only alphabets and symbols in Table 1;
- where first character is an uppercase character;
- and converts them into a text stream containing a series of comma-delimited positive integers, that can be decoded using the same method describe above.
- Example: An **uppercase** 'A' is encoded to 1, 'B' encoded to 2; a **lowercase** 'a' is also encoded to 1, 'b' also encoded to 2; an '!' is also encoded to 1, '?' encoded to 2.

Evidence 3: Your program code for function `encoder`.

Evidence 4: Produce a screenshot by running your program using the following input: "***What a wonderful day!***".

Task 3: The function `encoder` in task 2 generates a text stream that are first multiples of the positive integers. Amend your code for `encoder` so that it generates other multiples of positive integers randomly. Name the function `encoder_random`.

Evidence 5: Your amended program code for function `encoder_random`. Highlight your amended code in ***bold and italics***.