casesarcipher.py

Sally Poon A01232177

This module will be able to accept a string, accept a boolean to encode or decrypt, and a positive or negative number for the shift of the encode!

Start → accept a message, encode boolean value, and a number for shift → define output_translation= = "" → Should this message be encoded?

Yes → take each character of the message from beginning to end → *encode* by passing the character and the shift number to **CIPHER_CHARACTER** → concatenate result of **CIPHER_CHARACTER** to the end of output_translation → Is this the last character in the string?

No → proceed to the next character in the string

Is this the last character in the string? **Yes** →

No (from "Should this message be encoded?") → take each character of the message from beginning to end → *decrypt* by passing the character and **negative** shift number to **CIPHER_CHARACTER** → concatenate result of **CIPHER_CHARACTER** to the end of output_translation → Is this the last character in the string?

No → proceed to the next character in the string

Is this the last character in the string? **Yes** → RETURN output_translation

CIPHER_CHARACTER(character, shift)

Is the character an uppercase letter?

No → Is the character a lowercase letter?

No → Is the character a number?

No → *character is a whitespace character*

Is the character an uppercase letter? **Yes** → get the location of the uppercase letter and shift it the number of times indicated, loop back to the first or last alphabet when necessary → RETURN the result

Is the character a lowercase letter? **Yes** → get the location of the lowercase letter and shift it the number of times indicated, loop back to the first or last alphabet when necessary → RETURN the result

Is the character a number? **Yes** → get the location of the number and shift it the number of times indicated, loop back to the first or last number when necessary → RETURN the result

character is a whitespace character → RETURN the character