

EXAMEN DE PROGRAMACIÓN

Examen 1 · Python

INSTRUCCIONES

1. Lee cuidadosamente cada problema antes de comenzar a resolverlo.
2. Implementa las funciones solicitadas respetando los nombres y parámetros especificados.
3. Verifica que tu código produzca exactamente la salida esperada para cada caso.
4. No está permitido el uso de bibliotecas externas a menos que se indique lo contrario.
5. Puedes usar funciones auxiliares si lo consideras necesario.

1. Inversión de Lista

Crea una función en Python que reciba una lista de números enteros y retorne una nueva lista con los elementos en orden inverso. No utilices el método `.reverse()` ni slicing invertido.

ENTRADA

`lista = [1, 2, 3, 4, 5]`

SALIDA

`[5, 4, 3, 2, 1]`

2. Suma de Números Pares

Desarrolla una función que reciba una lista de números enteros y retorne la suma de todos los números pares contenidos en ella. Si no hay números pares, debe retornar 0.

ENTRADA	SALIDA
numeros = [1, 2, 3, 4, 5, 6, 7, 8]	20

3. Palíndromo

Implementa una función que determine si una cadena de texto es un palíndromo (se lee igual de izquierda a derecha que de derecha a izquierda). Ignora espacios y mayúsculas/minúsculas.

ENTRADA	SALIDA
texto = "Anita lava la tina"	True

4. Contador de Vocales

Crea una función que reciba una cadena de texto y retorne un diccionario con el conteo de cada vocal (a, e, i, o, u) presente en el texto. Las vocales deben contarse sin distinguir entre mayúsculas y minúsculas.

ENTRADA	SALIDA
texto = "Programacion en Python"	{'a': 2, 'e': 1, 'i': 1, 'o': 3, 'u': 0}

5. Números Primos

Desarrolla una función que reciba un número entero positivo n y retorne una lista con todos los números primos menores o iguales a n . Un número primo es aquel que solo es divisible por 1 y por sí mismo.

ENTRADA`n = 20`**SALIDA**`[2, 3, 5, 7, 11, 13, 17, 19]`

6. Factorial Recursivo

Implementa una función recursiva que calcule el factorial de un número entero no negativo. El factorial de n ($n!$) es el producto de todos los enteros positivos desde 1 hasta n . Por definición, $0! = 1$.

ENTRADA`numero = 5`**SALIDA**`120`

7. Eliminación de Duplicados

Crea una función que reciba una lista con elementos duplicados y retorne una nueva lista con los elementos únicos, manteniendo el orden de aparición original.

ENTRADA`lista = [1, 2, 2, 3, 4, 3, 5, 1, 6]`**SALIDA**`[1, 2, 3, 4, 5, 6]`

8. Serie de Fibonacci

Escribe una función que reciba un número entero n y retorne una lista con los primeros n números de la sucesión de Fibonacci. La sucesión comienza con 0 y 1, y cada número siguiente es la suma de los dos anteriores.

ENTRADA`n = 7`**SALIDA**`[0, 1, 1, 2, 3, 5, 8]`

9. Intersección de Listas

Desarrolla una función que reciba dos listas y devuelva una nueva lista que contenga solo los elementos que están presentes en ambas listas (intersección), sin duplicados en el resultado.

ENTRADA`lista1 = [1, 2, 3, 4], lista2 = [3,
4, 5, 6]`

SALIDA`[3, 4]`

10. Anagrama

Crea una función que reciba dos cadenas de texto y determine si son anagramas. Dos palabras son anagramas si contienen exactamente las mismas letras con la misma frecuencia, sin importar el orden.

ENTRADA`palabra1 = "amor", palabra2 = "roma"`

SALIDA`True`