

EXAMEN DE PROGRAMACIÓN

Lenguaje: Python

Nombre:

Fecha:

Problema 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 esperada:

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

Problema 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:

```
numeros = [1, 2, 3, 4, 5, 6, 7, 8]
```

Salida esperada:

```
suma_pares = 20
```

Problema 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:

```
texto = "Anita lava la tina"
```

Salida esperada:

```
es_palindromo = True
```

Problema 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:

```
texto = "Programacion en Python"
```

Salida esperada:

```
conteo = {'a': 2, 'e': 1, 'i': 1, 'o': 3, 'u': 0}
```

Problema 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 esperada:

```
primos = [2, 3, 5, 7, 11, 13, 17, 19]
```

Problema 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 esperada:

```
factorial = 120
```

Problema 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 esperada:

```
lista_unica = [1, 2, 3, 4, 5, 6]
```

Problema 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 esperada:

```
fibonacci = [0, 1, 1, 2, 3, 5, 8]
```

Problema 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 esperada:

```
interseccion = [3, 4]
```

Problema 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 esperada:

```
es_anagrama = True
```