

EJERCICIO PYTHON - CADENAS

CRPTOGRAFÍA

Dentro de la criptografía los profesionales distinguen entre *sistemas de código*, en los que se reemplaza cada palabra por otra o por un símbolo determinado, y *sistemas de cifrado* en los que la transformación se realiza carácter por carácter o bit por bit, sin importar la estructura lingüística del mensaje. La transformación consiste habitualmente en una serie de sustituciones y/o transposiciones.

Los mensajes a encriptar, conocidos como **texto plano**, se cifran mediante un **algoritmo de encriptación** parametrizado por una **clave** para obtener texto cifrado. En el destino, para recuperar el texto plano inicial, se aplica un algoritmo de desencriptado al texto cifrado dependiente a su vez de una clave.

Si las claves utilizadas para encriptar y desencriptar son diferentes se habla de criptografía asimétrica. En caso contrario decimos que la criptografía es simétrica.

Tu tarea es diseñar la función para encriptar un mensaje. Recibirá como parámetros un texto plano y la clave utilizada para cifrar el mensaje y devolverá el texto cifrado correspondiente. Debes diseñar también la función correspondiente de desencriptado, que recibirá el texto encriptado y la clave de desencriptado y devolverá el texto plano inicial. La clave de ambos métodos será la misma. (CLAVE SIMÉTRICA).

Pide al usuario que ingrese la clave a utilizar. La longitud de la clave no debe ser menor que 15 caracteres y debe contener mayúsculas, minúsculas, dígitos y caracteres especiales.

En tu algoritmo, debes realizar no menos de cuatro permutaciones y cuatro sustituciones, para lo que utilizarás métodos de tratamiento de cadenas de caracteres y operaciones con listas. El resultado obtenido debe ser diferente si la clave es diferente.

```
def validate(key)
    # return True si key cumple las condiciones establecidas
    # return False si key no cumple las condiciones

def encryption( msg, key ):
    # msg es el texto plano a encriptar
    # key es la clave a utilizar para encriptar
    # devuelve msg cifrado según key.
    #

def decoded (code, key)
    # code es el texto cifrado a desencriptar
    # key es la clave a utilizar para desencriptar
    # devuelve code decodificado según key.
```