



UNIVERSIDAD
NACIONAL DE SAN MARTÍN

GUÍA DE EJERCICIOS

Unidad 12: Ciclos y range

1 Ejercicios

1.1 Transpuesta de una matriz

Implementa una función que reciba una matriz (lista de listas) y devuelva su transpuesta.

1.2 Conteo de palabras únicas

Crea un programa que cuente la cantidad de palabras únicas en un texto ingresado por el usuario. Ignora las mayúsculas y minúsculas.

1.3 Frecuencia de caracteres

Escribe una función que reciba una cadena y devuelva un diccionario con la frecuencia de cada carácter en la cadena.

1.4 Intersección de dos listas

Implementa una función que reciba dos listas y devuelva una nueva lista con los elementos que aparecen en ambas listas, sin duplicados.

1.5 Verificar palíndromo

Escribe un programa que verifique si una cadena es un palíndromo, es decir, si se lee igual de adelante hacia atrás como de atrás hacia adelante.

1.6 Números perfectos

Crea un programa que encuentre todos los números perfectos en un rango dado. Un número perfecto es un número que es igual a la suma de sus divisores propios positivos (excluyendo el número mismo).

1.7 Encuentra el elemento faltante

Dada una lista de números consecutivos (sin orden), con un número faltante, escribe un programa para encontrar el número faltante.

1.8 Eliminar duplicados

Escribe una función que reciba una lista y devuelva una nueva lista sin elementos duplicados, manteniendo el orden original.

1.9 Inversión de palabras

Crea un programa que invierta las palabras en una oración. Por ejemplo, convierte "Hola Mundo" en "aloH odnuM".

1.10 Suma de matrices

Implementa una función que reciba dos matrices (listas de listas) de igual tamaño y devuelva su suma.

1.11 Número de palabras

Escribe un programa que cuente el número de palabras en una cadena ingresada por el usuario. Considera que las palabras están separadas por espacios.

1.12 Rotación de lista

Implementa una función que rote los elementos de una lista n posiciones hacia la izquierda. Por ejemplo, rotar [1, 2, 3, 4, 5] dos posiciones daría [3, 4, 5, 1, 2].

1.13 Conversión de mayúsculas y minúsculas

Escribe un programa que reciba una cadena y cambie todas las letras mayúsculas por minúsculas y viceversa.

1.14 Suma de números en cadena

Dada una cadena que contiene números y letras, escribe un programa que sume todos los números en la cadena. Por ejemplo, para la cadena "a1b2c3", el resultado debe ser 6.

1.15 Multiplicación de elementos de una lista

Escribe una función que reciba una lista de números y devuelva el producto de todos sus elementos.

1.16 Secuencia de Fibonacci

Implementa una función que genere una lista con los primeros n números de la secuencia de Fibonacci.

1.17 Filtrar palabras cortas

Crea un programa que reciba una lista de palabras y un número n , y devuelva una nueva lista con solo las palabras que tengan más de n caracteres.

1.18 Intercalar listas

Escribe una función que tome dos listas de igual longitud y las combine en una sola lista intercalando sus elementos. Por ejemplo, dadas las listas [1, 2, 3] y [a, b, c], el resultado debe ser [1, a, 2, b, 3, c].

1.19 Sumar elementos en índices pares

Escribe un programa que reciba una lista de números y devuelva la suma de los elementos que están en posiciones pares.

1.20 Combinaciones de caracteres

Escribe una función que tome una lista de caracteres y devuelva una lista con todas las combinaciones posibles de dos caracteres.

1.21 Encontrar sublistas

Crea un programa que reciba una lista y devuelva todas las posibles sublistas de esta lista. Por ejemplo, para la lista `[1, 2, 3]`, el resultado debe ser `[[], [1], [2], [3], [1, 2], [1, 3], [2, 3], [1, 2, 3]]`.

1.22 Invertir sublistas

Escribe una función que reciba una lista de listas y devuelva una nueva lista de listas donde cada sublista esté invertida. Por ejemplo, para la lista `[[1, 2], [3, 4], [5, 6]]`, el resultado debe ser `[[2, 1], [4, 3], [6, 5]]`.

1.23 Suma de dígitos

Implementa una función que reciba un número entero y devuelva la suma de sus dígitos. Usa un ciclo `while` para extraer y sumar los dígitos del número.

1.24 Menú interactivo

Crea un programa que muestre un menú con diferentes opciones al usuario (por ejemplo, agregar un elemento a una lista, eliminar un elemento de una lista, mostrar la lista, salir del programa) y que ejecute la opción seleccionada. Usa un ciclo `while` para mostrar el menú repetidamente hasta que el usuario elija salir del programa.

1.25 Suma de enteros positivos

Escribe un programa que solicite al usuario ingresar números enteros positivos hasta que el usuario ingrese un número negativo. Luego, muestra la suma de todos los números positivos ingresados.

1.26 Calculadora básica

Implementa un programa que permita al usuario realizar operaciones matemáticas básicas (suma, resta, multiplicación, división). El programa debe solicitar al

usuario ingresar dos números y la operación deseada, y luego mostrar el resultado. Usa un ciclo `while` para permitir al usuario realizar múltiples operaciones hasta que decida salir.

1.27 Promedio de números

Escribe un programa que solicite al usuario ingresar números uno por uno hasta que el usuario ingrese "0". Luego, calcula y muestra el promedio de los números ingresados, excluyendo el "0" final.

1.28 Simulación de colas

Simula una cola de espera en un banco. El programa debe permitir agregar personas a la cola, atender personas (eliminándolas de la cola) y mostrar el estado actual de la cola. Usa un ciclo `while` para mostrar un menú interactivo con estas opciones hasta que el usuario decida salir.