### **Ejercicio 1 – Serie Armónica y Constante de Euler–Mascheroni**

Este programa calcula la **serie armónica** hasta un número n dado por el usuario y la compara con la **aproximación teórica** ln(n) + γ, donde γ es la constante de Euler–Mascheroni.  
 La justificación es mostrar cómo la suma de los inversos de los números naturales crece lentamente y tiende a comportarse como el logaritmo natural, evidenciando una relación importante en análisis numérico y teoría de números.

### **Ejercicio 2 – Comparación de Fibonacci Recursivo e Iterativo**

Se implementan dos versiones para calcular el número de Fibonacci: una **recursiva** (ineficiente) y otra **iterativa** (eficiente).  
 El propósito es comparar sus **tiempos de ejecución** y evidenciar el impacto de la **complejidad temporal**, ya que la versión recursiva tiene crecimiento exponencial, mientras que la iterativa es lineal y mucho más rápida.

### **Ejercicio 3 – Estadística Básica con Números Aleatorios**

El programa genera una lista aleatoria de números enteros, la ordena y calcula sus **medidas estadísticas básicas**: media, mediana, moda y desviación estándar.  
 La justificación es aplicar conceptos de **estadística descriptiva** usando Python, mostrando cómo obtener e interpretar medidas de tendencia central y dispersión a partir de datos generados de forma aleatoria.

### **Ejercicio 4 – Distancia Máxima entre Puntos**

Dada una lista de puntos en el plano, el programa utiliza math.dist() y una función lambda dentro de max() para encontrar el **par de puntos más alejados entre sí**.  
 La justificación es practicar el uso de **funciones lambda**, **comprensiones de listas** y **herramientas del módulo math**, aplicadas al análisis geométrico de distancias.

### **Ejercicio 5 – Operaciones con Polinomios**

Se representan polinomios mediante listas de coeficientes y se implementan funciones para **evaluar** un polinomio en un valor de x y para **sumar** dos polinomios.  
 La justificación es comprender cómo manipular expresiones algebraicas mediante estructuras de datos en Python, aplicando operaciones aritméticas básicas sobre listas para simular cálculos simbólicos.

### **Ejercicio 6 – Operaciones con Conjuntos**

Este ejercicio utiliza **comprensiones de conjuntos** para definir A (múltiplos de 3) y B (múltiplos de 5) entre 1 y 50, y luego calcula su **intersección, unión y diferencia simétrica**.  
 La justificación es demostrar el uso de **operaciones de teoría de conjuntos** en Python, junto con el análisis de proporciones dentro de un universo numérico, reforzando conceptos de matemáticas discretas y lógica.