

Taller de Programación

Clase 08: Cifrado

Daniela Opitz
dopitz@udd.cl



Basada en presentaciones oficiales de libro Introduction to Programming in Python (Sedgewick, Wayne, Dondero).

Disponible en <https://introcs.cs.princeton.edu/python>

Cifrado

Cifrado: proceso mediante el cual la información (un texto por ejemplo) es cifrado de forma que este es ilegible a menos que se conozcan los datos necesarios para su lectura.

Ejemplos de información que usualmente que utiliza (o debe utilizar) cifrada (o encriptada):

- contraseñas
- números de tarjetas de crédito
- conversaciones privadas en redes sociales

Métodos Utiles

- `s.isdigit()`: verifica si un string s es un dígito, devuelve **True** si es un dígito y **False** si no lo es.
- `L.append(e)`: agrega un elemento a una lista L.
- `' '.join(L)`: convierte una lista L en un string en donde cada elemento es separado por string entre las comillas, en este caso un espacio. Por ejemplo si `L=['1', '2', '3']`, `' '.join(L)` devuelve `'1 2 3'`
- `s1.index(s2)`: retorna la posición en s1 donde esta el carácter s2. Por ejemplo si `s1='Hola'`, `s1.index('H')` devuelve 0.

Problema 1

Programe un código que encripte una palabra ingresada por teclado e imprima la versión encriptada. El programa debe recibir la palabra a encriptar y retornar la palabra encriptada. El proceso de encriptación que debe implementar se describe a continuación:

1. Existe la palabra mágica “**murcielago**”, que actúa como clave de encriptación.
2. La palabra a encriptar se compara, carácter por carácter con la palabra mágica, y dependiendo del resultado de la comparación, ciertos caracteres de la palabra a encriptar son reemplazados. Para esto existen dos casos:
 - Si el carácter de la palabra a encriptar existe en la palabra mágica, este se reemplazará por la posición en la palabra mágica donde el carácter se encuentra.
Si el carácter buscado no existe en la palabra mágica, el carácter de la palabra a encriptar no se cambiará.
 - El resultado final de los reemplazos de caracteres, dan origen a la palabra encriptada. **Ejemplo:** el programa al recibir la palabra ‘**mundo**’ retorna ‘**01nd9**’ . Esto es porque m está en la posición 0 de la palabra mágica, u está en la posición 1 , n y d no se encuentran, y o está en la posición 9 .

Problema 2

Programa un código que reciba una palabra encriptada ingresada por teclado e imprima la versión desencriptada. El programa debe **recibir** un string encriptado y **retornar** un string desencriptado.

El proceso de desencriptación se describe a continuación:

1. Existe la palabra mágica “**murcielago**”, que actúa como clave de desencriptación.
2. La palabra a desencriptar se recorre carácter por carácter, y si el carácter es un dígito, este se reemplaza por la letra de la clave en la posición que indica el dígito. Pero si el carácter no es un dígito, este se mantiene.
3. El resultado final de los reemplazos de caracteres, dan origen a la palabra desencriptada.

Ejemplo: El programa al recibir '01nd9' debe retornar 'mundo'. Esto es porque m está en la posición 0 de la palabra mágica, u está en la posición 1, n y d no se encuentran, y o está en la posición 9.

.

Problema 3

Para prevenir errores de digitación es frecuente el uso de un dígito verificador. Un caso puntual es el dígito verificador módulo 11, utilizado en la cédula nacional de identidad, el cual se calcula usando los siguientes pasos:

- 1- se multiplican los dígitos del RUT por la serie 2, 3, 4, 5, 6 y 7, y se suman los productos. Esta operación se realiza de derecha a izquierda, y cuando quedan dígitos del rut por multiplicar se utilizan nuevamente los elementos de la serie 2, 3, 4, 5, 6, y 7.
- 2- una vez obtenida la suma total de los productos, esta se divide por 11 y se calcula el resto de esa división.
- 3- finalmente el dígito verificador se obtiene restando a 11 el resto de la división obtenida en el punto anterior.
- 4- cuando el resultado del dígito verificador es igual a 10, al dígito verificador se le asigna la letra K. Y si el dígito verificador es igual a 11 se le asigna el número 0.

Realice un programa en Python que permita validar si el RUT de una persona fue ingresado correctamente. Considere:

- el programa debe recibir el **rut** en formato **string** y devolver **True** o **False** dependiendo si el **rut** es válido o no
- el formato de ingreso del **rut** es sin puntos, pero con guion, por ejemplo: **'12345678-9'**
- su programa debe funcionar con cualquier **rut**, no solo el del ejemplo ¡Pruebe con el suyo!

Problema 3

Por ejemplo, el rut 30686957 posee el dígito verificador 4, esto es porque:

$$7 \times 2 = 14$$

$$5 \times 3 = 15$$

$$9 \times 4 = 36$$

$$6 \times 5 = 30$$

$$8 \times 6 = 48$$

$$6 \times 7 = 42$$

$$0 \times 2 = 0$$

$$3 \times 3 = 9$$

, la suma de los productos $14 + 15 + 36 + 30 + 48 + 42 + 0 + 9 = 194$, el resto de la división de $194 \% 11 = 7$, y la resta de $11 - 7 = 4$.