**IT 102: Python Program Caesar’s Cipher**

Write a program that allows users to encrypt or decrypt messages using Caesar’s Cipher. The program will prompt the user for 1) the mode (encrypt or decrypt), 2) the message, and 3) the key. Sample output is shown here:

>>>

Do you wish to encrypt (e) or decrypt (d) a message?

e

Enter your message:

Meet me at the fountain Friday at 5PM

Enter the key number (1-26):

10

Your translated text is:

Wood wo kd dro pyexdksx Pbsnki kd 5ZW

>>>

Steps:

1. Define a function **getMessage()** that prompts the user for a message, and returns it.  
     
   Test your function before proceeding.
2. Define a function **getMode()** that asks the user whether they want to encrypt or decrypt. Continue prompting the user until they enter a valid value. The function should not be case sensitive. Return the answer.  
     
   Test your function before proceeding.
3. Define a function **getKey()** that prompts the user for a key between 1 and 26. Continue prompting the user until they enter a valid value. Return that value.  
     
   Test your function before proceeding.
4. Define a function **translateMessage()** that accepts three parameters – mode, message, and key. The function should implement the following logic:

If mode is decrypt

Key becomes negative

Initialize a variable *translated* to empty string

For each symbol in the message

If the symbol is a letter

Convert the letter to a number

Add the key to the number

If the symbol is upper case

If the number is greater than uppercase Z

Subtract 26 from the number

If the number is less than uppercase A

Add 26 to the number

Else if the symbol is lower case

If the number is greater than lowercase z

Subtract 26 from the number

If the number is less than lowercase a

Add 26 to the number

Convert the number to a character and append it to translated

Else

Append the symbol to translated

Return translated

1. Write your main program by calling the first three methods, then passing their return values to translateMessage. Print the return value.  
     
   Test your program thoroughly.

CHALLENGE: Allow the user to enter a third mode: brute force (b). Under brute force mode, the user will not be prompted for a key. Instead, the program will print out a decrypted version of the phrase with every key from 1 to 26.

