

Escuela de Programación - Python A2

Proyecto Final: Encriptar y desencriptar cadenas

Introducción

Este documento describe el proyecto final de programación para el nivel A2 de la Escuela de Programación. El proyecto consiste en la implementación de varias funciones que permiten encriptar y desencriptar información. Para su desarrollo se ponen en práctica los conocimientos adquiridos durante el curso Python A2.

Criterios de evaluación

- Demostrar que se han asimilado correctamente los contenidos teóricos y que se aplican con precisión.
- Razonar todas las respuestas de forma analítica y sintética.
- Estructurar correctamente el código.
- Esquematizar las respuestas mediante el uso de estructuras de control condicionales y de flujos.
- Capacidad del alumno de convertir pseudocódigo en una implementación concreta.

Contexto

La encriptación es un método para codificar la información de manera que no se pueda utilizar hasta que se desencripte. Solo las partes autorizadas tienen las claves para leer esos datos o acceder a ellos. La encriptación de archivos ayuda a evitar el acceso no autorizado o la manipulación de los datos mientras están en tránsito o en reposo, según el método utilizado.

Los métodos iniciales de encriptación consistían simplemente en desplazar las letras del alfabeto o sustituirlas por letras y símbolos, como el cifrado César o la máquina Enigma. Hoy en día, la encriptación de archivos es más sofisticada, segura y difícil de descifrar.

La desencriptación se considera el proceso opuesto a la encriptación e implica cambiar la información electrónica o las señales que se almacenaron, escribieron o enviaron en forma de un código secreto (un sistema de letras, números o símbolos) de nuevo a una forma que se pueda entender y usar.

Pregunta 1

En este ejercicio, deberás implementar una función llamada `encrypt(inputValue, keys, token)` que reciba una cadena y realice la encriptación. Realizaremos una encriptación básica, basada en sustitución de caracteres, pero te permitirá trabajar los conceptos fundamentales del curso. La función recibe tres parámetros:

- **inputValue**, que será un string con la cadena a encriptar.
- **keys**, que será un diccionario compuesto por pares donde la clave será una letra y el valor correspondiente un número, estos servirán durante la encriptación. Por ejemplo:

```
keys= {  
    "a" : 1,  
    "e" : 2,  
    "u" : 3  
}
```

- **token**, que será un carácter que nos permitirá identificar el inicio y fin de una letra encriptada. Así mismo los valores válidos para este será uno de los siguientes caracteres: `= * + # % /`

El mecanismo de encriptación será el siguiente. Dado una cadena de caracteres que corresponden al parámetro **inputValue**, si existe una letra en la cadena que corresponda con una clave del diccionario **keys**, esta deberá ser reemplazada con el valor numérico que le corresponda y será delimitado al principio y al final con el carácter utilizado como **token**.

Aquí te mostramos algunos ejemplos:

- `encrypt ("hola", {'a':1}, '*')` devolvería `"hol*1**"`
- `encrypt ("Mis amigos son GENIALES.", {'i':0, 'E':8, 'a':3, 'A':4}, '+')` devolvería `"M+0+s +3+m+0+gos son G+8+NI+4+L+8+S."`
- `encrypt ("hola 11", {'a':1}, '#')` devolvería `"hol#1# 11"`
- `encrypt ("16", {'A':1}, '%')` devolvería `"16"`
- `encrypt ("Hola mi celular es 09999999", {'e':5, 'H':7}, '/')` devolvería `"/7/ola mi c/5/lular /5/s 09999999"`
- `encrypt ("Opcion que permite generar una solicitud", {'O':120, 'o':20, 'e':17, 'p':100, 'u':30}, '*')` devolvería `"*120**100*ci*20*n q*30**17* *100**17*rmit*17* g*17*n*17*rar *30*na s*20*licit*30*d"`
- `encrypt ("Hola mi celular es 09999999", {'9':7, 'H':8}, '=')` devolvería `"No es posible encriptar la cadena. Verifique los parámetros de entrada"`, esto es debido a que el parámetro **keys** tiene un valor numérico.
- `encrypt ("Hola mi celular es 09999999", {'e':7, 'H':8}, '{')` devolvería `"No es posible encriptar la cadena. Verifique los parámetros de entrada"`, esto es debido a que el parámetro **token** tiene un valor que no es válido.
- `encrypt ("Hola ** mi celular es 09999999", {'e':7, 'H':8}, '*')` devolvería `"No es posible encriptar la cadena. Verifique los parámetros de entrada"`, esto es debido a que la cadena de entrada **inputValue**, tiene incluido en su texto caracteres iguales al parámetro **token**.

Restricciones

Para las siguientes restricciones siempre deberá finalizar la ejecución del programa y mostrar el siguiente mensaje de error: "No es posible encriptar la cadena. Verifique los parámetros de entrada".

- Para el diccionario **keys** las claves deben contener solamente letras y los valores relacionados deben ser solamente números, y en ambos casos éstos no pueden estar repetidos.
- La longitud del campo **token** es 1.
- Si una cadena previamente tiene incluido el carácter del parámetro **token** deberá mostrar el mensaje de error.
- Si el parámetro **token** difiere de uno de los posibles valores válidos.

Pregunta 2

En este ejercicio deberás implementar una función que haga el proceso opuesto, es decir, que permite descryptar una cadena previamente encriptada. Para ello debes implementar una función llamada **decrypt(inputValue, keys, token)** que reciba una cadena encriptada y realice la descryptación. La función recibe tres parámetros:

- **inputValue**, que será un string con la cadena encriptada (resultado del proceso anterior).
- **keys**, que será un diccionario compuesto por pares donde la clave será una letra y el valor correspondiente un número, estos servirán durante la encriptación. Por ejemplo:

```
keys = {  
    "a" : 1,  
    "e" : 2,  
    "u" : 3  
}
```

- **token**, que será un carácter que nos permitirá identificar el inicio y fin de una letra encriptada. Así mismo los valores válidos para este será uno de los siguientes caracteres: **= * + # % /**

Como resultado, la función devolverá una cadena descryptada. Para este proceso es necesario identificar el principio y fin de una letra encriptada el cual está delimitado por el parámetro utilizado como **token**.

Aquí te mostramos algunos ejemplos:

- `decrypt ("hol*1*", {'a':1}, '*')` devolvería "hola"
- `decrypt ("M+0+s +3+m+0+gos son G+8+NI+4+L+8+S.", {'i':0,'E':8,'a':3,'A':4}, '+')` devolvería "Mis amigos son GENIALES."
- `decrypt ("hol#1# 11", {'a':1}, '#')` devolvería "hola 11"
- `decrypt ("16", {'A':1}, '%')` devolvería "16"
- `decrypt ("/7/ola mi c/5/lular /5/s 09999999", {'e':5,'H':7}, '/')` devolvería "Hola mi celular es 09999999"
- `decrypt ("*120**100*ci*20*n q*30**17* *100**17*rmit*17* g*17*n*17*rar *30*na s*20*licit*30*d", {'O':120,'o':20,'e':17,'p':100,'u':30}, '*')` devolvería "Opcion que permite generar una solicitud"
- `decrypt ("Hola mi celular es 09999999", {'9':7,'H':8}, '=')` devolvería "No es posible encriptar la cadena. Verifique los parámetros de entrada", esto es debido a que el parámetro **keys** tiene un valor numérico.
- `decrypt ("Hola mi celular es 09999999", {'e':7,'H':8}, '{')` devolvería "No es posible descryptar la cadena. Verifique los parámetros de entrada", esto es debido a que el parámetro **token** tiene un valor que no es válido.

Restricciones

Las mismas del apartado anterior, únicamente cambiaremos el mensaje por: "No es posible descryptar la cadena. Verifique los parámetros de entrada".

- Para el diccionario **keys** las claves deben contener solamente letras y los valores relacionados deben ser solamente números, y en ambos casos éstos no pueden estar repetidos.
- La longitud del campo **token** es 1.
- Si el parámetro **token** difiere de uno de los posibles valores válidos.

Evaluación

La puntuación de cada pregunta es la siguiente:

- Pregunta 1: 50%
- Pregunta 2: 50%

En cada ejercicio se valorará la validez de la solución y la claridad de la argumentación.

Cada ejercicio tiene indicado en el enunciado su peso en la valoración final. Los criterios de evaluación para evaluar este ejercicio son los siguientes:

Pregunta	No logrado (C-)	Mínimamente logrado (C+)	Logrado (B)	Logrado de forma excelente (A)
Pregunta 1	La respuesta es incorrecta o no fue desarrollada	La respuesta es parcialmente correcta y está mínimamente justificada	La respuesta es correcta y describe ciertos pasos para llegar a la solución, y está justificada	La respuesta es correcta e indica todos los pasos para llegar a la solución. Se referencia correctamente el artículo y apuntes para justificar la respuesta
Pregunta 2	La respuesta es incorrecta o no fue desarrollada	La respuesta es parcialmente correcta y está mínimamente justificada	La respuesta es correcta y describe ciertos pasos para llegar a la solución, y está justificada	La respuesta es correcta e indica todos los pasos para llegar a la solución. Se referencia correctamente el artículo y apuntes para justificar la respuesta

Formato

Se sugiere, con el objetivo de estandarizar el formato, la actividad siga las siguientes restricciones:

- El ejercicio se desarrollará en la plataforma Vocareum.
- Se pueden realizar varias entregas en la plataforma
- El estudiante debe notificar al Docente Colaborador un mensaje una vez que considere que el ejercicio esté listo para la calificación.
- Incluir comentarios en el código.

Entrega

La entrega de la Actividad deberá realizarse en la sección:

Vocareum -> Python -> X.Final Proyecto Final / Projecte Final

💡 Únicamente se calificarán aquellos proyectos que hayan enviado el correo solicitando la revisión por parte del profesor colaborador (**Miguel Ponce**) mponce1@uoc.edu; en caso de no cumplir con este requisito y una vez finalizado el curso, **el estudiante suspenderá de manera automática el nivel.**

Bibliografía

- Fortra. "[Introducción a La Encriptación de Archivos.](#)"
- Cambridge Dictionary. "[Decrypt.](#)"