**Ejercicio:** El módulo deberá contener la función main(), tal como está definida en el archivo original (no debe ser modificada) y la definición de la siguiente función, debidamente “documentada” y probada:

- Una función denominada reparte\_txt que reciba por parámetros (QUE NO SE DEBEN VALIDAR) una cadena de caracteres (txt) y una tupla (tup\_bin), integrada por números 0 (cero) y números 1 (uno); la cadena y la tupla tienen igual cantidad de elementos. Debe devolver dos cadenas de caracteres, la primera contiene únicamente los caracteres de txt que se encuentren en las respectivas posiciones de los números 1 (uno) de tup\_bin; la segunda cadena de caracteres, contiene únicamente los caracteres de txt que se encuentren en las respectivas posiciones de los números 0 (cero) de tup\_bin.

Para realizar las pruebas, deberás ejecutar la función main(), porque incluye los resultados esperados para tres casos de prueba.

Considerá los primeros 17 ítems de la Matriz de Autoevaluación para revisar la calidad de tu producción.

~~#definir reparte\_txt con la siguiente documentación (verificar indentación):~~

def reparte\_txt(txt, tupla\_bin):

""" Recibe una cadena de caracteres txt y una tupla tup\_bin de igual longitud.

La tupla está integrada, exclusivamente, por números 0 (cero) y números 1 (uno).

Devuelve dos cadenas de caracteres, la primera contiene únicamente

los caracteres de txt que se encuentren en las respectivas posiciones

de los números 1 (uno) de tup\_bin; la segunda cadena de caracteres,

contiene únicamente los caracteres de txt que se encuentren en las

respectivas posiciones de los números 0 (cero) de tup\_bin

"""

cadena\_1 = cadena\_2 = ''

for indice in range(len(txt)):

if tupla\_bin[indice] == 1:

cadena\_1 = cadena\_1 + txt[indice]

else:

cadena\_2 = cadena\_2 + txt[indice]

return cadena\_1, cadena\_2

def main():

"""Prueba el comportamiento de la función reparte\_txt() """

binario = 1

cade = 'Ve-Murciélago'

tup\_unos = ()

tup\_ceros = ()

tup\_varios = ()

for car in cade:

if binario == 0:

binario = 1

else:

binario = 0

tup\_unos = tup\_unos + (1,)

tup\_ceros = tup\_ceros + (0,)

tup\_varios = tup\_varios + (binario,)

print ('Salidas esperadas de función reparte\_txt():')

print ('Cadena:->', cade,'-Tupla:->', tup\_unos)

print ('debe dar:----------------------->', (cade,''))

print ('Cadena:->', cade,'-Tupla:->', tup\_ceros)

print ('debe dar:----------------------->', ('',cade))

print ('Cadena:->', cade,'-Tupla:->', tup\_varios)

print ('debe dar:----------------------->', ('eMrilg','V-ucéao'))

print ('\nResultados reales de función reparte\_txt():')

print ('Cadena:->', cade,'-Tupla:->', tup\_unos)

print ('da:----------------------------->', reparte\_txt(cade,tup\_unos))

print ('Cadena:->', cade,'-Tupla:->', tup\_ceros)

print ('da:----------------------------->', reparte\_txt(cade,tup\_ceros))

print ('Cadena:->', cade,'-Tupla:->', tup\_varios)

print ('da:----------------------------->', reparte\_txt(cade,tup\_varios))

# help(reparte\_txt)

main()

**Prueba:**

======= RESTART: C:\Users\Daniel…

Salidas esperadas de función reparte\_txt():

Cadena:-> Ve-Murciélago -Tupla:-> (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)

debe dar:-----------------------> ('Ve-Murciélago', '')

Cadena:-> Ve-Murciélago -Tupla:-> (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)

debe dar:-----------------------> ('', 'Ve-Murciélago')

Cadena:-> Ve-Murciélago -Tupla:-> (0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0)

debe dar:-----------------------> ('eMrilg', 'V-ucéao')

Resultados reales de función reparte\_txt():

Cadena:-> Ve-Murciélago -Tupla:-> (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)

da:-----------------------------> ('Ve-Murciélago', '')

Cadena:-> Ve-Murciélago -Tupla:-> (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)

da:-----------------------------> ('', 'Ve-Murciélago')

Cadena:-> Ve-Murciélago -Tupla:-> (0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0)

da:-----------------------------> ('eMrilg', 'V-ucéao')

>>> help(reparte\_txt)

Help on function reparte\_txt in module \_\_main\_\_:

reparte\_txt(txt, tupla\_bin)

Recibe una cadena de caracteres txt y una tupla tup\_bin de igual longitud.

La tupla está integrada, exclusivamente, por números 0 (cero) y números 1 (uno).

Devuelve dos cadenas de caracteres, la primera contiene únicamente

los caracteres de txt que se encuentren en las respectivas posiciones

de los números 1 (uno) de tup\_bin; la segunda cadena de caracteres,

contiene únicamente los caracteres de txt que se encuentren en las

respectivas posiciones de los números 0 (cero) de tup\_bin

>>> car = 'X'

>>> tup = (0,)

>>> print ('Para:', tup,' y:',car,'-->', reparte\_txt(car,tup))

Para: (0,) y: X --> ('', 'X')

>>>

*La función* ***está documentada y******produce resultados esperados****.*

*Es un muy buen trabajo.*

***No se requiere nuevo envío.*** *Por favor, avisanos cuando hayas descargado este documento y conservalo en tu biblioteca de aprendizajes (las devoluciones pueden ser requeridas más adelante).*

*Saludos,*

*Daniel S. R.*