Instituto técnico Renault

Guía Python 1 - Programación II

Ejercicio 1

Convertir número decimal a binario

Implementar una función ToBinary que reciba por parámetro un número decimal y retorne la representación binaria correspondiente.

* Luego de implementar la función se debe invocar y mostrar el resultado en la consola del sistema.

Ejemplos:

ToBinary(43) -> 101011

ToBinary(189) -> 10111101

Ejercicio 2

Convertir número binario a decimal

Implementar una función ToDecimal que reciba por parámetro un número binario y retorne la representación decimal correspondiente.

* Luego de implementar la función se debe invocar y mostrar el resultado en la consola del sistema.

Ejemplos:

ToDecimal(10111101) -> 189

ToDecimal(1001100) -> 76

Ejercicio 3

Invertir una cadena de texto

Implementar una función InverseString que reciba por parámetro una cadena y retorne la versión inversa de la misma.

* Luego de implementar la función se debe invocar y mostrar el resultado en la consola del sistema.

Ejemplos:

InverseString(“arbolito”) -> “otilobra”

Ejercicio 4

Los factores primos

Implementar una función PrimeFactors que calcule reciba por parámetro un número entero positivo y devuelva un arreglo conformado por los factores primos.

Ejemplos:

primeFactors(60) -> [2,2,3,5]

Ejercicio 5

La regla de las gotas de lluvia

La tarea consiste en convertir un número en una cadena que contenga sonidos de gotas de lluvia correspondientes a ciertos factores potenciales. Un factor es un número que divide uniformemente a otro número, sin dejar ningún resto. La forma más sencilla de comprobar si un número es factor de otro es utilizar la operación módulo.

Las reglas de las gotas de lluvia son que si un número dado tiene 3 como factor, añade 'Pling' al resultado, en caso de tener 5 como factor, añade 'Plang' al resultado y finalmente si tiene 7 como factor, añade "Plong" al resultado.

En caso de no tener ninguno de los 3, 5 o 7 como factor, el resultado deben ser los dígitos del número.

* Luego de implementar la función se debe invocar y mostrar el resultado en la consola del sistema.

Ejemplos:

- 28 tiene 7 como factor, pero no 3 ni 5, por lo que el resultado sería "Plong".
- 30 tiene 3 y 5 como factores, pero no 7, por lo que el resultado sería "PlingPlang".
- 34 no tiene como factor 3, 5 o 7, por lo que el resultado sería "34".

Ejercicio 6

Notación posicional y bases

Implementar una función ToAnyBase que tomé por parámetro un número entero y una base n donde n>=2. La implementación de la función deberá permitir que logre convertir el número, representado como una secuencia de dígitos en la base recibida en el segundo parámetro.

Notación posicional

En notación posicional, un número en base b puede entenderse como una combinación lineal de potencias de b.

Ejemplos:

El número 42, en base 10, significa

(4 _ 10^1) + (2 _ 10^0)

El número 101010, en base 2, significa:

(1 _ 2^5) + (0 _ 2^4) + (1 _ 2^3) + (0 _ 2^2) + (1 _ 2^1) + (0 _ 2^0)

El número 1120, en base 3, significa

(1 _ 3^3) + (1 _ 3^2) + (2 _ 3^1) + (0 _ 3^0)

Ejercicio 7

Codificación de longitud de carrera (RLE)

Implementar una función que calcule la codificación y de decodificación utilizando el método de codificación de longitud de carrera.

Ayuda: La codificación de longitud de carrera (RLE) es una forma sencilla de compresión de datos, en la que las que los elementos de datos consecutivos se sustituyen por un solo valor de datos y un recuento.

Ejemplos:

Por ejemplo, podemos representar los 53 caracteres originales con sólo 13.

"WWWWWWWWWWWWBWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWBWWWWWWWWWWWWWW" ->
"12WB12W3B24WB"

Ejercicio 8

La conjetura de Collatz

La conjetura de Collatz o el problema de 3x+1 puede resumirse como sigue:

Tome cualquier número entero positivo n. Si n es par, dividan por 2 para obtener n / 2. Si n es impar, multipliquen por 3 y sume 1 para obtener 3n + 1.

Repite el proceso indefinidamente. La conjetura afirma que, independientemente del número con el que se empiece, siempre se llegará a 1 en algún momento.

Implementa una función Collatz que dado un número n, retorne el número de pasos necesarios utilizando la conjetura descrita con anterioridad para llegar a 1.

* Luego de implementar la función se debe invocar y mostrar el resultado en la consola del sistema.

Ejercicio 9

Anagramas

Implementar una función IsAnagram que tome por parámetro una cadena y un arreglo de cadenas. Luego se deben filtrar las cadenas del arreglo que sean anagramas de la cadena recibida para finalmente retornar las cadenas filtradas en un nuevo arreglo.

* Luego de implementar la función se debe invocar y mostrar el resultado en la consola del sistema.

Un anagrama es una reorganización de las letras de una palabra para formar una nueva.

Ejercicio 10

Cifrado Cesar

Crea una implementación del cifrado rotacional, también llamado a veces el cifrado César. El cifrado César es un simple cifrado por desplazamiento que se basa en la transposición de todas las letras del alfabeto utilizando una clave entera entre 0 y 26. El uso de una clave de 0 o 26 siempre dará el mismo resultado debido a la aritmética modular.

La letra se desplaza por tantos valores como el valor de la clave.

Teniendo en cuenta lo detallado anteriormente se solicita implementar una función Rotational que tome como entrada una cadena de texto y una clave de movimiento para obtener como resultado la cadena resultante de la rotación provista por los movimientos determinados en la clave.

Ejemplos:

- Rotational(‘omg’,5) -> trl

- Rotational(‘The quick brown fox jumps over the lazy dog’,13) -> gives Gur dhvpx oebja sbk whzcf bire gur ynml qbt

Prestar atención que el desplazamiento en el alfabeto debe ser circular i.e siempre se cae dentro del mismo sin importar la cantidad de movimientos realizados.

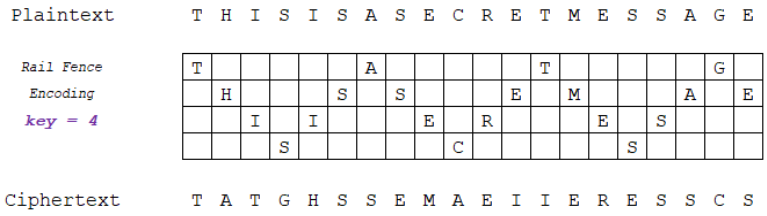
Ejercicio 11

Cifrado de cerca de riel

Implementar una función RailFence que tome por parámetros una cadena de texto y un número entero positivo mayor o igual a dos que represente la llave del cifrado.

Ayuda: En un cifrado de valla de riel, las letras no se cambian, sino que solo cambian con respecto a su posición en él. Este tipo de cifrado a menudo se denomina cifrado de transposición, porque las letras simplemente se transponen en términos de su ubicación. Estos tipos de cifrados se remontan a la Guerra Civil Americana, donde los soldados usarían el código para cifrar. En un cifrado de valla de riel, el escritor toma una frase y la escribe en líneas descendentes o "rieles". El cifrado de valla de riel a veces se denomina cifrado en zigzag si el escritor usa un patrón en zigzag o W para representar.

Ejemplo:



Ejercicio 12

Suma de múltiplos

Implementar una función SumOfMultiples que dado un número y un arreglos de múltiplos, determine la suma de todos los múltiplos únicos de números particulares hasta ese número pero sin incluirlo.

Ejemplos:

SumOfMultiples(20,[3,5]) -> 78

Notar que 78 se obtiene de sumar lo valores 3, 5, 6, 9, 10, 12, 15 y 18.

Donde n es la inversa multiplicativa modular de a mod m. Se puede encontrar más información sobre cómo encontrar un inverso multiplicativo modular y lo que significa aquí.

Ejercicio 13

Partida de Dragones y Mazmorras

En una partida de Dragones y Mazmorras, cada jugador comienza generando un personaje con el que puede jugar. Este personaje tiene, entre otras cosas, seis habilidades: fuerza, destreza, constitución, inteligencia, sabiduría y carisma. Estas seis habilidades tienen puntuaciones que se determinan al azar. Para ello, se tiran cuatro dados de 6 caras y se anota la suma de los tres dados más grandes. Lo haces seis veces, una por cada habilidad.

Los puntos de golpe iniciales de tu personaje son 10 + el modificador de constitución de tu personaje.

El modificador de constitución de tu personaje se obtiene restando 10 a la constitución de tu personaje, dividiendo por 2 y redondeando hacia abajo. Se solicita implementar una función GameGenerator que genere personajes aleatorios que sigan las reglas anteriores.

Por ejemplo, los seis lanzamientos de cuatro dados pueden ser como:

- 5, 3, 1, 6: Descartas el 1 y sumas 5 + 3 + 6 = 14, que asignas a la fuerza.
- 3, 2, 5, 3: Descartas el 2 y sumas 3 + 5 + 3 = 11, que asignas a la destreza.
- 1, 1, 1, 1: Descartas el 1 y sumas 1 + 1 + 1 = 3, que asignas a constitución.
- 2, 1, 6, 6: Descartas el 1 y sumas 2 + 6 + 6 = 14, que asignas a inteligencia.
- 3, 5, 3, 4: Descartas el 3 y sumas 5 + 3 + 4 = 12, que asignas a la sabiduría.
- 6, 6, 6, 6: Descartas el 6 y sumas 6 + 6 + 6 = 18, que asignas a carisma.

Como la constitución es 3, el modificador de constitución es -4 y los puntos de golpe son 6.