***Ejercicio 1***

Mi Código:

* Positivo (P):
  + Legibilidad: Mi código es fácil de entender debido a los comentarios descriptivos y nombres de variables significativos.
  + Funcionalidad: Calcula correctamente la suma de los números naturales desde 1 hasta el número ingresado por el usuario.
* Negativo (N):
  + Eficiencia: Utiliza un bucle for para sumar los números, lo que puede ser ineficiente para valores grandes de num.
  + Espacio de Memoria: La variable suma podría ser optimizada para evitar almacenar todos los valores intermedios.
* Interesante (I):
  + No se observan características particularmente interesantes en este código.

Código IA:

* Positivo (P):
  + Eficiencia: Utilizo la fórmula matemática para calcular directamente la suma de los números naturales.
  + Optimización de Memoria: No necesito almacenar todos los valores intermedios, lo que ahorra memoria.
* Negativo (N):
  + Legibilidad: Puede parecer menos intuitivo para alguien que no está familiarizado con la fórmula matemática.
  + Dependencia de Fórmula: Mi enfoque depende de la fórmula matemática y no es tan generalizable como un bucle.
* Interesante (I):
  + Utilizo la fórmula de la suma de una serie aritmética:  
     i=1∑n​i=2n(n+1)​

***Ejercicio 2*** *.*

* Mi Código:
  + Positivo (P):
    - Simplicidad: Mi código es conciso y fácil de entender.
    - Funcionalidad: Imprime correctamente los números impares en orden descendente.
    - Eficiencia: Utiliza un bucle for para iterar sobre los números impares, lo cual es eficiente.
  + Negativo (N):
  + Interesante (I):
    - No se observan características particularmente interesantes en este código.
* Código IA:
  + Positivo (P):
    - Eficiencia: Utilizo un bucle for similar al tuyo para imprimir los números impares.
    - Legibilidad: Mi código es simple y fácil de seguir.
    - Optimización de Memoria: No necesito almacenar valores intermedios, lo que ahorra memoria.
  + Negativo (N):
  + Interesante (I):
    - No se observan características particularmente interesantes en este código.

***Ejercicio 3***

Mi Código:

* Positivo (P):
  + Funcionalidad: Calcula correctamente la suma de los cuadrados de los números entre 1 y 100.
  + Legibilidad: Los comentarios descriptivos ayudan a entender el propósito del programa.
  + Eficiencia: Utiliza un bucle for para calcular los cuadrados, lo cual es eficiente.
* Negativo (N):
  + Espacio de Memoria: La variable suma podría optimizarse para evitar almacenar todos los valores intermedios.
* Interesante (I):
  + No se observan características particularmente interesantes en este código.

Código IA:

* Positivo (P):
  + Eficiencia: Utilizo la fórmula matemática para calcular directamente la suma de los cuadrados.
  + Optimización de Memoria: No necesito almacenar todos los valores intermedios, lo que ahorra memoria.
* Negativo (N):
  + Legibilidad: Puede parecer menos intuitivo para alguien que no está familiarizado con la fórmula matemática.
  + Dependencia de Fórmula: Mi enfoque depende de la fórmula matemática y no es tan generalizable como un bucle.
* Interesante (I):
  + Utilizo la fórmula para la suma de los cuadrados de una serie aritmética:  
     i=1∑n​i2=6n(n+1)(2n+1)​

***Ejercicio 4***

Mi Código:

* Positivo (P):
  + Funcionalidad: Calcula correctamente el promedio general de la sección.
  + Legibilidad: Los comentarios descriptivos ayudan a entender el propósito de las funciones.
* Negativo (N):
  + Espacio de Memoria: La variable suma podría ser optimizada para evitar almacenar todos los valores intermedios.
* Interesante (I):
  + No se observan características particularmente interesantes en este código.

Código IA:

* Positivo (P):
  + Eficiencia: Utilizo la fórmula matemática para calcular directamente el promedio.
  + Optimización de Memoria: No necesito almacenar todos los valores intermedios, lo que ahorra memoria.
* Negativo (N):
  + Legibilidad: Puede parecer menos intuitivo para alguien que no está familiarizado con la fórmula matemática.
  + Dependencia de Fórmula: Mi enfoque depende de la fórmula matemática y no es tan generalizable como un bucle.
* Interesante (I):
  + Utilizo la fórmula para el promedio de una serie aritmética:  
     Promedio=n∑i=1n​notai​​

***Ejercicio 5***

Mi Código:

* Positivo (P):
  + Funcionalidad: Calcula correctamente la cantidad de alumnos aprobados, la cantidad de alumnos reprobados y el promedio general del grupo.
  + Legibilidad: Los comentarios descriptivos ayudan a entender el propósito de las funciones.
* Negativo (N):
  + Espacio de Memoria: La variable suma podría ser optimizada para evitar almacenar todos los valores intermedios.
* Interesante (I):
  + No se observan características particularmente interesantes en este código.

Código IA:

* Positivo (P):
  + Eficiencia: Utilizo la fórmula matemática para calcular directamente el promedio.
  + Optimización de Memoria: No necesito almacenar todos los valores intermedios, lo que ahorra memoria.
* Negativo (N):
  + Legibilidad: Puede parecer menos intuitivo para alguien que no está familiarizado con la fórmula matemática.
  + Dependencia de Fórmula: Mi enfoque depende de la fórmula matemática y no es tan generalizable como un bucle.
* Interesante (I):
  + Utilizo la fórmula para el promedio de una serie aritmética:  
     Promedio=n∑i=1n​notai​​  
     .

***Ejercicio 6***

Mi Código:

* Positivo (P):
  + Funcionalidad: Calcula correctamente la suma de los números pares entre 100 y 200.
  + Legibilidad: Los comentarios descriptivos ayudan a entender el propósito del programa.
* Negativo (N):
  + Eficiencia: Utiliza un bucle for para calcular la suma, lo cual es eficiente.
  + Espacio de Memoria: La variable suma podría ser optimizada para evitar almacenar todos los valores intermedios.
* Interesante (I):
  + No se observan características particularmente interesantes en este código.

Código IA:

* Positivo (P):
  + Eficiencia: Utilizo la fórmula matemática para calcular directamente la suma de los números pares.
  + Optimización de Memoria: No necesito almacenar todos los valores intermedios, lo que ahorra memoria.
* Negativo (N):
  + Legibilidad: Puede parecer menos intuitivo para alguien que no está familiarizado con la fórmula matemática.
  + Dependencia de Fórmula: Mi enfoque depende de la fórmula matemática y no es tan generalizable como un bucle.
* Interesante (I):
  + Utilizo la fórmula para la suma de una serie aritmética de números pares:  
     i=1∑n​2i=2⋅2n(n+1)​=n(n+1)  
     .

**Ejercicio 7**

Mi Código:

* Positivo (P):
  + Funcionalidad: Calcula correctamente la suma de los dos vectores de igual longitud.
  + Legibilidad: Los comentarios descriptivos ayudan a entender el propósito de las funciones.
* Negativo (N):
  + Eficiencia: Utiliza un bucle for para calcular la suma, lo cual es eficiente.
  + Espacio de Memoria: La variable resultado podría ser optimizada para evitar almacenar todos los valores intermedios.
* Interesante (I):
  + No se observan características particularmente interesantes en este código.

Código IA:

* Positivo (P):
  + Eficiencia: Utilizo la fórmula matemática para calcular directamente la suma de los dos vectores.
  + Optimización de Memoria: No necesito almacenar todos los valores intermedios, lo que ahorra memoria.
* Negativo (N):
  + Legibilidad: Puede parecer menos intuitivo para alguien que no está familiarizado con la fórmula matemática.
  + Dependencia de Fórmula: Mi enfoque depende de la fórmula matemática y no es tan generalizable como un bucle.
* Interesante (I):
  + Utilizo la fórmula para la suma de dos vectores:  
     resultado[i]=vector1[i]+vector2[i]  
     .

**Ejercicio 8**

Mi Código:

* Positivo (P):
  + Funcionalidad: Calcula correctamente el producto punto de dos vectores de igual longitud.
  + Legibilidad: Los comentarios descriptivos ayudan a entender el propósito de las funciones.
* Negativo (N):
  + Eficiencia: Utiliza un bucle for para calcular el producto punto, lo cual es eficiente.
  + Espacio de Memoria: La variable resultado podría ser optimizada para evitar almacenar todos los valores intermedios.
* Interesante (I):
  + No se observan características particularmente interesantes en este código.

Código IA:

* Positivo (P):
  + Eficiencia: Utilizo la fórmula matemática para calcular directamente el producto punto de los dos vectores.
  + Optimización de Memoria: No necesito almacenar todos los valores intermedios, lo que ahorra memoria.
* Negativo (N):
  + Legibilidad: Puede parecer menos intuitivo para alguien que no está familiarizado con la fórmula matemática.
  + Dependencia de Fórmula: Mi enfoque depende de la fórmula matemática y no es tan generalizable como un bucle.
* Interesante (I):
  + Utilizo la fórmula para el producto punto de dos vectores:  
     resultado=i=1∑n​vector1[i]⋅vector2[i]  
     .

***Ejercicio 9***

Mi Código:

* Positivo (P):
  + Funcionalidad: Calcula correctamente la multiplicación de dos matrices.
  + Legibilidad: Los comentarios descriptivos ayudan a entender el propósito de las funciones.
* Negativo (N):
  + Eficiencia: Utiliza bucles for anidados para calcular la multiplicación, lo cual es eficiente.
  + Espacio de Memoria: Las matrices matriz1 y matriz2 podrían ser optimizadas para evitar almacenar todos los valores intermedios.
* Interesante (I):
  + No se observan características particularmente interesantes en este código.

Código IA:

* Positivo (P):
  + Eficiencia: Utilizo la fórmula matemática para calcular directamente la multiplicación de matrices.
  + Optimización de Memoria: No necesito almacenar todos los valores intermedios, lo que ahorra memoria.
* Negativo (N):
  + Legibilidad: Puede parecer menos intuitivo para alguien que no está familiarizado con la fórmula matemática.
  + Dependencia de Fórmula: Mi enfoque depende de la fórmula matemática y no es tan generalizable como un bucle.
* Interesante (I):
  + Utilizo la fórmula para la multiplicación de matrices:  
     resultado[i][j]=k=1∑n​matriz1[i][k]⋅matriz2[k][j]

**Ejercicio 10**

Mi Código:

* Positivo (P):
  + Funcionalidad: Calcula correctamente la transposición de una matriz.
  + Legibilidad: Los comentarios descriptivos ayudan a entender el propósito de las funciones.
* Negativo (N):
  + Eficiencia: Utiliza bucles for anidados para calcular la transposición, lo cual es eficiente.
  + Espacio de Memoria: La matriz matriz\_transpuesta podría ser optimizada para evitar almacenar todos los valores intermedios.
* Interesante (I):
  + No se observan características particularmente interesantes en este código.

Código IA:

* Positivo (P):
  + Eficiencia: Utilizo la fórmula matemática para calcular directamente la transposición de la matriz.
  + Optimización de Memoria: No necesito almacenar todos los valores intermedios, lo que ahorra memoria.
* Negativo (N):
  + Legibilidad: Puede parecer menos intuitivo para alguien que no está familiarizado con la fórmula matemática.
  + Dependencia de Fórmula: Mi enfoque depende de la fórmula matemática y no es tan generalizable como un bucle.
* Interesante (I):
  + Utilizo la fórmula para la transposición de una matriz:  
     matriz\_transpuesta[i][j]=matriz[j][i]  
     .