

Ejercicios Python

Ejercicio 1:

Escriba una función `frecuencia_digitos(n)` que reciba como entrada un número entero positivo `n` y devuelva como salida una lista de 10 números donde cada número deberá indicar la cantidad de veces que aparece cada dígito en el número. Por ejemplo, si el número de entrada es `913294207820`, la función deberá retornar la lista `[2,1,3,1,1,0,0,1,1,2]`, ya que hay dos 0, un 1, tres 2, un 3, un 4, ningún 5, ningún 6, un 7, un 8 y dos 9.

Ejercicio 2:

Escriba una función `cuenta_pares(n)` que reciba como entrada un número entero positivo `n` y devuelva como salida el número de dígitos pares que contiene. Por ejemplo, si el número de entrada es `123204`, la función deberá retornar `4`. Recuerde que el 0 es par.

Ejercicio 3:

Escriba una función `invierte_impares(n)` que reciba como entrada un número entero positivo `n` y devuelva como salida otro número que contenga únicamente los dígitos impares de `n` pero en orden inverso. Por ejemplo, si el número de entrada es `2741309`, la función deberá retornar el número `9317`.

Ejercicio 4:

Escriba una función `primeros_divisores(n)` que reciba como entrada un número entero positivo `n` y devuelva como salida una lista con los primeros 5 divisores (como máximo) de `n`. Por ejemplo, si el número de entrada es `24`, la función deberá retornar la lista `[1,2,3,4,6]`. Tenga en cuenta que hay números que pueden tener menos de 5 divisores. Por ejemplo, los divisores de 8 son 1, 2, 4 y 8, por lo que la función deberá retornar la lista `[1,2,4,8]`.

Ejercicio 5:

Escriba una función `vocales_frase(s)` que reciba como entrada una frase `s` y devuelva como salida una lista de números donde cada número deberá indicar la cantidad de vocales que hay en cada palabra de la frase. Por ejemplo, si la frase de entrada es `'Say no more'`, la función deberá retornar la lista `[1,1,2]`, ya que hay una vocal en la primera palabra, una vocal en la segunda palabra y dos vocales en la tercera palabra.

Ejercicio 6:

Escriba una función `reemplaza_suma(l)` que reciba como entrada una lista de números positivos de tres dígitos `l` y devuelva como salida otra lista donde cada número deberá reemplazarse por la suma de sus dígitos. Por ejemplo, si la lista de entrada es `[271,139,310]`, la función deberá retornar `[10,13,4]`, ya que $10 = 2 + 7 + 1$, $13 = 1 + 3 + 9$ y $4 = 3 + 1 + 0$.

Ejercicio 7:

Escriba una función `forma_entero(d)` que reciba como entrada un número decimal positivo `d` y devuelva como salida el número entero formado omitiendo el punto decimal. Por ejemplo, si el número de entrada es `91.467`, la función deberá retornar `91467`.

Ejercicio 8:

Escriba una función `filtra_pares(n)` que reciba como entrada un número entero positivo `n` y devuelva como salida otro número donde cada dígito par esté reemplazado por el dígito 1. Por ejemplo, si el número de entrada es `30852`, la función deberá retornar `31151`.

Ejercicio 9:

Escriba una función `combina_unicos(a, b)` que reciba dos listas de números, `a` y `b`, y retorne una nueva lista formada por la concatenación de `a` y `b` pero incluyendo sólo una vez aquellos elementos repetidos. Por ejemplo, para `[1,2,3,1,2]` y `[2,4,5,6,4]`, la función deberá retornar `[1,2,3,4,5,6]`.

Ejercicio 10:

Escriba una función `segundo_maximo(l)` que reciba una lista de números `l` y retorne el segundo número más grande de la lista. Por ejemplo, para la lista `[1,3,7,5]`, la función deberá retornar `5`.

Ejercicio 11:

Escriba una función `indices_especiales(s)` que reciba una cadena `s` y retorne una lista de índices correspondientes a las posiciones de los caracteres especiales de `s`. Recuerde que un caracter especial es aquel no es ni una letra ni un dígito, por ejemplo: `'*'`, `' '`, `'_'`, etc. Entonces, si la cadena de entrada es `'abc 123*_567'`, la función deberá retornar `[3,7,8]`.

Ejercicio 12:

Escriba una función `rota_izquierda(l, p)` que reciba una lista de números `l` y un número entero `p`, y retorne una nueva lista rotada hacia la izquierda `p` veces. La rotación a la izquierda es el desplazamiento de los elementos de la lista a una posición a la izquierda y la copia del primer elemento al último. Por ejemplo, para `[1,2,3,4,5]` y `1`, la versión rotada hacia la izquierda sería `[2,3,4,5,1]`. Y para `[1,2,3,4,5]` y `2`, la versión rotada hacia la izquierda sería `[3,4,5,1,2]`. Asuma que `p` es mayor o igual que 0 y menor que la longitud de la lista `l`.

Ejercicio 13:

Escriba una función `rota_derecha(l, p)` que reciba una lista de números `l` y un número entero `p`, y retorne una nueva lista rotada hacia la derecha `p` veces. La rotación a la derecha es el desplazamiento de los elementos de la lista a una posición a la derecha y la copia del último elemento al primero. Por ejemplo, para `[1,2,3,4,5]` y `1`, la versión rotada hacia la derecha sería `[5,1,2,3,4]`. Y para `[1,2,3,4,5]` y `2`, la versión rotada hacia la izquierda sería `[4,5,1,2,3]`. Asuma que `p` es mayor o igual que 0 y menor que la longitud de la lista `l`.

Ejercicio 14:

Escriba una función `producto_triple(l)` que reciba una lista de números de tres dígitos `l` y retorne `True` si al menos un producto de tres dígitos termina en 2 o `False` en caso contrario. Por ejemplo, para la lista `[253, 341, 613]` deberá retornar `True`, ya que $3 \times 4 \times 1 = 12$ termina en 2. En cambio, para `[253, 342, 613]` deberá retornar `False`, ya que ningún producto termina en 2.

Ejercicio 15:

Escriba una función `suma_potencia(p)` que reciba un número entero positivo `p` y calcule la suma de los dígitos de 2^p . Por ejemplo, si `p` es 4, entonces $2^4 = 16$ y la suma de los dígitos es $1 + 6 = 7$.